

## 0.9 - 配置编译器：构建配置

亚历克斯于2015年2月17日 | 最后由ALEX于2018年10月25日修改

一个**构建配置**（也称为**构建目标**）是项目设置的集合，它决定了你的IDE将如何建立你的项目。构建配置通常包括可执行文件将被命名的内容，IDE将查找其他代码和库文件的目录，是否保留或删除调试信息，编译器优化程序的程度等等...通常，您将希望将这些设置保留为默认值，除非您有特定的理由进行更改。

在IDE中创建新项目时，大多数IDE将为您设置两种不同的构建配置：发布配置和调试配置。

该**调试配置**，旨在帮助您调试程序，通常是编写程序时要使用的一个。此配置会关闭所有优化，并包含调试信息，这会使您的程序变得更大，更慢，但更容易调试。默认情况下，调试配置通常选择为活动配置。我们将在后面的课程中详细讨论调试技术。

该**版本的配置**设计释放你的程序给公众时使用。此版本通常针对大小和性能进行了优化，并且不包含额外的调试信息。由于发布配置包括所有优化，因此此模式对于测试代码的性能也很有用（我们将在后面的教程系列中向您展示如何操作）。

当使用Visual Studio构建Hello World程序（来自第0.7课-[编译第一个程序](#)）时，调试配置中生成的可执行文件为65kb，而发行版中生成的可执行文件为12kb。差异主要是由于调试版本中保留了额外的调试信息。

虽然您可以创建自己的自定义构建配置，但除非您想要比较使用不同编译器设置进行的两个构建，否则您很少有理由。

### 最佳实践

在开发程序时使用**调试**版本配置。当您准备将可执行文件发布给其他人或想要测试性能时，请使用**发布**版本配置。

## 在构建配置之间切换

### 对于Visual Studio用户

在Visual Studio中有多种方法可以在**调试**和**发布**之间切换。最简单的方法是直接从**标准工具栏**选项的**解决方案配置**下拉列表中设置您的选择：



现在将其设置为**Debug**。

您也可以通过选择访问配置管理器对话框**生成菜单 > 配置管理器**，并更改**活动的解决方案配置**。

### 对于Code :: Blocks用户

在Code :: Blocks中，您应该在“**编译器**”工具栏中看到一个名为**Build Target**的项：



现在将其设置为**Debug**。



[0.10 - 配置编译器：编译器扩展](#)



[指数](#)



[0.8 - 一些常见的C ++问题](#)

分享这个：



Facebook的



推特



G+ 谷歌



Pinterest的

广告 X



腾讯云新用户域名抢购1元起

买就送证书和解析，更有多种爆款组合选购。

C++教程 | 打印本文

## 75条评论转至0.9 - 配置编译器：构建配置

«旧评论 (1) (2)



标记

2018年10月24日下午12:35 · 回复

Since you inserted a new lesson prior to this one (0.8—A few common C++ problems), the following sentence in the fifth paragraph needs to reflect that.

"When the Hello World program from >>the previous lesson<< was built using Visual Studio, ..."

Should be corrected to read:

"When the Hello World program from >>lesson 0.7<< was built using Visual Studio, ..."



Alex

October 25, 2018 at 12:36 pm · Reply

Indeed. Thanks for pointing this out. Fixed!



Ryan

September 23, 2018 at 12:08 am · Reply

Hello! I believe I found a typo in the bottom section. Where you said 'In Code::Blocks, you should see aitem called Build Target', I think you meant to say 'an item' instead of 'aitem'.



Alex

September 24, 2018 at 7:52 am · Reply

Definitely. Thanks for pointing this out.



Mark

May 17, 2018 at 2:36 pm · Reply

Hi,

I can't find the Build target toolbar, because I am using codeblocks 17.12. Another question that is when and why do you need to use the highlight option? And last question is what are the "match case" and "use regex" options used for?

Thanks.



Willie

March 10, 2018 at 1:30 pm · Reply

Hello Alex.

Is there a easy to read section where the variables are listed in a table form so the values can be seen and whatever else ....possibly how the variables are written?

Trying to make it easy.



Alex

March 11, 2018 at 12:26 pm · Reply

I don't think I understand what you mean. Variables aren't mentioned in this lesson, so I'm not sure what problem you are trying to solve for.



Willie

March 11, 2018 at 1:21 pm · Reply

Hello ,

I wasn't trying to solve or solving anything I'm writing an easy method for reference so I can see tools and how I could use them or more appropriate - how they are used.



Alex

March 11, 2018 at 3:53 pm · Reply

You're looking for an easy way to see all the variables in scope and their values?

If so, then some IDEs offer this as a debugging tool. In debugging mode, Visual Studio's watch windows auto tab does this.



Willie  
[March 12, 2018 at 1:52 am · Reply](#)  
 Thanks Alex.



Prynce  
[March 10, 2018 at 9:46 am · Reply](#)

Hello!!!

Thanks so much I was able to write my first c++ program using Dev c++

"Hello world!" Was displayed on my screen



**Prabhat Kumar Roy**  
[December 4, 2017 at 4:59 am · Reply](#)

Dear Sir!

I am revising your lessons after completing the chapter 1 (including its sub-chapters).

I have downloaded and installed Code Block-version 16.01 (Release 16.01 rev 10702 (2016-01-25 19:50:14) gcc 4.9.2 Windows/unicode - 32 bit).

I failed to locate any 'Build target' there even after pasting my 'HellowWorld' program which otherwise runs perfectly.

Can you please guide me in this respect?

Thank you.

Best regards.



Alex  
[December 6, 2017 at 7:29 pm · Reply](#)

Did you set up both a release and debug release configuration when your project was created? Check out the screenshot here:

<https://i.ytimg.com/vi/ccYmIFRLCWE/maxresdefault.jpg> – it shows where you should be able to see your active build target on the interface (in this screenshot, the Debug configuration is selected)



Kushal Singh  
[November 20, 2017 at 10:10 am · Reply](#)

Thanks ☐



Kushal Singh  
[November 8, 2017 at 12:31 pm · Reply](#)

Hello Alex,

When I navigate to Debug dir of my project folder , I find files with extensions: .idb,.pdb and .pch. what are these files for and when are they generated ?

Regards



Alex  
[November 8, 2017 at 11:54 pm · Reply](#)

.idb and .pdb are debug files produced by Visual Studio. .pch is a precompiled header file. These are all generated when you compile your program in debug configuration with precompiled headers enabled (which they are by default).



Kushal Singh  
[November 8, 2017 at 9:27 am · Reply](#)

How can I check the size of executable produced in debug/release configuration ?



Alex  
[November 8, 2017 at 11:53 pm · Reply](#)

Use your OS file explorer/browser and find where your IDE put the executable for each of your configurations. This is most likely in a subfolder inside your project directory.



phong nguyen  
[June 16, 2017 at 8:44 pm · Reply](#)

Hi, I just want to say thank you for the continuous efforts over the past couple of years. You rock!!!



Ahtazaz Khan  
[May 3, 2017 at 8:51 pm · Reply](#)

Hi, i'm new here, i love your effort. i'm doing bachelor in Computer science. I have a little bit programming skills. But now i have a aim to learn Programming from the 0.I found your work best. Amazing tutorials. I have a question about building configurations. I have don't clear the build-configuration. Mean, What is the purpose of Build Configuration. How and when debug and release configurations use...?

Alex

[May 4, 2017 at 11:10 am · Reply](#)



You'll generally use a debug configuration when developing your code or running it for your own purposes. This enables all debugging functionality, making it easier to find errors. If you decide to release your executable to others, you'll build it using a release configuration. This strips out all debugging information and uses more compiler optimizations, making your code smaller and faster.

You can define custom release configurations as well, but there's no need unless you have special circumstances.



Ahtazaz Khan

[May 4, 2017 at 8:50 pm](#) · [Reply](#)

Awesome response...i got it... Thanks

[« Older Comments](#) [1](#) [2](#)



# Python大型免费公开课

上课时间20: 00-23: 00

点击