

RealView[®] Development Suite

3.1 版

Eclipse 插件 用户指南

ARM[®]

RealView Development Suite

Eclipse 插件用户指南

版权所有 © 2006-2007 ARM Limited。保留所有权利。

版本信息

本手册进行了以下更改。

更改历史记录

日期	发行号	保密性	更改
2006 年 3 月	A	非保密	RVDS v3.0 的 1.0 版
2007 年 3 月	B	非保密	RVDS v3.1 的 2.0 版

所有权声明

除非本所有权声明在下面另有说明，否则带有®或™标记的词语和徽标是 ARM Limited 在欧洲和其他国家/地区的注册商标或商标。此处提及的其他品牌和名称可能是其各自所有者的商标。

除非事先得到版权所有人的书面许可，否则不得以任何形式修改或复制本文档包含的部分或全部信息以及产品说明。

本文档描述的产品还将不断发展和完善。ARM 将如实提供本文档所述产品的所有特性及其使用方法。但是，所有暗示或明示的担保，包括但不限于对特定用途适销性或适用性的担保，均不包括在内。

本文档的目的仅在于帮助读者使用产品。对于因使用本文档中的任何信息、文档信息出现任何错误或遗漏或者错误使用产品造成的任何损失或损害，ARM 公司概不负责。

使用 ARM 一词时，它表示“ARM 或其任何相应的子公司”。

保密状态

本文档的内容是非保密的。根据 ARM 与 ARM 将本文档授予的参与方的协议条款，使用、复制和公开本文档内容的权利可能会受到许可限制的制约。

产品状态

本文档的信息是开发的产品的最新信息。

网址

<http://www.arm.com>

目录

RealView Development Suite 3.1 版

Eclipse 插件 用户指南

	前言	
	关于本手册	vi
	反馈	viii
第 1 章	简介	
	1.1 关于 RVDS Eclipse 插件	1-2
	1.2 入门	1-4
	1.3 安装 RVDS Eclipse 插件的软件更新	1-5
第 2 章	创建 ARM 工程	
	2.1 为 ARM 创建 RVDS 工程	2-2
	2.2 向工程中添加新文件	2-9
	2.3 生成工程	2-11
第 3 章	导入现有的 Eclipse 工程	
	3.1 选择并生成现有的 Eclipse 工程	3-2

第 4 章	导入 CodeWarrior 工程	
	4.1 关于 CodeWarrior 导入程序	4-2
	4.2 导入 CodeWarrior 工程	4-3
第 5 章	ARM 工程类型	
	5.1 关于 ARM 工程类型	5-2
	5.2 关于预定义配置	5-3
第 6 章	配置生成设置	
	6.1 访问配置面板	6-2
	6.2 访问适用于特定文件的配置面板	6-4
	6.3 配置 RealView 工具	6-5
	6.4 使用 ARM fromelf 实用程序	6-6
	6.5 恢复缺省值	6-8
	6.6 访问在线帮助	6-9
第 7 章	使用 RealView Debugger	
	7.1 将可执行映像加载到 RealView Debugger 中	7-2
	7.2 创建调试配置	7-4
	7.3 设置调试配置	7-5
	7.4 使用调试配置启动 RealView Debugger	7-8

前言

本前言介绍 *RealView Development Suite Eclipse 插件用户指南*。它包含以下几节:

- 第vi页的关于本手册
- 第viii页的反馈

关于本手册

本手册为用户提供有关 *RealView Development Suite (RVDS) Eclipse* 插件的信息。书中介绍了如何使用 *Eclipse 集成开发环境 (IDE)* 为 ARM® 目标配置和生成工程。本手册的目的不是让用户熟悉 *Eclipse IDE* 的所有方面。

适用对象

本手册是专门为使用 *Eclipse IDE* 在 Microsoft Windows (2000和XP) 或 Unix 下管理面向 ARM 的开发工程的所有开发人员而编写的。它假定您是一位有经验的软件开发人员,熟悉 ARM 开发工具。它并不要求您熟悉 *Eclipse IDE*。

使用本手册

本手册由以下章节组成

第1章 简介

本章简要介绍 *RealView Development Suite Eclipse* 插件。

第2章 创建 ARM 工程

本章介绍如何创建 ARM 工程。

第3章 导入现有的 Eclipse 工程

本章介绍如何将现有的 *Eclipse* 工程导入到 *Eclipse IDE* 中。

第4章 导入 CodeWarrior 工程

本章介绍如何将现有的 *CodeWarrior®* 工程导入到 *Eclipse IDE* 中。

第5章 ARM 工程类型

本章介绍如何使用 *Eclipse IDE* 中提供的各种 ARM 工程类型。

第6章 配置生成设置

本章介绍如何配置 ARM *RealView* 工具,以修改工程的生成方式。

第7章 使用 RealView Debugger

本章介绍如何在 *Eclipse* 中使用 *RealView Debugger*。

本手册假定 ARM 软件安装在缺省位置。例如,在 Windows 上,这可能为 卷 : \Program Files\ARM。引用路径名时,假定该位置为 *install_directory*。例如, *install_directory\Documentation\...*。如果已将 ARM 软件安装在其他位置,则可能需要更改此路径。

印刷约定

本手册使用了以下印刷约定

<i>斜体</i>	突出显示重要说明, 介绍专用术语, 表示内部交叉引用和引文。
粗体	突出显示界面元素, 如菜单名称。表示 ARM 处理器信号名称。必要时还用于说明列表中的术语。
monospace	表示可以从键盘输入的文本, 如命令、文件和程序名以及源代码。
<u>monospace</u>	表示允许的命令或选项缩写。可只输入带下划线的文本, 无需输入命令或选项的全名。
<i>monospace italic</i>	表示此处的命令和函数的自变量可用特定值代替。
等宽粗体	表示在示例代码以外使用的语言关键字。

更多参考出版物

本部分列出了 ARM 公司和第三方发布的、可提供有关 ARM 系列处理器开发代码的附加信息的出版物。

ARM 将定期对其文档进行更新和更正。有关最新勘误表、附录和 ARM 常见问题 (FAQ), 请访问 <http://www.arm.com>。

ARM 出版物

有关详细介绍 RVDS 中各个组件的文档, 请参阅下列出版物

- 《*RealView Development Suite 入门指南*》(ARM DUI 0255)
- 《*RVCT 要点指南*》(ARM DUI 0202)
- 《*RVCT 开发指南*》(ARM DUI 0203)
- 《*RVCT 汇编程序指南*》(ARM DUI 0204)
- 《*RVCT 编译器用户指南*》(ARM DUI 0205)
- 《*RVCT 链接器和实用程序指南*》(ARM DUI 0206)
- 《*RVCT 编译器参考指南*》(ARM DUI 0348)
- 《*RVCT 库和浮点支持指南*》(ARM DUI 0349)
- 《*RVCT NEON 编译器指南*》(ARM DUI 0350)

- 《RealView Debugger 要点指南》(ARM DUI 0181)
- 《RealView Debugger 用户指南》(ARM DUI 0153)
- 《RealView Development Suite 词汇表》(ARM DUI 0324)

其他出版物

本手册介绍特定于 ARM 提供的插件的信息。有关 Eclipse IDE 的详细信息，请访问 Eclipse 网站（网址为 <http://www.eclipse.org> 和 <http://www.eclipse.org/cdt>）。

反馈

ARM Limited 欢迎用户就 RVDS Eclipse 插件及其文档提供反馈。

有关 RealView Development Suite Eclipse 插件的反馈

如果您对 RVDS Eclipse 插件有任何问题，请与供应商联系。为便于他们快速提供有用的回复，请提供

- 您的姓名和公司名称。
- 您的 RVDS 产品的序列号和版本号。
- **Eclipse C/C++ 开发工具**和 **RVDS Eclipse 插件的版本号**。要获取此版本号，请在 Eclipse 中选择 **Help** → **About Eclipse Platform**，然后单击 **Plug-in Details**。
- 所有已安装组件的详细信息和版本号，如硬件平台、操作系统、GNU make 和 Java 运行时环境。
- 一小段能够重现问题的独立代码示例。
- 您期望发生和实际已经发生的情况的详细说明。
- 您使用的命令，包括所有命令行选项。
- 能说明问题的示例输出。
- 工具的版本字符串，包括版本号和内部版本号。

关于本手册的反馈

如果您对本手册有任何问题，请发送电子邮件至 errata@arm.com，并提供

- 文档标题
- 文档编号
- 您有疑问的页码
- 问题的简要说明

我们还欢迎您对需要增加和改进之处提出建议。

第 1 章 简介

本手册介绍 *RealView Development Suite (RVDS) Eclipse* 插件，并说明如何在 Eclipse 中用此插件为 ARM® 目标开发软件。本手册并不介绍 Eclipse 集成开发环境 (IDE)。有关如何使用 Eclipse IDE 的完整说明，请参阅 Eclipse “Help” 菜单中的信息或 Eclipse 网站上的在线文档，网址是 <http://www.eclipse.org>。

本章介绍 RVDS Eclipse 插件并包含以下内容：

- 第 1-2 页的关于 *RVDS Eclipse 插件*
- 第 1-4 页的入门
- 第 1-5 页的 *安装 RVDS Eclipse 插件的软件更新*

1.1 关于 RVDS Eclipse 插件

RVDS Eclipse 插件可将 RealView 开发工具集成到 Eclipse IDE 中。利用此插件，软件开发人员可以使用 Eclipse IDE 作为工程管理器，以便为 ARM 目标创建、生成、调试和管理 C、C++ 和汇编语言工程。Eclipse 插件提供了一系列 ARM 工程类型，可简化 ARM 工程的创建。此外，它还提供了全套配置面板，可用于指定 ARM 编译器、汇编程序、链接器、调试器及其他工具的选项。

1.1.1 使用限制

本节列出使用 Eclipse 时适用的特定限制和特点。

组织工程

工程源文件的建议结构是在工程文件下的子目录中创建它们。如果创建源文件时使用的目录结构高于工程目录，Eclipse 将创建该文件的绝对链接。

打开现有的 Eclipse 工程

您必须使用导入向导。有关详细信息，请参阅第3章 *导入现有的 Eclipse 工程*。

CodeWarrior 子工程

不支持 CodeWarrior 子工程。您必须逐个导入每个工程。有关详细信息，请参阅第4章 *导入 CodeWarrior 工程*。

工程间的相关性

不支持嵌套工程。每个工程都必须组织为单独的实体。通过引用驻留在工作区中的其他工程，可以设置工程间的相关性。通过在 Eclipse 主菜单中选择 **Project** → **Properties** → **Project References**，可以手动添加引用。

链接顺序

使用 Eclipse 时，无法在同一工程内部指定目标文件的链接顺序。一种解决方法是将目标文件拆分到不同的工程中，然后便可以指定工程的生成顺序。选择 **Windows** → **Preferences...** → **General** → **Workspace** → **Build Order**。

恢复缺省值

恢复工程的缺省值将丢弃不属于工程类型的所有信息。这意味着，在 ARM New Project Wizard 中更改的任何设置都将丢失。

使用 Eclipse 目标配置启动 RealView Debugger

当使用具有现有 Eclipse 目标配置的工程在 Eclipse 中启动 RealView Debugger 时，可能会显示“RealView Debugger”对话框，如图1-1所示。单击 **No**，以使用 Eclipse 中的目标连接属性。

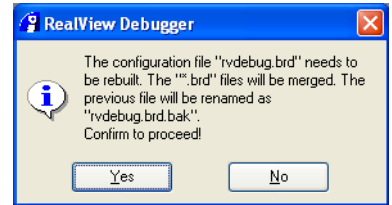


图1-1 “RealView Debugger”对话框

1.2 入门

本节概述使用 Eclipse 所需的各种组件。

1.2.1 RVDS 安装

RVDS 从 CD 安装 RVDS v3.1 时, 将为您安装所有必需的组件, 包括 Eclipse。

1.2.2 自定义安装

如果您有 Eclipse 的自定义安装, 则在使用 RVDS Eclipse 插件之前, 应确保已安装下列组件。

Eclipse 从 <http://www.eclipse.org> 下载并安装 Eclipse v3.2 或更高版本。

RVDS Eclipse 插件

使用 Eclipse 的软件更新功能安装来自 <http://www.arm.com/eclipse> 的 RVDS Eclipse 插件。请参阅第 1-5 页的 *安装 RVDS Eclipse 插件的软件更新*。

JRE 下载并安装最适合您的平台的 Java 运行时环境 (JRE) 版本。请参阅 <http://www.java.com> 或 <http://www.eclipse.org>。

CDT C 和 C++ 开发工具 (CDT) 是一个可将 C 和 C++ 生成环境集成到 Eclipse IDE 中的插件。必须安装此插件, 因为 RealView 工具将插入到 CDT 中。要求使用 CDT v3.1。请安装来自 <http://www.eclipse.org> 的最新版本的 CDT。

GNU make 要求使用 GNU make v3.80。请安装 MinGW (来自 <http://www.mingw.org>) 或 Cygwin (来自 <http://www.cygwin.com>)。建议安装 MinGW。

1.3 安装 RVDS Eclipse 插件的软件更新

如果您是从 CD 安装的 RVDS v3.1, 则所有必需的 Eclipse 组件均已安装。

要安装 RVDS Eclipse 插件的最新功能, 必须使用 Eclipse 中的软件更新功能。首次使用软件更新功能时, 必须创建要从中下载该插件的更新站点:

1. 启动 Eclipse IDE。
2. 选择 **Help** → **Software Updates** → **Find and Install...**。
3. 在 **Install/Update** 对话框中, 选择 **Search for new features to install**。单击 **Next**。
4. 您必须创建一个新站点才能找到 Eclipse 插件。在 **Install** 对话框中, 单击 **New Remote Site...**。
5. 在 **New Update Site** 对话框中, 输入任意名称, 例如 Eclipse 插件 for RVDS。在 URL 字段中, 输入 <http://www.arm.com/eclipse>, 然后单击 **OK**。
6. Eclipse 会自动从 **Sites to include in search** 列表中选择新创建的更新站点。单击 **Finish**。
7. 在 **Updates** 对话框中, 选择您的 RVDS Eclipse 插件更新站点, 然后单击 **Next**。
8. 阅读 ARM 许可协议并接受它。如果您不接受该许可协议, 则无法安装 Eclipse 插件。单击 **Next**, 然后单击 **Finish**。
9. 在“Verification”对话框中, 单击 **Install All**。Eclipse 随即将安装 RVDS Eclipse 插件。
10. Eclipse 会询问您是否要重新启动。单击 **Yes** 重新启动 Eclipse。这样便安装了最新版本的 RVDS Eclipse 插件。

要在以后更新 Eclipse 插件, 请按照 Eclipse 插件的安装步骤操作, 但不用创建新的远程站点。此过程还可以用于安装其他插件。

注意

您可以在 **Preferences** 面板中启用自动更新。要显示“Preferences”面板, 请在 Eclipse 主菜单中选择 **Window** → **Preferences...**。

第 2 章

创建 ARM 工程

可以使用 Eclipse 为 ARM 目标创建 RVDS 工程。Eclipse 会为 RVDS 工程生成相应的 makefile。此外，您还可以创建托管生成和标准生成工程。有关详细信息，请参阅 Eclipse 帮助中的《C/C++ 开发用户指南》。

本章介绍如何为 ARM 创建新的 RVDS 工程。它包含以下几节：

- 第 2-2 页的 *为 ARM 创建 RVDS 工程*
- 第 2-9 页的 *向工程中添加新文件*
- 第 2-11 页的 *生成工程*

2.1 为 ARM 创建 RVDS 工程

您可以使用随 RVDS 提供的任何工程类型创建新的 ARM 工程。本节介绍如何创建新的可执行 (ARM) 工程

1. 启动 Eclipse IDE。
2. 在 Eclipse 主菜单中, 选择 **File** → **New** → **RVDS Project for ARM** (请参阅图2-1)。

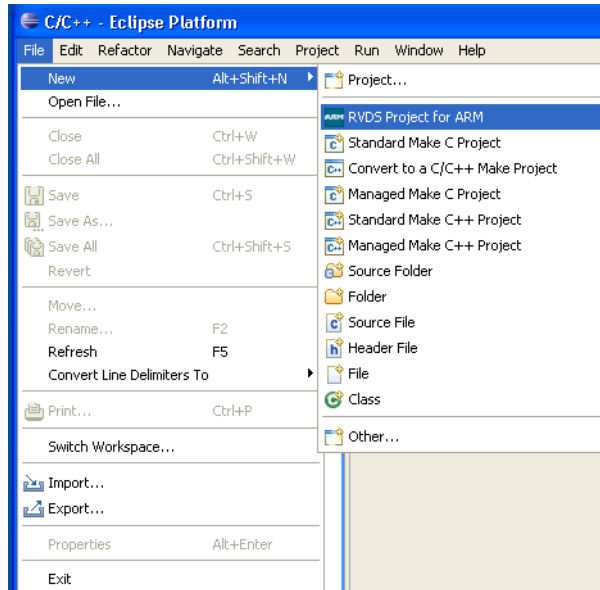


图2-1 创建新的 ARM 工程

注意

ARM 的 RVDS 工程集托管生成 C 工程与托管生成 C++ 工程的功能于一身。同一个工程中可以同时存在 .c、.cpp 和 .s 文件。

3. 输入工程的名称, 请参阅第 2-3 页的图2-2。

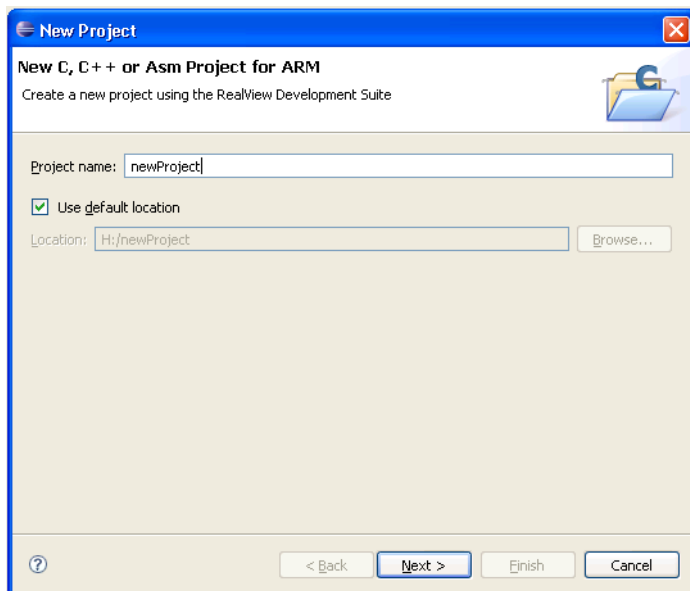


图2-2 为工程命名

4. 将 **Use Default** 选项保持选中状态，这样将在显示的缺省目录中创建工程。此外还可以取消选择 **Use Default** 选项，然后为新工程的内容输入一个目录。单击 **Next**。
5. 第2-4 页的图2-3 显示了 **Project** 面板。选择 **Project Type**。

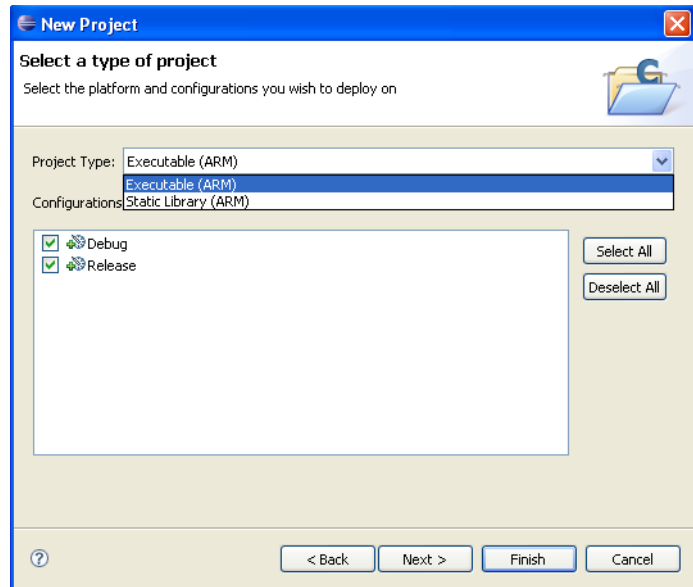


图2-3 选择工程类型

6. 在 **Configurations** 面板中，将 **Debug** 和 **Release** 选项保持选中状态，以便能够更改其他配置的工程设置。单击 **Next**。
7. 第2-5 页的图2-4 显示了 **Additional Project Settings** 面板。如果您有相互依赖的工程，请选择适用于此工程的任何选项，然后单击 **Next**。

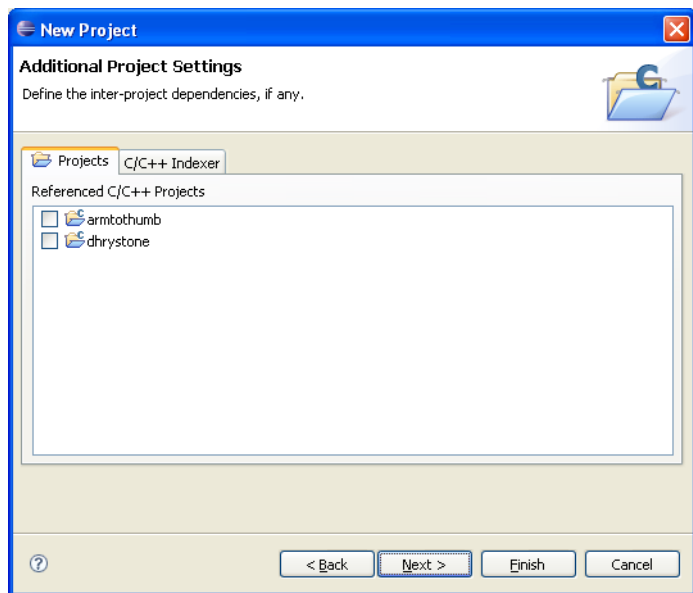


图2-4 其他工程设置

8. 图2-5 显示了典型的 **Target Settings** 面板。根据需要选择 **Target**。单击 **Next**。

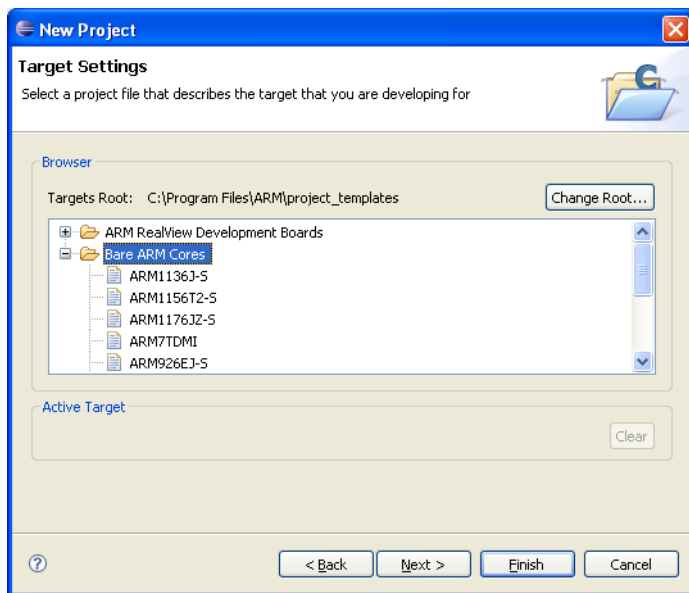


图2-5 典型的目标设置

——注意——

在“Target Settings”面板中选择目标后，将安装供 RealView Debugger 使用的关联信息。在从 Eclipse 中启动 RealView Debugger 以运行或调试可执行映像时，将传送此信息。

- 图2-6 显示了 **Fine-tune Target Settings** 面板。选择 **Architecture or Processor (--cpu=)**、**Floating Point (--fpu=)** 和 **Byte Order**。

——注意——

要在此处选择 VFP 目标，需要具有初始化代码和可能的支持代码，以便在工程中使用 VFP。有关详细信息，请参阅《应用程序注释 133: 在 RVDS 中使用 VFP》或尝试主示例目录中随 RVDS 提供的 VFP 示例 `install_directory\RVDS\Examples\...\vfpsupport`。

单击 **Next**。

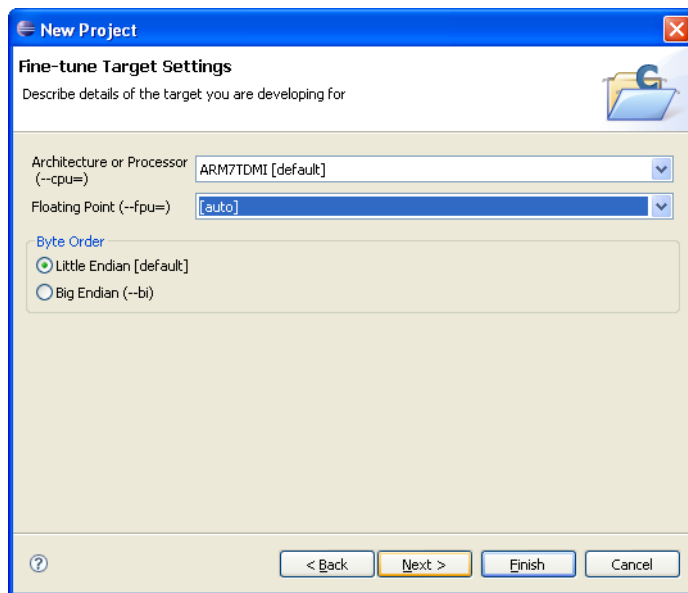


图2-6 优化目标设置

10. 第2-8 页的图2-7 显示了 **Language Settings** 面板。请根据需要选择 **Initial State** 和 **Source Language**。使用缺省设置 **[auto]** 可让编译器为工程选择最佳选项。单击 **Finish** 创建工程。

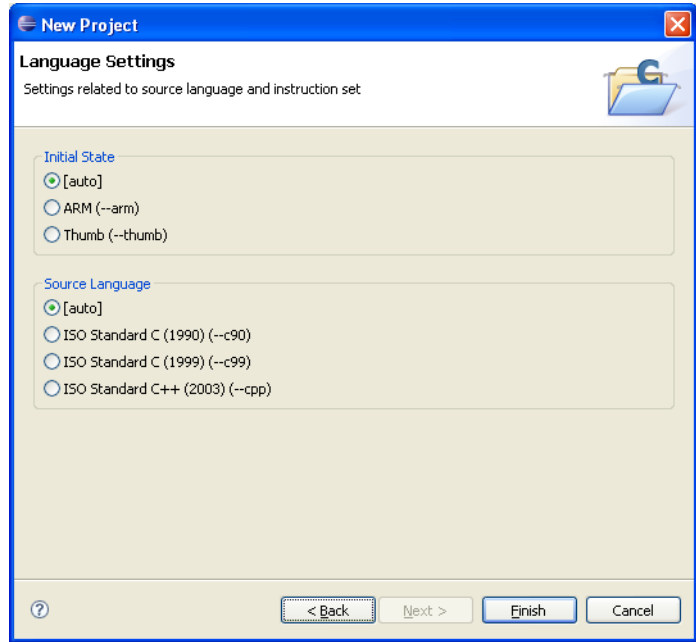


图2-7 语言设置

11. 新工程将显示在 **C/C++ Projects** 视图中。

2.2 向工程中添加新文件

要向新创建的工程中添加空白源文件，请执行下列操作：

1. 在 **C/C++ Projects** 视图中右击工程。
2. 选择 **New** → **File**，以显示 **New file** 对话框（请参阅图2-8）。

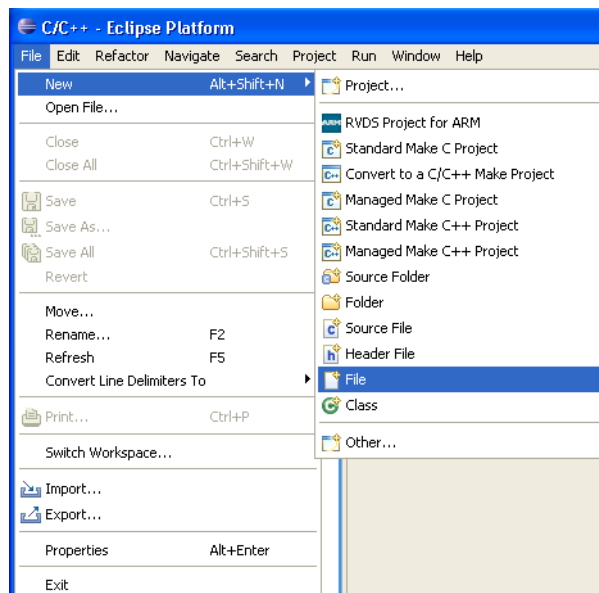


图2-8 添加新的源文件

3. 在 **Enter or select the parent folder** 字段中，选择您的工程（请参阅第2-10 页的图2-9）。
4. 在 **File name** 字段中，输入源文件的名称和扩展名。

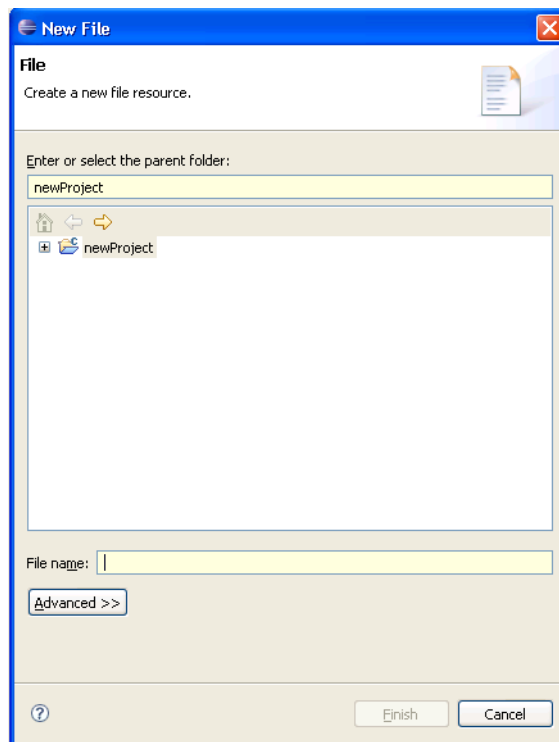


图2-9 为文件命名

5. 单击 **Finish**。源文件将显示在 **C/C++ Projects** 视图中。

——**注意**——

- 利用文件资源管理器，您还可以在工程目录中创建文件或直接将文件拖放到工程目录中。要在 Eclipse 中查看这些文件，请在 **C/C++ Projects** 视图中右击该工程，然后选择 **Refresh**。
- 在 Eclipse 的 **C/C++ Projects** 视图中，您也可以将文件直接拖放到工程目录中。

2.3 生成工程

在向工程中添加文件后，如果在 Eclipse 主菜单上选择了 **Project → Build Automatically**，则会自动为您生成工程。您可以通过下列方式手动生成工程

1. 在 **C/C++ Projects** 视图中选择工程目录。
2. 在 Eclipse 主菜单中，选择 **Project → Build Project**。此外，您也可以右击工程，然后从上下文菜单中选择 **Build Project**。

缺省情况下，生成输出将使用相应工程的名称。工程目录中将为活动的生成配置创建一个子目录，生成输出便保存在此子目录中。缺省情况下，活动的生成配置称为 Debug。请参阅第5-3页的*关于预定义配置*。

——注意——

缺省情况下，Eclipse 配置为保存文件时自动执行生成。要禁用此功能，请确保 **Project → Build Automatically** 未处于选中状态。

请注意，执行手动生成时，将不会自动保存文件。

第 3 章

导入现有的 Eclipse 工程

Eclipse 环境使用导入过程在 Eclipse 工作区中选择并生成现有工程。本章介绍工程的导入过程。它包含以下一节:

- 第 3-2 页的 *选择并生成现有的 Eclipse 工程*

3.1 选择并生成现有的 Eclipse 工程

本节介绍如何将现有的 Eclipse 工程导入到 Eclipse 工作区中。

1. 启动 Eclipse IDE。
2. 在 Eclipse 主菜单中，选择 **File** → **Import...**（请参阅图3-1）。

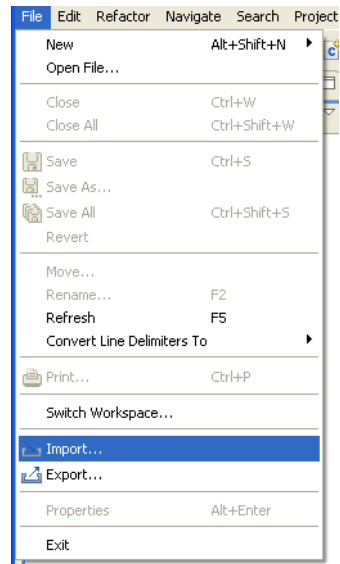


图3-1 Eclipse 中的导入选项

3. 要导入现有的工程，请选择 **Existing Project into Workspace**。单击 **Next**。请参阅第3-3 页的图3-2。

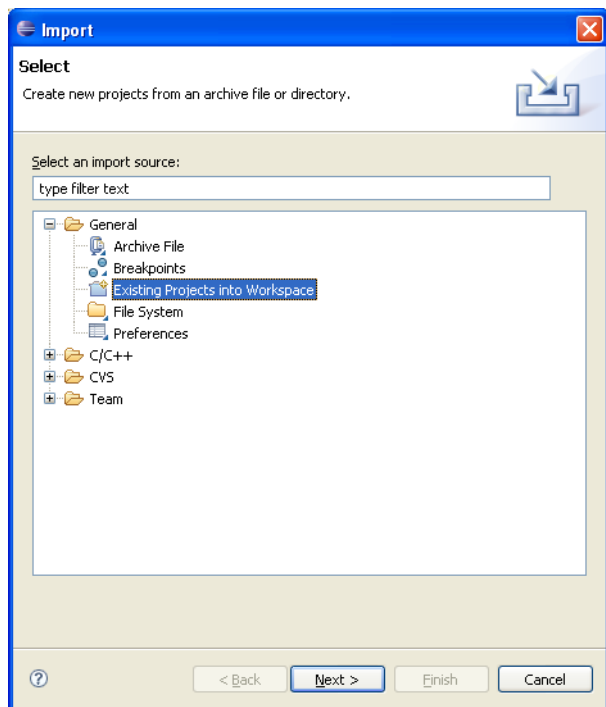


图3-2 选择导入源类型

- 单击 **Browse** 选择工程根目录。确保已在 **Projects** 面板中选择了所有需要导入的工程。根据需要选择 **Copy projects into workspace**，或取消选择此选项以创建指向现有工程及关联文件的链接。请参阅第 3-4 页的图 3-3。

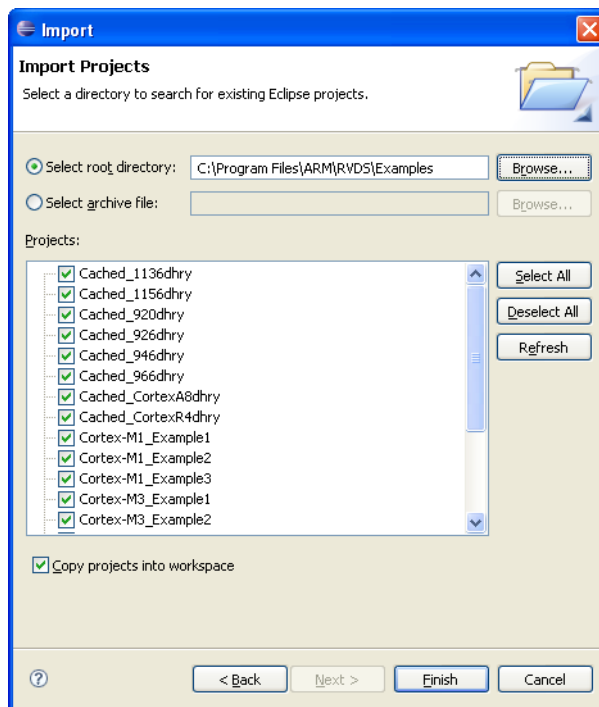


图3-3 选择要导入的现有 Eclipse 工程

5. 单击 **Finish**。请参阅图3-3。

——**注意**——

如果选择了 **Build Automatically**，则当您单击 **Finish** 时将生成工程。如果未选择该选项，则需要在 Eclipse 主菜单中选择 **Project** → **Build Project** 才能完成生成步骤。

第 4 章

导入 CodeWarrior 工程

本章介绍如何将现有的 CodeWarrior® ARM 工程导入到 Eclipse IDE 中。它包含以下内容

- 第4-2 页的关于 *CodeWarrior* 导入程序
- 第4-3 页的导入 *CodeWarrior* 工程

4.1 关于 CodeWarrior 导入程序

利用 RVDS Eclipse 插件，可以将使用 RVDS CodeWarrior 创建的工程导入到 Eclipse 环境中。为了实现此目的，必须先将 CodeWarrior 工程导出为可扩展标记语言 (XML) 格式 (文件扩展名为 .xml)。

——注意——

该 .xml 文件必须创建在 CodeWarrior 工程文件 (文件扩展名为 .mcp) 所在的目录中

有关如何将 CodeWarrior 工程导出为 XML 的详细信息，请参阅《*RealView Development Suite CodeWarrior IDE 指南*》。在 Eclipse 中导入 CodeWarrior 工程时，将创建一个新的 Eclipse 托管生成工程。新的 Eclipse 工程将使用 CodeWarrior 工程中的 ARM RealView 工具设置。

4.1.1 局限性

CodeWarrior 导入程序存在一些局限性

- 不支持子工程。您必须逐个导入每个工程。
- 子目标的导入方式与文件相同。有关详细信息，请参阅第2-9页的*向工程中添加新文件*。
- 未映射到 Eclipse 面板设置的命令行选项作为额外选项包括在 **Tool Settings** 面板中 (**Extra** → **Extra options**)。有关详细信息，请参阅第6-2页的*访问配置面板*。

4.2 导入 CodeWarrior 工程

要将 CodeWarrior 工程导入到 Eclipse 中，请执行下列操作：

1. 在 Eclipse 主菜单中，选择 **File** → **Import**，以显示 **Import** 对话框。
2. 在 **Import** 对话框中，选择 **CodeWarrior Project exported as XML**，请参阅图4-1。单击 **Next**。

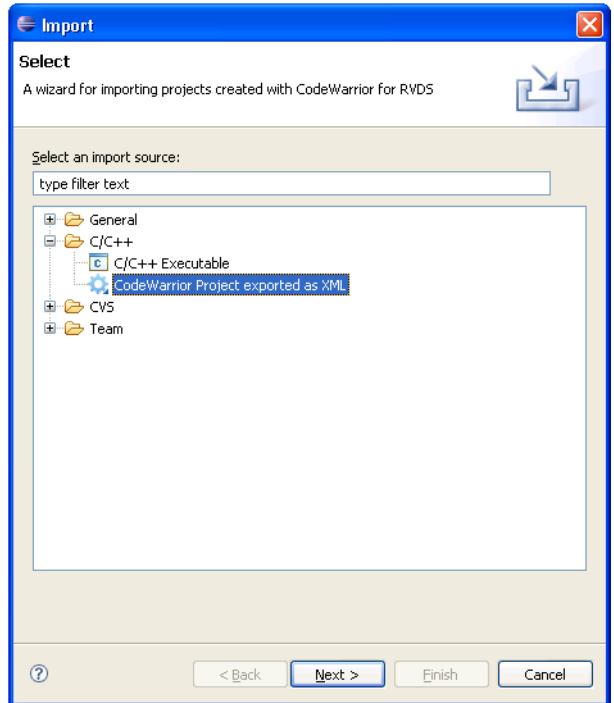


图4-1 选择 CodeWarrior XML

3. 使用 **Browse** 按钮选择包含 .xml 文件的目录（请参阅第4-4 页的图4-2）。然后，请在 **Projects** 面板中选择工程。

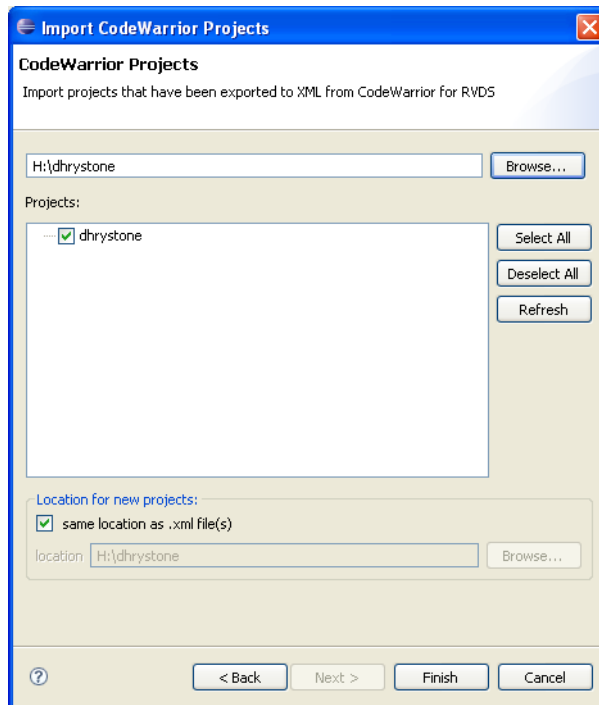


图4-2 导入 CodeWarrior 工程

4. **CodeWarrior Projects** 对话框显示可导入到 Eclipse 中的工程。缺省情况下，可从所选目录中导入的全部工程都处于选定状态。如果 **Projects** 面板显示了多个工程，请取消选择不需要导入的工程。
5. 在 **location** 文本框中，您可以输入另一个目录，也可以使用 **Browse** 按钮选择要在其中创建 Eclipse 工程的目录。

——注意——

如果选择了 **same location as .xml file(s)** 选项，将在 .xml 文件所在的位置创建 Eclipse 工程文件，而不再单独创建工程目录。

6. 单击 **Finish**。Eclipse 将在所选的位置创建一个新的工程目录，然后将工程文件保存到该目录中。使用 **C/C++ Projects** 视图可以查看导入的工程。

第 5 章

ARM 工程类型

本章提供有关由 RVDS Eclipse 插件提供的各种 ARM 工程类型的信息。它包含以下几节:

- 第5-2 页的关于*ARM 工程类型*
- 第5-3 页的关于*预定义配置*

5.1 关于 ARM 工程类型

RVDS Eclipse 插件提供下列 ARM 工程类型

- executable
- static library

创建新的托管生成工程时，可以从下拉框中选择一种工程类型，请参阅图5-1。以下几节介绍 ARM 工程类型。

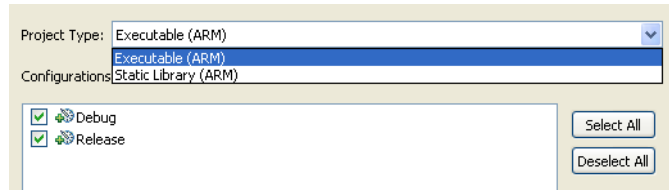


图5-1 ARM 工程类型

5.1.1 Executable

使用 **Executable** 工程类型可基于 ARM 或 Thumb 代码创建可执行的 ELF 映像。**Executable** 工程使用

- ARM 编译器 (armcc)，用于在 ARM 或 Thumb 状态下编译 C 和 C++ 源文件。
- ARM 汇编程序 (armasm)，用于汇编文件扩展名为 .s 的文件。
- ARM 链接器 (armlink)，用于链接可执行的 ELF 映像。
- RealView Debugger，用于调试和运行由工程生成输出的可执行映像。

您也可以将 Eclipse 配置为调用 ARM fromelf 实用程序来基于可执行的 ELF 映像创建二进制文件。请参阅第6-6页的 *使用 ARM fromelf 实用程序*。

5.1.2 Static library

使用 **Static Library** 工程类型可生成 ELF 对象格式成员库。此工程类型与 **Executable** 类似。二者的主要区别在于：

- **Static Library** 工程使用 ARM 库管理程序 (armar) 输出对象库，而不是使用 ARM 链接器 (armlink) 输出可执行的 ELF 映像。
- 只有将独立的库文件链接到某个映像后，才能对其进行调试或运行。

5.2 关于预定义配置

使用 ARM 工程类型创建的任何工程都提供两个单独的生成配置:

Debug 调试目标配置为生成完全可调试的输出二进制文件, 但优化程度较低。它将编译器优化设置配置为最低 (级别 0), 以便为代码开发提供理想的调试视图。

Release 发布目标配置为生成高度优化的输出二进制文件, 但调试视图的质量较差。它将编译器优化设置配置为高 (级别 2)。调试信息仍包括在生成的映像中。

缺省情况下, 所有 ARM 工程类型的 **Debug** 配置都设置为活动。活动配置可以在 **C/C++ Build** 配置面板中更改。有关修改工具选项的信息, 请参阅第 6 章 *配置生成设置*。

第 6 章

配置生成设置

本章介绍如何使用配置面板配置生成属性。这些设置决定了 ARM RealView 工具生成 ARM 可执行映像或库的方式。它包含以下几节：

- 第 6-2 页的 *访问配置面板*
- 第 6-4 页的 *访问适用于特定文件的配置面板*
- 第 6-5 页的 *配置 RealView 工具*
- 第 6-6 页的 *使用 ARM fromelf 实用程序*
- 第 6-8 页的 *恢复缺省值*
- 第 6-9 页的 *访问在线帮助*

6.1 访问配置面板

使用配置面板，可以针对工程中的特定生成配置设置 RealView 工具。您可以设置应用于工程中的所有源文件的选项，也可以设置应用于特定文件的选项。本节介绍如何访问影响工程中所有源文件的生成配置面板：

1. 将现有的某一 ARM 工程导入到工作区或使用 ARM 工程类型创建新的工程（请参阅第2-2页的 *为ARM 创建 RVDS 工程*）。
2. 在 **C/C++ Projects** 视图中，选择要配置的 ARM 工程。
3. 在 Eclipse 主菜单中，选择 **Project → Properties**。
4. 在 **Properties** 对话框中的属性列表中，选择 **C/C++ Build**。
5. 可供工程使用的 RealView 工具及其相应的配置面板显示在 **Configuration Settings** 面板的 **Tool Settings** 选项卡中（第6-3 页的图6-1）。
6. **Active configuration** 面板显示当前的工程类型和配置。
7. 使用 **Configuration** 下拉菜单选择 **Release**。这将显示用于发布生成的 RealView 工具设置。
8. 使用 **Configuration** 下拉菜单选择 **Debug**。这将显示用于调试生成的 RealView 工具设置。

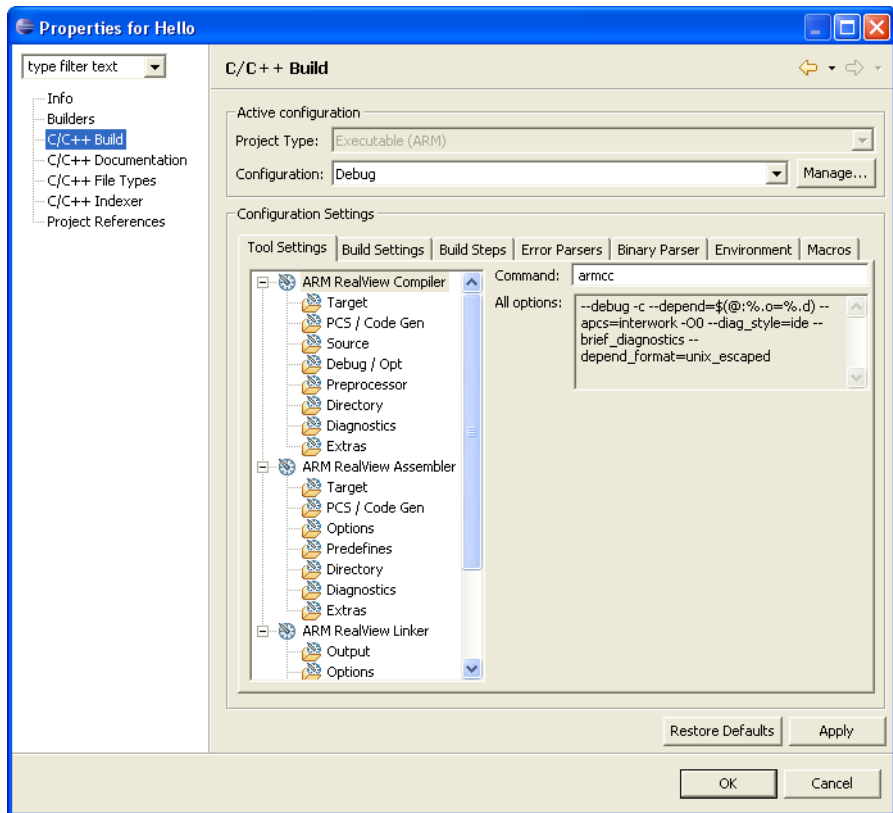


图6-1 ARM 工程的生成配置面板

注意

通过在 **Properties** 对话框中的 **Active configuration** 面板中选择 **Manage**，可以创建新的生成配置。

6.2 访问适用于特定文件的配置面板

对于每个源文件，可以单独设置 RealView 编译器和 RealView 汇编程序。如果为源文件指定了不同的工具选项，它将覆盖在适用于所有源文件的通用配置面板中指定的选项。有关访问通用配置面板的信息，请参阅第 6-2 页的 *访问配置面板*。本节介绍如何访问工程中任意特定源文件的配置面板：

1. 将现有的某一 ARM 工程导入到工作区或使用 ARM 工程类型创建新的工程（请参阅第 2-2 页的 *为 ARM 创建 RVDS 工程*）。
2. 如果工程中没有源文件，请向工程中添加一个新的源文件（扩展名为 .c、.cpp 或 .s。请参阅第 2-9 页的 *向工程中添加新文件*）。
3. 右击工程中的某个源文件，然后从上下文菜单中选择 **Properties**，以显示该文件的 **Properties** 对话框。
4. 在 **Properties** 对话框中，选择 **C/C++ Build**。
5. 这将显示特定于所选文件的配置面板。可以配置的 RealView 工具显示在 **Tool Settings** 选项卡中。如果源文件的扩展名为 .s，**Tool Settings** 选项卡将显示 **ARM RealView Assembler** 设置。如果源文件的扩展名为 .c 或 .cpp，**Tool Settings** 选项卡将显示 **ARM RealView Compiler** 设置。

6.3 配置 RealView 工具

为了让您方便地查看和设置各个选项，每个 ARM RealView 工具都提供了一系列面板，例如编译器 (armcc)、汇编程序 (armasm)、链接器 (armlink) 和库管理程序 (armar)。要访问这些面板，请参阅第6-2 页的 *访问配置面板*。

注意

- 只有可执行文件才能显示 **ARM RealView Linker** 面板。
- 只有 static library 类型才能显示 **ARM RealView Librarian** 面板。

有关详细信息，请参阅第5章 *ARM 工程类型*。

有关每个面板中所显示的选项的说明，请参阅在线帮助（参阅第6-9 页的 *访问在线帮助*）或 RealView 编译工具文档套件中的相应书籍。每个工具的父面板都包含一个 **All options** 文本框，其中显示可为该工具设置的所有选项。调用此选项列表时，它将被传递给相应的 RealView 工具。

注意

- **All options** 文本框还包含一些附加选项，用于支持 RealView 工具在 Eclipse 环境中正常运行。
- 设置为 RealView 工具缺省值的选项在调用时不需要传递给相应的工具，因此不包括在 **All options** 文本框中。例如，如果将编译器优化级别设置为缺省值（级别 2），则 -O2 选项将不会出现在 **All options** 文本框中。

每个工具都有一个 **Extras** 面板，可用于指定无法在其他面板中设置的选项。使用 **Extras** 面板设置的选项将覆盖在其他面板中设置的选项。

6.4 使用 ARM fromelf 实用程序

本节包含以下内容

- 将 ELF 映像转换为其他格式
- 第6-7 页的反汇编代码

6.4.1 将 ELF 映像转换为其他格式

ARM fromelf 实用程序可将 ARM 链接器生成的可执行和可链接格式 (ELF) 映像文件转换为适用于 ROM 工具的其他格式，以直接加载到内存中。有关使用 fromelf 的详细信息（包括有关输出格式的信息），请参阅《RealView 编译工具链接器和实用程序指南》。要将 Eclipse 配置为基于可执行 ELF 映像创建无格式二进制文件，请执行下列操作：

1. 转到工程的生成配置面板。请参阅第6-2 页的访问配置面板。
2. 在 C/C++ Build 面板中，单击 Build Steps 选项卡。
3. 在 Post-build step 的 Command 文本框中，输入 fromelf --bin --output output.bin *inputfile*.axf，其中 *inputfile* 是可执行 ELF 映像的名称。请参阅图6-2。
4. 生成工程时，将创建该二进制文件。该二进制文件将以 output.bin 形式保存在活动生成配置的子目录中，例如 Debug。

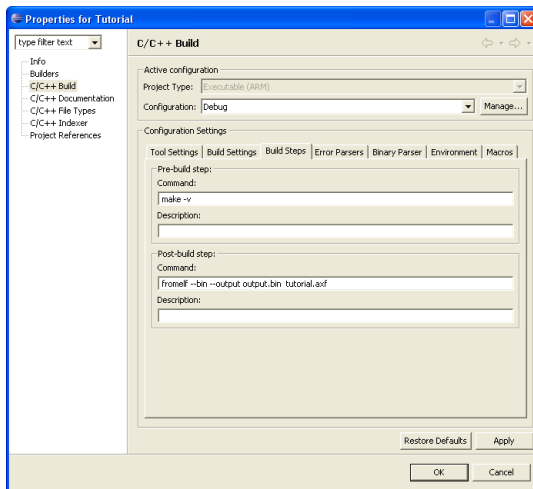


图6-2 在 Eclipse 中调用 fromelf

6.4.2 反汇编代码

还可以使用 ARM `fromelf` 实用程序反汇编 ELF 目标文件并显示各种信息。要反汇编目标文件，请执行下列操作：

1. 在 **C/C++ Projects** 视图中展开工程，以显示各个目标文件。
2. 右击要反汇编的目标文件，然后选择 **OpenWith → FromElfOpener**。或者，您只需双击目标文件即可。
3. 经过反汇编的目标文件显示在 Eclipse 文件编辑器中。
4. 您可以对反汇编后的文件进行保存。

6.5 恢复缺省值

可以使用配置面板（请参阅第 6-2 页的 *访问配置面板*）上的 **Restore Defaults** 按钮重置 **Tool Settings** 选项卡中显示的所有 RealView 工具的设置。工具选项将还原为特定于所选工程类型的缺省值。它只影响活动的生成配置。例如，如果活动配置是 **Debug**，则只重置针对调试配置的所有工具设置。发布配置并不受影响。

—— 注意 ——

- 如果单击通用配置面板（请参阅第 6-2 页的 *访问配置面板*）上的 **Restore Defaults**，将不会重置使用文件特定的配置面板（请参阅第 6-4 页的 *访问适用于特定文件的配置面板*）设置的任何选项。
- 如果在文件特定的配置面板上单击 **Restore Defaults**，此面板上的设置将还原为通用配置面板上的设置。它同样不影响使用工程中其他文件的配置面板设置的任何选项。

6.6 访问在线帮助

要访问有关 ARM RealView 工具及其选项的在线帮助，请执行下列操作：

1. 在 Eclipse 主菜单中，选择 **Help** → **Help Contents**。
2. 随即会打开一个帮助窗口。在 **Contents** 中，选择 **RealView Development Suite Plug-ins**。单击要查看其帮助的 ARM RealView 工具。

第 7 章

使用 RealView Debugger

本章介绍如何在 Eclipse IDE 中启动 RealView® 调试器以及如何配置 RealView Debugger 的连接设置。它包含以下内容:

- 第7-2 页的 *将可执行映像加载到 RealView Debugger 中*
- 第7-4 页的 *创建调试配置*
- 第7-5 页的 *设置调试配置*
- 第7-8 页的 *使用调试配置启动 RealView Debugger*

7.1 将可执行映像加载到 RealView Debugger 中

利用 Eclipse 插件，可以将目标配置与工程文件一起进行设置和保存。这些配置文件存储在工程中的 `rvd` 子目录中。

如果工程包含目标配置文件，则会自动向 RealView Debugger 传送连接详细信息。有关详细信息，请参阅 *使用现有的 Eclipse 目标配置进行加载*。

如果工程不包含目标配置文件，则必须手动设置指向目标的连接。有关详细信息，请参阅第 7-3 页的 *在不使用 Eclipse 目标配置的情况下加载*。

7.1.1 使用现有的 Eclipse 目标配置进行加载

在从 Eclipse 中启动 RealView Debugger 以运行或调试可执行映像之前，应确保 RealView Debugger 处于关闭状态。在从 Eclipse 中启动 RealView Debugger 时，将自动配置 RealView Debugger 中的连接属性并将所有控制传递给 RealView Debugger。您必须使用 RealView Debugger 界面来执行步进、插入断点以及检查内存等调试操作。有关使用 RealView Debugger 的信息，请参阅 *RealView Debugger 用户指南*。要将可执行映像加载到 RealView Debugger 中，请执行下列操作。

1. 在 Eclipse 中，确保工程已生成并包含可执行映像（请参阅第 2-11 页的 *生成工程*）。
2. 在 Eclipse 的 **C/C++ Projects** 视图中，右击工程目录中 **Binaries** 子目录中的可执行文件。
3. 在上下文菜单中，选择 **Debug As → Load into RealView Debugger**。
4. 对于模拟目标，Eclipse 将自动启动 RealView Debugger 并将可执行映像加载到其中。如果目标不是模拟的，则会自动配置 RealView Debugger 中的连接属性，但需要进行手动连接和加载。

Eclipse 会记住上次加载的可执行映像。如果需要从同一工程中重新加载该可执行映像，只需按键盘上的 F11 即可。这会根据需要重新生成工程，并将可执行映像重新加载到 RealView Debugger 中。

——注意——

如果要生成并调试新的目标，则必须关闭 RealView Debugger，然后再加载其他可执行映像。这将允许从 Eclipse 中传送新的目标配置。

7.1.2 在不使用 Eclipse 目标配置的情况下加载

通过从 Eclipse 中启动 RealView Debugger，可以运行或调试可执行映像。首先必须在 RealView Debugger 中创建指向目标的连接。在从 Eclipse 中启动 RealView Debugger 时，会将映像加载到目标中，同时将所有控制传递给 RealView Debugger。您必须使用 RealView Debugger 界面来执行步进、插入断点以及检查内存等调试操作。有关使用 RealView Debugger 的信息，请参阅 *RealView Debugger 用户指南*。要将可执行映像加载到 RealView Debugger 中，请执行下列操作：

1. 启动 RealView Debugger。
2. 在 RealView Debugger 中，创建指向目标的连接。
3. 关闭 RealView Debugger。
4. 在 Eclipse 中，确保工程已生成并包含可执行映像（请参阅第2-11页的*生成工程*）。
5. 在 Eclipse 的 **C/C++ Projects** 视图中，右击工程目录中 **Binaries** 子目录中的可执行文件。
6. 在上下文菜单中，选择 **Debug As → Load into RealView Debugger**。
7. Eclipse 将自动启动 RealView Debugger 并将可执行映像加载到其中。

Eclipse 会记住上次加载的可执行映像。如果要从同一工程中重新加载该可执行映像，只需按键盘上的 F11 即可。这会根据需要重新生成工程，并将可执行映像重新加载到 RealView Debugger 中。

7.2 创建调试配置

您可以为 Eclipse 中的每个可执行映像创建并设置自己的调试配置。要为可执行映像创建新的调试配置，请执行下列操作：

1. 在 **C/C++ Projects** 视图中选择您的工程。
2. 在 Eclipse 主菜单中选择 **Run → Debug**，以显示 **Create, manage, and run configurations** 对话框。
3. 在左侧的配置框中，选择 **RealView Debugger**。
4. 右击并选择 **New**，以便为工程创建新的调试配置。可在 **C/C++ Application** 框中看到可执行映像的名称。

——注意——

如果工程中不包含可执行映像，或者包含多个可执行映像，则“C/C++ Application”框将保留为空白。Eclipse 将显示程序不存在的警告，并禁用面板上的 **Debug** 按钮。请参阅第 7-5 页的 *选择另一个要调试的映像*，了解如何设置要调试的可执行映像。

5. 缺省情况下，将使用工程名称创建一个新的调试配置。它将显示在 **Configurations** 框中的 **RealView Debugger** 下。

7.3 设置调试配置

要调试可执行映像，既可以创建新的调试配置，也可以使用现有的配置。如果尚未创建调试配置，请参阅第 7-4 页的 *创建调试配置*，以创建新的配置。本节介绍如何设置现有的调试配置。

7.3.1 选择现有的调试配置

要选择现有的调试配置，请执行下列操作：

1. 在 Eclipse 主菜单中选择 **Run → Debug**，以显示 **Create, manage, and run configurations** 对话框。
2. 在 **Configurations** 框中，展开 **RealView Debugger**。
3. 选择要使用的调试配置。

RVDS Eclipse 插件提供了四个选项卡式面板，可用于设置或修改选定的配置。以下几节将介绍 **Main**、**Arguments** 和 **Connection** 选项卡。

7.3.2 选择另一个要调试的映像

Main 选项卡提供的选项可将另一个工程或可执行映像与选定的调试配置关联在一起。

Project 您可以在 **Project** 字段中键入工程的名称，也可以单击 **Browse...** 按钮并从可用工程的列表中进行选择。

——注意——

只能选择当前在 Eclipse IDE 中处于打开状态的工程。

C/C++ Application

您可以在 **C/C++ Application** 字段中键入要调试的映像的名称，也可以单击 **Browse...** 按钮并选择映像。单击 **Search Project...** 按钮将显示工程中可用的可执行映像的列表，以供您从中进行选择。来自不同生成配置（例如调试和发布）的可执行映像将显示在此列表中。

7.3.3 配置 RealView Debugger 连接设置

使用 **Connection** 选项卡可以配置 RealView Debugger 加载选项

Connections

如果 RealView Debugger 已连接到多个目标，则可以使用 **Connections** 下拉列表选择映像要加载到的目标。单击 **Get Connections** 按钮，可使 Eclipse 从 RealView Debugger 中获取可用连接的列表。取消选择 **Load into first target** 可查看可用的连接并选择所需的连接。如果选中了 **Load into first target**，Eclipse 会将可执行映像加载到可用连接列表中显示的第一个目标中。

Parts to load

使用此选项可选择要将映像的哪一部分加载到目标中：

Symbols and Image

使用此选项可加载所有的调试符号和程序映像。

Image Only

使用此选项可只加载程序映像，而不加载调试符号。

Symbols Only

使用此选项可只加载符号，而不加载程序映像。

Loading mode

使用此选项可选择是否替换目标中已有的可执行映像

Append 使用此选项可向现有映像附加新的可执行映像。

Replace 使用此选项可将现有映像替换为正在加载的映像。

Sections 可以使用 **Sections** 字段指定在加载映像时要加载的节。此字段通常用于在启动程序时重新加载已初始化的数据节。选择 **Load all sections** 选项作为缺省设置。

Set Program Counter (PC) to start address from object module

可以使用此选项对 PC 进行设置，以使其成为 ELF 映像每次被加载到 RealView Debugger 中时指定的起始地址。

7.3.4 指定执行参数

在 **Arguments** 选项卡中，可以指定可执行映像的参数。

Program Arguments

在 **C/C++ Program Arguments** 文本框中，可以为可执行映像指定一个以空格分隔的参数列表。

Variables 可以将 Eclipse 提供的变量用作可执行映像的参数。单击 **Variables...** 可选择所需的变量，您也可以创建自己的变量。

7.4 使用调试配置启动 RealView Debugger

对于要加载到 RealView Debugger 中的可执行映像，如果您不想更改其调试配置，请按照第 7-2 页的 *将可执行映像加载到 RealView Debugger 中* 中的简单步骤操作。在将可执行映像加载到 RealView Debugger 中之前，如果您希望更改任何调试配置设置，请执行下列操作：

1. 确保 RealView Debugger 已连接至目标。有关详细信息，请参阅 *RealView Debugger 用户指南*。关闭仍在运行的所有 RealView Debugger 实例。
2. 在 Eclipse 主菜单中，选择 **Run → Debug**。
3. 在“Configurations”框中选择您的 RealView Debugger 配置，或创建新的调试配置。请参阅第 7-4 页的 *创建调试配置*。
4. 如果您想更改要调试的工程或映像，请使用 **Main** 选项卡选择另一个工程或映像。
5. 如果要修改可执行映像的参数或向其传递参数，请使用 **Arguments** 选项卡。请参阅第 7-7 页的 *指定执行参数*。
6. 选择 **Connections** 选项卡。单击 **Get Connections**。随即会启动 RealView Debugger，并且 Eclipse 将尝试连接到 RealView Debugger。
7. 当 Eclipse 连接至 RealView Debugger 后，请返回到 Eclipse **Debug** 对话框中的 **Connection** 选项卡。如果已将多个目标连接至 RealView Debugger，请取消选择 **Load into first target** 选项。可用目标连接显示在 **Connection** 下拉列表中。选择要使用的连接（请参阅第 7-9 页的图 7-1）。根据需要修改加载设置。请参阅第 7-6 页的 *配置 RealView Debugger 连接设置*。

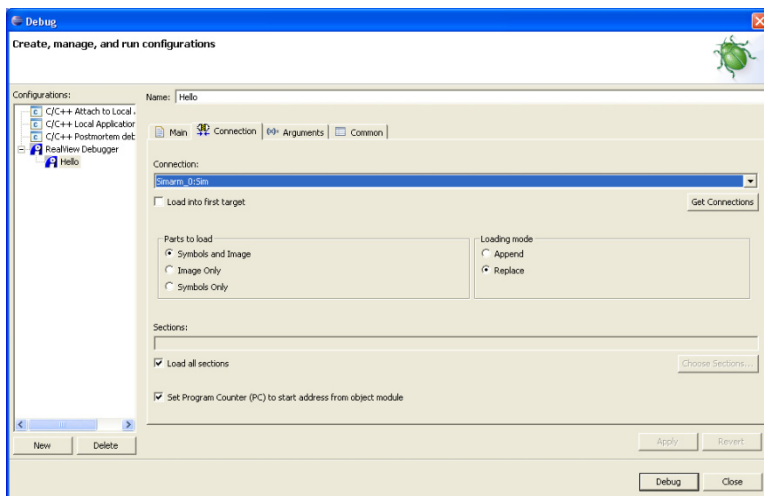


图7-1 “Debug” 面板

8. 单击 **Debug**。根据需要重新生成工程。
9. 您的可执行映像将加载到目标中，随后便可以在 RealView Debugger 中对其进行调试了。

