

Preprocessor Directives

Preprocessor directives must be the first non-whitespace text specified on a line. All directives are prefixed with the pound or number-sign character ('#'). For example:

```
#pragma PRINT
#include <stdio.h>
#define DEBUG 1
```

Whitespace is allowed before and after the number-sign ('#'). A number-sign ('#') that appears alone on a line is interpreted as a null preprocessor directive. For example:

```
# define myfavnum 45
#
# include
```

The entire preprocessor directive must be contained in a single source line. Line continuations, backslash ('\') followed by a new-line character, may be used in preprocessor directives since these are removed by the preprocessor. For example:

```
#define mycode
{
volatile unsigned char i;
for (i=0; i<100; i++);
}
```

The number-sign ('#') and the preprocessor directive must be explicitly specified and may not come from a macro expansion. For example:

```
#define mydef #define otherdef 16

mydef
```

In this case, mydef is expanded to **define otherdef 16** since # is interpreted as a stringize operator. This expansion is processed by the compiler and a syntax error is generated.

The following table lists the preprocessor directives and gives a brief description of each.

Directive	Description
#define	Defines a preprocessor macro or constant.
#elif	Initiates an alternative branch of the if condition, when the previous #if , #ifdef , #ifndef , or #elif branch was not taken.
#else	Initiates an alternative branch when the previous #if , #ifdef , or #ifndef branch was not taken.
#endif	Ends a #if , #ifdef , #ifndef , #elif , or #else block.
#error	Outputs an error message defined by the user.
#ifdef	Evaluates an expression for conditional compilation. The argument to be evaluated is the name of a definition.
#ifndef	Same as #ifdef but the evaluation succeeds if the definition is not defined.
#if	Evaluates an expression for conditional compilation.
#include	Reads source text from an external file. The notation sequence determines the search sequence of the included files. The compiler searches for include files specified with less-than/greater-than symbols ('<', '>') in the include file directory. Include files specified with double-quotes (" ") are searched for in the current directory.

#line	Specifies a line number and an optional filename. This specification is used in error messages to identify the error position.
#message	Outputs a information message defined by the user.
#pragma	Allows you to specify directives that may be included on the compiler command line. Pragmas may contain the same directives that are specified on the command line.
#undef	Deletes a preprocessor macro or constant definition.
#warning	Outputs a warning message defined by the user.

Copyright © Keil, An ARM Company. All rights reserved.