

UNREFERENCED_PARAMETER的作用计算机等级考试 PDF
转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_UNREFERENCED_PARAMETER_c97_645770.htm

UNREFERENCED_PARAMETER(pParam). 跟踪到宏定义

```
#define UNREFERENCED_PARAMETER(P) (P) #define
```

```
DBG_UNREFERENCED_PARAMETER(P) (P)百考试题论坛
```

```
#define DBG_UNREFERENCED_LOCAL_VARIABLE(V) (V) 使用
```

```
的方法比如: int SomeFunction(int arg1, int arg2) {
```

```
UNREFERENCED_PARAMETER(arg2) ... }
```

UNREFERENCED_PARAMETER 展开传递的参数或表达式。

其目的是避免编译器关于未引用参数的警告。许多程序员，包括我在内，喜欢用最高级别的警告 Level 4 (/W4) 进行编译。Level 4 属于“能被安全忽略的事件”的范畴。虽然它们可能使你难堪，但很少破坏你的代码。例如，在你的程序中可能会有这样一些代码行：int x=1. 但你从没用过 x。也许这一行是你以前使用 x 时留下来的，只删除了使用它的代码，而忘了删除这个变量。Warning Level 4 能找到这些小麻烦。所以，为什么不让编译器帮助你完成可能是最高级别的专业化呢？用Level 4 编译是展示你工作态度的一种方式。如果你为公众使用者编写库，Level 4 则是社交礼节上需要的。你不想强迫你的开发人员使用低级选项清洁地编译他们的代码。问题是，Level 4 实在是太过于注意细节，在 Level 4 上，编译器连未引用参数这样无伤大雅的事情也要抱怨（当然，除非你真的有意使用这个参数，这时便相安无事）。假设你有一个函数带来两个参数，但你只使用其中一个：int

SomeFunction(int arg1, int arg2) { return arg1 5. } 使用 /W4 , 编译器抱怨 : “ warning C4100: arg2 : unreferenced formal parameter. ” 为了骗过编译器 , 你可以加上 UNREFERENCED_PARAMETER(arg2)。现在编译器在编译你的引用 arg2 的函数时便会住口。并且由于语句 : arg2. 实际上不做任何事情 , 编译器不会为之产生任何代码 , 所以在空间和性能上不会有任何损失。编辑特别推荐: 送给正在学习C 朋友的50条建议 C sizeof使用规则及陷阱分析 emplates之类模板的继承 VC编程入门浅谈 C 笔试考前练习 100Test 下载频道开通 , 各类考试题目直接下载。详细请访问 www.100test.com