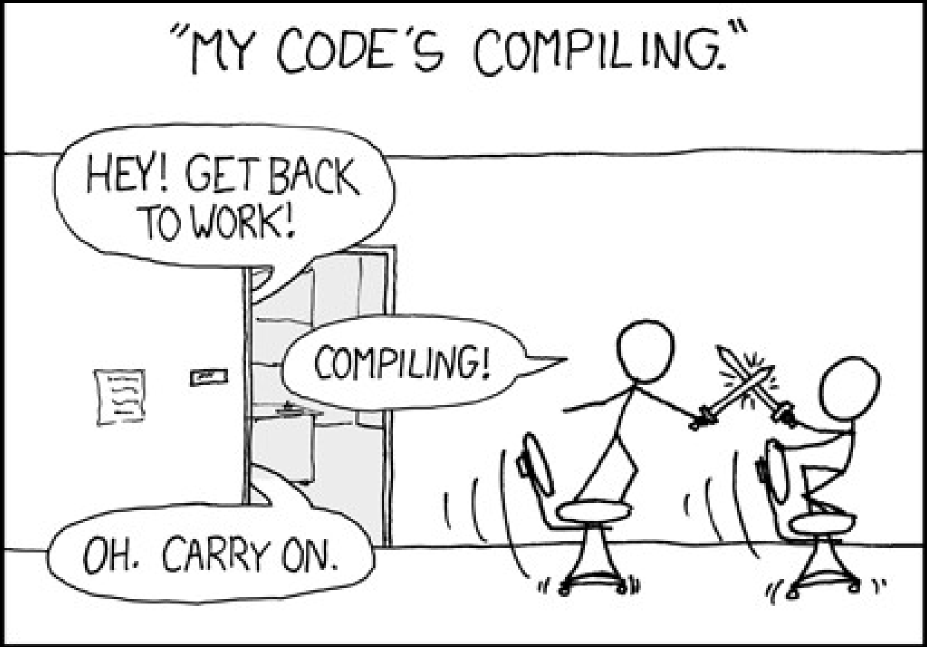
CPPINCLUDE

2020, internal talk

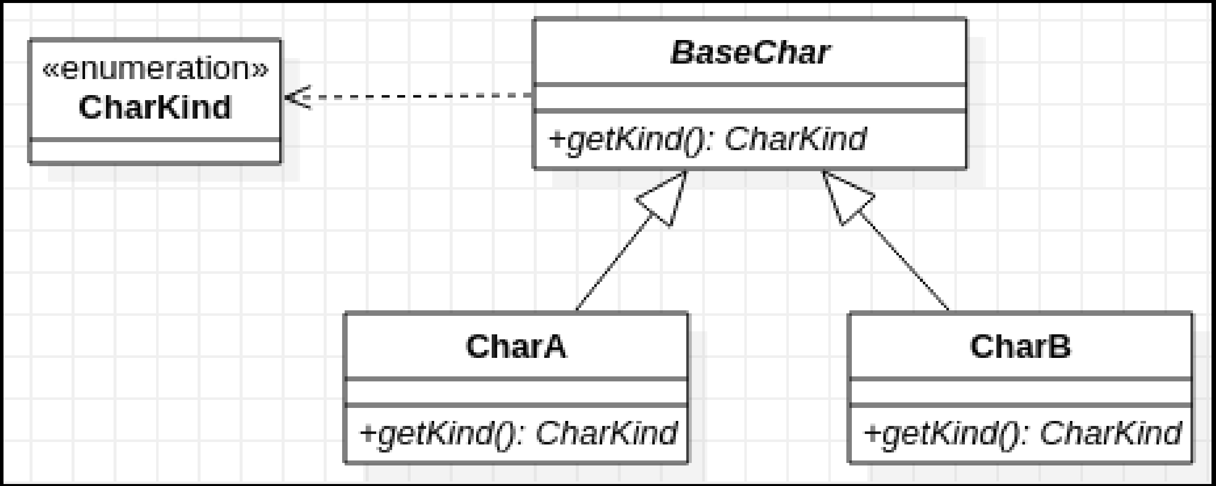
KEEP CALM and COMPILE

> touch big_header_that_include_in_all_files.hpp

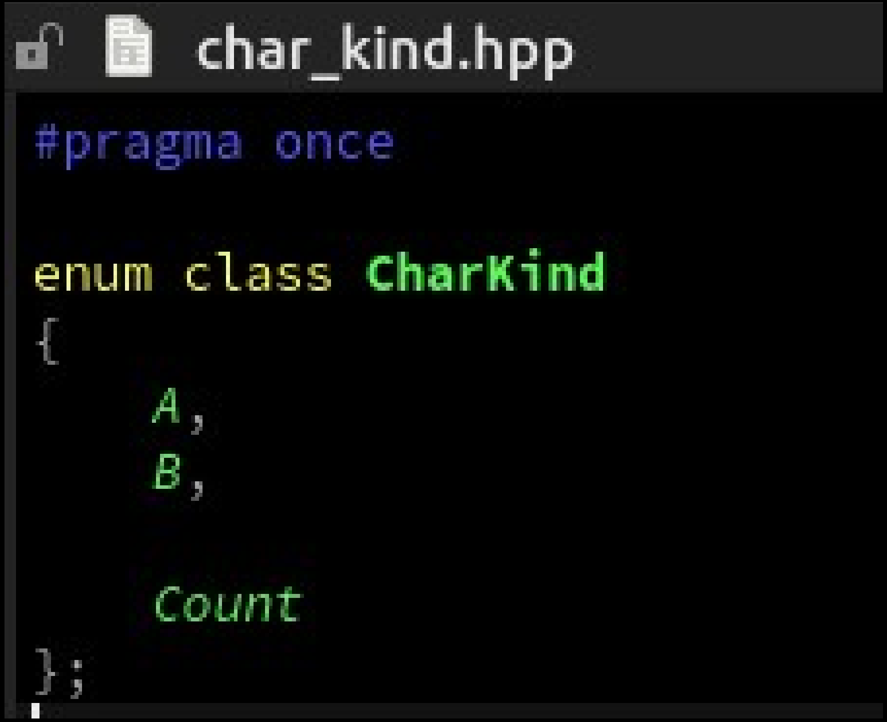


Simple example. Objects

Example from *<project_folder>/docs/examples/simple_example/*



Simple example. *BaseChar*

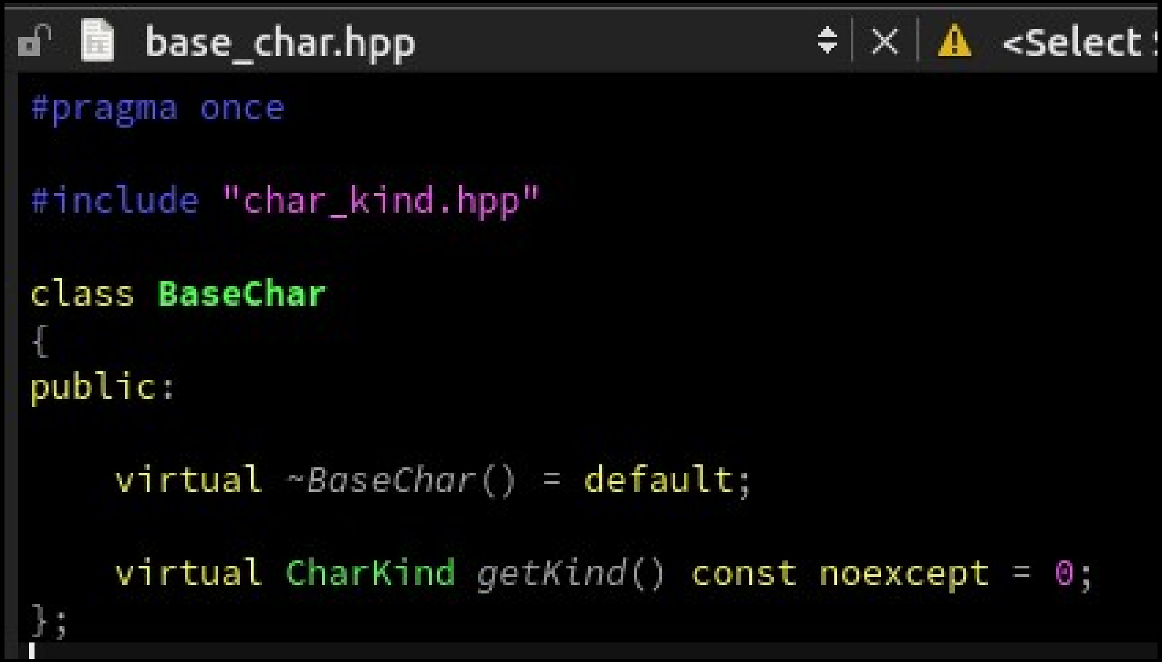


char_kind.hpp

```
#pragma once

enum class CharKind
{
    A,
    B,

    Count
};
```

Example from *cppinclude/docs/examples/simple_example/*

Simple example. *BaseChar*

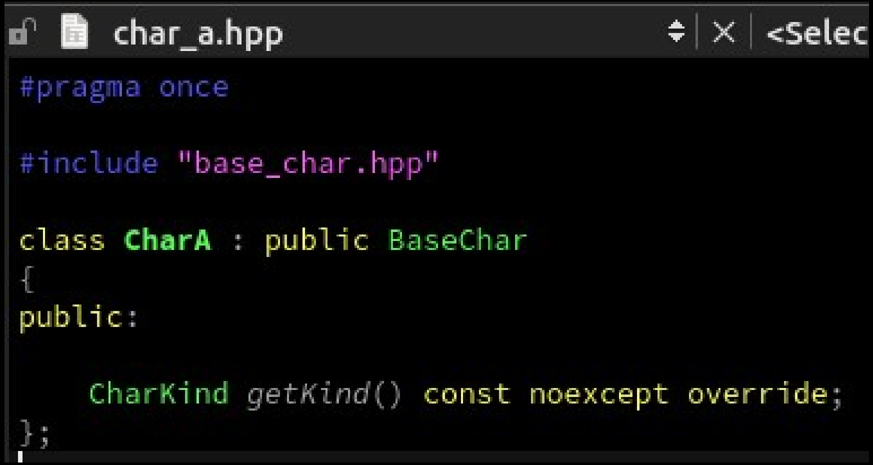


```
#pragma once

#include "char_kind.hpp"

class BaseChar
{
public:

    virtual ~BaseChar() = default;

    virtual CharKind getKind() const noexcept = 0;
};
```

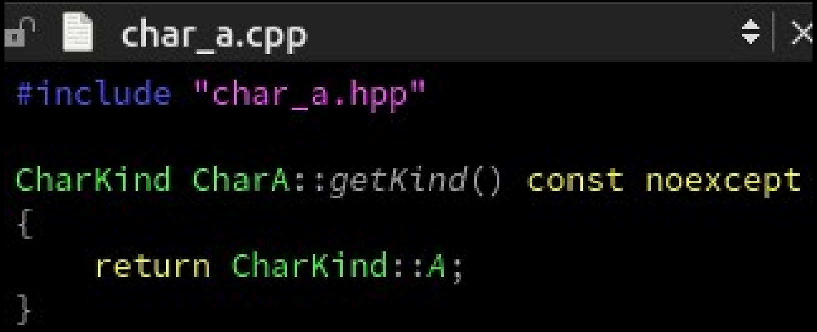
Example from *cppinclude/docs/examples/simple_example/*

Simple example. *CharA*

Example from *cppinclude/docs/examples/simple_example/*



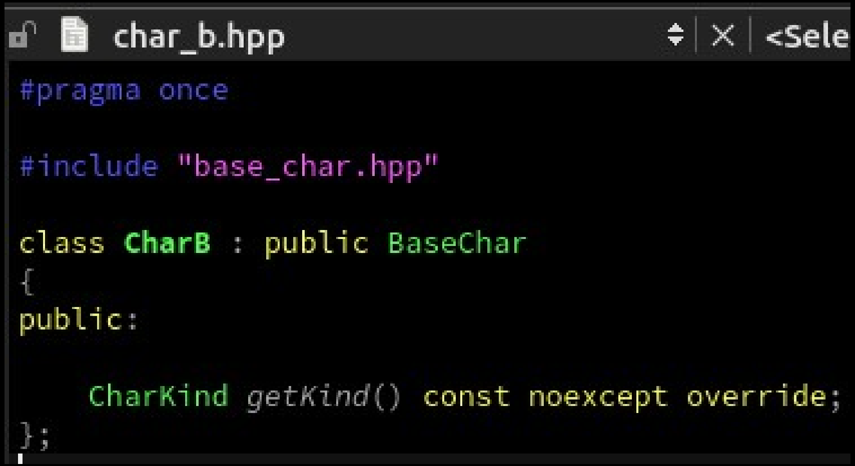
```
#pragma once

#include "base_char.hpp"

class CharA : public BaseChar
{
public:

    CharKind getKind() const noexcept override;
};
```



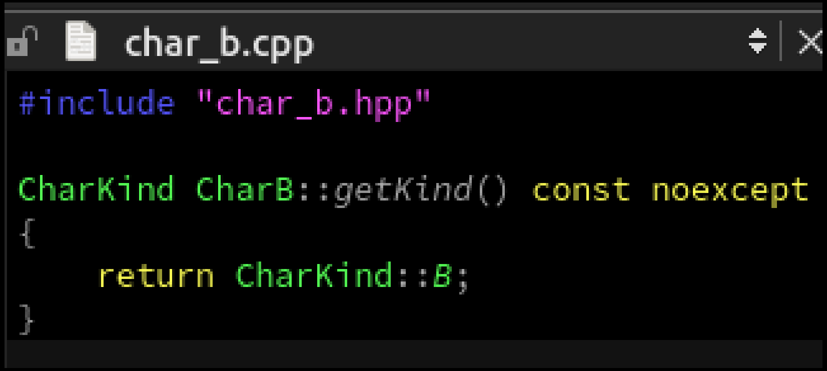
```
#include "char_a.hpp"

CharKind CharA::getKind() const noexcept
{
    return CharKind::A;
}
```

Simple example. *CharB*

Example from *cppinclude/docs/examples/simple_example/*



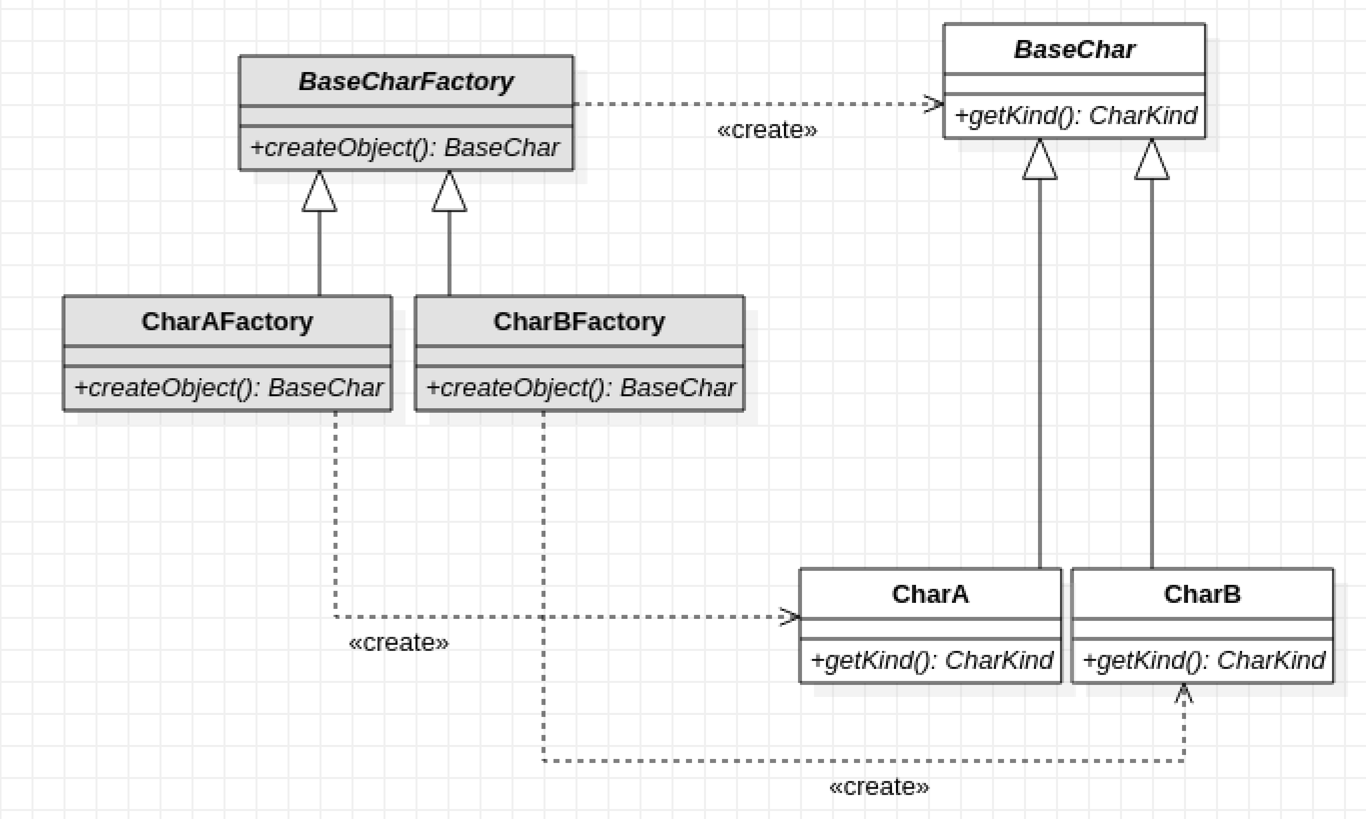
```cpp
#pragma once

#include "base_char.hpp"

class CharB : public BaseChar
{
public:

    CharKind getKind() const noexcept override;
};
```



```cpp
#include "char_b.hpp"

CharKind CharB::getKind() const noexcept
{
    return CharKind::B;
}
```

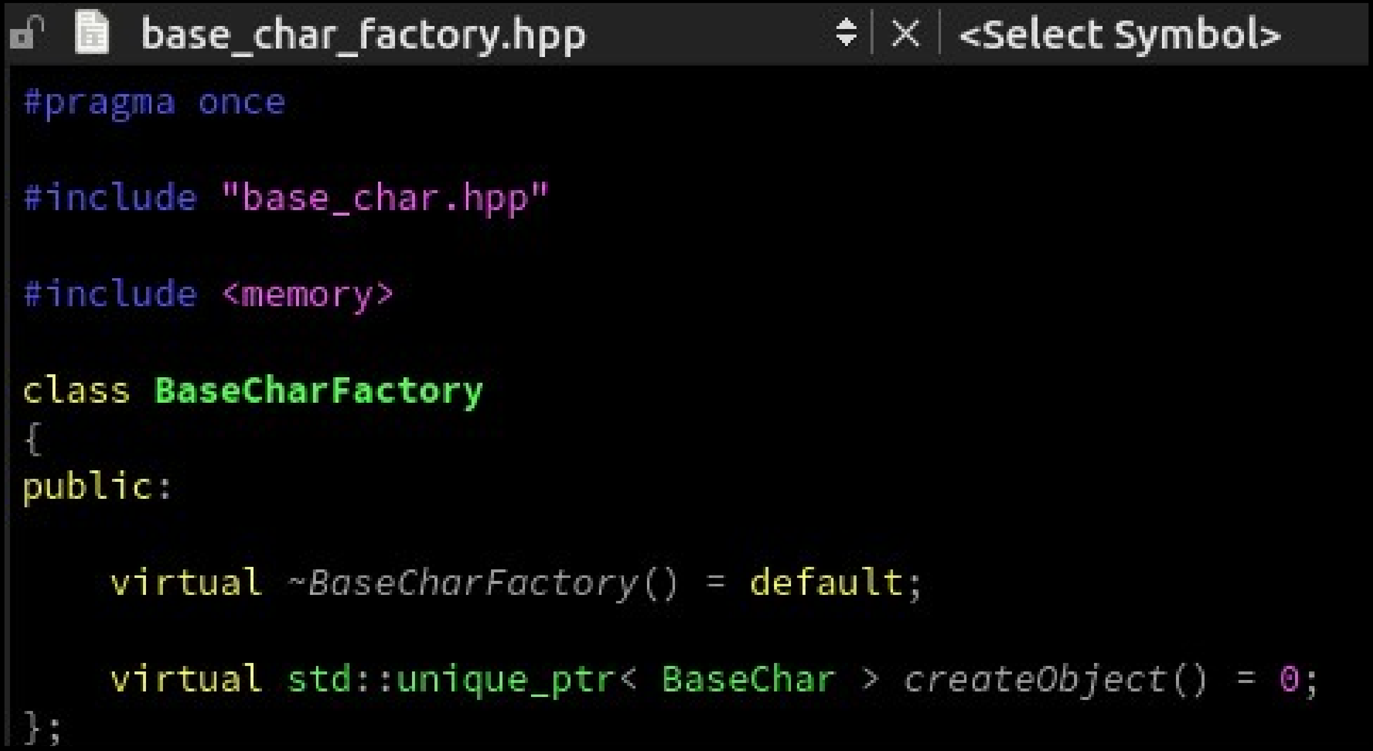
Simple example. *Factories*

Example from *cppinclude/docs/examples/simple_example/*



Simple example. *BaseCharFactory*

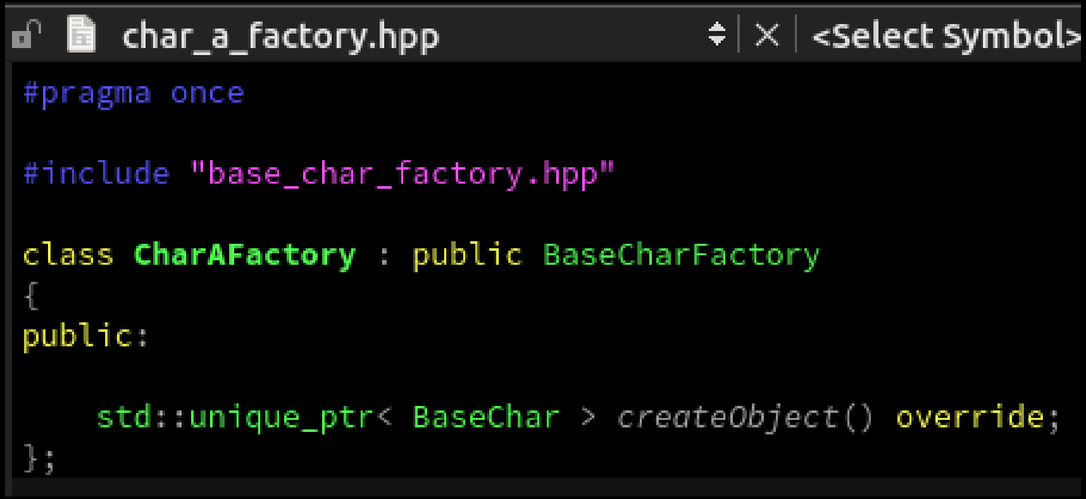
Example from *cppinclude/docs/examples/simple_example/*



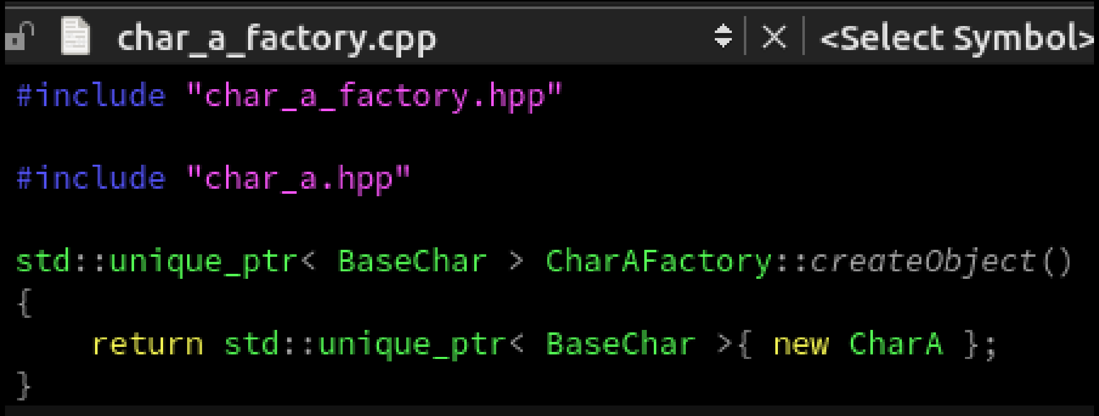
```
#pragma once

#include "base_char.hpp"

#include <memory>

class BaseCharFactory
{
public:

    virtual ~BaseCharFactory() = default;

    virtual std::unique_ptr< BaseChar > createObject() = 0;
};
```

Simple example. *CharAFactory*

Example from *cppinclude/docs/examples/simple_example/*



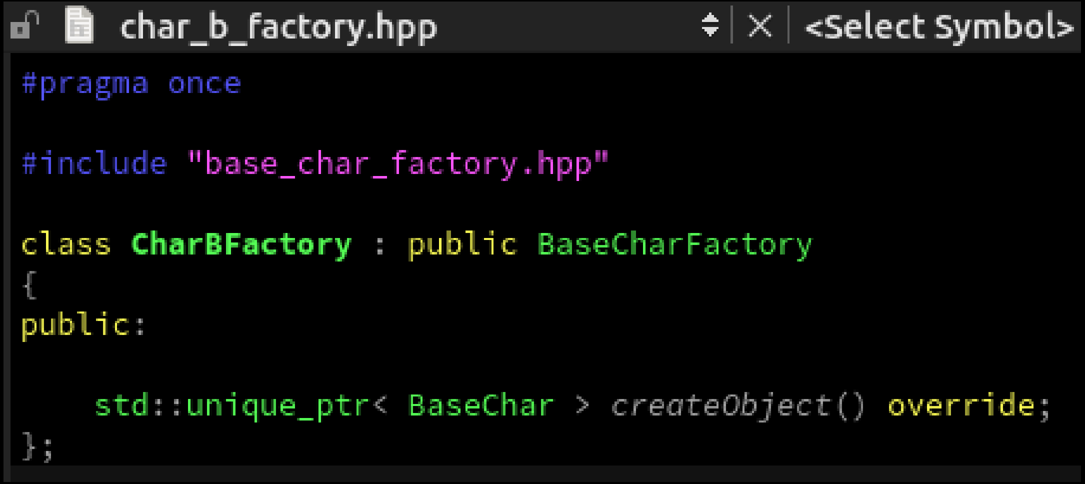
```cpp
#pragma once

#include "base_char_factory.hpp"

class CharAFactory : public BaseCharFactory
{
public:

    std::unique_ptr< BaseChar > createObject() override;
};
```



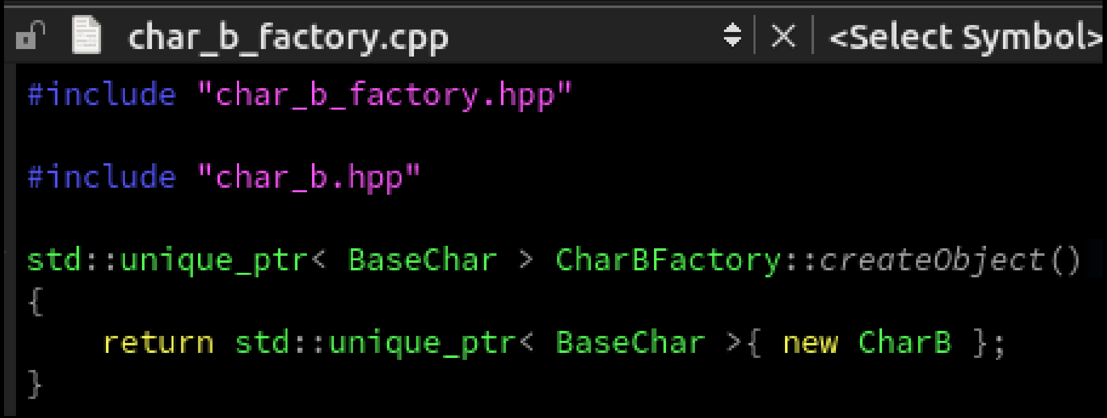
```cpp
#include "char_a_factory.hpp"

#include "char_a.hpp"

std::unique_ptr< BaseChar > CharAFactory::createObject()
{
    return std::unique_ptr< BaseChar >{ new CharA };
}
```

Simple example. *CharBFactory*

Example from *cppinclude/docs/examples/simple_example/*



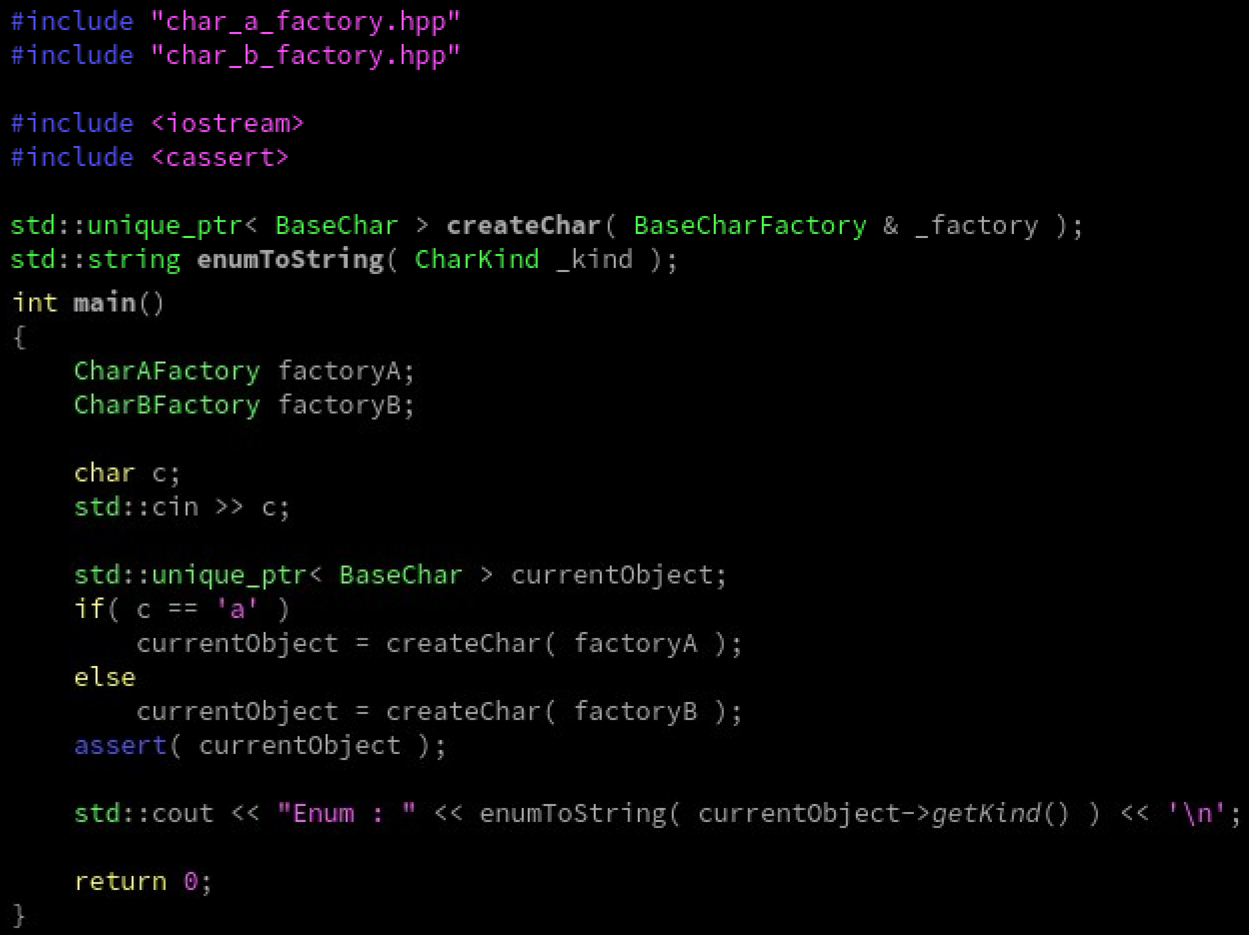
```cpp
#pragma once

#include "base_char_factory.hpp"

class CharBFactory : public BaseCharFactory
{
public:

    std::unique_ptr< BaseChar > createObject() override;
};
```



```cpp
#include "char_b_factory.hpp"

#include "char_b.hpp"

std::unique_ptr< BaseChar > CharBFactory::createObject()
{
    return std::unique_ptr< BaseChar >{ new CharB };
}
```

Simple example. *main*

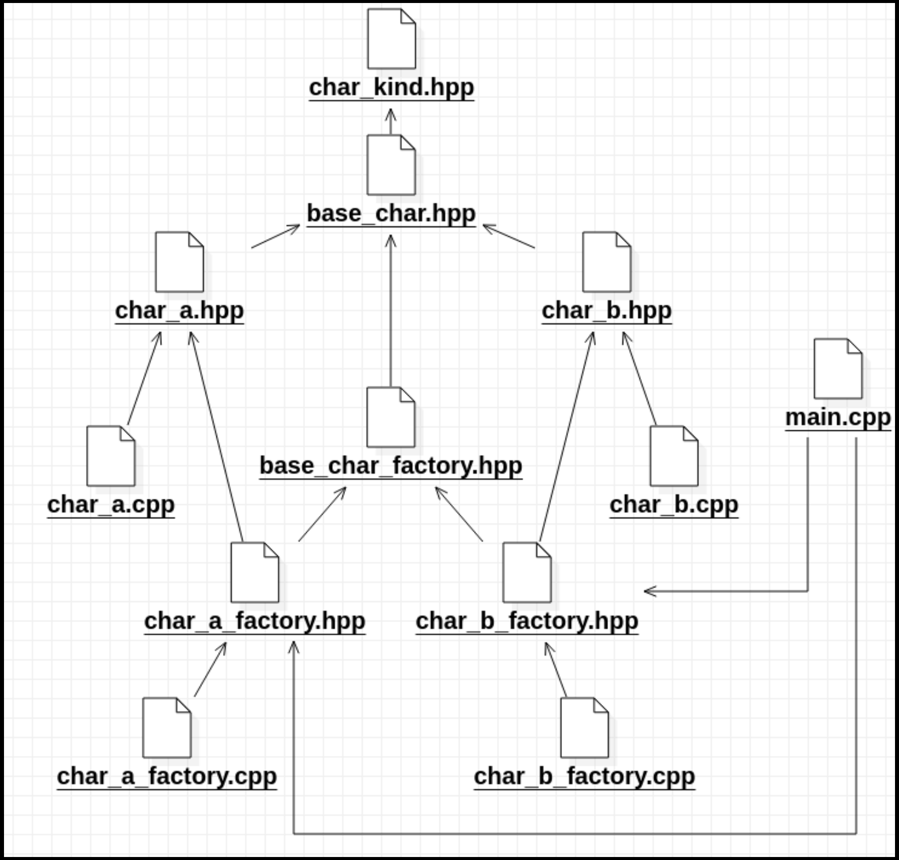
Example from *cppinclude/docs/examples/simple_example/*



```
#include "char_a_factory.hpp"
#include "char_b_factory.hpp"

#include <iostream>
#include <cassert>

std::unique_ptr< BaseChar > createChar( BaseCharFactory & _factory );
std::string enumToString( CharKind _kind );
int main()
{
    CharAFactory factoryA;
    CharBFactory factoryB;

    char c;
    std::cin >> c;

    std::unique_ptr< BaseChar > currentObject;
    if( c == 'a' )
        currentObject = createChar( factoryA );
    else
        currentObject = createChar( factoryB );
    assert( currentObject );

    std::cout << "Enum : " << enumToString( currentObject->getKind() ) << '\n';

    return 0;
}
```

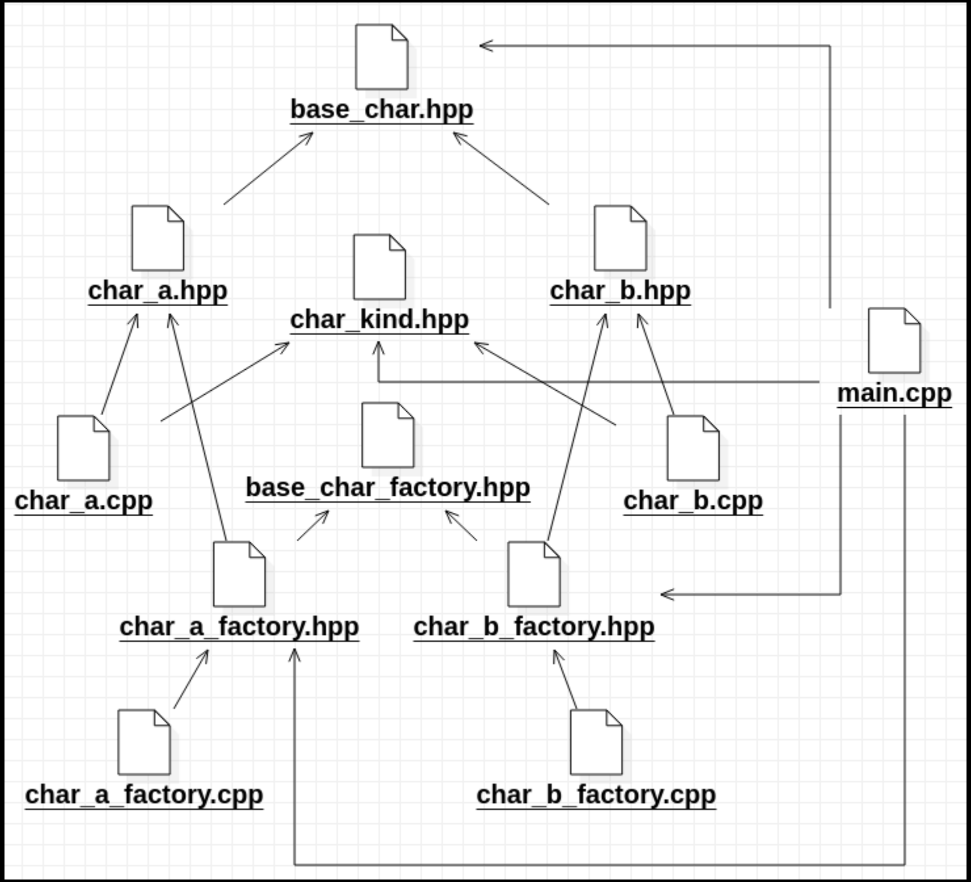
Simple example. Include hierarchy

Example from *cppinclude/docs/examples/simple_example/*

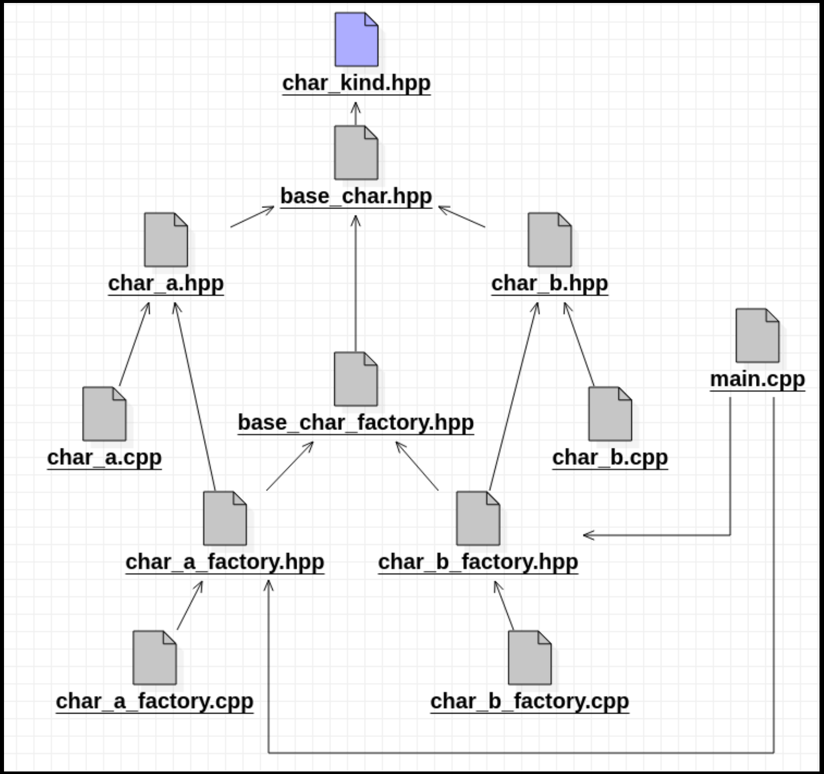


Simple example. Forward declaration

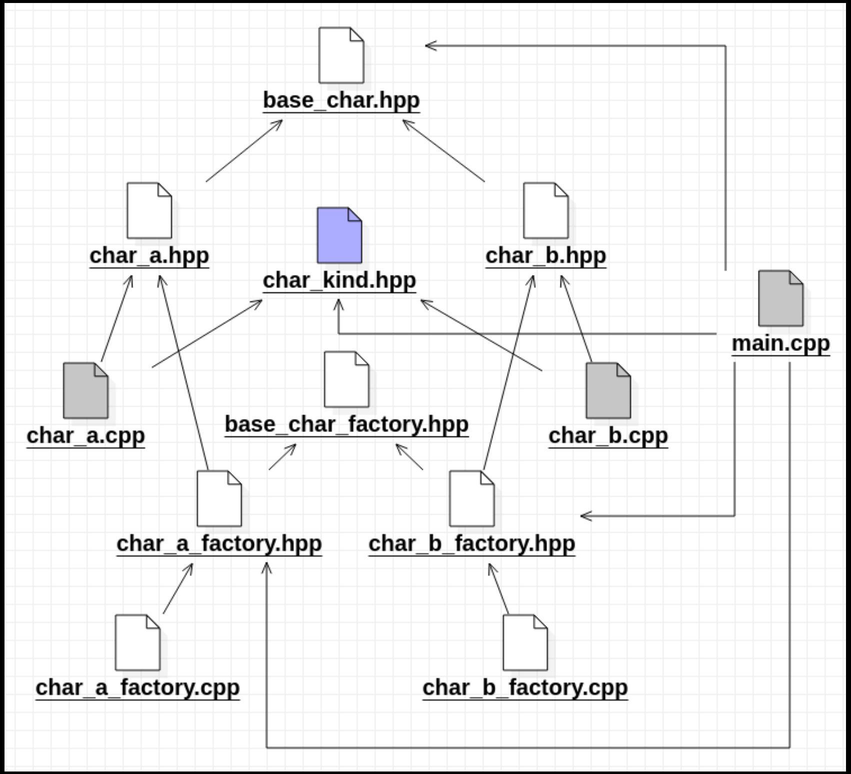
Example from *cppinclude/docs/examples/simple_example_with_forward_declaration/*



Change enum and compile



Without forward declaration



With forward declaration

Using *cppinclude* for simple example

```
> cppinclude

Start initialization project ...
Start analyze sources ...
Start report results ...
Most impact files:
1 : "char_kind.hpp" impact to 11 file(s)
Included by:
        1 : "base_char.hpp" line 3, impact to 10 file(s)
2 : "base_char.hpp" impact to 10 file(s)
Included by:
        1 : "base_char_factory.hpp" line 3, impact to 5 file(s)
        2 : "char_a.hpp" line 3, impact to 2 file(s)
        3 : "char_b.hpp" line 3, impact to 2 file(s)
3 : "base_char_factory.hpp" impact to 5 file(s)
Included by:
        1 : "char_a_factory.hpp" line 3, impact to 2 file(s)
        2 : "char_b_factory.hpp" line 3, impact to 2 file(s)

...
```

Using *cppinclude* for *vlc*

Example from *cppinclude/docs/examples/vlc/*

```
Start initialization project ...
Start analyze sources ...
Start report results ...
Most impact files:
1 : "modules/demux/adaptive/ID.hpp" impact to 78 file(s)
Included by:
        1 : "modules/demux/adaptive/http/Chunk.h" line 30, impact to 73 file(s)
        2 : "modules/demux/adaptive/playlist/Inheritables.hpp" line 26, impact to 63 file(s)
        3 : "modules/demux/adaptive/ID.cpp" line 24
        4 : "modules/demux/adaptive/playlist/BaseRepresentation.cpp" line 34
2 : "modules/demux/adaptive/http/ConnectionParams.hpp" impact to 78 file(s)
Included by:
        1 : "modules/demux/adaptive/http/Chunk.h" line 29, impact to 73 file(s)
        2 : "modules/demux/adaptive/http/HTTPConnection.hpp" line 28, impact to 5 file(s)
        3 : "modules/demux/adaptive/http/AuthStorage.cpp" line 25
        4 : "modules/demux/adaptive/http/ConnectionParams.cpp" line 24
        5 : "modules/demux/adaptive/http/HTTPConnection.cpp" line 25
        6 : "modules/demux/adaptive/http/HTTPConnectionManager.cpp" line 30

...
```

Using *cppinclude* for *Clementine*

Example from *cppinclude/docs/examples/vlc/*

```
Start initialization project ...
Start analyze sources ...
Start report results ...
Most impact files:
1 : "src/playlist/playlistsequence.h" impact to 185 file(s)
Included by:
        1 : "src/playlist/playlist.h" line 25, impact to 129 file(s)
        2 : "src/widgets/osd.h" line 30, impact to 76 file(s)
        3 : "src/networkremote/incomingdataparser.cpp" line 27
        4 : "src/playlist/playlistsequence.cpp" line 18
        5 : "src/ui/mainwindow.cpp" line 88
        6 : "tests/mpris1_test.cpp" line 24
2 : "src/playlist/playlistitem.h" impact to 179 file(s)
Included by:
        1 : "src/playlist/playlist.h" line 24, impact to 129 file(s)
        2 : "src/smartplaylists/generator.h" line 21, impact to 31 file(s)
        3 : "src/ui/mainwindow.h" line 33, impact to 13 file(s)
        4 : "src/library/libraryplaylistitem.h" line 22, impact to 12 file(s)

...
```

Using *cppinclude* for *gcc*

Example from *cppinclude/docs/examples/gcc/*

```
Start initialization project ...
Start analyze sources ...
Start report results ...
Most impact files:
1 : "gcc/config/arm/arm_mve_types.h" impact to 2496 file(s)
Included by:
	1 : "gcc/config/arm/arm_mve.h" line 35, impact to 2484 file(s)
	2 : "gcc/config/arm/arm_cde.h" line 140, impact to 10 file(s)
2 : "gcc/config/arm/arm_mve.h" impact to 2484 file(s)
Included by:
	1 : "gcc/testsuite/gcc.target/arm/mve/intrinsics/asrl.c" line 5
	2 : "gcc/testsuite/gcc.target/arm/mve/intrinsics/lsll.c" line 5
	3 : "gcc/testsuite/gcc.target/arm/mve/intrinsics/mve_fp_fpu1.c" line 6
	4 : "gcc/testsuite/gcc.target/arm/mve/intrinsics/mve_fp_fpu2.c" line 6
	5 : "gcc/testsuite/gcc.target/arm/mve/intrinsics/mve_fpu1.c" line 6
	6 : "gcc/testsuite/gcc.target/arm/mve/intrinsics/mve_fpu2.c" line 6

...
```

First run

1. Detect unresolved files:

```
> cppinclude --report unresolved
```

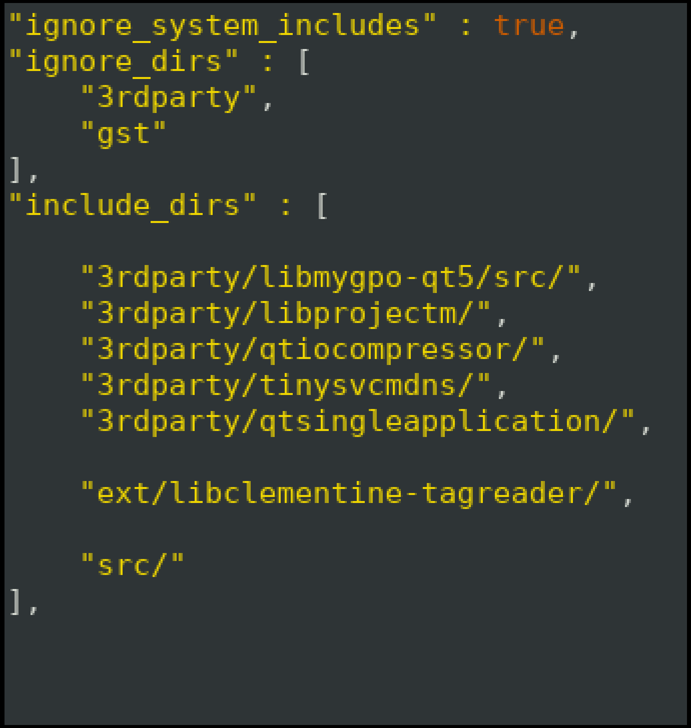
2. If a lot of files then apply limits to reports

```
> cppinclude --report unresolved --report_limit=5 --report_details_limit=3
```

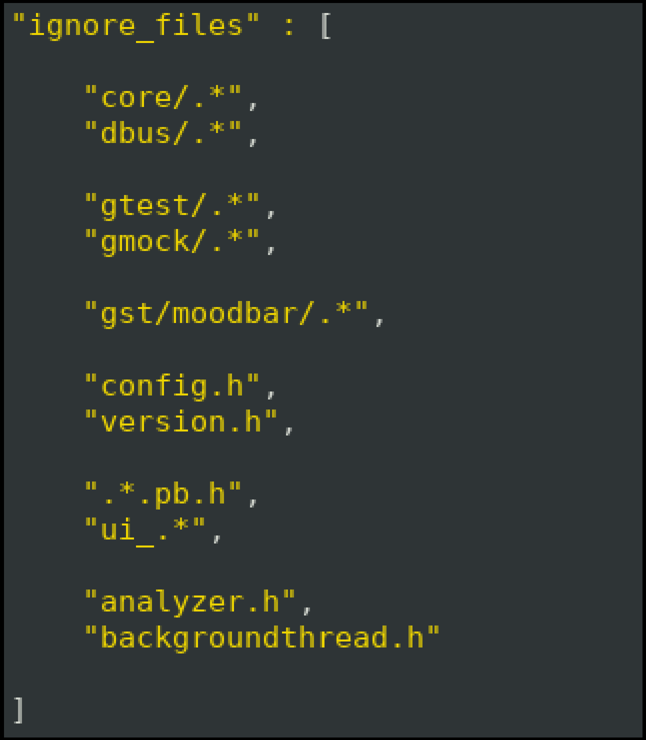
- *report_limit=5* – max. 5 unresolved files
- *report_details_limit=3 – max. 3 files that include unresolved file*

3. Create .*cppinclude.json* in project directory or customize via command arguments

First run.Customization in json



```
"ignore_system_includes" : true,
"ignore_dirs" : [
    "3rdparty",
    "gst"
],
"include_dirs" : [

    "3rdparty/libmygpo-qt5/src/",
    "3rdparty/libprojectm/",
    "3rdparty/qtiocompressor/",
    "3rdparty/tinysvcmdns/",
    "3rdparty/qtsingleapplication/",

    "ext/libclementine-tagreader/",

    "src/"
],
```



```
"ignore_files" : [

    "core/.*",
    "dbus/.*",

    "gtest/.*",
    "gmock/.*",

    "gst/moodbar/.*",

    "config.h",
    "version.h",

    ".*.pb.h",
    "ui_.*",

    "analyzer.h",
    "backgroundthread.h"

]
```

Example from *cppinclude/docs/examples/clementine/.cppinclude.json*

First run.Customization via arguments

```
>cppinclude –help

Usage:
  [OPTION...]

      --project_dir arg            Project directory (default: .)
    --file_extensions arg          Extensions C++ files (default: .cpp,*.hpp,*.c,*.h,*.cxx,*.hxx)
    --include_dirs arg             Include directories (default: .)
    --ignore_dirs arg              Directories that will be ignored (default: )
    --ignore_system_includes       Ignore headers in <>
    --ignore_files arg             Files will be ignored by regexp (default: )
    --configuration_file arg       Path to configuration file (default: cppinclude.json)
    --report arg                   List of reports (default: unresolved,most_impact)
    --report_limit arg             Maximum elements in report, 0 - unlimited (default: 10)
    --report_details_limit arg     Maximum details in report, 0 - unlimited (default: 10)
    --help                         Show usage
    --verbose                      Verbose mode
    --version                      Show application version
```

Q&A

Feedback & Questions

cppinclude@yandex.com