

怎样从现有代码创建项目并设置代码帮助计算机二级考试

PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/597/2021_2022__E6_80_8E_E6_A0_B7_E4_BB_8E_E7_c97_597135.htm

要获取更多有关在 NetBeans IDE 中使用 C/C 应用程序的信息，请参见 NetBeans 网站上的 C/C 应用程序学习课程页。教程需求在继续前，请确保您重新检查了本节中的需求。先决条件 本教程假设您已经有一些 C 和 C 的编程经验。本教程所需的软件在开始前，您需要安装 NetBeans 6.0。C/C 下载选项包括 NetBeans IDE 6.0 和 C/C 支持。问题是什么？IDE 的一项职责就是代码模型的构建。代码模型对 IDE 功能很重要，例如：代码完成 超链接 导航 为了正确生成模型，IDE 需要关于项目的信息，例如：编译的文件 使用哪个编译器 用户包含路径和每个编译的文件中的宏 使用的头文件 没有这些信息，IDE 无法生成恰当的模型，这将造成 IDE 的功能工作出错。当然，所有这些信息都包含在 makefile 里。问题是由于以下原因，难以判断出必要的信息：复杂的开源项目通常有一个层次文件结构，主 makefile 引用了子目录下的 makefile。Makefile 常常使用外壳脚本命令。有时候 make 实用工具的行为取决于它前面的操作；例如，基于前一步中所创建的文件的存在性。Netbeans IDE 提供了另一种方法来获取模型的必要信息，它基于使用下列文件中的编译器注释或调试信息：对象文件 静态或动态库 可执行文件 如果工作的？让我们检查在为某个开源应用程序新建 Netbeans IDE 项目时的典型情况。这种情况始于开始下载某个开源项目。我们使用 MySQL 项目。以下指令都假设我们已经载入了该开源项目并安装了必要的软件。现在我们

来配置这个项目。 要保证编译器注释的完整，我们提供附加的编译标志：`./configure CFLAGS="-g3 -gdwarf-2" CXXFLAGS="-g3 -gdwarf-2"` 这些标志有什么用？我相信 `-g` 标志广为人知。该标志告诉编译器在对象文件中包括调试信息。该标志的 `-g3` 变体使调试信息更完整。该标志使得能够获取用户宏的信息。`-gdwarf-2` 标志告诉编译器使用版本 2 的 DWARF 调试信息格式。我们用 `make` 命令生成应用程序。基本来说，我们必须确保应用程序用以上提到的标志来编译。怎样告诉 `make` 实用工具去使用必要的标志取决于特定的开源项目。有时候有必要导出环境变量：`export CFLAGS="-g3 -gdwarf-2" export CXXFLAGS="-g3 -gdwarf-2"` 对于不灵活的项目，有必要查看 `makefile` 的内容，以便了解怎样提供标志。现在我们的开源项目已经成功生成。所以能够创建一个 NetBeans IDE 项目。启动 NetBeans IDE。通过选择“文件” `gt.` “C/C” `gt.` “代码帮助” `gt.` “常规”，或者“属性” `gt.` “C 编译器” `gt.` “显示失败的 `#include` 目录”。可以配置一个现已存在的 IDE 项目吗？可以。首先，用上面提到的必要标志生成开源项目。然后右键单击项目节点，选择“代码帮助” `>` “配置代码帮助”。“配置代码帮助”向导以两种模式工作：简易和高级。简易模式重复新建项目向导的操作。高级模式让您选择获取编译器注释的不同方法，并具有一个分析结果的预览。有可能让一个 IDE 项目具备不同平台/设备相关的配置吗？可以。您可以在项目属性中管理配置。对每个配置，您都可以启动“配置代码帮助”向导。在向导中指定不同平台/设备的生成结果。向导将设置当前的活动配置。100Test 下载频道开通，各类考试题目直接下载。详细请访

