cae

0.0.0

Generated by Doxygen 1.11.0

1 cae 1
1.1 Cross-API-Engine | Rendering Engine with multiple dynamic backends . . . 1
1.1.1 Prerequisites . . . 2
1.1.2 External Libraries . . . 2
1.1.3 Contributing . . . 2
1.1.4 License . . . 2

2 Commit Norms 3

3 Namespace Index 5
3.1 Namespace List . . . 5

4 Hierarchical Index 7
4.1 Class Hierarchy . . . 7

5 Class Index 9
5.1 Class List . . . 9

6 File Index 11
6.1 File List . . . 11

7 Namespace Documentation 13
7.1 cae Namespace Reference . . . 13
7.2 cae::Audio Namespace Reference . . . 13
7.2.1 Variable Documentation . . . 14
7.2.1.1 MUTED . . . 14
7.2.1.2 VOLUME . . . 14
7.3 cae::Network Namespace Reference . . . 14
7.3.1 Variable Documentation . . . 14
7.3.1.1 HOST . . . 14
7.3.1.2 PORT . . . 14
7.4 cae::User Namespace Reference . . . 14
7.4.1 Variable Documentation . . . 14
7.4.1.1 NAME . . . 14
7.5 cae::Window Namespace Reference . . . 15
7.5.1 Variable Documentation . . . 15
7.5.1.1 FULLSCREEN . . . 15
7.5.1.2 HEIGHT . . . 15
7.5.1.3 MAX_FPS . . . 15
7.5.1.4 NAME . . . 15
7.5.1.5 VSYNC . . . 15
7.5.1.6 WIDTH . . . 15

8 Class Documentation 17
8.1 cae::Application Class Reference . . . 17
8.1.1 Detailed Description . . . 18

8.1.2 Constructor & Destructor Documentation . . . 18
8.1.2.1 Application() [1/3] . . . 18
8.1.2.2 ~Application() . . . 18
8.1.2.3 Application() [2/3] . . . 18
8.1.2.4 Application() [3/3] . . . 18
8.1.3 Member Function Documentation . . . 18
8.1.3.1 operator=() [1/2] . . . 18
8.1.3.2 operator=() [2/2] . . . 18
8.2 cae::ArgsConfig Struct Reference . . . 19
8.2.1 Detailed Description . . . 19
8.2.2 Member Data Documentation . . . 19
8.2.2.1 run . . . 19
8.3 cae::ArgsHandler Class Reference . . . 20
8.3.1 Detailed Description . . . 20
8.3.2 Constructor & Destructor Documentation . . . 21
8.3.2.1 ArgsHandler() [1/3] . . . 21
8.3.2.2 ~ArgsHandler() . . . 21
8.3.2.3 ArgsHandler() [2/3] . . . 21
8.3.2.4 ArgsHandler() [3/3] . . . 21
8.3.3 Member Function Documentation . . . 21
8.3.3.1 operator=() [1/2] . . . 21
8.3.3.2 operator=() [2/2] . . . 21
8.3.3.3 ParseArgs() . . . 21
8.3.3.4 ParseEnv() . . . 22
8.4 cae::EnvConfig Struct Reference . . . 22
8.4.1 Detailed Description . . . 22
8.5 cae::IRenderer Interface Reference . . . 23
8.5.1 Detailed Description . . . 24
8.5.2 Constructor & Destructor Documentation . . . 24
8.5.2.1 ~IRenderer() . . . 24
8.6 cae::IWindow Interface Reference . . . 25
8.6.1 Detailed Description . . . 26
8.6.2 Constructor & Destructor Documentation . . . 26
8.6.2.1 ~IWindow() . . . 26
8.7 cae::OPGL Class Reference . . . 26
8.7.1 Detailed Description . . . 28
8.7.2 Constructor & Destructor Documentation . . . 28
8.7.2.1 OPGL() [1/3] . . . 28
8.7.2.2 ~OPGL() . . . 28
8.7.2.3 OPGL() [2/3] . . . 28
8.7.2.4 OPGL() [3/3] . . . 28
8.7.3 Member Function Documentation . . . 28
8.7.3.1 getName() . . . 28

8.7.3.2 getType() . . . 28
8.7.3.3 operator=() [1/2] . . . 28
8.7.3.4 operator=() [2/2] . . . 29
8.8 cae::VULKN Class Reference . . . 29
8.8.1 Detailed Description . . . 31
8.8.2 Constructor & Destructor Documentation . . . 31
8.8.2.1 VULKN() [1/3] . . . 31
8.8.2.2 ~VULKN() . . . 31
8.8.2.3 VULKN() [2/3] . . . 31
8.8.2.4 VULKN() [3/3] . . . 31
8.8.3 Member Function Documentation . . . 31
8.8.3.1 getName() . . . 31
8.8.3.2 getType() . . . 31
8.8.3.3 operator=() [1/2] . . . 31
8.8.3.4 operator=() [2/2] . . . 31

9 File Documentation 33
9.1 /home/masina/Projects/Cross-API-Engine/CONTRIBUTING.md File Reference . . . 33
9.2 /home/masina/Projects/Cross-API-Engine/include/CAE/Application.hpp File Reference . . . 33
9.2.1 Detailed Description . . . 34
9.3 Application.hpp . . . 34
9.4 /home/masina/Projects/Cross-API-Engine/include/CAE/ArgsHandler.hpp File Reference . . . 34
9.4.1 Detailed Description . . . 35
9.5 ArgsHandler.hpp . . . 35
9.6 /home/masina/Projects/Cross-API-Engine/include/CAE/Common.hpp File Reference . . . 36
9.6.1 Detailed Description . . . 36
9.7 Common.hpp . . . 36
9.8 /home/masina/Projects/Cross-API-Engine/include/CAE/Generated/Version.hpp File Reference . . . 37
9.8.1 Macro Definition Documentation . . . 37
9.8.1.1 BUILD_TYPE . . . 37
9.8.1.2 GIT_COMMIT_HASH . . . 38
9.8.1.3 GIT_TAG . . . 38
9.8.1.4 PROJECT_NAME . . . 38
9.8.1.5 PROJECT_VERSION . . . 38
9.8.1.6 PROJECT_VERSION_MAJOR . . . 38
9.8.1.7 PROJECT_VERSION_MINOR . . . 38
9.8.1.8 PROJECT_VERSION_PATCH . . . 38
9.9 Version.hpp . . . 39
9.10 /home/masina/Projects/Cross-API-Engine/include/CAE/Interfaces/IRenderer.hpp File Reference . . . 39
9.10.1 Detailed Description . . . 40
9.11 IRenderer.hpp . . . 40

9.12 /home/masina/Projects/Cross-API-Engine/include/CAE/Interfaces/IWindow.hpp File Reference . . . 41
9.12.1 Detailed Description . . . 41
9.13 IWindow.hpp . . . 41
9.14 /home/masina/Projects/Cross-API-Engine/plugins/Renderer/OpenGL/include/OPGL/↩ OPGL.hpp File Reference . . . 42
9.14.1 Detailed Description . . . 43
9.15 OPGL.hpp . . . 43
9.16 /home/masina/Projects/Cross-API-Engine/plugins/Renderer/OpenGL/src/entrypoint.cpp File Reference . . . 44
9.16.1 Function Documentation . . . 44
9.16.1.1 entryPoint() . . . 44
9.17 entrypoint.cpp . . . 45
9.18 /home/masina/Projects/Cross-API-Engine/plugins/Renderer/Vulkan/src/entrypoint.cpp File Reference . . . 45
9.18.1 Function Documentation . . . 45
9.18.1.1 entryPoint() . . . 45
9.19 entrypoint.cpp . . . 46
9.20 /home/masina/Projects/Cross-API-Engine/plugins/Renderer/Vulkan/include/VULKN/↩ VULKN.hpp File Reference . . . 46
9.20.1 Detailed Description . . . 47
9.21 VULKN.hpp . . . 47
9.22 /home/masina/Projects/Cross-API-Engine/README.md File Reference . . . 48
9.23 /home/masina/Projects/Cross-API-Engine/src/application.cpp File Reference . . . 48
9.23.1 Macro Definition Documentation . . . 48
9.23.1.1 PLUGINS_EXTENSION . . . 48
9.24 application.cpp . . . 49
9.25 /home/masina/Projects/Cross-API-Engine/src/argsHandler.cpp File Reference . . . 49
9.25.1 Macro Definition Documentation . . . 50
9.25.1.1 APP_EXTENSION . . . 50
9.25.2 Variable Documentation . . . 50
9.25.2.1 ARGS_MAP . . . 50
9.25.2.2 HELP_MSG . . . 50
9.25.2.3 VERSION_MSG . . . 50
9.26 argsHandler.cpp . . . 51
9.27 /home/masina/Projects/Cross-API-Engine/src/main.cpp File Reference . . . 51
9.27.1 Function Documentation . . . 52
9.27.1.1 main() . . . 52
9.28 main.cpp . . . 53

Index 55

Chapter 1

cae

1.1 Cross-API-Engine | Rendering Engine with multiple dynamic backends

Cross-API-Engine is a rendering engine designed to support multiple graphics APIs dynamically. It allows developers to switch between different rendering backends such as OpenGL, Vulkan, DirectX at runtime. It is useful to do benchmarks during development or to support multiple platforms with different graphics APIs.

```
flowchart LR
subgraph main
    subgraph App
        A[Engine]
        A -->|.so/.dylib/.dll| B[IAudio]
        A -->|.so/.dylib/.dll| C[INetwork]
        A -->|.so/.dylib/.dll| D[IRenderer]
        A -->|.so/.dylib/.dll| E[IWindow]
    end

    subgraph Plugins
        subgraph audio impl
            F[WASAPI]
            K[ALSA]
            L[CoreAudio]
        end
        subgraph network impl
            G[Winsock2]
            O[Berkeley Sockets]
        end
        subgraph renderer impl
            H[OpenGL]
            I[Vulkan]
        end
        subgraph window impl
            J[WIN32]
            M[WayLand]
            N[Cocoa]
            subgraph Input devices
                P[IInput]
                Q[KeyBoard]
                R[Mouse]
                S[Controller]
            end
        end
    end

    B -.-> F
    B -.-> K
    B -.-> L
    C -.-> G
    C -.-> O
    D -.-> H
```

```
    D -.-> I
    E -.-> P
    E -.-> J
    E -.-> M
    E -.-> N
    P -.-> S
    P -.-> R
    P -.-> Q
end
```

1.1.1 Prerequisites

Make sure you have the following dependencies installed on your system:

- CMake 4.0.0
- C++23
- Vulkan SDK

1.1.2 External Libraries

- GLFW: For creating windows, receiving input, and managing OpenGL and Vulkan contexts.
- Google Test: A testing framework for C++.
- ImGui: Immediate Mode Graphical User Interface for real-time debugging and tool development.
- stb: A set of single-file public domain libraries for graphics, image loading, and more.

1.1.3 Contributing

Want to contribute? See CONTRIBUTING.md.

1.1.4 License

This project is licensed under the MIT License - see the LICENSE file for details.

Chapter 2

Commit Norms

Commit Type	Description
build	Changes that affect the build system or external dependencies (npm, make, etc.)
ci	Changes related to integration files and scripts or configuration (Travis, Ansible, BrowserStack, etc.)
feat	Addition of a new feature
fix	Bug fix
perf	Performance improvements
refactor	Modification that neither adds a new feature nor improves performance
style	Change that does not affect functionality or semantics (indentation, formatting, adding space, renaming a variable, etc.)
docs	Writing or updating documentation
test	Addition or modification of tests

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

cae 13
cae::Audio 13
cae::Network 14
cae::User 14
cae::Window 15

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

cae::Application 17
cae::ArgsConfig 19
cae::ArgsHandler 20
cae::EnvConfig 22
utl::IPlugin
  cae::IRenderer 23
    cae::OPGL 26
    cae::VULKN 29
  cae::IWindow 25

Chapter 5

Class Index

5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

cae::Application
: Main class . . . 17

cae::ArgsConfig . . . 19

cae::ArgsHandler
: Class to handle command line arguments . . . 20

cae::EnvConfig . . . 22

cae::IRenderer
: Interface for renderer . . . 23

cae::IWindow
: Interface for window . . . 25

cae::OPGL
: Class for the OpenGL plugin . . . 26

cae::VULKN
: Class for the Vulkan plugin . . . 29

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

/home/masina/Projects/Cross-API-Engine/include/CAE/Application.hpp
This file contains the Application class declaration 33
/home/masina/Projects/Cross-API-Engine/include/CAE/ArgsHandler.hpp
This file contains the ArgsHandler class declaration 34
/home/masina/Projects/Cross-API-Engine/include/CAE/Common.hpp
This file contains 36
/home/masina/Projects/Cross-API-Engine/include/CAE/Generated/Version.hpp 37
/home/masina/Projects/Cross-API-Engine/include/CAE/Interfaces/IRenderer.hpp
This file contains the Renderer interface 39
/home/masina/Projects/Cross-API-Engine/include/CAE/Interfaces/IWindow.hpp
This file contains the Window interface 41
/home/masina/Projects/Cross-API-Engine/plugins/Renderer/OpenGL/include/OPGL/OPGL.hpp
This file contains the OPGL class declaration 42
/home/masina/Projects/Cross-API-Engine/plugins/Renderer/OpenGL/src/entrypoint.cpp . . 44
/home/masina/Projects/Cross-API-Engine/plugins/Renderer/Vulkan/include/VULKN/VULKN.hpp
This file contains the VULKN class declaration 46
/home/masina/Projects/Cross-API-Engine/plugins/Renderer/Vulkan/src/entrypoint.cpp . . . 45
/home/masina/Projects/Cross-API-Engine/src/application.cpp 48
/home/masina/Projects/Cross-API-Engine/src/argsHandler.cpp 49
/home/masina/Projects/Cross-API-Engine/src/main.cpp 51

Chapter 7

Namespace Documentation

7.1 cae Namespace Reference

Namespaces

- namespace Audio
- namespace Network
- namespace User
- namespace Window

Classes

- class Application
  Main class.
- struct ArgsConfig
- class ArgsHandler
  Class to handle command line arguments.
- struct EnvConfig
- interface IRenderer
  Interface for renderer.
- interface IWindow
  Interface for window.
- class OPGL
  Class for the OpenGL plugin.
- class VULKN
  Class for the Vulkan plugin.

7.2 cae::Audio Namespace Reference

Variables

- constexpr auto VOLUME = 50.F
- constexpr auto MUTED = false

7.2.1 Variable Documentation

7.2.1.1 MUTED

auto cae::Audio::MUTED = false [inline], [constexpr]

Definition at line 14 of file Common.hpp.

7.2.1.2 VOLUME

auto cae::Audio::VOLUME = 50.F [inline], [constexpr]

Definition at line 13 of file Common.hpp.

7.3 cae::Network Namespace Reference

Variables

- constexpr auto HOST = "127.0.0.1"
- constexpr auto PORT = 4242

7.3.1 Variable Documentation

7.3.1.1 HOST

auto cae::Network::HOST = "127.0.0.1" [inline], [constexpr]

Definition at line 18 of file Common.hpp.

7.3.1.2 PORT

auto cae::Network::PORT = 4242 [inline], [constexpr]

Definition at line 19 of file Common.hpp.

7.4 cae::User Namespace Reference

Variables

- constexpr auto NAME = "User"

7.4.1 Variable Documentation

7.4.1.1 NAME

auto cae::User::NAME = "User" [inline], [constexpr]

Definition at line 23 of file Common.hpp.

7.5 cae::Window Namespace Reference

Variables

- constexpr auto HEIGHT = 1920
- constexpr auto WIDTH = 1080
- constexpr auto NAME = "CAE - Cross API Engine"
- constexpr auto FULLSCREEN = false
- constexpr auto VSYNC = false
- constexpr auto MAX_FPS = 90

7.5.1 Variable Documentation

7.5.1.1 FULLSCREEN

auto cae::Window::FULLSCREEN = false [inline], [constexpr]

Definition at line 30 of file Common.hpp.

7.5.1.2 HEIGHT

auto cae::Window::HEIGHT = 1920 [inline], [constexpr]

Definition at line 27 of file Common.hpp.

7.5.1.3 MAX_FPS

auto cae::Window::MAX_FPS = 90 [inline], [constexpr]

Definition at line 32 of file Common.hpp.

7.5.1.4 NAME

auto cae::Window::NAME = "CAE - Cross API Engine" [inline], [constexpr]

Definition at line 29 of file Common.hpp.

7.5.1.5 VSYNC

auto cae::Window::VSYNC = false [inline], [constexpr]

Definition at line 31 of file Common.hpp.

7.5.1.6 WIDTH

auto cae::Window::WIDTH = 1080 [inline], [constexpr]

Definition at line 28 of file Common.hpp.

Chapter 8

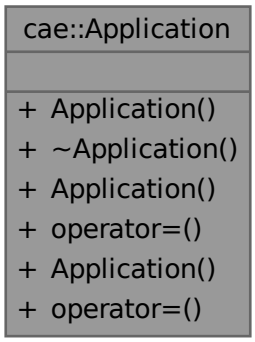
Class Documentation

8.1 cae::Application Class Reference

Main class.

#include <Application.hpp>

Collaboration diagram for cae::Application:



Public Member Functions

- Application ()
- ~Application ()=default
- Application (const Application &)=delete
- Application & operator= (const Application &)=delete
- Application (Application &&)=delete
- Application & operator= (Application &&)=delete

8.1.1 Detailed Description

Main class.

Definition at line 17 of file Application.hpp.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 Application() [1/3]

cae::Application::Application ()

Definition at line 17 of file application.cpp.

References PLUGINS_EXTENSION.

8.1.2.2 ~Application()

cae::Application::~Application () [default]

8.1.2.3 Application() [2/3]

cae::Application::Application (
const Application &) [delete]

8.1.2.4 Application() [3/3]

cae::Application::Application (
Application &&) [delete]

8.1.3 Member Function Documentation

8.1.3.1 operator=() [1/2]

Application & cae::Application::operator= (
Application &&) [delete]

8.1.3.2 operator=() [2/2]

Application & cae::Application::operator= (
const Application &) [delete]

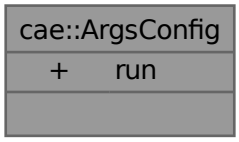
The documentation for this class was generated from the following files:

- /home/masina/Projects/Cross-API-Engine/include/CAE/Application.hpp
- /home/masina/Projects/Cross-API-Engine/src/application.cpp

8.2 cae::ArgsConfig Struct Reference

#include <ArgsHandler.hpp>

Collaboration diagram for cae::ArgsConfig:



Public Attributes

- bool run {false}

8.2.1 Detailed Description

Definition at line 12 of file ArgsHandler.hpp.

8.2.2 Member Data Documentation

8.2.2.1 run

bool cae::ArgsConfig::run {false}

Definition at line 14 of file ArgsHandler.hpp.

Referenced by cae::ArgsHandler::ParseArgs().

The documentation for this struct was generated from the following file:

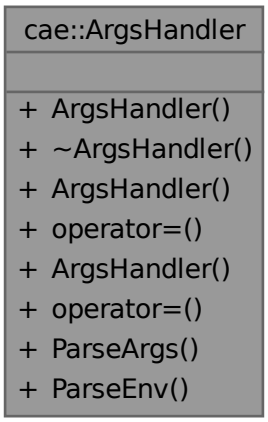
- /home/masina/Projects/Cross-API-Engine/include/CAE/ArgsHandler.hpp

8.3 cae::ArgsHandler Class Reference

Class to handle command line arguments.

#include <ArgsHandler.hpp>

Collaboration diagram for cae::ArgsHandler:



Public Member Functions

- ArgsHandler ()=default
- ~ArgsHandler ()=default
- ArgsHandler (const ArgsHandler &)=delete
- ArgsHandler & operator= (const ArgsHandler &)=delete
- ArgsHandler (ArgsHandler &&)=delete
- ArgsHandler & operator= (ArgsHandler &&)=delete

Static Public Member Functions

- static ArgsConfig ParseArgs (int argc, const char *const *argv)
- static EnvConfig ParseEnv (const char *const *envp)

8.3.1 Detailed Description

Class to handle command line arguments.

Definition at line 25 of file ArgsHandler.hpp.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 ArgsHandler() [1/3]

cae::ArgsHandler::ArgsHandler () [default]

8.3.2.2 ~ArgsHandler()

cae::ArgsHandler::~ArgsHandler () [default]

8.3.2.3 ArgsHandler() [2/3]

cae::ArgsHandler::ArgsHandler (
const ArgsHandler &) [delete]

8.3.2.4 ArgsHandler() [3/3]

cae::ArgsHandler::ArgsHandler (
ArgsHandler &&) [delete]

8.3.3 Member Function Documentation

8.3.3.1 operator=() [1/2]

ArgsHandler & cae::ArgsHandler::operator= (
ArgsHandler &&) [delete]

8.3.3.2 operator=() [2/2]

ArgsHandler & cae::ArgsHandler::operator= (
const ArgsHandler &) [delete]

8.3.3.3 ParseArgs()

cae::ArgsConfig cae::ArgsHandler::ParseArgs (
int argc,
const char *const * argv) [static]

Definition at line 30 of file argsHandler.cpp.

References ARGS_MAP, and cae::ArgsConfig::run.

Referenced by main().

Here is the caller graph for this function:

8.3.3.4 ParseEnv()

cae::EnvConfig cae::ArgsHandler::ParseEnv (
const char *const * envp) [static]

Definition at line 45 of file argsHandler.cpp.

Referenced by main().

Here is the caller graph for this function:

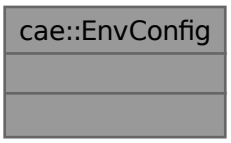
The documentation for this class was generated from the following files:

- /home/masina/Projects/Cross-API-Engine/include/CAE/ArgsHandler.hpp
- /home/masina/Projects/Cross-API-Engine/src/argsHandler.cpp

8.4 cae::EnvConfig Struct Reference

#include <ArgsHandler.hpp>

Collaboration diagram for cae::EnvConfig:



8.4.1 Detailed Description

Definition at line 16 of file ArgsHandler.hpp.

The documentation for this struct was generated from the following file:

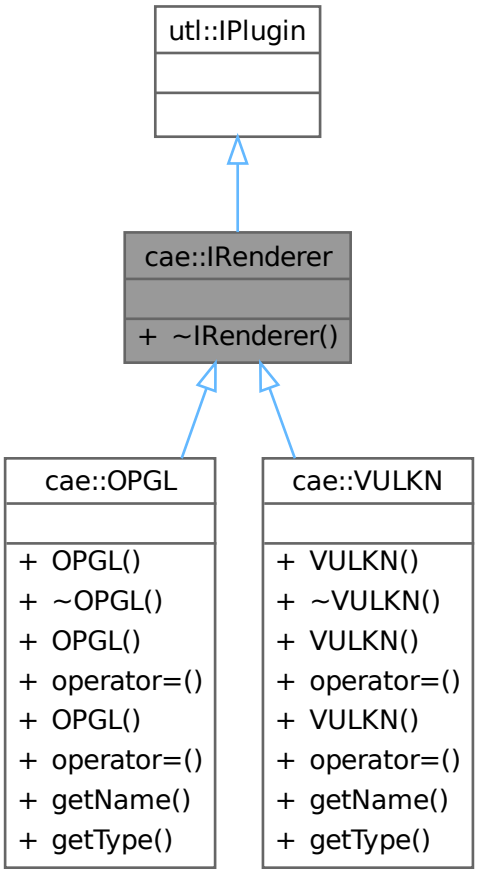
- /home/masina/Projects/Cross-API-Engine/include/CAE/ArgsHandler.hpp

8.5 cae::IRenderer Interface Reference

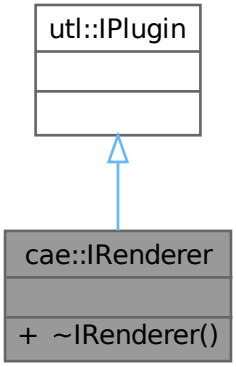
Interface for renderer.

#include <IRenderer.hpp>

Inheritance diagram for cae::IRenderer:



Collaboration diagram for cae::IRenderer:



Public Member Functions

- ~IRenderer () override=default

8.5.1 Detailed Description

Interface for renderer.

Definition at line 19 of file IRenderer.hpp.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 ~IRenderer()

cae::IRenderer::~IRenderer () [override], [default]

The documentation for this interface was generated from the following file:

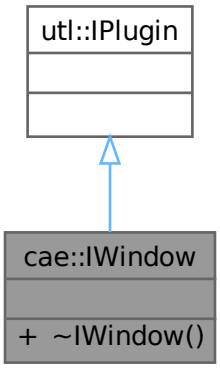
- /home/masina/Projects/Cross-API-Engine/include/CAE/Interfaces/IRenderer.hpp

8.6 cae::IWindow Interface Reference

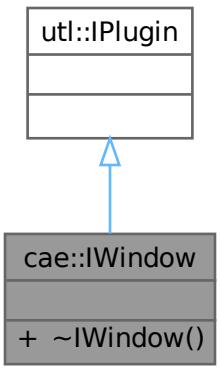
Interface for window.

#include <IWindow.hpp>

Inheritance diagram for cae::IWindow:



Collaboration diagram for cae::IWindow:



Public Member Functions

- ~IWindow () override=default

8.6.1 Detailed Description

Interface for window.

Definition at line 19 of file IWindow.hpp.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 ~IWindow()

cae::IWindow::~IWindow () [override], [default]

The documentation for this interface was generated from the following file:

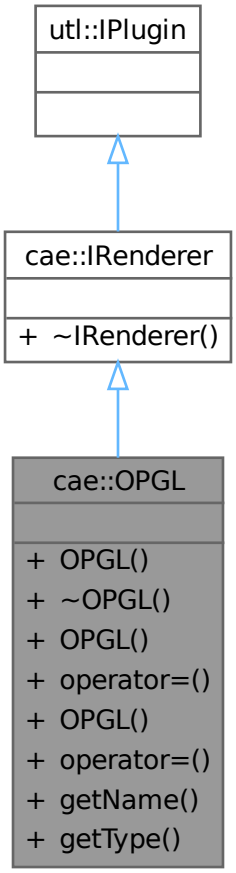
- /home/masina/Projects/Cross-API-Engine/include/CAE/Interfaces/IWindow.hpp

8.7 cae::OPGL Class Reference

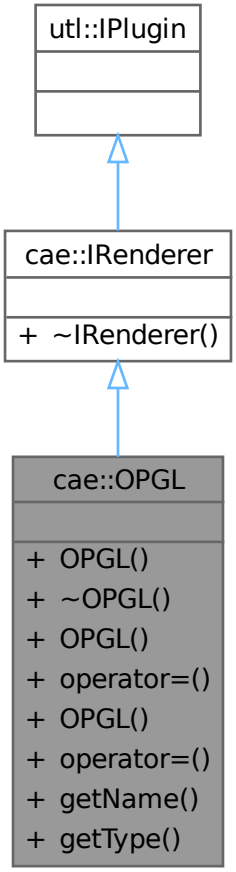
Class for the OpenGL plugin.

#include <OPGL.hpp>

Inheritance diagram for cae::OPGL:



Collaboration diagram for cae::OPGL:



Public Member Functions

- OPGL ()=default
- ~OPGL () override=default
- OPGL (const OPGL &)=delete
- OPGL & operator= (const OPGL &)=delete
- OPGL (OPGL &&)=delete
- OPGL & operator= (OPGL &&)=delete
- std::string getName () const override
- utl::PluginType getType () const override

Public Member Functions inherited from cae::IRenderer

- ~IRenderer () override=default

8.7.1 Detailed Description

Class for the OpenGL plugin.

Definition at line 19 of file OPGL.hpp.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 OPGL() [1/3]

cae::OPGL::OPGL () [default]

8.7.2.2 ~OPGL()

cae::OPGL::~OPGL () [override], [default]

8.7.2.3 OPGL() [2/3]

cae::OPGL::OPGL (
const OPGL &) [delete]

8.7.2.4 OPGL() [3/3]

cae::OPGL::OPGL (
OPGL &&) [delete]

8.7.3 Member Function Documentation

8.7.3.1 getName()

std::string cae::OPGL::getName () const [inline], [nodiscard], [override]

Definition at line 31 of file OPGL.hpp.

8.7.3.2 getType()

utl::PluginType cae::OPGL::getType () const [inline], [nodiscard], [override]

Definition at line 32 of file OPGL.hpp.

8.7.3.3 operator=() [1/2]

OPGL & cae::OPGL::operator= (
const OPGL &) [delete]

8.7.3.4 operator=() [2/2]

OPGL & cae::OPGL::operator= (
OPGL &&) [delete]

The documentation for this class was generated from the following file:

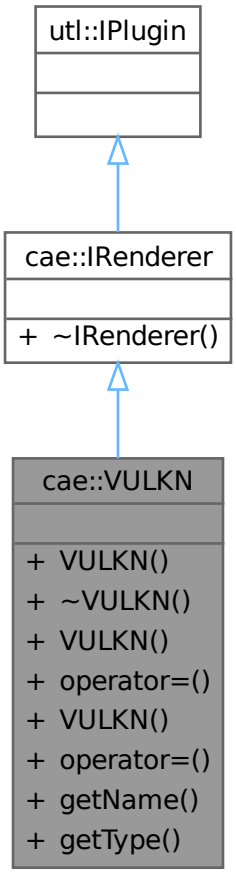
- /home/masina/Projects/Cross-API-Engine/plugins/Renderer/OpenGL/include/OPGL/OPGL.hpp

8.8 cae::VULKN Class Reference

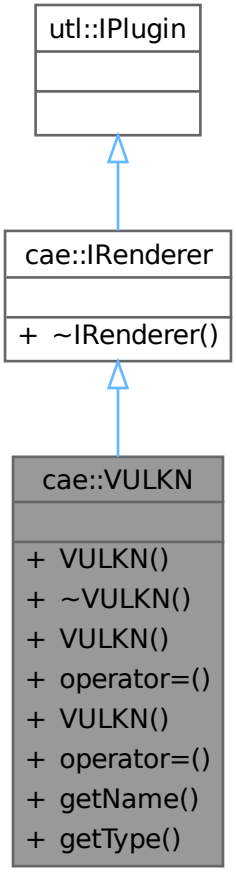
Class for the Vulkan plugin.

#include <VULKN.hpp>

Inheritance diagram for cae::VULKN:



Collaboration diagram for cae::VULKN:



Public Member Functions

- VULKN ()=default
- ~VULKN () override=default
- VULKN (const VULKN &)=delete
- VULKN & operator= (const VULKN &)=delete
- VULKN (VULKN &&)=delete
- VULKN & operator= (VULKN &&)=delete
- std::string getName () const override
- utl::PluginType getType () const override

Public Member Functions inherited from cae::IRenderer

- ~IRenderer () override=default

8.8.1 Detailed Description

Class for the Vulkan plugin.

Definition at line 19 of file VULKN.hpp.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 VULKN() [1/3]

cae::VULKN::VULKN () [default]

8.8.2.2 ~VULKN()

cae::VULKN::~VULKN () [override], [default]

8.8.2.3 VULKN() [2/3]

cae::VULKN::VULKN (
const VULKN &) [delete]

8.8.2.4 VULKN() [3/3]

cae::VULKN::VULKN (
VULKN &&) [delete]

8.8.3 Member Function Documentation

8.8.3.1 getName()

std::string cae::VULKN::getName () const [inline], [nodiscard], [override]

Definition at line 31 of file VULKN.hpp.

8.8.3.2 getType()

utl::PluginType cae::VULKN::getType () const [inline], [nodiscard], [override]

Definition at line 32 of file VULKN.hpp.

8.8.3.3 operator=() [1/2]

VULKN & cae::VULKN::operator= (
const VULKN &) [delete]

8.8.3.4 operator=() [2/2]

VULKN & cae::VULKN::operator= (
VULKN &&) [delete]

The documentation for this class was generated from the following file:

- /home/masina/Projects/Cross-API-Engine/plugins/Renderer/Vulkan/include/VULKN/VULKN.hpp

Chapter 9

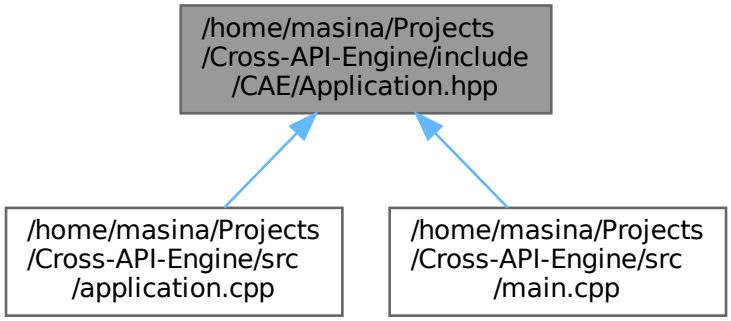
File Documentation

9.1 /home/masina/Projects/Cross-API-Engine/CONTRIBUTING.md File Reference

9.2 /home/masina/Projects/Cross-API-Engine/include/CAE/↩ Application.hpp File Reference

This file contains the Application class declaration.

This graph shows which files directly or indirectly include this file:



Classes

- class cae::Application

  Main class.

Namespaces

- namespace cae

9.2.1 Detailed Description

This file contains the Application class declaration.

Definition in file Application.hpp.

9.3 Application.hpp

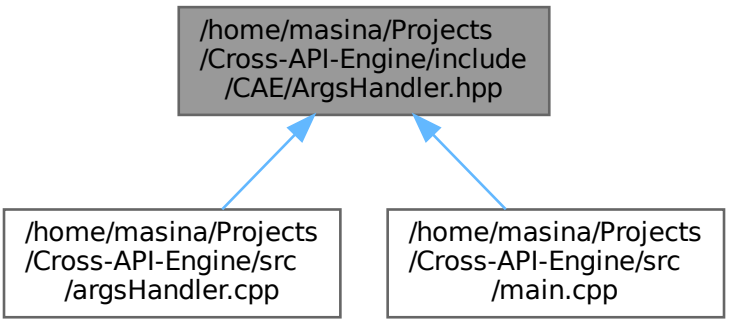
Go to the documentation of this file.

```
00001 ///
00002 /// @file Application.hpp
00003 /// @brief This file contains the Application class declaration
00004 /// @namespace cae
00005 ///
00006
00007 #pragma once
00008
00009 namespace cae
00010 {
00011
00012     ///
00013     /// @class Application
00014     /// @brief Main class
00015     /// @namespace cae
00016     ///
00017     class Application
00018     {
00019
00020         public:
00021             Application();
00022             ~Application() = default;
00023
00024             Application(const Application &) = delete;
00025             Application &operator=(const Application &) = delete;
00026             Application(Application &&) = delete;
00027             Application &operator=(Application &&) = delete;
00028
00029     }; // class Application
00030
00031 } // namespace cae
```

9.4 /home/masina/Projects/Cross-API-Engine/include/CAE/ArgsHandler.hpp File Reference

This file contains the ArgsHandler class declaration.

This graph shows which files directly or indirectly include this file:



Classes

- struct cae::ArgsConfig
- struct cae::EnvConfig
- class cae::ArgsHandler

  Class to handle command line arguments.

Namespaces

- namespace cae

9.4.1 Detailed Description

This file contains the ArgsHandler class declaration.

Definition in file ArgsHandler.hpp.

9.5 ArgsHandler.hpp

Go to the documentation of this file.

```
00001 ///
00002 /// @file ArgsHandler.hpp
00003 /// @brief This file contains the ArgsHandler class declaration
00004 /// @namespace cae
00005 ///
00006
00007 #pragma once
00008
00009 namespace cae
00010 {
00011
00012     struct ArgsConfig
00013     {
00014         bool run{false};
00015     };
00016     struct EnvConfig
00017     {
00018     };
00019
00020     ///
00021     /// @class ArgsHandler
00022     /// @brief Class to handle command line arguments
00023     /// @namespace cae
00024     ///
00025     class ArgsHandler
00026     {
00027
00028         public:
00029             ArgsHandler() = default;
00030             ~ArgsHandler() = default;
00031
00032             ArgsHandler(const ArgsHandler &) = delete;
00033             ArgsHandler &operator=(const ArgsHandler &) = delete;
00034             ArgsHandler(ArgsHandler &&) = delete;
00035             ArgsHandler &operator=(ArgsHandler &&) = delete;
00036
00037             static ArgsConfig ParseArgs(int argc, const char *const *argv);
00038             static EnvConfig ParseEnv(const char *const *envp);
00039
00040         private:
00041     }; // class ArgsHandler
00042
00043 } // namespace cae
```

9.6 /home/masina/Projects/Cross-API-Engine/include/CAE/↩ Common.hpp File Reference

This file contains.

Namespaces

- namespace cae
- namespace cae::Audio
- namespace cae::Network
- namespace cae::User
- namespace cae::Window

Variables

- constexpr auto cae::Audio::VOLUME = 50.F
- constexpr auto cae::Audio::MUTED = false
- constexpr auto cae::Network::HOST = ”127.0.0.1”
- constexpr auto cae::Network::PORT = 4242
- constexpr auto cae::User::NAME = ”User”
- constexpr auto cae::Window::HEIGHT = 1920
- constexpr auto cae::Window::WIDTH = 1080
- constexpr auto cae::Window::NAME = ”CAE - Cross API Engine”
- constexpr auto cae::Window::FULLSCREEN = false
- constexpr auto cae::Window::VSYNC = false
- constexpr auto cae::Window::MAX_FPS = 90

9.6.1 Detailed Description

This file contains.

Definition in file Common.hpp.

9.7 Common.hpp

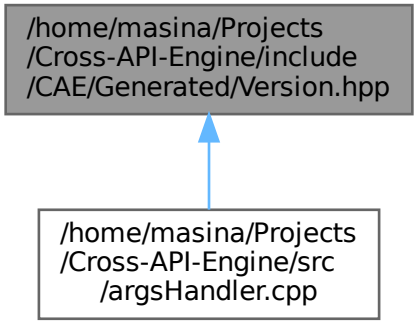
Go to the documentation of this file.

```
00001 ///
00002 /// @file Common.hpp
00003 /// @brief This file contains
00004 /// @namespace cae
00005 ///
00006
00007 #pragma once
00008
00009 namespace cae
00010 {
00011     namespace Audio
00012     {
00013         inline constexpr auto VOLUME = 50.F;
00014         inline constexpr auto MUTED = false;
00015     } // namespace Audio
00016     namespace Network
00017     {
00018         inline constexpr auto HOST = ”127.0.0.1”;
00019         inline constexpr auto PORT = 4242;
00020     } // namespace Network
00021     namespace User
```

```
00022     {
00023         inline constexpr auto NAME = "User";
00024     }
00025     namespace Window
00026     {
00027         inline constexpr auto HEIGHT = 1920;
00028         inline constexpr auto WIDTH = 1080;
00029         inline constexpr auto NAME = "CAE - Cross API Engine";
00030         inline constexpr auto FULLSCREEN = false;
00031         inline constexpr auto VSYNC = false;
00032         inline constexpr auto MAX_FPS = 90;
00033     } // namespace Window
00034 } // namespace cae
```

9.8 /home/masina/Projects/Cross-API-Engine/include/CAE/Generated/Version.hpp File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define PROJECT_NAME "cae"
- #define PROJECT_VERSION "0.0.0"
- #define PROJECT_VERSION_MAJOR "0"
- #define PROJECT_VERSION_MINOR "0"
- #define PROJECT_VERSION_PATCH "0"
- #define GIT_COMMIT_HASH "b72d711"
- #define GIT_TAG "b72d711"
- #define BUILD_TYPE "Release"

9.8.1 Macro Definition Documentation

9.8.1.1 BUILD_TYPE

#define BUILD_TYPE "Release"

Definition at line 15 of file Version.hpp.

9.8.1.2 GIT_COMMIT_HASH

#define GIT_COMMIT_HASH "b72d711"

Definition at line 13 of file Version.hpp.

9.8.1.3 GIT_TAG

#define GIT_TAG "b72d711"

Definition at line 14 of file Version.hpp.

9.8.1.4 PROJECT_NAME

#define PROJECT_NAME "cae"

Definition at line 7 of file Version.hpp.

9.8.1.5 PROJECT_VERSION

#define PROJECT_VERSION "0.0.0"

Definition at line 8 of file Version.hpp.

9.8.1.6 PROJECT_VERSION_MAJOR

#define PROJECT_VERSION_MAJOR "0"

Definition at line 9 of file Version.hpp.

9.8.1.7 PROJECT_VERSION_MINOR

#define PROJECT_VERSION_MINOR "0"

Definition at line 10 of file Version.hpp.

9.8.1.8 PROJECT_VERSION_PATCH

#define PROJECT_VERSION_PATCH "0"

Definition at line 11 of file Version.hpp.

9.9 Version.hpp

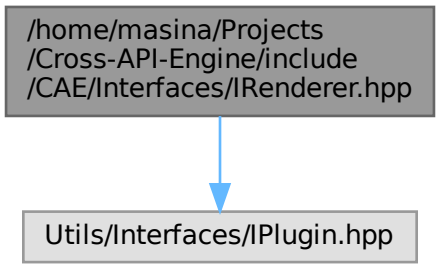
Go to the documentation of this file.

```
00001 #pragma once
00002
00003 //
      ================================================================================
00004 // DO NOT EDIT THIS FILE MANUALLY. IT IS GENERATED BY CMAKE DURING THE BUILD PROCESS.
00005 //
      ================================================================================
00006
00007 #define PROJECT_NAME "cae"
00008 #define PROJECT_VERSION "0.0.0"
00009 #define PROJECT_VERSION_MAJOR "0"
00010 #define PROJECT_VERSION_MINOR "0"
00011 #define PROJECT_VERSION_PATCH "0"
00012
00013 #define GIT_COMMIT_HASH "b72d711"
00014 #define GIT_TAG "b72d711"
00015 #define BUILD_TYPE "Release"
```

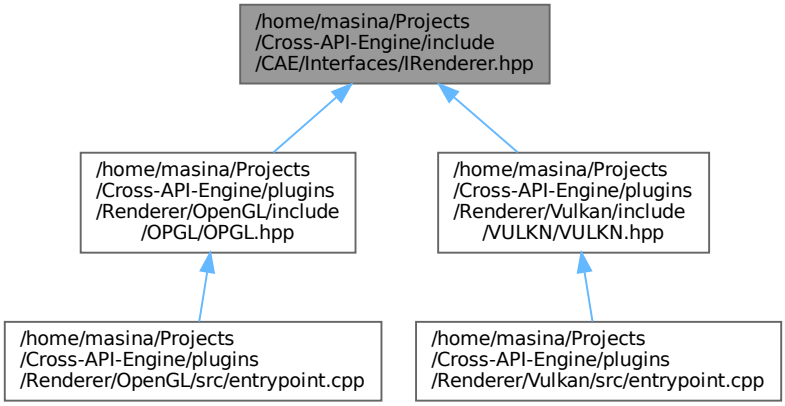
9.10 /home/masina/Projects/Cross-API-Engine/include/CAE/Interfaces/IRenderer.hpp File Reference

This file contains the Renderer interface.

#include "Utils/Interfaces/IPlugin.hpp"
Include dependency graph for IRenderer.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- interface cae::IRenderer

  Interface for renderer.

Namespaces

- namespace cae

9.10.1 Detailed Description

This file contains the Renderer interface.

Definition in file IRenderer.hpp.

9.11 IRenderer.hpp

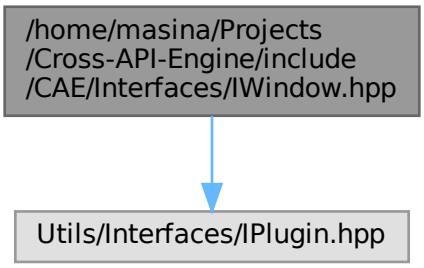
Go to the documentation of this file.

```
00001 ///
00002 /// @file IRenderer.hpp
00003 /// @brief This file contains the Renderer interface
00004 /// @namespace cae
00005 ///
00006
00007 #pragma once
00008
00009 #include "Utils/Interfaces/IPlugin.hpp"
00010
00011 namespace cae
00012 {
00013
00014     ///
00015     /// @interface IRenderer
00016     /// @brief Interface for renderer
00017     /// @namespace cae
00018     ///
00019     class IRenderer : public utl::IPlugin
00020     {
00021
00022         public:
00023             ~IRenderer() override = default;
00024
00025     }; // interface IRenderer
00026
00027 } // namespace cae
```

9.12 /home/masina/Projects/Cross-API-Engine/include/CAE/↩ Interfaces/IWindow.hpp File Reference

This file contains the Window interface.

#include "Utils/Interfaces/IPlugin.hpp"
Include dependency graph for IWindow.hpp:



Classes

- interface cae::IWindow

  Interface for window.

Namespaces

- namespace cae

9.12.1 Detailed Description

This file contains the Window interface.

Definition in file IWindow.hpp.

9.13 IWindow.hpp

Go to the documentation of this file.

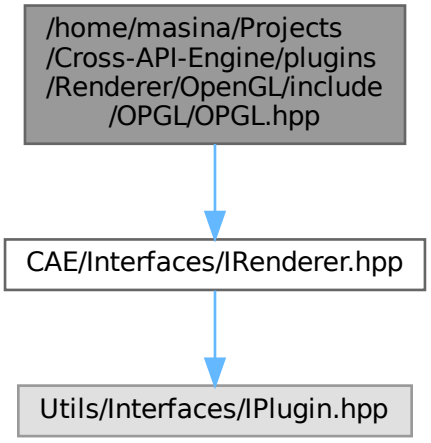
```
00001 ///
00002 /// @file IWindow.hpp
00003 /// @brief This file contains the Window interface
00004 /// @namespace cae
00005 ///
00006
00007 #pragma once
00008
00009 #include "Utils/Interfaces/IPlugin.hpp"
00010
00011 namespace cae
00012 {
```

```
00013
00014     ///
00015     /// @interface IWindow
00016     /// @brief Interface for window
00017     /// @namespace cae
00018     ///
00019     class IWindow : public utl::IPlugin
00020     {
00021
00022         public:
00023             ~IWindow() override = default;
00024
00025     }; // interface IWindow
00026
00027 } // namespace cae
```

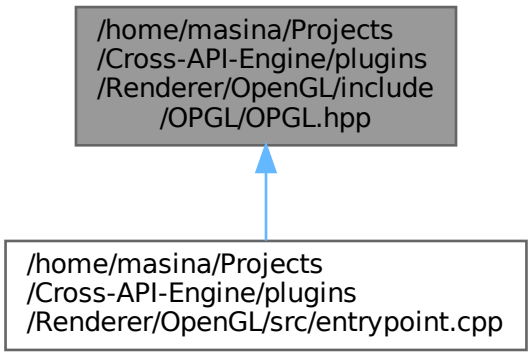
9.14 /home/masina/Projects/Cross-API-Engine/plugins/Renderer/OpenGL/include/OPGL/OPGL.hpp File Reference

This file contains the OPGL class declaration.

#include "CAE/Interfaces/IRenderer.hpp"
Include dependency graph for OPGL.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class cae::OPGL

  Class for the OpenGL plugin.

Namespaces

- namespace cae

9.14.1 Detailed Description

This file contains the OPGL class declaration.

Definition in file OPGL.hpp.

9.15 OPGL.hpp

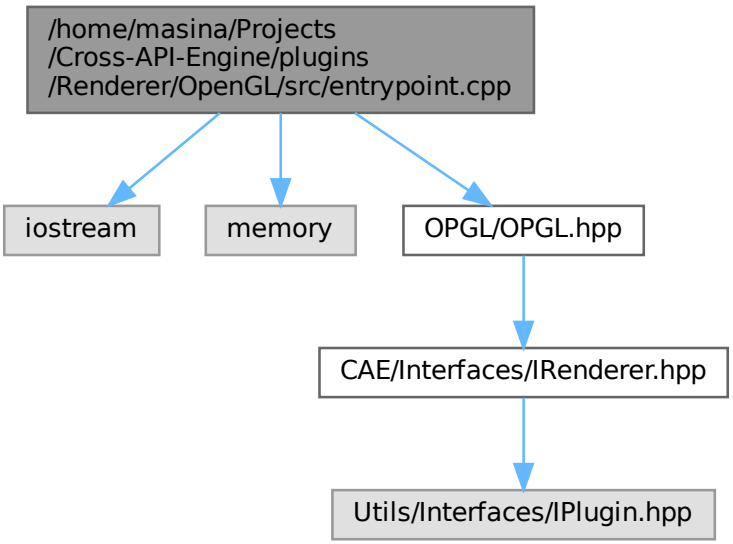
Go to the documentation of this file.

```
00001 ///
00002 /// @file OPGL.hpp
00003 /// @brief This file contains the OPGL class declaration
00004 /// @namespace cae
00005 ///
00006
00007 #pragma once
00008
00009 #include "CAE/Interfaces/IRenderer.hpp"
00010
00011 namespace cae
00012 {
00013
00014     ///
00015     /// @class OPGL
00016     /// @brief Class for the OpenGL plugin
00017     /// @namespace cae
00018     ///
00019     class OPGL final : public IRenderer
```

```
00020     {
00021
00022         public:
00023             OPGL() = default;
00024             ~OPGL() override = default;
00025
00026             OPGL(const OPGL &) = delete;
00027             OPGL &operator=(const OPGL &) = delete;
00028             OPGL(OPGL &&) = delete;
00029             OPGL &operator=(OPGL &&) = delete;
00030
00031             [[nodiscard]] std::string getName() const override { return "OpenGL"; }
00032             [[nodiscard]] utl::PluginType getType() const override { return utl::PluginType::RENDERER; }
00033
00034     }; // class OPGL
00035
00036 } // namespace cae
```

9.16 /home/masina/Projects/Cross-API-Engine/plugins/Renderer/OpenGL/src/entrypoint.cpp File Reference

#include <iostream>
#include <memory>
#include "OPGL/OPGL.hpp"
Include dependency graph for entrypoint.cpp:



Functions

- cae::IRenderer * entryPoint ()

9.16.1 Function Documentation

9.16.1.1 entryPoint()

cae::IRenderer * entryPoint ()

Definition at line 8 of file entrypoint.cpp.

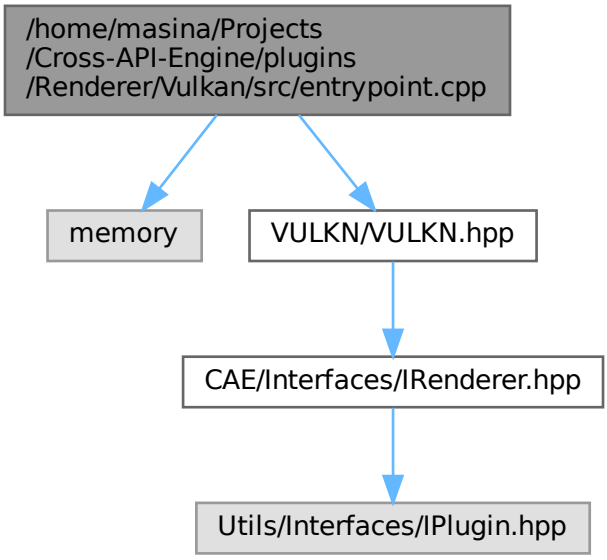
9.17 entrypoint.cpp

Go to the documentation of this file.

```
00001 #include <iostream>
00002 #include <memory>
00003
00004 #include "OPGL/OPGL.hpp"
00005
00006 extern "C"
00007 {
00008     cae::IRenderer *entryPoint() { return std::make_unique<cae::OPGL>().release(); }
00009 }
```

9.18 /home/masina/Projects/Cross-API-Engine/plugins/Renderer/Vulkan/src/entrypoint.cpp File Reference

#include <memory>
#include "VULKN/VULKN.hpp"
Include dependency graph for entrypoint.cpp:



Functions

- cae::IRenderer * entryPoint ()

9.18.1 Function Documentation

9.18.1.1 entryPoint()

cae::IRenderer * entryPoint ()

Definition at line 7 of file entrypoint.cpp.

9.19 entrypoint.cpp

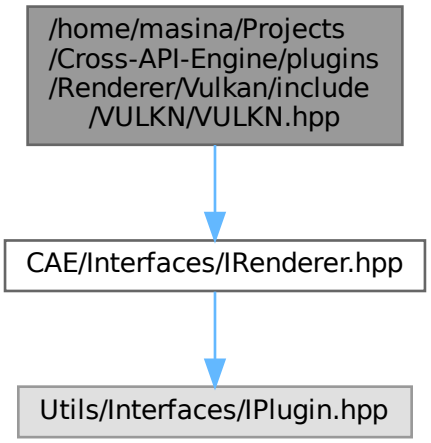
Go to the documentation of this file.

```
00001 #include <memory>
00002
00003 #include "VULKN/VULKN.hpp"
00004
00005 extern "C"
00006 {
00007     cae::IRenderer *entryPoint() { return std::make_unique<cae::VULKN>().release(); }
00008 }
```

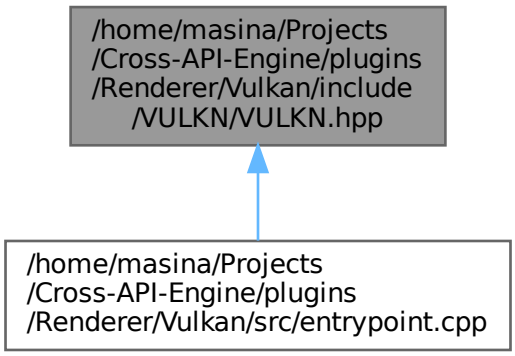
9.20 /home/masina/Projects/Cross-API-Engine/plugins/Renderer/Vulkan/include/VULKN/VULKN.hpp File Reference

This file contains the VULKN class declaration.

#include "CAE/Interfaces/IRenderer.hpp"
Include dependency graph for VULKN.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class cae::VULKN
  Class for the Vulkan plugin.

Namespaces

- namespace cae

9.20.1 Detailed Description

This file contains the VULKN class declaration.

Definition in file VULKN.hpp.

9.21 VULKN.hpp

Go to the documentation of this file.

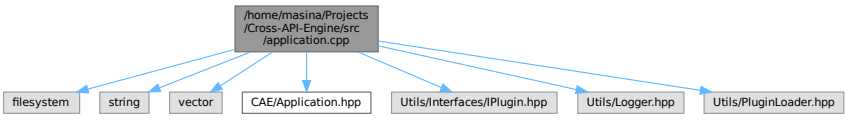
```
00001 ///
00002 /// @file VULKN.hpp
00003 /// @brief This file contains the VULKN class declaration
00004 /// @namespace cae
00005 ///
00006
00007 #pragma once
00008
00009 #include "CAE/Interfaces/IRenderer.hpp"
00010
00011 namespace cae
00012 {
00013
00014     ///
00015     /// @class VULKN
00016     /// @brief Class for the Vulkan plugin
00017     /// @namespace cae
00018     ///
00019     class VULKN final : public IRenderer
```

```
00020     {
00021
00022         public:
00023             VULKN() = default;
00024             ~VULKN() override = default;
00025
00026             VULKN(const VULKN &) = delete;
00027             VULKN &operator=(const VULKN &) = delete;
00028             VULKN(VULKN &&) = delete;
00029             VULKN &operator=(VULKN &&) = delete;
00030
00031             [[nodiscard]] std::string getName() const override { return "Vulkan"; }
00032             [[nodiscard]] utl::PluginType getType() const override { return utl::PluginType::RENDERER; }
00033
00034     }; // class VULKN
00035
00036 } // namespace cae
```

9.22 /home/masina/Projects/Cross-API-Engine/README.md File Reference

9.23 /home/masina/Projects/Cross-API-Engine/src/application.cpp File Reference

#include <filesystem>
#include <string>
#include <vector>
#include "CAE/Application.hpp"
#include "Utils/Interfaces/IPlugin.hpp"
#include "Utils/Logger.hpp"
#include "Utils/PluginLoader.hpp"
Include dependency graph for application.cpp:



Macros

- #define PLUGINS_EXTENSION ".so"

9.23.1 Macro Definition Documentation

9.23.1.1 PLUGINS_EXTENSION

#define PLUGINS_EXTENSION ".so"

Definition at line 9 of file application.cpp.

Referenced by cae::Application::Application().

9.24 application.cpp

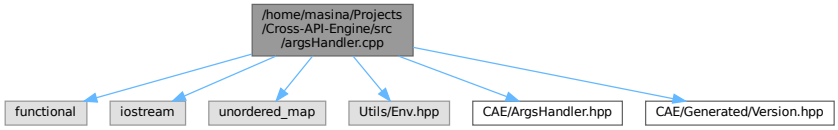
Go to the documentation of this file.

```
00001 #include <filesystem>
00002 #include <string>
00003 #include <vector>
00004
00005 #ifdef _WIN32
00006 #include <windows.h>
00007 #define PLUGINS_EXTENSION ".dll"
00008 #else
00009 #define PLUGINS_EXTENSION ".so"
00010 #endif
00011
00012 #include "CAE/Application.hpp"
00013 #include "Utils/Interfaces/IPlugin.hpp"
00014 #include "Utils/Logger.hpp"
00015 #include "Utils/PluginLoader.hpp"
00016
00017 cae::Application::Application()
00018 {
00019     try
00020     {
00021         const std::filesystem::path pluginDir{PLUGINS_DIR};
00022         utl::PluginLoader pluginManager;
00023         std::vector<std::string> loadedPlugins;
00024         for (const auto &entry : std::filesystem::directory_iterator(pluginDir))
00025         {
00026             if (!entry.is_regular_file() || entry.path().extension() != PLUGINS_EXTENSION)
00027             {
00028                 continue;
00029             }
00030             if (const std::string pluginPath = entry.path().string();
00031                 pluginManager.loadPlugin<utl::IPlugin>(pluginPath) != nullptr)
00032             {
00033                 loadedPlugins.push_back(entry.path().filename().string());
00034             }
00035             else
00036             {
00037                 utl::Logger::log("Failed to load plugin: " + pluginPath, utl::LogLevel::WARNING);
00038             }
00039         }
00040         if (loadedPlugins.empty())
00041         {
00042             utl::Logger::log("No plugins loaded from directory: " + pluginDir.string(), utl::LogLevel::WARNING);
00043         }
00044     }
00045     catch (const std::exception &e)
00046     {
00047         std::cerr « "Error: " « e.what() « '\n';
00048     }
00049 }
```

9.25 /home/masina/Projects/Cross-API-Engine/src/argsHandler.cpp File Reference

```
#include <functional>
#include <iostream>
#include <unordered_map>
#include "Utils/Env.hpp"
#include "CAE/ArgsHandler.hpp"
#include "CAE/Generated/Version.hpp"
```

Include dependency graph for argsHandler.cpp:



Macros

- #define APP_EXTENSION ""

Variables

- static constexpr std::string_view HELP_MSG
- static constexpr std::string_view VERSION_MSG
- static const std::unordered_map< std::string, std::function< void()> > ARGS_MAP

9.25.1 Macro Definition Documentation

9.25.1.1 APP_EXTENSION

#define APP_EXTENSION ""

Definition at line 8 of file argsHandler.cpp.

9.25.2 Variable Documentation

9.25.2.1 ARGS_MAP

const std::unordered_map<std::string, std::function<void()> > ARGS_MAP [static]

Initial value:
```
= {
    {"-h", []() { std::cout « HELP_MSG; }},
    {"--help", []() { std::cout « HELP_MSG; }},
    {"-v", []() { std::cout « VERSION_MSG; }},
    {"--version", []() { std::cout « VERSION_MSG; }}}
```

Definition at line 24 of file argsHandler.cpp.

Referenced by cae::ArgsHandler::ParseArgs().

9.25.2.2 HELP_MSG

std::string_view HELP_MSG [static], [constexpr]

Initial value:
```
= "Usage: " PROJECT_NAME APP_EXTENSION " [options]\n\n"
                                    "Options:\n"
                                    "  -h, --help       Show this help message\n"
                                    "  -v, --version    Show version information\n"
```

Definition at line 16 of file argsHandler.cpp.

9.25.2.3 VERSION_MSG

std::string_view VERSION_MSG [static], [constexpr]

Initial value:
```
= PROJECT_NAME
    " v" PROJECT_VERSION " " BUILD_TYPE " (" GIT_TAG ", commit " GIT_COMMIT_HASH ") " __DATE__ " "
        __TIME__ "\n"
```

Definition at line 21 of file argsHandler.cpp.

9.26 argsHandler.cpp

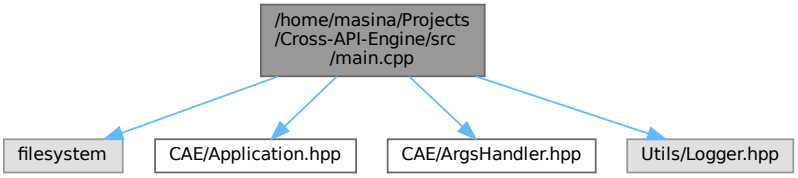
Go to the documentation of this file.

```
00001 #include <functional>
00002 #include <iostream>
00003 #include <unordered_map>
00004
00005 #ifdef _WIN32
00006 #define APP_EXTENSION ".exe"
00007 #else
00008 #define APP_EXTENSION ""
00009 #endif
00010
00011 #include "Utils/Env.hpp"
00012
00013 #include "CAE/ArgsHandler.hpp"
00014 #include "CAE/Generated/Version.hpp"
00015
00016 static constexpr std::string_view HELP_MSG = "Usage: " PROJECT_NAME APP_EXTENSION " [options]\n\n"
00017                                              "Options:\n"
00018                                              "  -h, --help       Show this help message\n"
00019                                              "  -v, --version    Show version information\n";
00020
00021 static constexpr std::string_view VERSION_MSG = PROJECT_NAME
00022     " v" PROJECT_VERSION " " BUILD_TYPE " (" GIT_TAG ", commit " GIT_COMMIT_HASH ") " __DATE__ "
      " __TIME__ "\n";
00023
00024 static const std::unordered_map<std::string, std::function<void()» ARGS_MAP = {
00025     {"-h", []() { std::cout « HELP_MSG; }},
00026     {"--help", []() { std::cout « HELP_MSG; }},
00027     {"-v", []() { std::cout « VERSION_MSG; }},
00028     {"--version", []() { std::cout « VERSION_MSG; }}};
00029
00030 cae::ArgsConfig cae::ArgsHandler::ParseArgs(const int argc, const char *const *argv)
00031 {
00032     if (argc <= 1)
00033     {
00034         return {.run = true};
00035     }
00036     const std::string arg1{argv[1]};
00037     if (const auto it = ARGS_MAP.find(arg1); it != ARGS_MAP.end())
00038     {
00039         it->second();
00040         return {.run = false};
00041     }
00042     throw std::runtime_error("Unknown argument: " + arg1 + ". Use -h or --help to see available options.");
00043 }
00044
00045 cae::EnvConfig cae::ArgsHandler::ParseEnv(const char *const *envp)
00046 {
00047     for (const auto &[fst, snd] : utl::getEnvMap(envp))
00048     {
00049         std::cout « "var:" « fst « ":" « snd « '\n';
00050     }
00051     return {};
00052 }
```

9.27 /home/masina/Projects/Cross-API-Engine/src/main.cpp File Reference

```
#include <filesystem>
#include "CAE/Application.hpp"
#include "CAE/ArgsHandler.hpp"
#include "Utils/Logger.hpp"
```

Include dependency graph for main.cpp:



Functions

- int main (const int argc, const char *const *argv, const char *const *envp)

9.27.1 Function Documentation

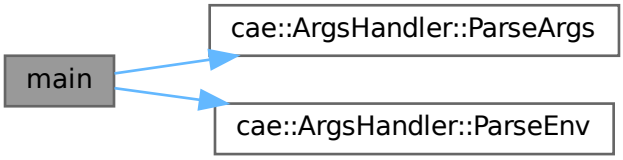
9.27.1.1 main()

int main (
const int argc,
const char *const * argv,
const char *const * envp)

Definition at line 7 of file main.cpp.

References cae::ArgsHandler::ParseArgs(), and cae::ArgsHandler::ParseEnv().

Here is the call graph for this function:



9.28 main.cpp

Go to the documentation of this file.

```
00001 #include <filesystem>
00002
00003 #include "CAE/Application.hpp"
00004 #include "CAE/ArgsHandler.hpp"
00005 #include "Utils/Logger.hpp"
00006
00007 int main(const int argc, const char *const *argv, const char *const *envp)
00008 {
00009     utl::Logger::init();
00010     cae::ArgsHandler argsHandler{};
00011     try
00012     {
00013         auto [run] = cae::ArgsHandler::ParseArgs(argc, argv);
00014         auto env = cae::ArgsHandler::ParseEnv(envp);
00015         if (!run)
00016         {
00017             return EXIT_SUCCESS;
00018         }
00019         cae::Application app;
00020     }
00021     catch (const std::exception &e)
00022     {
00023         std::cerr « "Error: " « e.what() « '\n';
00024         return EXIT_FAILURE;
00025     }
00026     return EXIT_SUCCESS;
00027 }
```

Index

/home/masina/Projects/Cross-API-Engine/CONTRIBUTING.md, 33
/home/masina/Projects/Cross-API-Engine/README.md, 48
/home/masina/Projects/Cross-API-Engine/include/CAE/Application.hpp, 33, 34
/home/masina/Projects/Cross-API-Engine/include/CAE/ArgsHandler.hpp, 34, 35
/home/masina/Projects/Cross-API-Engine/include/CAE/Common.hpp, 36
/home/masina/Projects/Cross-API-Engine/include/CAE/Generated/Version.hpp, 37, 39
/home/masina/Projects/Cross-API-Engine/include/CAE/Interfaces/IRenderer.hpp, 39, 40
/home/masina/Projects/Cross-API-Engine/include/CAE/Interfaces/IWindow.hpp, 41
/home/masina/Projects/Cross-API-Engine/plugins/Renderer/OpenGL/include/OPGL/OPGL.hpp, 42, 43
/home/masina/Projects/Cross-API-Engine/plugins/Renderer/OpenGL/src/entrypoint.cpp, 44, 45
/home/masina/Projects/Cross-API-Engine/plugins/Renderer/Vulkan/include/VULKN/VULKN.hpp, 46, 47
/home/masina/Projects/Cross-API-Engine/plugins/Renderer/Vulkan/src/entrypoint.cpp, 45, 46
/home/masina/Projects/Cross-API-Engine/src/application.cpp, 48, 49
/home/masina/Projects/Cross-API-Engine/src/argsHandler.cpp, 49, 51
/home/masina/Projects/Cross-API-Engine/src/main.cpp, 51, 53
~Application
  cae::Application, 18
~ArgsHandler
  cae::ArgsHandler, 21
~IRenderer
  cae::IRenderer, 24
~IWindow
  cae::IWindow, 26
~OPGL
  cae::OPGL, 28
~VULKN
  cae::VULKN, 31

APP_EXTENSION
  argsHandler.cpp, 50
Application
  cae::Application, 18
application.cpp
  PLUGINS_EXTENSION, 48
ARGS_MAP
  argsHandler.cpp, 50
ArgsHandler
  cae::ArgsHandler, 21
argsHandler.cpp
  APP_EXTENSION, 50
  ARGS_MAP, 50
  HELP_MSG, 50
  VERSION_MSG, 50
BUILD_TYPE
  Version.hpp, 37

cae, 1, 13
cae::Application, 17
  ~Application, 18
  Application, 18
  operator=, 18
cae::ArgsConfig, 19
  run, 19
cae::ArgsHandler, 20
  ~ArgsHandler, 21
  ArgsHandler, 21
  operator=, 21
  ParseArgs, 21
  ParseEnv, 21
cae::Audio, 13
  MUTED, 14
  VOLUME, 14
cae::EnvConfig, 22
cae::IRenderer, 23
  ~IRenderer, 24
cae::IWindow, 25
  ~IWindow, 26
cae::Network, 14
  HOST, 14
  PORT, 14
cae::OPGL, 26
  ~OPGL, 28
  getName, 28
  getType, 28
  operator=, 28
  OPGL, 28
cae::User, 14
  NAME, 14
cae::VULKN, 29
  ~VULKN, 31
  getName, 31
  getType, 31
  operator=, 31
  VULKN, 31

cae::Window, 15
FULLSCREEN, 15
HEIGHT, 15
MAX_FPS, 15
NAME, 15
VSYNC, 15
WIDTH, 15
Commit Norms, 3

entryPoint
entrypoint.cpp, 44, 45
entrypoint.cpp
entryPoint, 44, 45

FULLSCREEN
cae::Window, 15

getName
cae::OPGL, 28
cae::VULKN, 31
getType
cae::OPGL, 28
cae::VULKN, 31
GIT_COMMIT_HASH
Version.hpp, 37
GIT_TAG
Version.hpp, 38

HEIGHT
cae::Window, 15
HELP_MSG
argsHandler.cpp, 50
HOST
cae::Network, 14

main
main.cpp, 52
main.cpp
main, 52
MAX_FPS
cae::Window, 15
MUTED
cae::Audio, 14

NAME
cae::User, 14
cae::Window, 15

operator=
cae::Application, 18
cae::ArgsHandler, 21
cae::OPGL, 28
cae::VULKN, 31
OPGL
cae::OPGL, 28

ParseArgs
cae::ArgsHandler, 21
ParseEnv
cae::ArgsHandler, 21
PLUGINS_EXTENSION
application.cpp, 48
PORT
cae::Network, 14
PROJECT_NAME
Version.hpp, 38
PROJECT_VERSION
Version.hpp, 38
PROJECT_VERSION_MAJOR
Version.hpp, 38
PROJECT_VERSION_MINOR
Version.hpp, 38
PROJECT_VERSION_PATCH
Version.hpp, 38

run
cae::ArgsConfig, 19

Version.hpp
BUILD_TYPE, 37
GIT_COMMIT_HASH, 37
GIT_TAG, 38
PROJECT_NAME, 38
PROJECT_VERSION, 38
PROJECT_VERSION_MAJOR, 38
PROJECT_VERSION_MINOR, 38
PROJECT_VERSION_PATCH, 38
VERSION_MSG
argsHandler.cpp, 50
VOLUME
cae::Audio, 14
VSYNC
cae::Window, 15
VULKN
cae::VULKN, 31

WIDTH
cae::Window, 15