Types of pragma directives in C

Last Updated : 04 Apr, 2021

Pragma Directives: The pragma directive is used to control the actions of the compiler in a particular portion of a program without affecting the program as a whole.

- Pragma directives are included in the C program to take effect.
- The effect of pragma will be applied from the point where it is included to the end of the compilation unit or until another pragma changes its status.
- A **#pragma** directive is an instruction to the compiler and is usually ignored during preprocessing.

Syntax:

#pragma string

Here, the string can be one of the instructions given to the compiler with any required parameters.

Instruction	Description
COPYRIGHT	To specify a copyright string
COPYRIGHT_DATE	To specify a copyright date for the copyright string
OPTIMIZE	To turn the optimization feature on or off
LOCALITY	To name a coded subspace
OPT_LEVEL	To set the level of optimization
HP_SHLIB_VERSION	To create a version of a shared library routine
VERSION ID	To specify a version string
ONCE	Specify that file opened only once

Types of Pragma Directives:

pragma COPYRIGHT:

The **syntax** of pragma copyright is:

#pragma copyright "string"

- Here, the string specifies the set of characters included in the copyright message in the object file.
- If no data is specified during pragma copyright then the current year is used in the copyright message:

Example: If copy write is written in a below way:

#pragma copyright "GFG Private Limited"

- Then the following string is placed in the object code (assuming the current year is 2020):

© Copyright GFG private limited, 2020.
All rights reserved. No part of this program may be copied, reproduced, or transmitted with

out the prior written consent of GFG Private Limited.

Note: To see the COPYRIGHT string as well as any other strings in the object file, use the strings(1) command with the -a option.

Example:

strings -a ObjectFileName.o

pragma COPYRIGHT_DATE:

The **syntax** of pragma copyrighted is:

#pragma COPYRIGHT_DATE “string”

Here, the string is a date that will be used by the COPYRIGHT pragma.
Consider the following example given below:

#pragma COPYRIGHT_DATE “2011-2020”
#pragma copyright “GFG Private Limited”

The above pragma will place the following string in the object code:

© Copyright GFG Private Limited, 2011-2020.
All right reserved no part of this program may be photocopied reproduced, or transmitted w ithout the prior written consent of GFG private limited.

Note: To see the **COPYRIGHT_DATE** string as well as any other strings in the object file, use the strings(1) command with the **-a** option.
Example:

strings -a ObjectFileName.o

pragma OPT_LEVEL:

The **syntax** of pragma OPT_LEVEL which is used to set the optimization level to 1, 2, 3 or 4 is:

 # pragma OPT_LEVEL 1
 # pragma OPT_LEVEL 2
 # pragma OPT_LEVEL 3
 # pragma OPT_LEVEL 4

Like the optimization pragma, even the pragma cannot be used in a function. Finally, OPT_LEVEL 3 and 4 are allowed only at the beginning of the file.

Below is an example code snippet in C illustrating the use of pragma opt level:

C

```
aCC - O prog.C

#pragma OPT_LEVEL 1

      // Optimise func1() at level 1
      void
      Func1()
{
}

#pragma OPT_LEVEL 2

// Optimize Func2() to at level 2
void Func2()
{
}
```

pragma OPTIMIZE:

The **syntax** of using the pragma OPTIMIZE is:

#pragma OPTIMIZE ON
#pragma OPTIMIZE OFF

- The pragma optimize is basically used to turn ON/OFF optimization in the section of the source program.
- However, when using the pragma specifies one of the optimization options on the ACC command (while giving the command to compile the program), otherwise, this pragma is ignored.
- Also, remember that the pragma optimize cannot be used within a function.