Oracle® Solaris Studio 12.3 IDE 快速入门教程

2011 年 12 月，E26461-01

ORACLE®

创建项目

借助 Oracle Solaris Studio IDE，您可以使用生成的 makefile 创建 C、C++ 和 Fortran 应用程序和库项目，也可以基于现有源代码和 makefile 以及现有二进制文件创建项目。

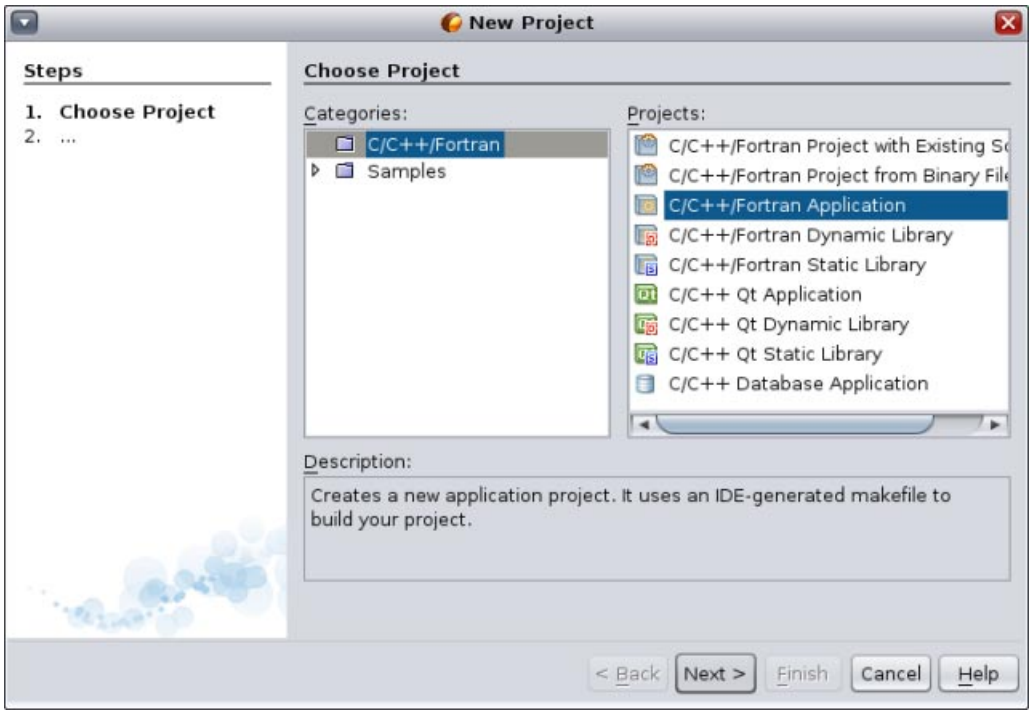
您可以在本地主机（从中启动 IDE 的系统）上生成、运行和调试项目，也可以在运行 Solaris 操作系统或 Linux 操作系统的远程主机上生成、运行和调试项目。

通过 C/C++/Fortran 应用程序、动态库、静态库或 Oracle 数据库项目，IDE 可以控制应用程序生成、运行和调试的各个方面。可在创建项目时指定项目设置，也可以在 "Project Properties"（项目属性）对话框中指定。IDE 会生成一个 makefile，您的所有设置都存储在其中。

使用 makefile 基于现有源代码生成项目。

创建应用程序项目

1. 通过选择 "File"（文件）> "New Project"（新建项目）(Ctrl+Shift+N)，打开 "New Project"（新建项目）向导。
2. 在向导中选择 "C/C++/Fortran" 类别。
3. 该向导提供了多种项目类型。选择 "C/C++/Fortran Application"（C/C++/Fortran 应用程序），然后单击 "Next"（下一步）。



4. 通过向导使用缺省值创建 C/C++/Fortran 应用程序新项目。您可以选择项目的名称和位置。
5. 单击 "Finish"（完成）退出向导。

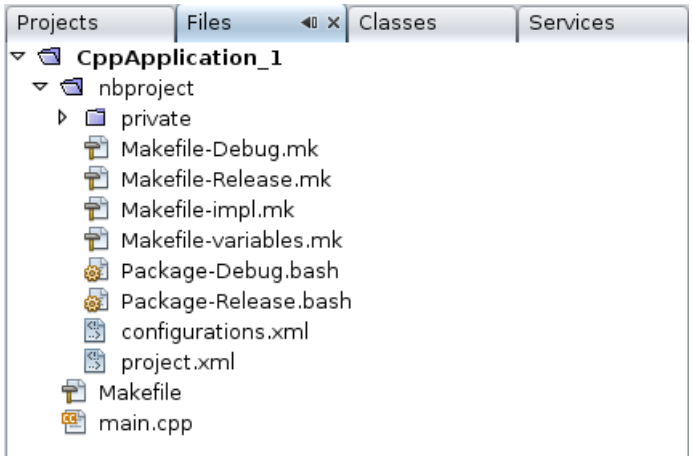
现在便创建了一个包含几个逻辑文件夹的项目。逻辑文件夹不是目录，而是一种文件组织方式，不反映文件在磁盘上的物理存储位置。添加到逻辑文件夹的文件会自动成为项目的组成部分，并在您生成项目时进行编译。

添加到 "Important Files"（重要文件）文件夹的文件不是项目的组成部分，在您生成项目时不会进行编译。这些文件仅供参考，如果您有一个包含现有 makefile 的项目，这些文件将非常方便。

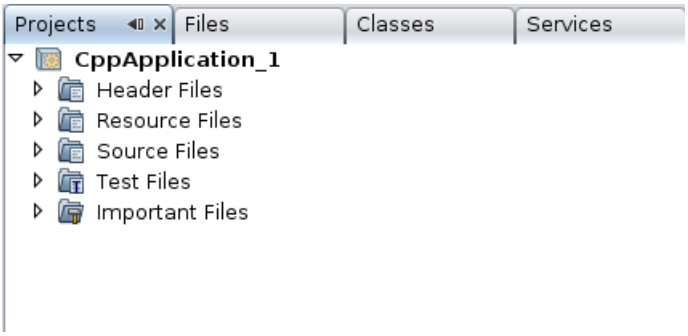
在项目的逻辑视图和物理视图之间切换

项目既有逻辑视图，也有物理视图。您可以在项目的逻辑视图和物理视图之间切换。

1. 选择 "Files"（文件）选项卡。此窗口显示项目的物理视图。该视图显示文件和文件夹，就像存储在磁盘上一样。



2. 选择 "Projects"（项目）选项卡。此窗口显示项目的逻辑视图。



向项目添加文件和文件夹

可以向项目添加逻辑文件夹。

1. 右键单击 `CppApplication_1` 项目的项目节点，然后选择 "New Logical Folder"（新建逻辑文件夹）。项目中将添加一个新逻辑文件夹。
2. 右键单击新的逻辑文件夹，然后选择 "Rename"（重命名）。键入您希望新文件夹使用的名称。

可以向现有文件夹添加文件和文件夹。逻辑文件夹可以嵌套。

向项目添加新文件

可以向项目添加新文件。

1. 右键单击 "Source Files"（源文件）文件夹，然后选择 "New"（新建）> "C Main File"（C 主文件）。
2. 在 "Name and Location"（名称和位置）页面上，"File Name"（文件名）字段中显示 `newmain`。
3. 单击 "Finish"（完成）。

这将在磁盘上的项目目录中创建 `newmain.c` 文件，并将其添加到 "Source Files"（源文件）文件夹中。可以向此文件夹添加任何种类的文件，而不仅仅是源文件。

注 – 还可以从此文件夹中删除文件。在本例中，您不需要在创建项目时缺省添加的 `main.cpp` 文件。要从项目中删除此文件，请右键单击此文件名称，然后选择 "Remove From Project"（从项目中删除）。

向项目添加更多新文件

1. 右键单击 "Header Files" 文件夹，然后选择 "New"（新建）> "C Header File"（C 头文件）。
2. 在 "Name and Location"（名称和位置）页面上，"File Name"（文件名）字段中显示 `newfile`。
3. 单击 "Finish"（完成）。

此时在磁盘上的项目目录中创建了 `newfile.h` 文件，并添加到 "Header Files" 文件夹中。

向项目添加现有文件

可以使用两种方法向项目添加现有文件：

- 右键单击 "Source Files"（源文件）文件夹，然后选择 "Add Existing Item"（添加现有项）。可以使用 "Select Item"（选择项）对话框指向磁盘上的某个现有文件，然后将该文件添加到项目中。
- 右键单击 "Source Files"（源文件）文件夹，然后选择 "Add Existing Items from Folders"（从文件夹添加现有项）。使用 "Add Folders"（添加文件夹）对话框添加包含现有文件的文件夹。

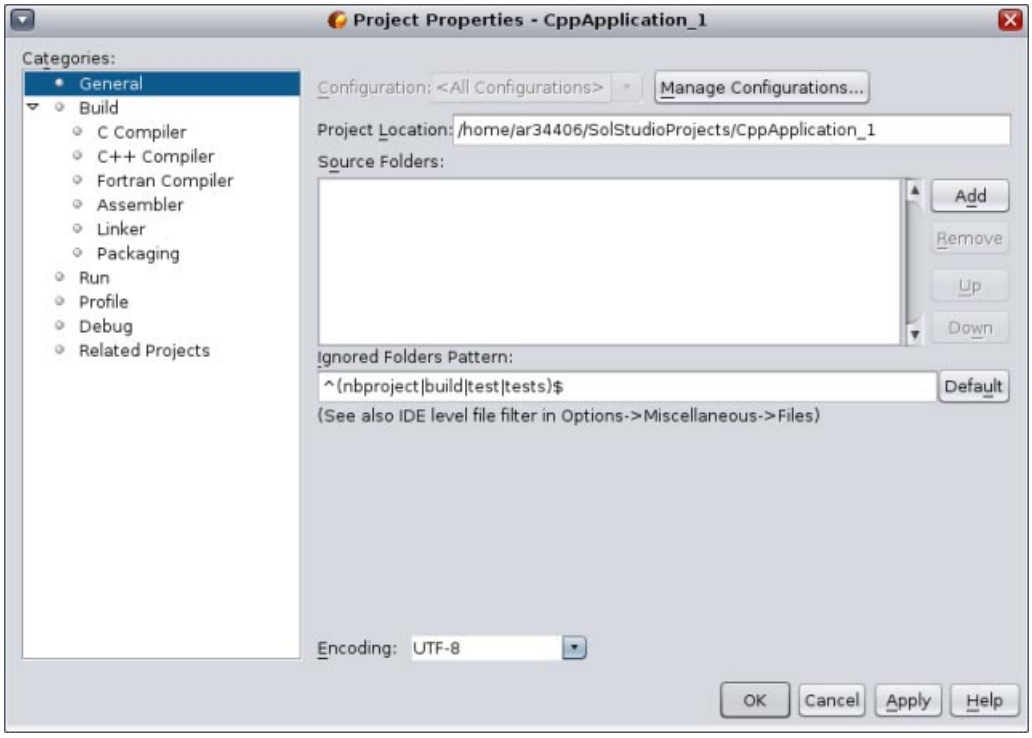
请不要使用 "New"（新建）菜单项来添加现有项。"Name and Location"（名称和位置）面板将显示该文件已存在。

设置项目属性

在创建项目时，有两个配置："Debug"（调试）和 "Release"（发行）。配置是用于项目的设置集合，使您能够轻松地在许多属性设置之间快速切换。"Debug"（调试）配置生成一个包含调试信息的应用程序版本。"Release"（发行）配置生成一个优化版本。

"Project Properties"（项目属性）对话框包含项目的生成信息和配置信息。打开 "Project Properties"（项目属性）对话框：

- 右键单击 "Application"（应用程序）项目的项目节点，然后选择 "Properties"（属性）。



通过在 "Project Properties"（项目属性）对话框的左面板中选择一个节点然后在右面板中修改属性，可以修改编译器设置和其他配置设置。选择其中一些节点和属性值，请注意可以设置的属性。设置 "General"（常规）属性时，这些属性将应用到项目的所有配置。设置 "Build"（生成）、"Run"（运行）或 "Debug"（调试）属性时，这些属性仅应用到当前选定的配置。

管理配置

在 "Project Properties"（项目属性）对话框中更改过的属性存储在当前配置的 makefile 中。您可以编辑缺省配置或创建新的配置。创建新配置：

1. 在 "Project Properties"（项目属性）对话框中，单击 "Manage Configurations"（管理配置）按钮。
2. 在 "Configurations"（配置）对话框中，选择与所需配置最为匹配的配置。这种情况下，请选择 "Release" 配置，并单击 "Copy"（复制）按钮，然后单击 "Rename"（重命名）。
3. 在 "Rename"（重命名）对话框中，将配置重命名为 "PerformanceRelease"。单击 "OK"（确定）。
4. 在 "Configurations"（配置）对话框中，单击 "OK"（确定）。
5. 在 "Project Properties"（项目属性）对话框的左面板中选择 "C Compiler"（C 编译器）节点。请注意，在 "Configuration"（配置）下拉式列表中已选中 "PerformanceRelease" 配置。
6. 在右面板的属性表中，将 "Development Mode"（开发模式）由 "Release" 更改为 "PerformanceRelease"。单击 "OK"（确定）。

您现已创建一个新配置，该配置将使用一组不同的选项来编译应用程序。

设置源文件属性

设置项目的项目属性时，相关属性将应用到项目中的所有文件。可以为某个特定文件设置一些属性。

1. 右键单击 `newmain.c` 源文件，然后选择 "Properties"（属性）。
2. 在 "Categories"（类别）面板中单击 "General"（常规）节点，您将可以选择一个不同的编译器或其他工具来生成此文件。还可以选择一个复选框，以便从当前选定项目配置的生成中排除该文件。
3. 单击 "C Compiler"（C 编译器）节点，您将可以覆盖项目编译器设置以及此文件的其他属性。
4. 取消 "Project Properties"（项目属性）对话框。

设置主项目

在 "Projects"（项目）窗口中右键单击某个项目节点时，IDE 将显示一个弹出式菜单，其中包含您可以对选定项目执行的操作。如果同时打开了多个项目，对某个项目节点显示弹出菜单表明您正在针对该项目执行操作。

菜单栏和工具栏上大多数与项目相关的操作都针对主项目。在 "Projects"（项目）窗口中，主项目节点以粗体显示。

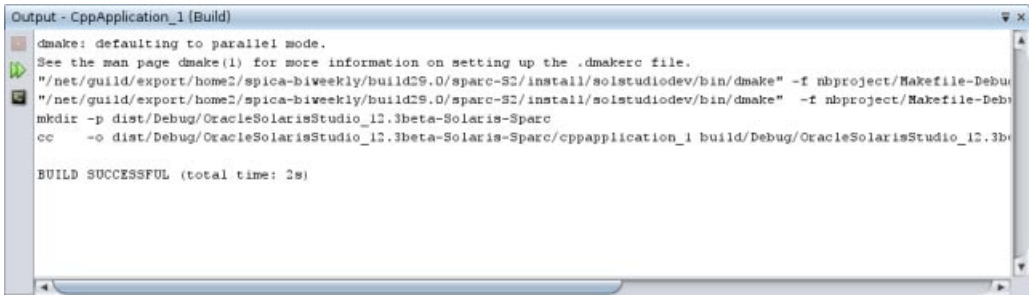
在 IDE 中更改主项目：

- 右键单击所需的项目节点，然后选择 "Set as Main Project"（设置为主项目）。此项目现在是 IDE 中的主项目，菜单栏和工具栏中的操作都针对此项目。

生成项目

生成项目：

1. 右键单击项目，然后选择 "Build"（生成），即可生成项目。生成输出显示在 "Output"（输出）窗口中。



2. 在主工具栏中的配置下拉式列表中，将配置从 "Debug"（调试）切换到 "PerformanceRelease"。现在将使用 "PerformanceRelease" 配置生成项目。
3. 右键单击项目，然后选择 "Build"（生成），即可生成项目。生成输出显示在 "Output"（输出）窗口中。

要同时生成项目的多个配置，请选择 "Run"（运行）> "Batch Build Main Project"（批量生成主项目），然后在 "Batch Build"（批量生成）对话框中选择要生成的配置。

通过右键单击项目并从菜单中选择操作，可以生成项目、清除项目或者清除并生成项目。项目还会将不同配置中的目标文件和可执行文件分开，这样您就不必担心来自多个配置的文件混在一起。

编译单一文件

编译单一源文件：

- 右键单击 `newmain.c` 文件，然后选择 "Compile File"（编译文件）。将仅编译此文件。

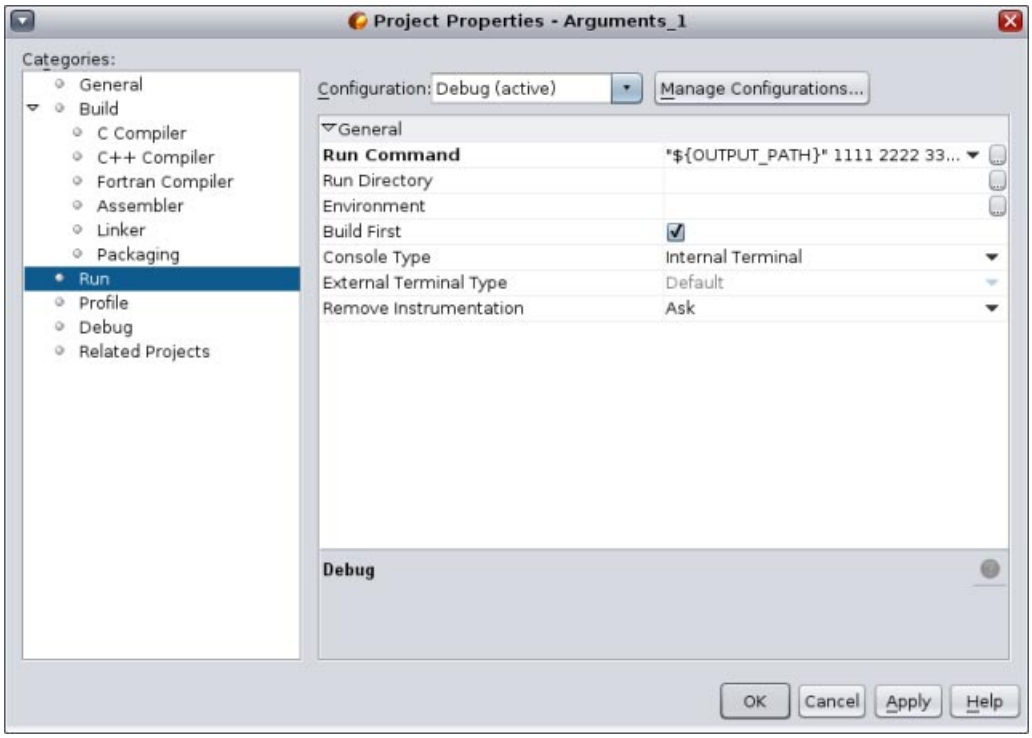
注 – 项目类型 "C/C++/Fortran Project From Existing Code"（源自现有代码的 C/C++/Fortran 项目）不支持单一文件编译。

运行项目

`Arguments` 样例程序会输出命令行参数。在运行程序之前，您需要在当前配置中设置一些参数，然后运行程序。

要创建 `Arguments_1` 项目，请设置一些参数，然后运行该项目：

1. 选择 "File"（文件）> "New Project"（新建项目）。
2. 在项目向导中，展开 "Samples"（样例）类别。
3. 选择 "C/C++" 子类别，然后选择 `Arguments` 项目。单击 "Next"（下一步），然后单击 "Finish"（完成）。
4. 右键单击 `Arguments_1` 项目节点，然后选择 "Build"（生成），即可生成项目。
5. 右键单击 `Arguments_1` 项目节点，然后选择 "Properties"（属性）。
6. 在 "Project Properties"（项目属性）对话框中，选择 "Run"（运行）节点。
7. 在 "Run Command"（运行命令）文本字段中，在输出路径后面键入 `1111 2222 3333`。单击 "OK"（确定）。



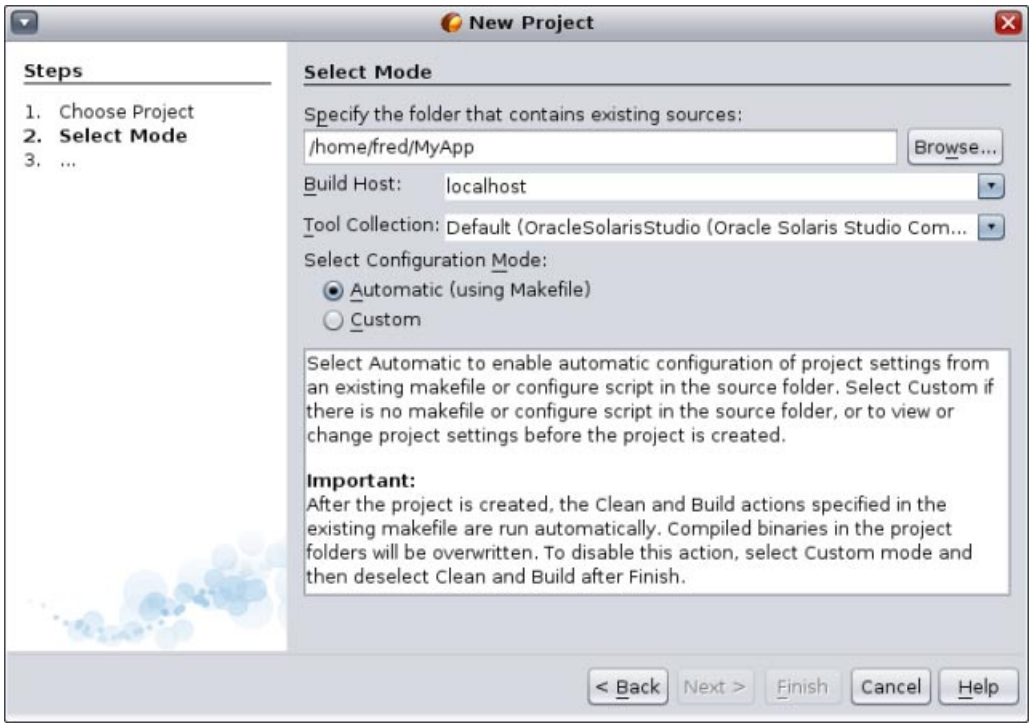
8. 选择 "Run"（运行）> "Run Main Project"（运行主项目）。应用程序将生成并运行。您的参数显示在一个外部窗口中。

提示 – 当您运行项目时，"Run Monitor"（运行监视器）选项卡将打开，并显示用于观察应用程序行为的分析工具。您可以在 "Project Properties"（项目属性）对话框中的 "Profile"（分析）类别中关闭这些分析工具。

基于现有源代码创建项目

对于 "C/C++/Fortran Project From Existing Sources"（基于现有源代码的 C/C++/Fortran 项目），IDE 可依靠现有的 makefile 获取有关如何编译和运行应用程序的说明。

1. 选择 "File"（文件）> "New Project"（新建项目）。
2. 选择 "C/C++/Fortran" 类别。
3. 选择 "C/C++/Fortran Project From Existing Sources"（基于现有源代码的 C/C++/Fortran 项目），然后单击 "Next"（下一步）。
4. 在 "New Project"（新建项目）向导的 "Select mode"（选择模式）页面上，单击 "Browse"（浏览）按钮。在 "Select Project Folder"（选择项目文件夹）对话框中，导航到源代码所在的目录。单击 "Select"（选择）。



5. 使用缺省配置模式 "Automatic"（自动）。单击 "Finish"（完成）。
6. 项目将在 "Projects"（项目）窗口中创建并打开，然后 IDE 会自动运行在现有 makefile 中指定的 "Clean"（清除）和 "Build"（生成）部分。项目还会自动配置以获取代码帮助。

项目将在 "Projects"（项目）窗口中创建并打开。您现已创建一个项目，是包装现有代码的一个瘦包装器。

生成和重新生成项目

生成项目：

- 右键单击项目的项目节点，然后选择 "Build"（生成）。

重新生成项目：

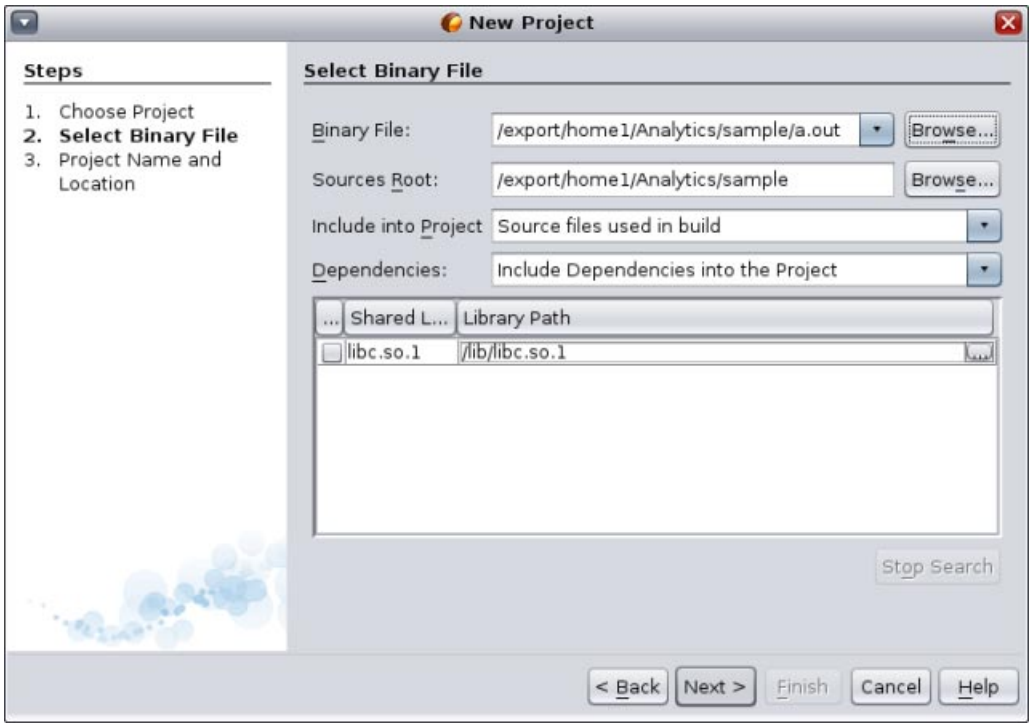
- 右键单击项目的项目节点，然后选择 "Clean and Build"（清除并生成）。

基于二进制文件创建项目

对于 "C/C++/Fortran project from a binary file"（基于二进制文件的 C/C++/Fortran 项目），您可以基于现有的二进制文件创建项目。

1. 选择 "File"（文件）> "New Project"（新建项目）。
2. 选择 "C/C++/Fortran" 类别。
3. 选择 "C/C++/Fortran Project from Binary File"（基于二进制文件的 C/C++/Fortran 项目），然后单击 "Next"（下一步）。
4. 在 "New Project"（新建项目）向导的 "Select Binary File"（选择二进制文件）页面上，单击 "Browse"（浏览）按钮。在 "Select Binary File"（选择二进制文件）对话框中，导航到要基于其创建项目的二进制文件。

   生成二进制文件时所基于的源文件的根目录是自动填充的。缺省情况下，项目中只包括生成二进制文件时所基于的源文件。缺省情况下，项目中包括相关项。项目所需的共享库会自动列出。



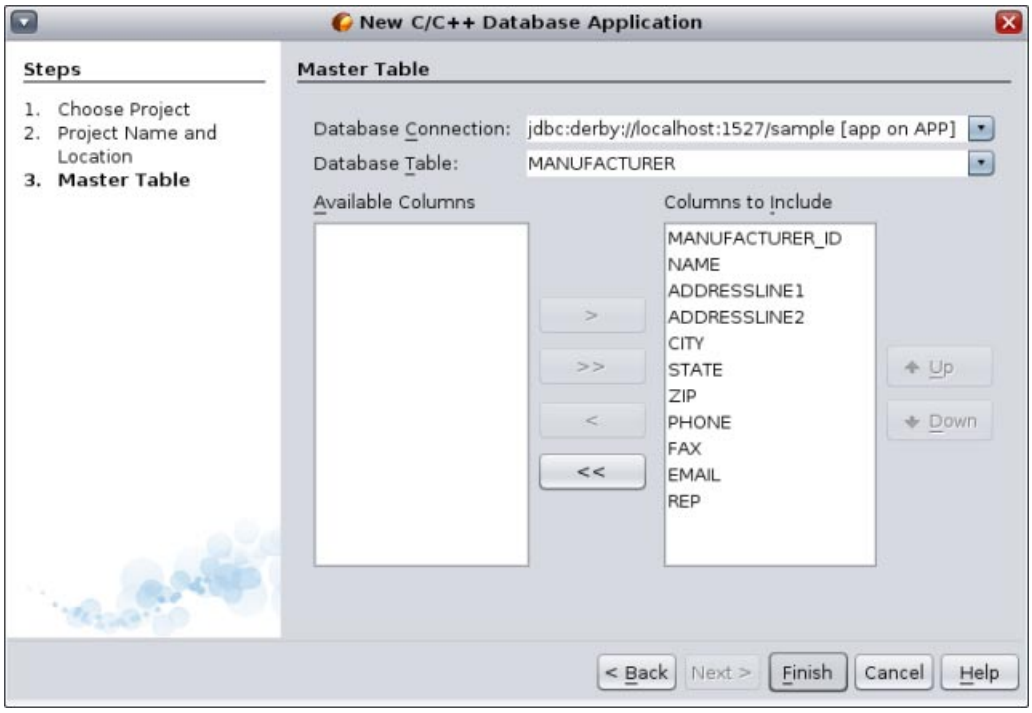
5. 单击 "Next"（下一步）。
6. 在 "Project Name and Location"（项目名称和位置）页面上，可以选择项目的名称和位置。单击 "Finish"（完成）。

创建 Oracle 数据库项目

您可以为 Oracle 数据库应用程序创建项目。要执行此操作，您使用的 Oracle Solaris Studio 安装必须包括可选的 Oracle Instant Client 组件。

1. 选择 "File"（文件）> "New Project"（新建项目）。
2. 在 "New Project"（新建项目）对话框中，选择 "C/C++/Fortran" 类别和 "C/C++ Database Application"（C/C++ 数据库应用程序）项目。单击 "Next"（下一步）。
3. 在 "Project Name and Location"（项目名称和位置）页面上，可以选择项目的名称和位置。单击 "Next"（下一步）。

4. 在 "Master table"（主表）页面上，从 "Database Connection"（数据库连接）下拉式列表中选择 `jdbc:derby://localhost:1527/sample`。IDE 将连接到数据库。从 "Database Table"（数据库表）下拉式列表中选择项目的主表。使用 "Available Columns"（可用列）和 "Columns to Include"（要包含的列）列表之间的方向键选择要在项目中包括的表列。



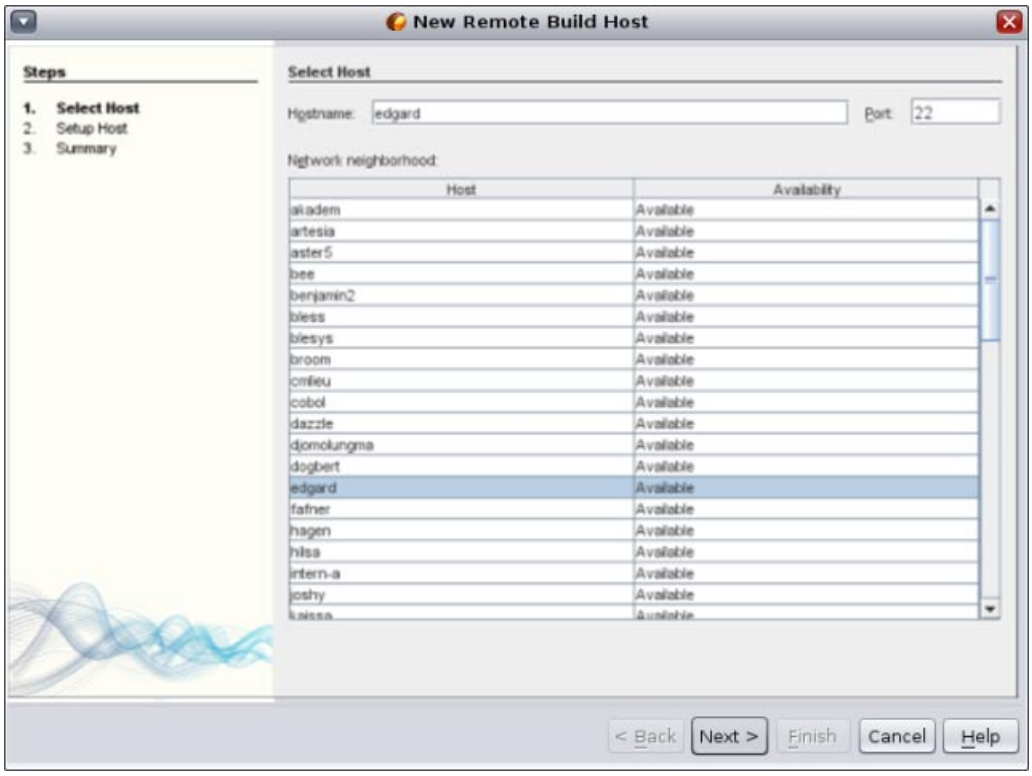
5. 单击 "Finish"（完成）。

执行远程开发

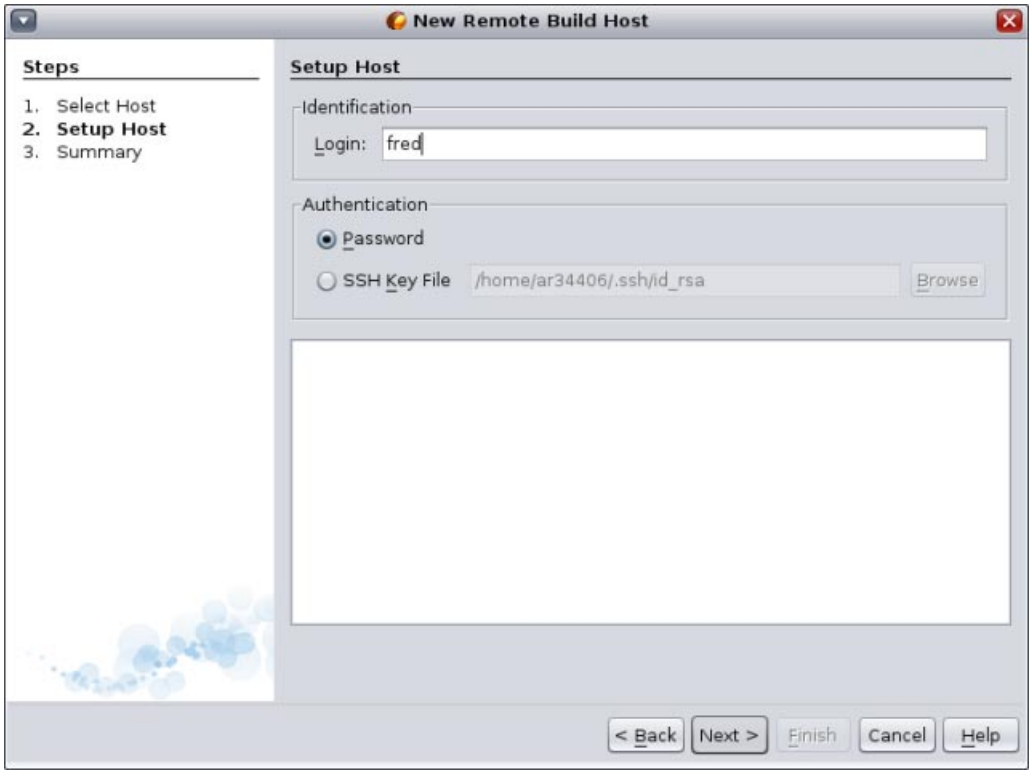
您可以在本地主机（从中启动 IDE 的系统）上生成、运行和调试项目，也可以在运行 UNIX® 操作系统的远程主机上生成、运行和调试项目。借助 "Remote development"（远程开发），您可以在熟悉的桌面环境中本地运行 IDE，同时使用远程服务器的计算能力和开发工具来生成您的项目。

您可以在 "Options"（选项）对话框的 "Build Tools"（生成工具）选项卡中配置远程开发主机。添加远程主机：

1. 选择 "Tools"（工具）> "Options"（选项），然后单击 "C/C++" 类别。
2. 在 "Options"（选项）对话框的 "Build Tools"（生成工具）选项卡中，单击 "Edit"（编辑）。
3. 在 "Build Hosts Manager"（生成主机管理器）对话框中，单击 "Add"（添加）。
4. 在 "New Remote Build Host"（新建远程生成主机）向导的 "Select Host"（选择主机）页面上，在 "Hostname"（主机名）字段中键入主机的系统名称，或者双击 "Network neighborhood"（网上邻居）列表中的某个可用主机以将其选中。单击 "Next"（下一步）。



5. 在 "Setup Host"（设置主机）页面上，在 "Login"（登录）字段中键入登录名称，然后单击 "Next"（下一步）。



6. 向导将提示您输入口令，连接到主机，并显示 "Summary"（摘要）页面。单击 "Finish"（完成）。

7. 在主机添加到 "Build Hosts Manager"（生成主机管理器）对话框中的 "Build Hosts"（生成主机）列表中后，单击 "OK"（确定）。
8. 您可以在 "Services"（服务）窗口中设置用来指定 IDE 如何使用远程主机的属性。展开 "C/C++ Build Hosts"（C/C++ 生成主机）节点，右键单击远程主机，然后选择 "Properties"（属性）。在 "Host Properties"（主机属性）对话框中设置所需的属性。
9. 要将远程主机设置为缺省生成主机，请在 "Services"（服务）窗口中右键单击 "C/C++ Build Hosts"（C/C++ 生成主机）节点中的主机，然后选择 "Set as Default"（设置为缺省）。

要在远程主机上开发项目，该项目必须位于在本地主机和远程主机上都可以看到的共享文件系统上。通常，这样的文件系统是使用 NFS 或 Samba 进行共享的。定义远程主机时，您可以在项目源文件的本地路径和远程路径之间定义映射。

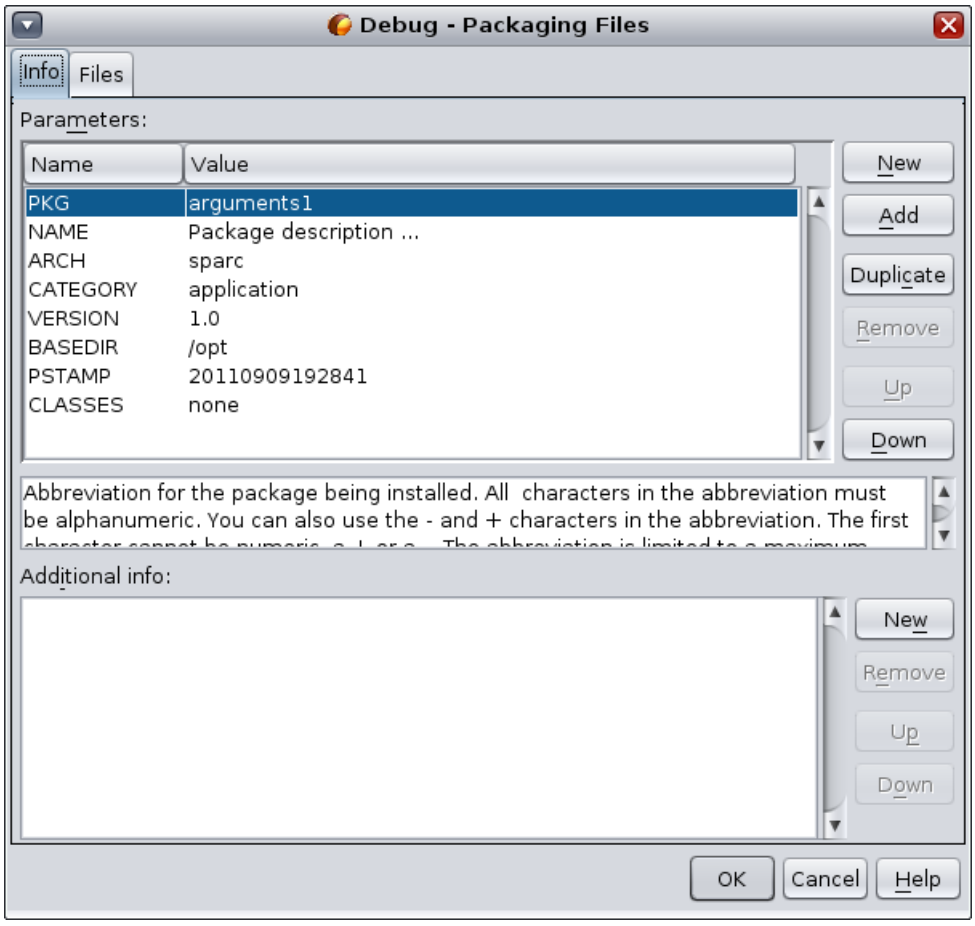
在创建项目时，缺省生成主机将被选作项目的生成主机。可以在 "Project Properties"（项目属性）对话框的 "Build"（生成）面板上更改项目的生成主机。还可以在调试可执行文件或信息转储文件时指定生成主机。

要在本地主机上处理位于远程主机上的项目，请选择 "File"（文件）> "Open Remote C/C++ Project"（打开远程 C/C++ 项目）。

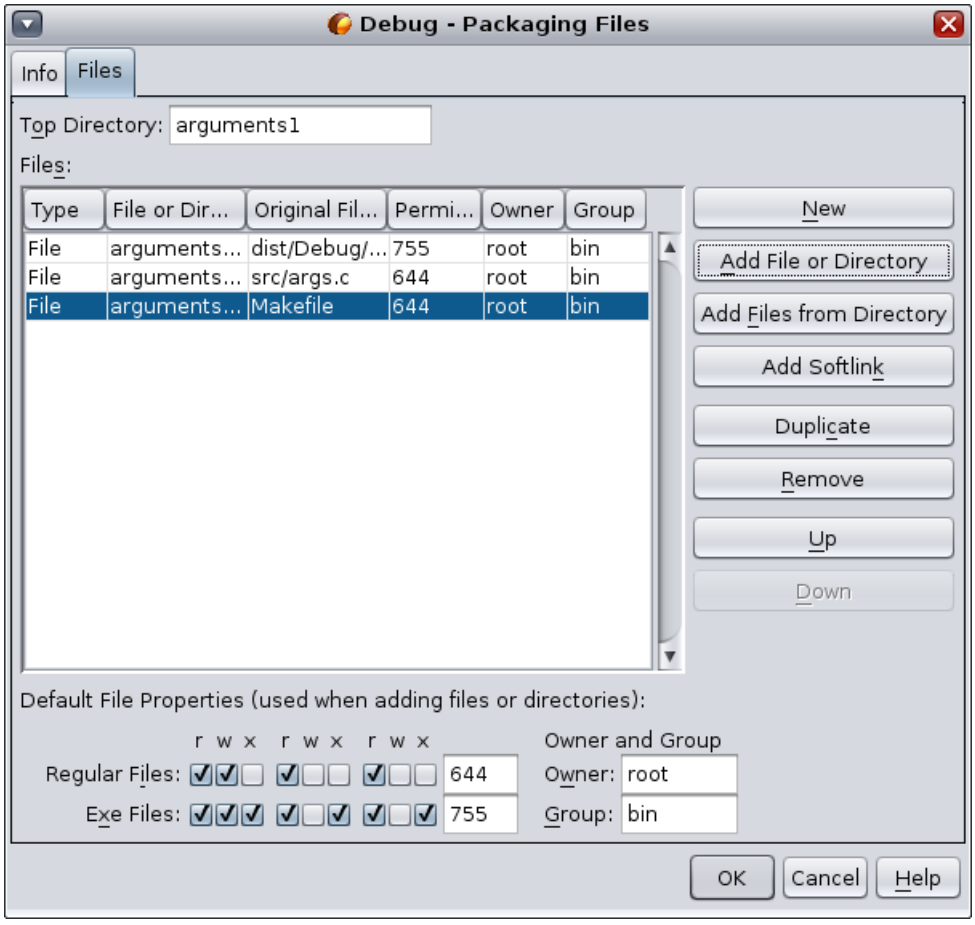
将应用程序打包

可以将完成的应用程序打包为 tar 文件、zip 文件、Solaris SVR4 软件包、RPM 或 Debian 软件包。

1. 右键单击 `Arguments_1` 项目，然后选择 "Properties"（属性）。
2. 在 "Project Properties"（项目属性）对话框中，选择 "Packaging"（打包）节点。
3. 从下拉式列表中选择 "Solaris SVR4" 软件包类型。
4. 如果想要为软件包使用其他目标目录或文件名，请更改输出路径。
5. 单击 "Packaging Files Browse"（打包文件浏览）按钮。在 "Packaging Files"（打包文件）对话框中（对于 SVR4 软件包），根据需要在 "Info"（信息）选项卡上修改软件包参数。



6. 对于所有软件包类型，请使用 "Files"（文件）选项卡上的按钮将文件添加到软件包中。对于每个文件，可以在 "Files"（文件）列表的 "File or Directory Path in Package"（软件包中的文件或目录路径）列中指定希望包含在软件包中的路径。"Files"（文件）列表完成后，单击 "OK"（确定）。



7. 如果希望通过单击复选框来完成，请关闭详细模式。
8. 单击 "OK"（确定）。
9. 要生成软件包，请右键单击项目，然后选择 "More Build Commands"（更多生成命令）> "Build Package"（生成软件包）。

编辑源文件

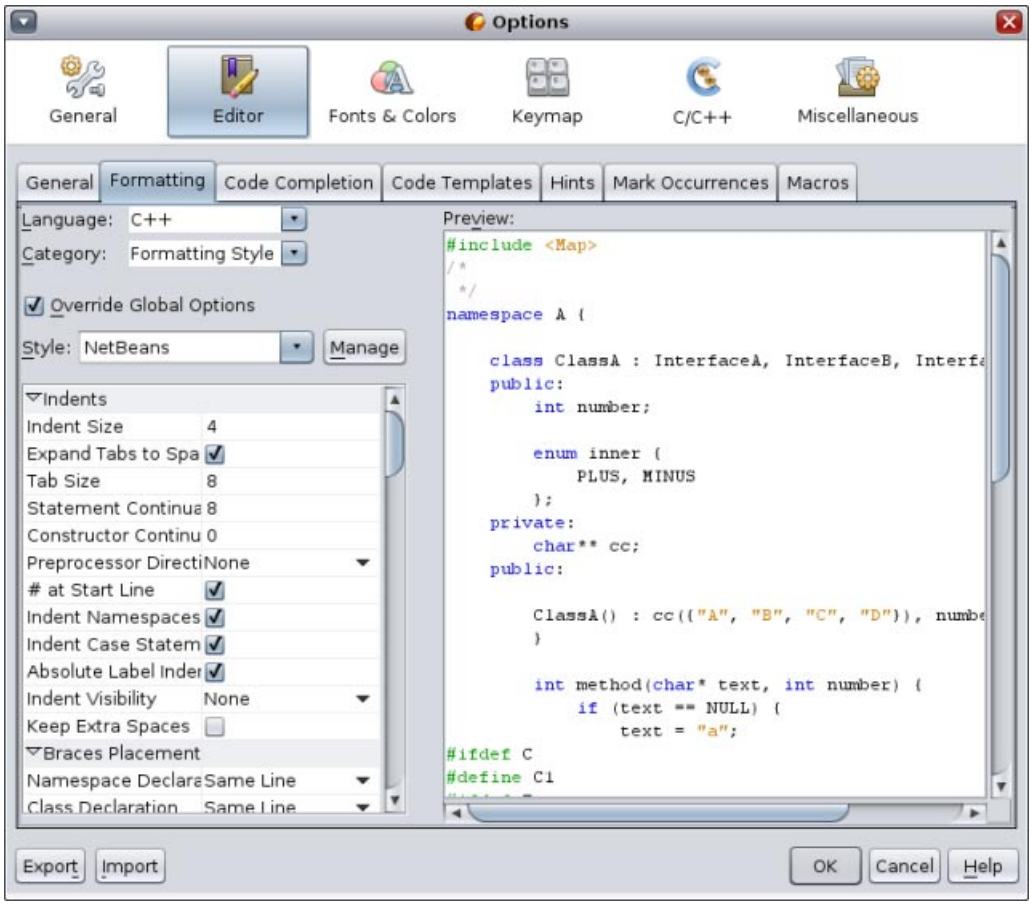
Oracle Solaris Studio IDE 提供了高级编辑和代码帮助功能，可帮助您查看和修改源代码。要了解这些功能，请使用 `Quote` 项目：

1. 选择 "File"（文件）> "New Project"（新建项目）。
2. 在项目向导中，展开 "Samples"（样例）类别和 "C/C++" 子类别，然后选择 `Quote` 项目。单击 "Next"（下一步），然后单击 "Finish"（完成）。

设置格式化样式

可以使用 "Options"（选项）对话框为项目配置缺省格式化样式。

1. 选择 "Tools"（工具）> "Options"（选项）。
2. 在对话框的顶部窗格中，单击 "Editor"（编辑器）。
3. 单击 "Formatting"（格式化）选项卡。
4. 从 "Language"（语言）下拉式列表中，选择您需要设置格式化样式的语言。
5. 从 "Style"（样式）下拉式列表中，选择您需要设置的样式。



6. 根据需要修改样式属性。

在 C 和 C++ 文件中折叠代码块

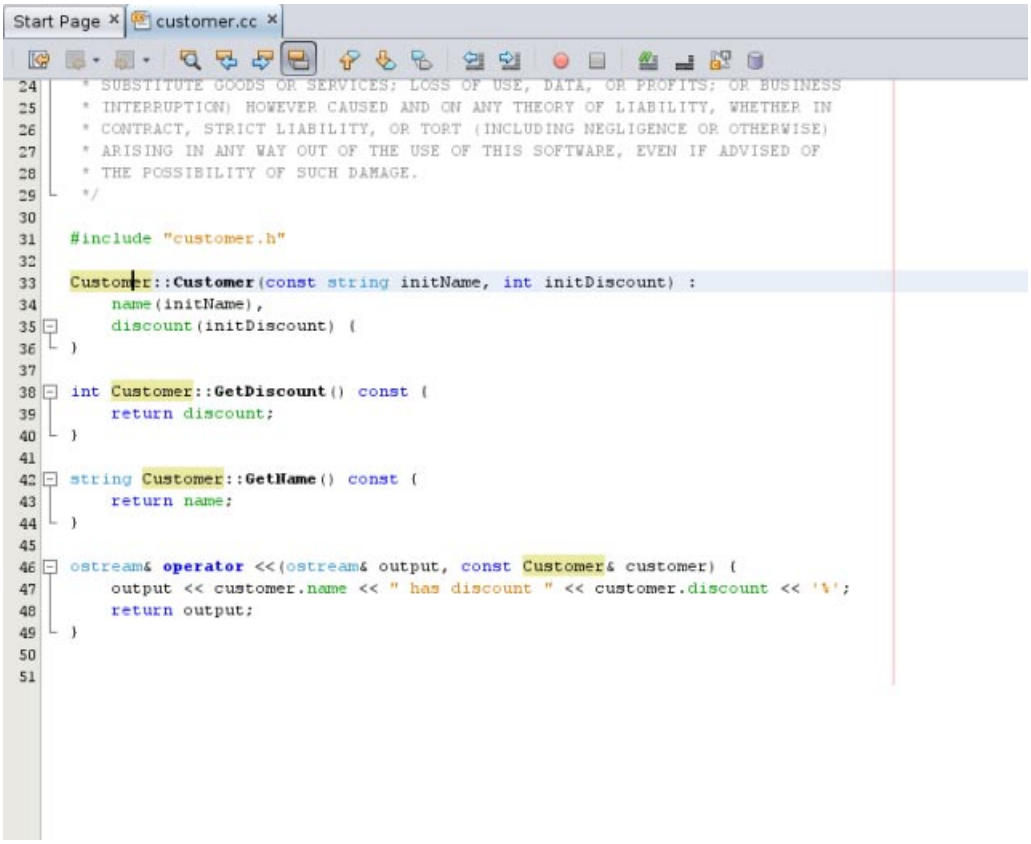
对于某些文件类型，您可以使用代码折叠功能折叠代码块，这样，只有块的第一行显示在源代码编辑器中。

1. 在 `Quote_1` 应用程序项目中，打开 "Source Files"（源文件）文件夹，然后双击 `cpu.cc` 文件以在源代码编辑器中将其打开。
2. 在左边界中单击折叠图标（带有减号的小框）可折叠其中一种方法的代码。
3. 将鼠标悬停到折叠块右侧的 {...} 符号上方可显示块中的代码。

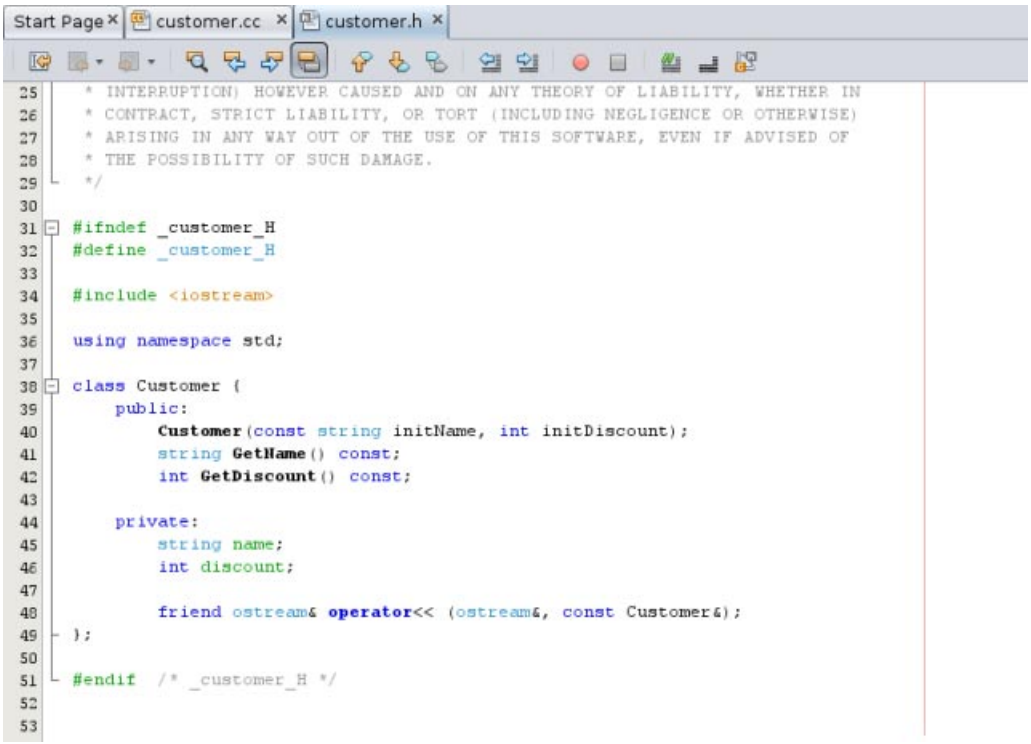
使用语义突出显示

您可以设置一个选项，以便当您单击某个类、函数、变量或宏时，该类、函数、变量或宏在当前文件中的每个实例都会突出显示。

1. 选择 "Tools"（工具）> "Options"（选项）。
2. 在对话框的顶部窗格中，单击 "C/C++"。
3. 单击 "Highlighting"（突出显示）选项卡。
4. 确保所有复选框都具有复选标记。
5. 单击 "OK"（确定）。
6. 在 `Quote_1` 项目的 `customer.cc` 文件中，请注意函数名称以粗体突出显示。
7. 单击 `Customer` 类的某个实例。
8. 文件中 `Customer` 类的所有实例都会以黄色背景突出显示。



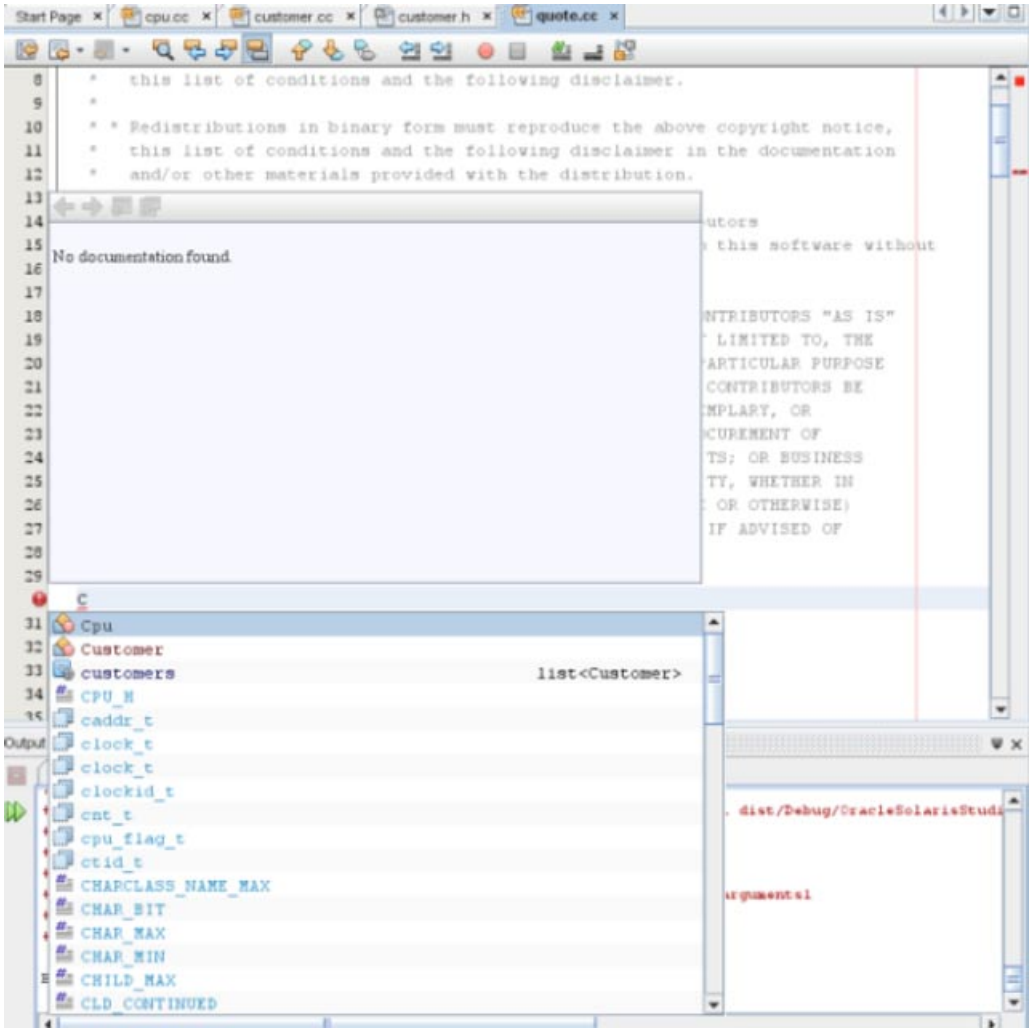
9. 在 customer.h 文件中，请注意类字段以粗体突出显示。



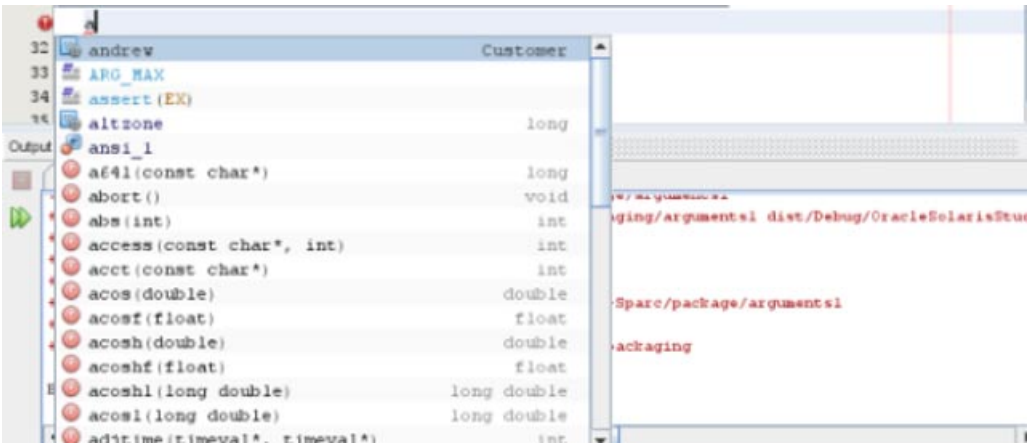
使用代码完成

IDE 有一个动态的 C 和 C++ 代码完成功能。通过该功能，您在键入一个或多个字符后，可以看到一个列表，其中列出了可用于完成表达式的可能的类、方法和变量等。

1. 打开 `Quote_1` 项目中的 `quote.cc` 文件。
2. 在 `quote.cc` 文件的第一个空白行上，键入大写的 C，然后按 Ctrl + 空格键。代码完成框将显示一个包含 `Cpu` 类和 `Customer` 类的简短列表。还会打开一个文档窗口，并显示消息 "No documentation found because the project source code does not include documentation"（找不到文档，因为项目源代码不包含文档）。
3. 通过再次按 Ctrl + 空格键，展开代码完成列表。



4. 从列表中选择标准库函数（如 `calloc()`），如果 IDE 可访问该函数的手册页，文档窗口将显示相应手册页。
5. 选择 `Customer` 类，然后按 Enter 键。
6. 通过键入 `andrew;` 完成 `Customer` 类的新实例。在下一行上，键入字母 `a`，然后按 Ctrl + 空格键。代码完成框将显示一个以字母 a 开头的选项列表（如方法参数、类字段和全局名称），这些选项可从当前上下文访问。



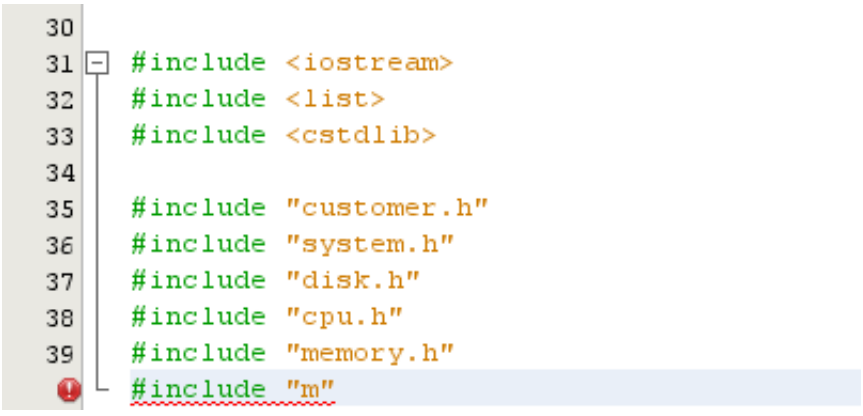
7. 双击 andrew 选项以接受它，然后在后面键入一个句点。系统会自动为您提供一个包含 Customer 类的公共方法和字段的列表。

8. 删除所添加的代码。

使用静态代码错误检查

当您在源代码编辑器中的源文件或头文件中键入代码时，编辑器会在您键入时执行静态代码错误检查，如果它检测到错误，则它会在左边界处显示一个错误图标  。

1. 在 Quote_1 项目的 quote.cc 文件中，在第 40 行键入 #include "m，则会看到一个错误图标出现在边界处。

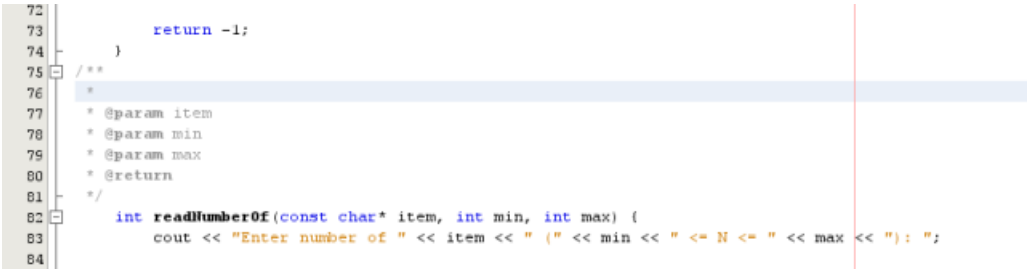


2. 退格至第二个引号，然后键入 odule.h" 来完成此语句，您会看到错误图标在语句引用现有的头文件后立即消失。
3. 删除所添加的语句。

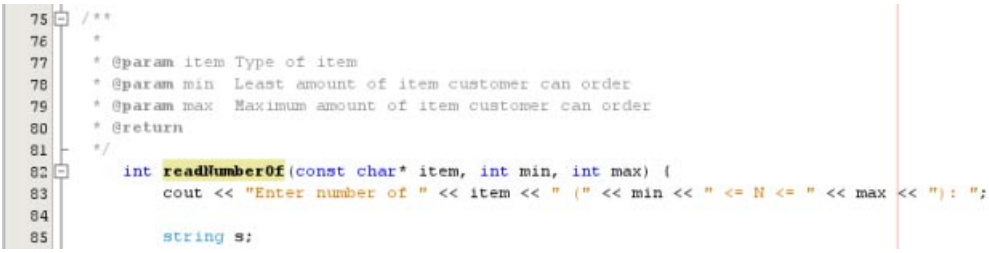
添加源代码文档

您可以向代码添加注释，以便为函数、类和方法生成文档。IDE 可识别使用 Doxygen 语法的注释，并自动生成文档。还可以自动生成注释块来记录注释下方的函数。

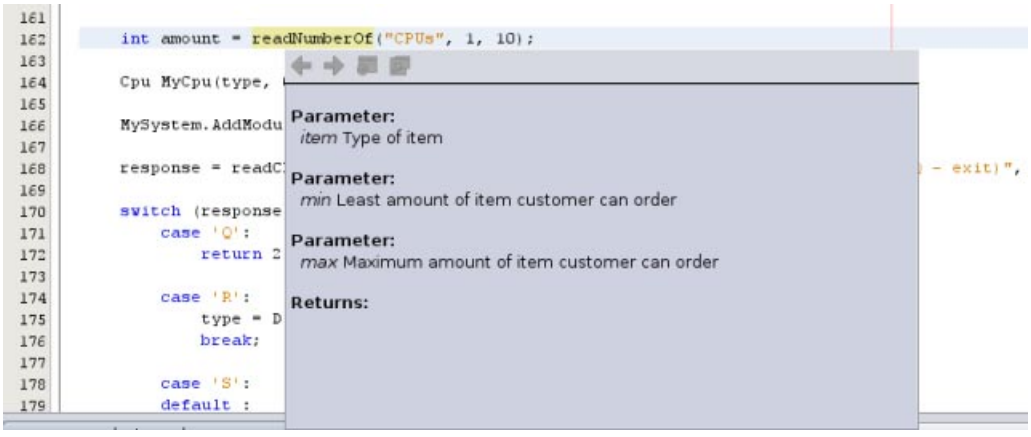
1. 在 `quote.cc` 文件中，将光标置于行 `int readNumberOf(const char* item, int min, int max) {` 上面的行上。
2. 键入一个斜杠和两个星号，然后按 Enter 键。编辑器将为 `readNumberOf` 类插入一个 Doxygen 格式的注释。



3. 向每个 `@param` 行添加一些描述性文本，然后保存文件。
4. 单击 `readNumberOf` 类将其以黄色突出显示，然后单击右侧的其中一个标记实例跳到使用该类的某个位置。



5. 单击所跳到的行中的 `readNumberOf` 类，然后按 Ctrl + Shift + 空格键显示刚才为这些参数添加的文档。



6. 单击文件中的任何其他地方关闭文档窗口，然后再次单击 `readNumberOf` 类。
7. 选择 "Source"（源）> "Show documentation"（显示文档）再次为该类打开文档窗口。

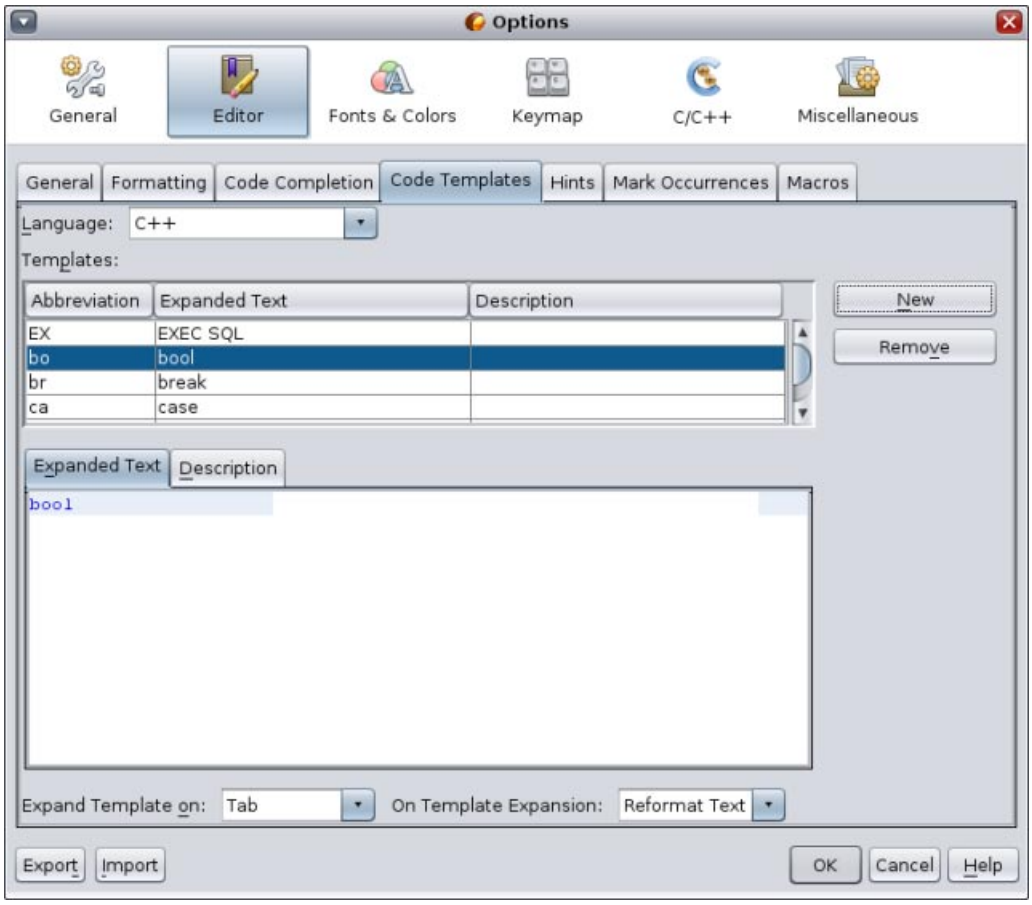
使用代码模板

源代码编辑器有一组用于 C、C++ 和 Fortran 代码的常见代码片段的可定制代码模板。您可以通过键入缩写然后按 Tab 键来生成完整的代码片段。例如，在 `Quote_1` 项目的 `quote.cc` 文件中：

- 键入 `uns` 然后按 Tab 键，`uns` 会展开为 `unsigned`。
- 键入 `iff` 然后按 Tab 键，`iff` 会展开为 `if (exp) {}`。
- 键入 `ifs` 然后按 Tab 键，`ifs` 会展开为 `if (exp) {} else {}`。
- 键入 `fori` 然后按 Tab 键，`fori` 会展开为 `for (int i=0; i< size; i++) { Object size = array[i] ; }`。

要看到所有可用的代码模板，修改这些模板，创建自己的代码模板，或选择其他键来展开模板，请执行下列操作：

1. 选择 "Tools"（工具）> "Options"（选项）。
2. 在 "Options"（选项）对话框中，选择 "C/C++"，然后单击 "Code Templates"（代码模板）选项卡。
3. 从 "Language"（语言）下拉式列表中选择一种语言。



使用配对完成

当您编辑 C 和 C++ 源文件时，源代码编辑器会对成对字符（如括号、圆括号和引号）进行智能匹配。当您键入这些字符中的其中一个时，源代码编辑器会自动插入封闭字符。

1. 在 `Quote_1` 项目中，将光标放到 `module.cc` 文件的第 116 行上的 `{` 后面，然后按 Return 键打开一个新行。
2. 键入 `enum state {`，然后按 Return 键，将自动添加封闭花括号和分号，且光标位于两个花括号之间。

3. 键入 `invalid=0, success=1` 来完成枚举。
4. 在该行上枚举的封闭 `};` 后面，键入 `if (`，将自动添加封闭圆括号，且光标位于两个圆括号之间。
5. 键入 `v==null`。在右圆括号后键入 `i` 并换行，将自动添加封闭括号。
6. 删除所添加的代码。

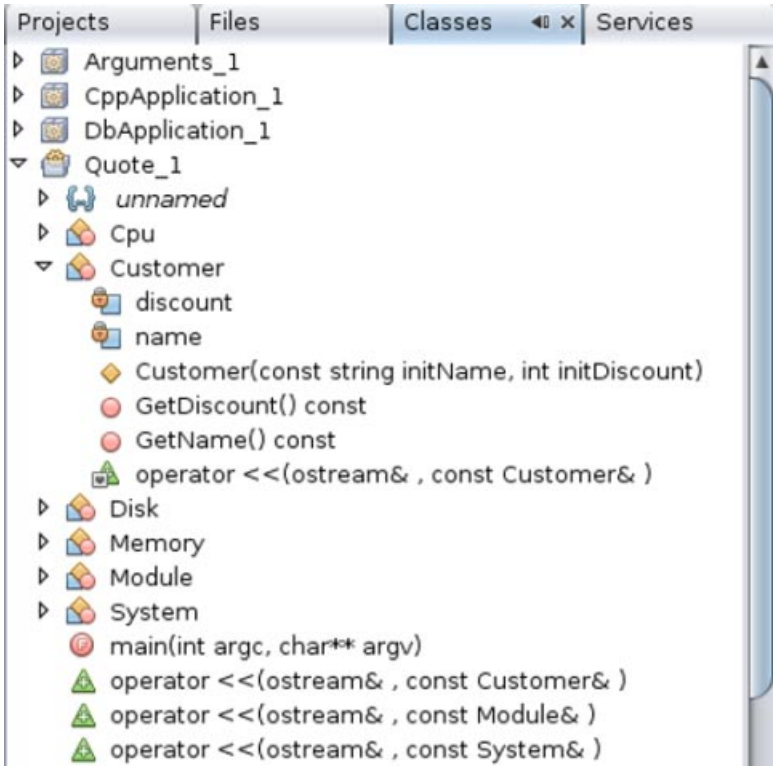
导航源文件

IDE 提供了用于查看源代码的高级导航功能。要了解这些功能，请继续使用 `Quote_1` 项目。

使用 "Classes"（类）窗口

使用 "Classes"（类）窗口可看到项目中的所有类以及每个类的成员和字段。

1. 单击 "Classes"（类）选项卡可显示 "Classes"（类）窗口。
2. 展开 `Quote_1` 节点。会列出项目中的所有类。
3. 展开 `Customer` 类。

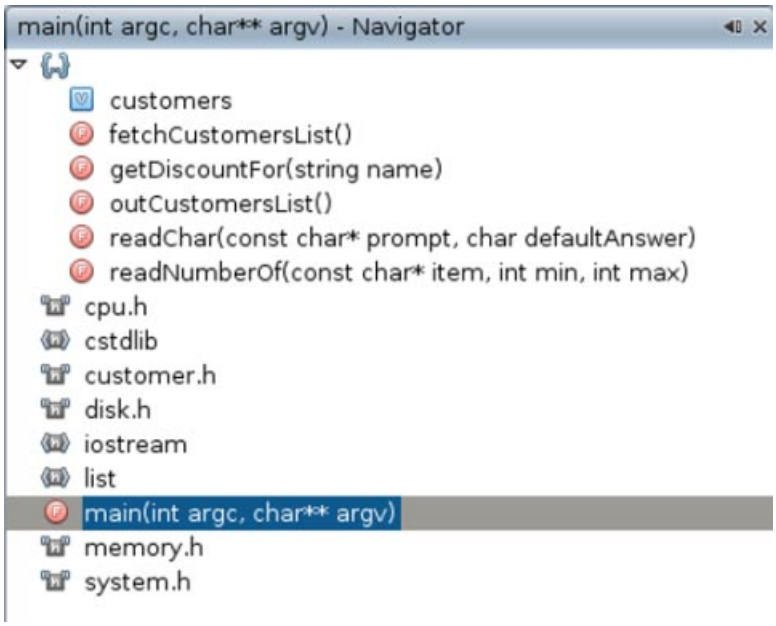


4. 双击 `name` 变量可打开 `customer.h` 头文件。

使用 "Navigator"（导航器）窗口

"Navigator"（导航器）窗口提供了当前选定文件的紧凑视图，并简化了在文件的不同部分之间的导航。如果 "Navigator"（导航器）窗口未打开，请选择 "Window"（窗口）> "Navigating"（导航）> "Navigator"（导航器）打开该窗口。

1. 在 "Editor"（编辑器）窗口中，单击 `quote.cc` 文件的任意位置。
2. 在 "Navigator"（导航器）窗口中会显示该文件的紧凑视图。单击窗口顶部的节点可展开该视图。

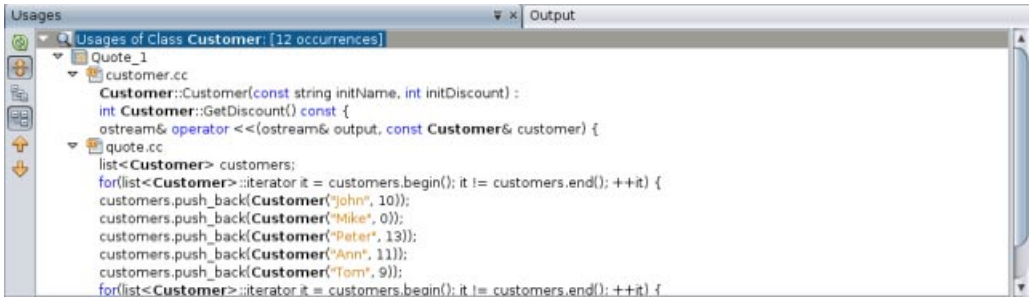


3. 要导航到文件的某个元素，请在 "Navigator"（导航器）窗口中双击该元素，"Editor"（编辑器）窗口中的光标将移到该元素。
4. 在 "Navigator"（导航器）窗口中右键单击，可看到用于在窗口中对元素进行排序、对项目进行分组或过滤的选项。
5. 要查看 "Navigator"（导航器）窗口的图标所表示的意义，请通过选择 "Help"（帮助）> "Help Contents"（帮助内容）打开 IDE 联机帮助。在 "Help"（帮助）浏览器中，单击 "Search"（搜索）选项卡，然后在 "Find"（查找）字段中键入 `navigator icons`（导航器图标）。

查找类、方法和字段使用实例

您可以使用 "Usages"（使用实例）窗口显示在项目的源代码中使用类（结构）、函数、变量、宏或文件的所有位置。

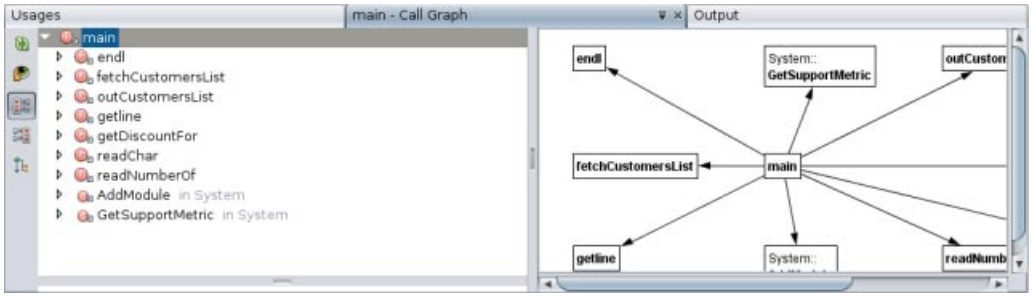
1. 在 `customer.cc` 文件中，右键单击第 42 行上的 `Customer` 类，然后选择 "Find Usages"（查找使用实例）。
2. 在 "Find Usages"（查找使用实例）对话框中，单击 "Find"（查找）。
3. "Usages"（使用实例）窗口将打开，并显示项目的源文件中 `Customer` 类的所有使用实例。



使用调用图

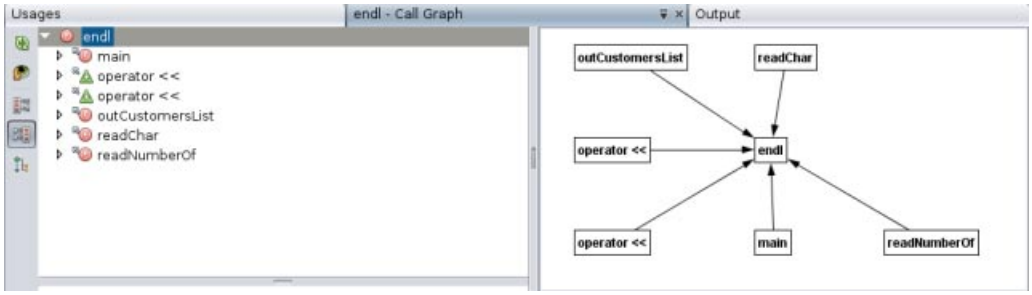
"Call Graph"（调用图）窗口显示类中函数之间的调用关系的两个视图。树视图显示从某个选定函数调用的函数，或者调用该选定函数的函数。图形视图使用箭头来显示被调用函数与调用函数之间的调用关系。

1. 在 `quote.cc` 文件中，右键单击 main 函数，然后选择 "Show Call Graph"（显示调用图）。
2. "Call Graph"（调用图）窗口将打开，并显示从 `main` 函数调用的所有函数的树视图和图形视图。

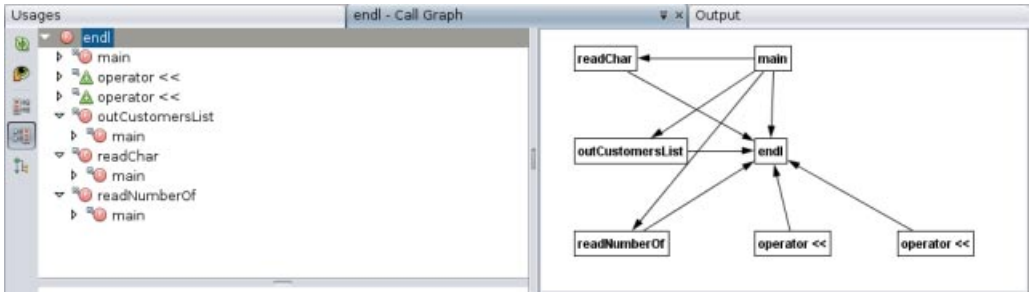


如果屏幕抓图中没有显示所有函数，请单击 "Call Graph"（调用图）窗口左侧的第三个按钮以显示哪个函数是从此函数调用的。

3. 展开 `end1` 节点以显示该函数所调用的函数。请注意图形已经更新，添加了 `end1` 所调用的函数。
4. 选择 `end1` 节点，单击窗口左侧的第二个按钮以将焦点放到 `endl` 函数上，然后单击第四个按钮来查看调用 `end1` 函数的所有函数。



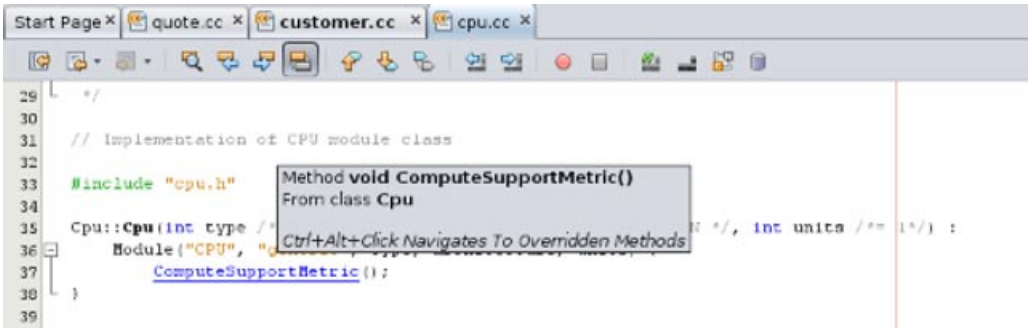
5. 展开树中的某些节点可看到更多函数。



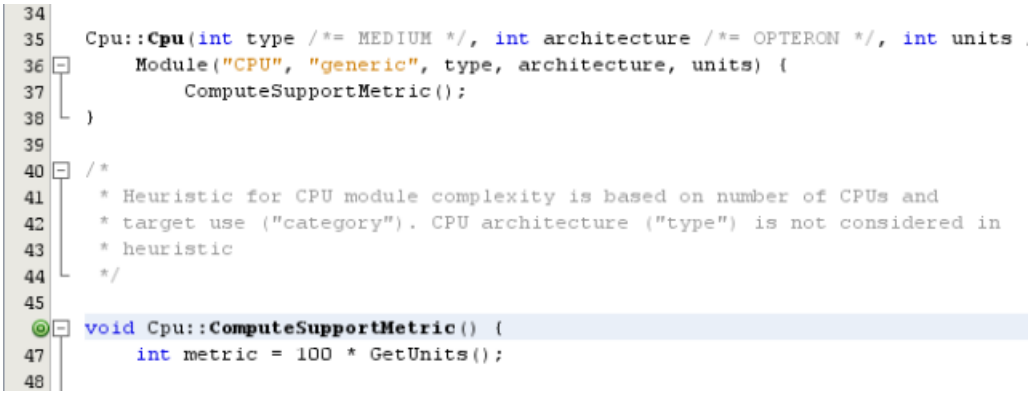
使用超级链接

使用超级链接导航，可以从类、方法、变量或常量的调用跳到声明，也可以从声明跳到定义。使用超级链接，还可以从某个被覆盖的方法跳到覆盖它的方法，反之亦然。

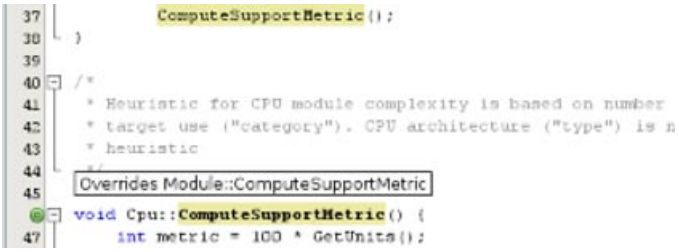
1. 在 `Quote_1` 项目的 `cpu.cc` 文件中，在按 Ctrl 键的同时将鼠标移到第 37 行上。将突出显示 `ComputeSupportMetric` 函数，还会有一条注释显示有关该函数的信息。



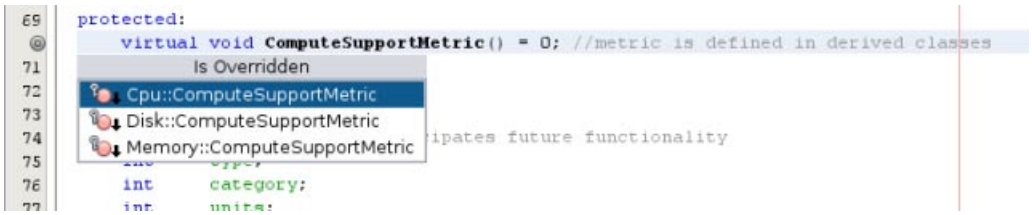
2. 单击超级链接，编辑器将跳到函数的定义。



3. 在按住 Ctrl 键的同时将鼠标移到定义上方，然后单击超级链接。编辑器将跳到 `cpu.h` 头文件中函数的声明。
4. 单击编辑器工具栏中的左箭头，编辑器将跳回到 `cpu.cc` 中的定义。
5. 将鼠标光标悬停在左边界处的绿色圆圈  上，将会看到指出此方法覆盖另一方法的注释。



6. 单击绿色圆圈可转到被覆盖的方法，编辑器会跳到 `module.h` 头文件，该头文件在边界中显示一个灰色圆圈，表示该方法已被覆盖。
7. 单击灰色圆圈，编辑器会显示覆盖此方法的方法列表。

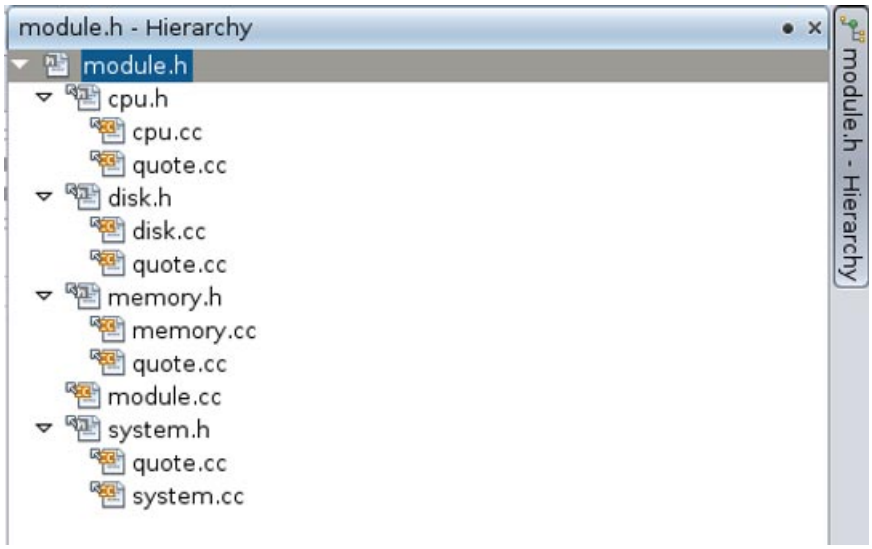


8. 单击 `Cpu::ComputerSupportMetric` 项，编辑器会跳回到 `cpu.h` 头文件中该方法的声明。

使用包含分层结构

使用 "Include Hierarchy"（包含分层结构）窗口，可以检查直接或间接包含在源文件中的所有头文件和源文件，或者直接或间接包含头文件的所有源文件和头文件。

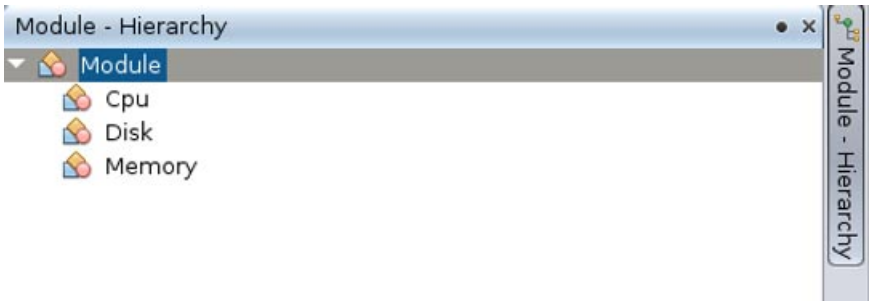
1. 在 `Quote_1` 项目中，在源代码编辑器中打开 `module.cc` 文件。
2. 右键单击文件中的 `#include "module.h"` 行，然后选择 "Navigate"（导航）> "View Includes Hierarchy"（查看包含分层结构）。
3. 缺省情况下，"Hierarchy"（分层结构）窗口显示直接包含头文件的文件的普通列表。单击窗口底部最右侧的按钮可将显示更改为树视图。单击右侧第二个按钮可将显示更改为包含的所有文件或被包含的所有文件。展开树视图中的节点可看到包含头文件的所有源文件。



使用类型分层结构 ）

使用 "Type Hierarchy"（类型分层结构）窗口，可以检查类的所有子类型或父类型。

1. 在 `Quote_1` 项目中，打开 `module.h` 文件。
2. 右键单击 `Module` 类的声明，然后选择 "Navigate"（导航）> "View Type Hierarchy"（查看类型分层结构）。
3. "Hierarchy"（分层结构）窗口将显示 `Module` 类的所有子类型。

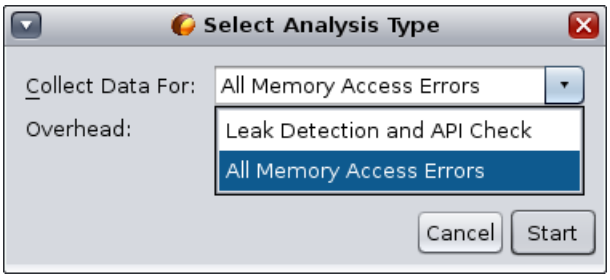


对项目运行内存访问检查

您可以使用内存分析工具查找项目中的内存访问错误。使用该工具，您可以定位源代码中出现的各个错误的确切位置，从而轻松地找到这些错误。

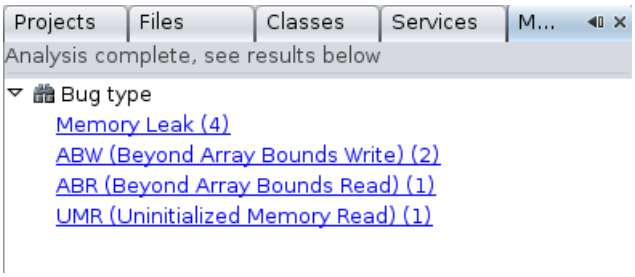
内存分析工具会在程序执行期间动态捕捉并报告内存访问错误，因此，如果您的代码的某个部分在运行时未执行，则不会报告该部分的错误。

1. 如果您尚未执行此操作，请从位于 `http://www.oracle.com/technetwork/server-storage/solarisstudio/downloads/solaris-studio-samples-1408618.html` 的 Oracle Solaris Studio 12.3 Sample Applications（Oracle Solaris Studio 12.3 样例应用程序）Web 页面下载样例应用程序 zip 文件，然后将该文件解压缩到您选择的位置中。`memorychecks` 应用程序位于 `SolarisStudioSampleApplications` 目录的 `CodeAnalyzer` 子目录中。
2. 使用 `memorychecks` 应用程序基于现有源代码创建项目。
3. 右键单击项目，然后选择 "Properties"（属性）。在 "Project Properties"（项目属性）对话框中，选择 "Run"（运行）节点，然后在 "Run Command"（运行命令）的输出路径后面键入 Customer.db。单击 "OK"（确定）。
4. 运行项目。
5. 现在在采用内存分析检测的情况下生成项目。
   a. 确保您的 `memorychecks` 项目设置为主项目。
   b. 单击 "Profile Project"（分析项目）按钮旁边的向下箭头  右侧的向下箭头，然后从下拉式列表中选择 "Profile Project to find Memory Access Errors"（分析项目以查找内存访问错误）。
   c. 在 "Select Analysis Type"（选择分析类型）对话框中，从下拉式列表中选择 "All Memory Access Errors"（所有内存访问错误）。

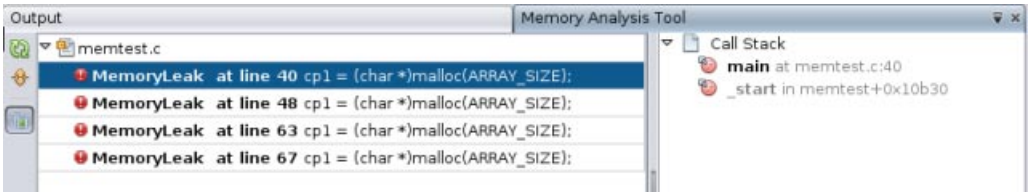


   "Overhead"（开销）字段中会显示 "High"（高）或 "Moderate"（中等）来表示系统将承受的负载。当系统开销很大时（例如，您同时检测数据争用和死锁时），在系统上运行的其他程序的性能可能会受到影响。
   d. 单击 "Start"（启动）。
6. 会打开 "Run Memory Profile"（运行内存分析）对话框，让您知道将对您的二进制文件进行检测。单击 "OK"（确定）。

7. 此时将生成并检测项目。应用程序开始运行并且 "Memory Analysis"（内存分析）窗口将打开。项目运行完成时，"Memory Analysis"（内存分析）窗口会列出在项目中找到的内存访问错误类型。在错误类型后面会在括号中显示每种类型的错误数。



8. 单击某一错误类型时，该类型的错误将显示在 "Memory Analysis Tool"（内存分析工具）窗口中。



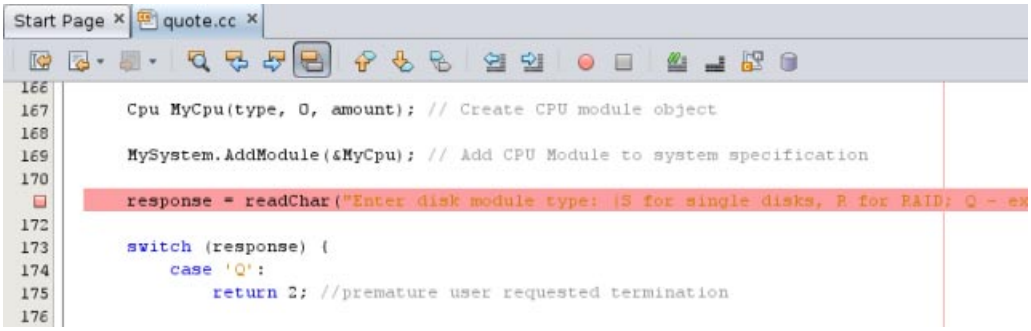
缺省情况下，错误按它们所在的源文件分组。单击某一错误时，将显示该错误的调用堆栈。双击堆栈中的某个函数调用可显示源文件中的相关行。

创建断点

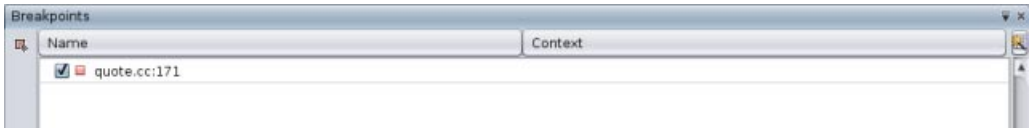
您可以随时创建和处理代码中的断点。

创建和移除行断点

1. 在 `Quote_1` 项目中，打开 `quote.cc` 文件。
2. 通过在 "Editor"（编辑器）窗口左边界中的第 171 行 (`response = readChar("Enter disk module type: (S for single disks, R for RAID; Q - exit)", 'S');`) 的旁边单击，设置一个行断点。该行将以红色突出显示，表示已设置断点。

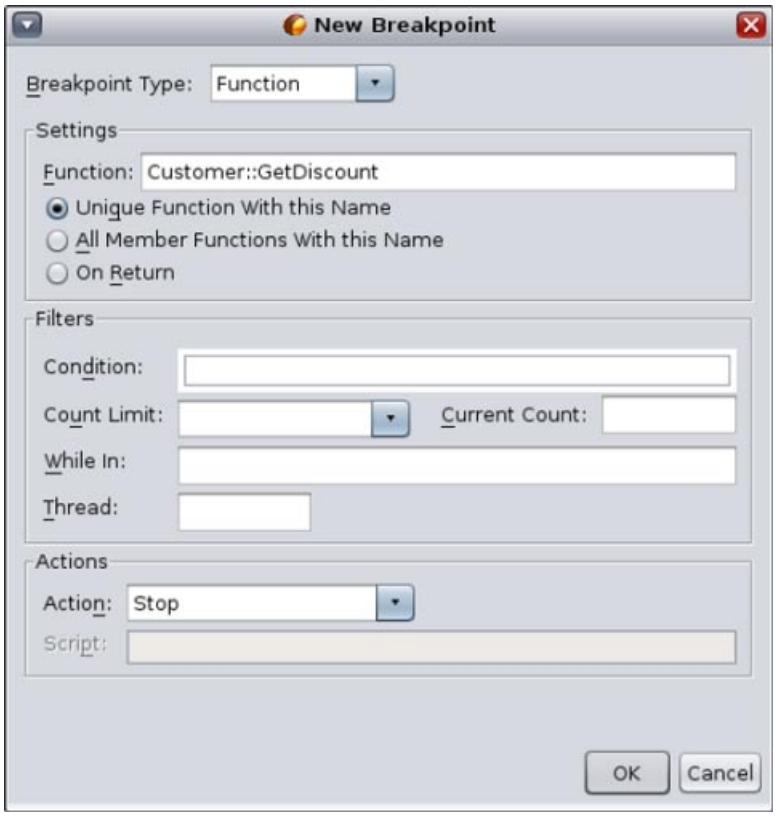


3. 您可以通过单击左边界中的图标移除断点。
4. 选择 "Window"（窗口）> "Debugging"（调试）> "Breakpoints"（断点）可打开 "Breakpoints"（断点）窗口。该窗口中会列出您的行断点。

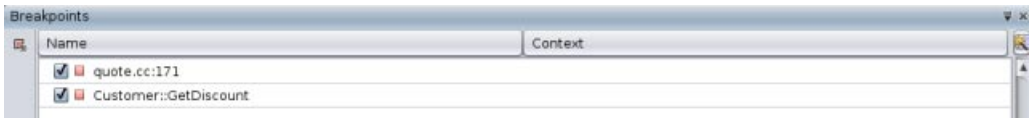


创建函数断点

1. 选择 "Debug"（调试）> "New Breakpoint"（新建断点）(Ctrl+Shift+F8) 可打开 "New Breakpoint"（新建断点）对话框。
2. 在 "Breakpoint Type"（断点类型）下拉式列表中，将类型设置为 "Function"（函数）。
3. 在 "Function"（函数）文本字段中，键入函数名称 `Customer::GetDiscount`。单击 "OK"（确定）。



4. 您的函数断点现已设置完毕，并添加到了 "Breakpoints"（断点）窗口中的列表中。

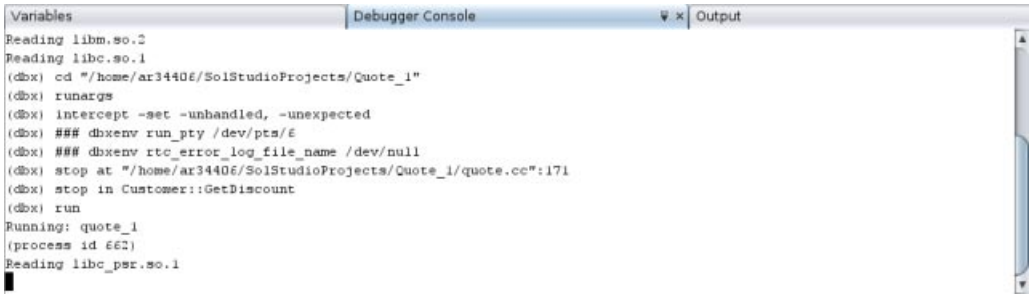


调试项目

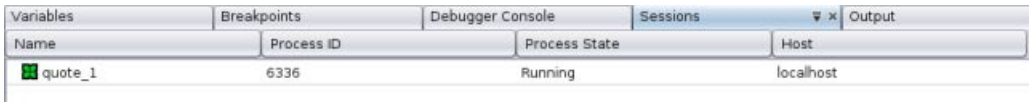
当您启动调试会话时，IDE 将启动项目的关联工具集中的调试器（缺省情况下为 dbx 调试器），然后在该调试器内运行应用程序。IDE 会自动打开调试器窗口，并将调试器输出内容输出到 "Debugger Console"（调试器控制台）窗口。

启动调试会话

1. 通过右键单击项目节点然后选择 "Debug"（调试），可为 Quote_1 项目启动一个调试会话。调试器将启动，应用程序开始运行，同时 "Variables"（变量）和 "Debugger Console"（调试器控制台）窗口将打开。

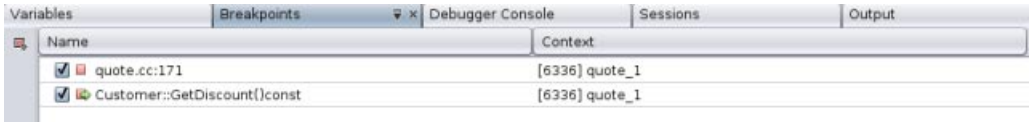


2. 通过选择 "Window"（窗口）> "Debugging"（调试）> "Sessions"（会话）可打开 "Sessions"（会话）窗口。调试会话将显示在此窗口中。



检查应用程序的状态

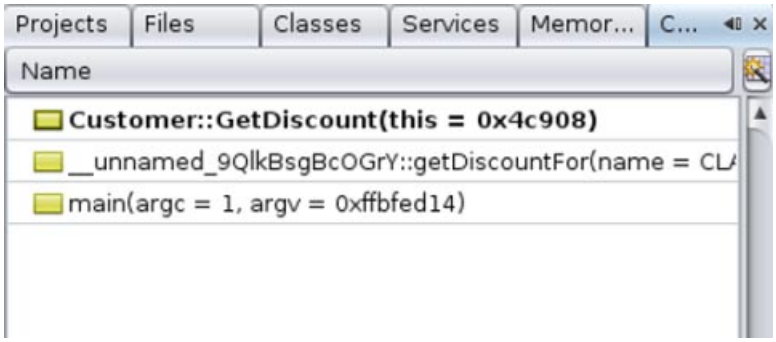
1. Quote_1 应用程序在 "Output"（输出）窗口中提示您进行输入。
2. 在 Enter customer name: 提示符后输入客户名称。
3. 应用程序将在先前设置的函数断点处停止。"Breakpoints"（断点）窗口列出了先前设置的两个断点。绿色程序计数器箭头显示在函数断点的断点图标顶部。



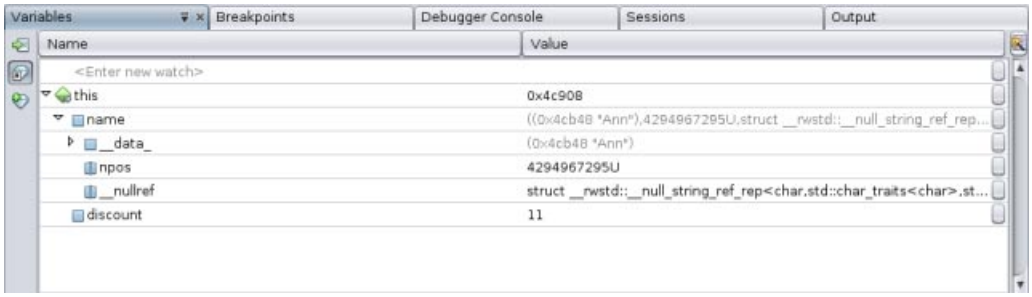
4. 在 customer.cc 文件中，绿色程序计数器箭头显示在 GetDiscount 函数第一行上的断点图标顶部。

```
30
31  #include "customer.h"
32
33  Customer::Customer(const string initName, int initDiscount) :
34      name(initName),
35      discount(initDiscount) {
36  }
37
38  int Customer::GetDiscount() const {
39      return discount;
40  }
41
42  string Customer::GetName() const {
43      return name;
44  }
45
```

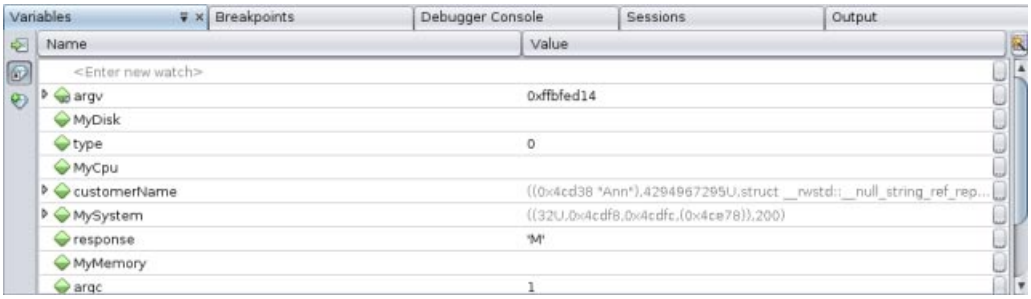
5. 打开 "Call Stack"（调用堆栈）窗口。调用堆栈将显示三个帧。



6. 单击 "Variables"（变量）窗口，请注意会显示一个变量。单击节点可展开结构。



7. 单击 "Continue"（继续）按钮。`GetDiscount` 函数将执行，它将客户折扣输出到 "Output"（输出）窗口中。然后系统会提示您进行输入。
8. 请根据提示输入内容。程序会在下一个断点（先前设置的行断点）处停止。单击 "Variables"（变量）窗口，注意长长的局部变量列表。



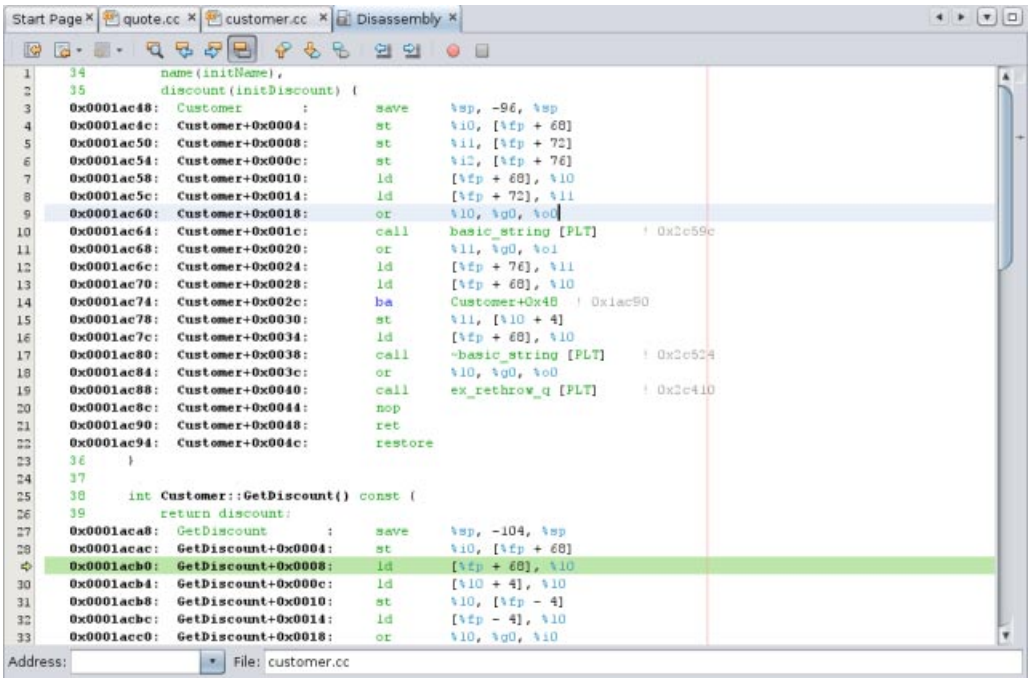
9. 查看 "Call Stack"（调用堆栈）窗口，您会看到，堆栈中只有一个帧。

10. 单击 "Continue"（继续）  ，根据 "Output"（输出）窗口中的提示继续输入内容，直到程序完成。将最后的输入内容输入到程序中后，调试会话将结束。要在程序执行完成之前结束调试会话，可以在 "Sessions"（会话）窗口中右键单击该会话，然后选择 "Finish"（完成）。

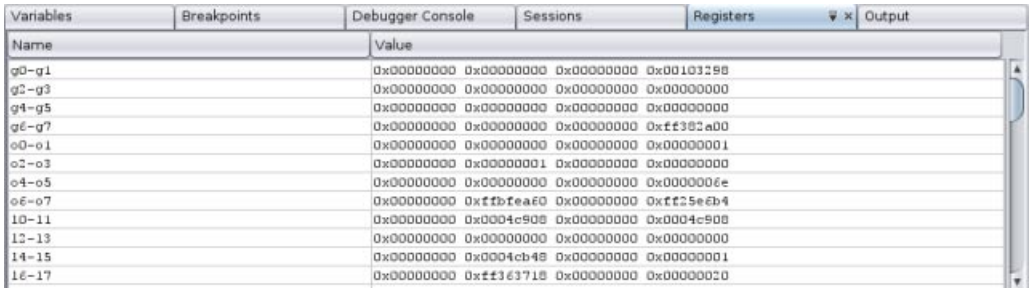
在机器指令级调试

调试器提供了用于在机器指令级调试项目的窗口。

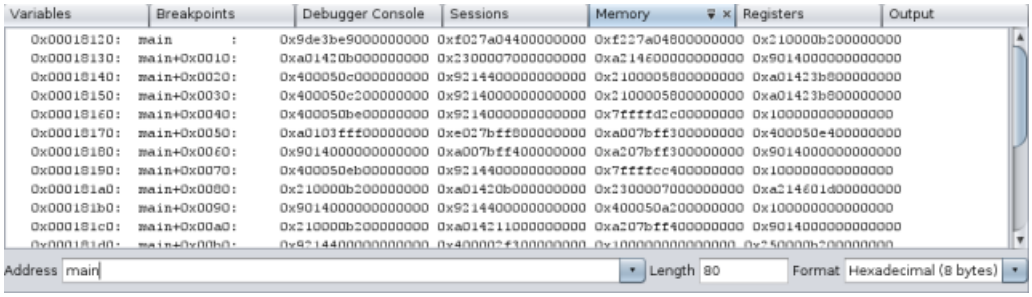
1. 右键单击 `Quote_1` 项目，然后选择 "Debug"（调试）。
2. 在 "Output"（输出）窗口中，根据提示输入客户名称。
3. 当程序在 `GetDiscount` 函数上的断点处暂停时，选择 "Window"（窗口）> "Debugging"（调试）> "Disassembly"（反汇编）可打开 "Disassembly"（反汇编）窗口，如同在 "Editor"（编辑器）窗口中一样。绿色程序计数器箭头将显示在暂停程序的指令处的断点图标顶部。



4. 选择 "Window"（窗口）> "Debugging"（调试）> "Registers"（寄存器）可打开 "Registers"（寄存器）窗口，该窗口显示寄存器的内容。



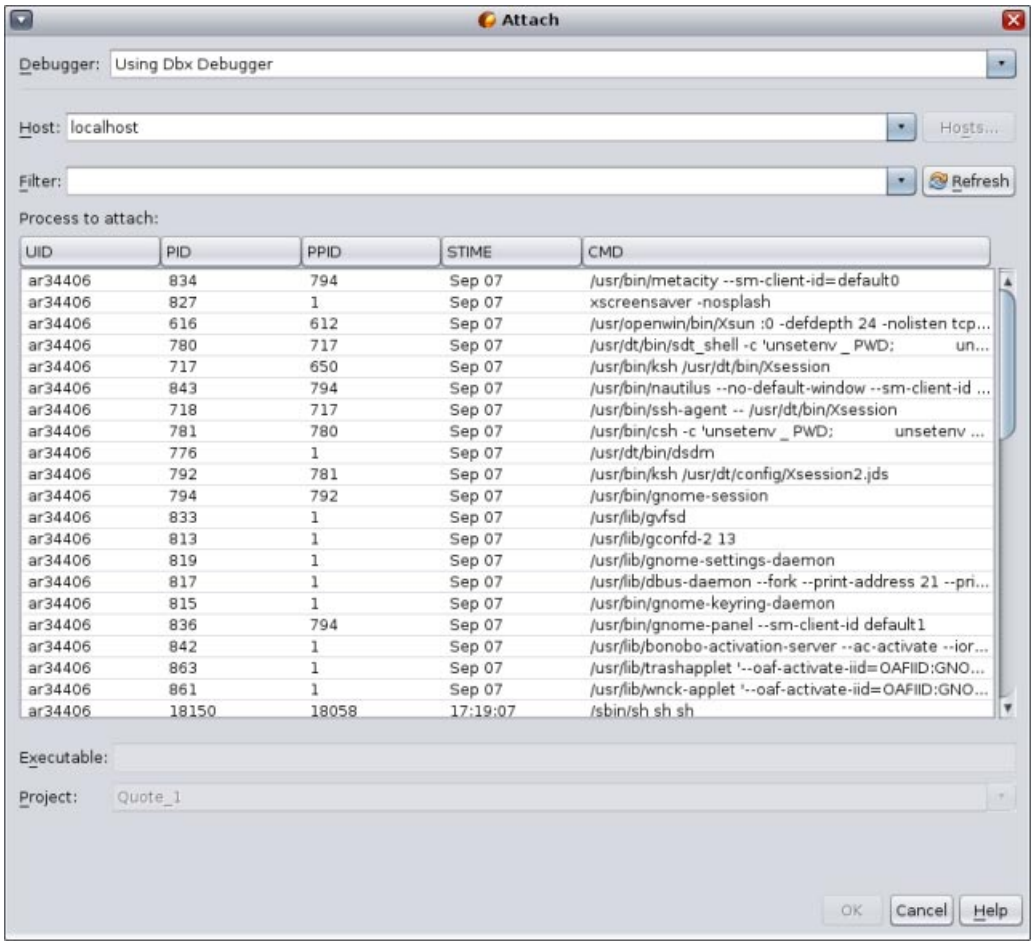
5. 选择 "Window"（窗口）> "Debugging"（调试）> "Memory"（内存）可打开 "Memory"（内存）窗口，该窗口显示项目当前使用内存的内容。在窗口底部，可以指定要浏览的内存地址、更改内存浏览的长度，或者更改内存信息的格式。



通过附加到某个正在运行的程序对其进行调试

如果要调试某个已在运行的程序，可以将调试器附加到相应的进程。

1. 选择 "File"（文件）> "New Project"（新建项目）。
2. 在 "New Project"（新建项目）向导中，展开 "Samples"（样例）节点，然后选择 "C/C++" 类别。
3. 选择 "Freeway Simulator"（Freeway 仿真器）项目。单击 "Next"（下一步），然后单击 "Finish"（完成）。
4. 右键单击所创建的 `Freeway_1` 项目，然后选择 "Run"（运行）。项目将生成，`Freeway` 应用程序将启动。在 `Freeway` GUI 窗口中，选择 "Actions"（操作）> "Start"（启动）。
5. 在 IDE 中，选择 "Debug"（调试）> "Attach Debugger"（附加调试器）。



6. 在 "Attach"（附加）对话框中，在 "Filter"（过滤器）字段中键入 Freeway 以过滤处理器列表。
7. 从过滤后的列表中选择 Freeway 进程。
8. 单击 "OK"（确定）。
9. 将启动一个调试会话，Freeway 进程的执行会在调试器附加到该进程的位置暂停。
10. 单击 "Continue"（继续）  继续执行 Freeway，它当前在调试器控制下运行。如果单击 "Pause"（暂停）  ，Freeway 将暂停执行，然后您可以检查变量、调用堆栈，等等。
11. 再次单击 "Continue"（继续），然后单击 "Finish Debugger Session"（完成调试器会话）  。调试器会话将结束，但 Freeway 进程会继续执行。在 Freeway GUI 中选择 "File"（文件）> "Exit"（退出）以退出应用程序。

调试信息转储文件

如果程序即将崩溃，您可能需要调试信息转储文件（程序崩溃时的内存映像）。将信息转储文件装入调试器：

1. 选择 "Debug"（调试）> "Debug core file"（调试信息转储文件）。
2. 在 "Debug core file"（调试信息转储文件）字段中键入信息转储文件的完整路径，或者在 "Select Core File"（选择信息转储文件）对话框中单击 "Browse"（浏览）并导航到您的信息转储文件。

3. 如果调试器无法将您指定的信息转储文件与某个可执行文件相关联，它将显示一条错误消息。如果出现这种情况，请在 "Executable"（可执行文件）文本框中键入可执行文件的路径名，或者单击 "Browse"（浏览）按钮并使用 "Executable"（可执行文件）对话框选择可执行文件。
4. 缺省情况下，"Project"（项目）文本字段会显示 `<no project>` 或者与可执行文件名称完全匹配的某个现有项目的名称。如果需要为可执行文件创建一个新项目，请选择 `<create new project>`。
5. 单击 "Debug"（调试）。

有关调试的更深入的教程，请参见《Oracle Solaris Studio 12.3： dbxtool 教程》。

Oracle Corporation 500 Oracle Parkway, Redwood City, CA 94065 U.S.A.

ORACLE