# Precompiled headers and modules in clang

Ivan Murashko

## Contents



## Introduction

The doc describes precompiled headers (pch) and modules (pcm) in clang, their internals and how they can be used

The source code for examples can be found in the article git repository [1] in the folder **pch/src**.

# 1 User guide

## 1.1 Precompiled headers

Generate you pch file is simple. Suppose you have a header file with name **header.h**:

```
#pragma once

#include <iostream>

void foo() {
  std::cout << "foo" << std::endl;
}
```

then you can generate a pch for it with

```
clang -x c++-header header.h -o header.pch
```

the option **-x c++-header** was used there. The option says that the header file has to be treated as a c++ header file. The output file is **header.pch**.

The precompiled headers generation is not enough and you may want to start using them. Typical C++ source file that uses the header may look like

```
// test pchs
```

```
#include "header.h"

int main() {
  foo();
  return 0;
}
```

As you may see, the header is included as follows

```
  ...
#include "header.h"
  ...
```

By default clang will not use a pch at the case and you have to specify it explicitly with

```
clang -include-pch header.pch main.cpp -o main -lstdc++
```

### 1.2 Modules

TBD

## 2 Internals

Actual pch file read is done via **clang::ASTReader::ReadASTCore** method. **clang::CompilerInstance::createPCHExternalASTSource -¿ clang::ASTReader::ReadAST -¿ clang::ASTReader::ReadASTCore -¿** method.

## References

[1] Ivan Murashko. Articles git repository, 2022.