#pragma

```
#pragma compiler specific extension
```

The pragma directive is used to access compiler-specific preprocessor extensions. A common use of #pragma is the #pragma once directive, which asks the compiler to include a header file only a single time, no matter how many times it has been imported:

```
#pragma once
// header file code
```

In this example, using #pragma once is equivalent to an include guard that prevents the file from being processed multiple times.

```
#ifndef _FILE_NAME_H_
#define _FILE_NAME_H_

/* code */

#endif // #ifndef _FILE_NAME_H_
```

#pragma once is available on many major compilers, including Clang, GCC, the Intel C++ compiler and MSVC.

The #pragma directive can also be used for other compiler-specific purposes. #pragma is commonly used to suppress warnings. For example, in MSVC

```
#pragma warning (disable : 4018 )
```

Can be used to disable warning 4018, warning of signed/unsigned mismatch. While you should be reluctant to suppress warnings sometimes it is necessary.

For more uses of the #pragma directive, consult your compiler's documentation.

Related

C preprocessor tutorial