RealView® Development Suite

4.0 版

入门指南

ARM®

ARM DUI 0255HC

RealView Development Suite
入门指南

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	变更
2003 年 9 月	A	非保密	ARM® RealView® Developer Suite v2.0 版
2004 年 1 月	B	非保密	RealView Developer Suite v2.1 版
2004 年 12 月	C	非保密	RealView Developer Suite v2.2 版
2005 年 5 月	D	非保密	RealView Developer Suite v2.2 SP1 版
2006 年 3 月	E	非保密	RealView Development Suite v3.0 版
2007 年 3 月	F	非保密	RealView Development Suite v3.1 版
2008 年 2 月	G	非保密	RealView Development Suite v3.1 Professional 版
2008 年 9 月	H	非保密	RealView Development Suite v4.0 版

所有权声明

保密状态

本文档的内容是非保密的。根据 ARM 与 ARM 将本文档交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。

受限访问是一种 ARM 内部分类。

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
入门指南

前言

关于本手册 viii
反馈 xiii

第 1 章 简介

1.1 RealView Development Suite 组件 1-2
1.2 RealView Development Suite 授权 1-10
1.3 RealView Development Suite 文档 1-12
1.4 RealView Development Suite 示例 1-14
1.5 ARM Profiler 示例 （仅适用于 RVDS Professional 版） 1-16
1.6 RealView Debugger 中的调试接口支持 1-17
1.7 解决 RVDS 环境的问题 1-18

第 2 章 RealView Development Suite 入门

2.1 生成和调试任务概述 2-2
2.2 使用示例项目 2-5
2.3 ARM Profiler 入门 （仅适用于 RVDS Professional 版） 2-6

第 3 章 **RealView Development Suite 的变化**
3.1 处理器支持 3-2
3.2 模拟器支持 3-3
3.3 RealView 编译工具 3-4
3.4 RealView Debugger 3-5
3.5 RealView ICE、 RealView Trace 和 RealView Trace 2 3-6
3.6 ARM Profiler 3-7
3.7 IDE 支持 3-8
3.8 文档 3-9
3.9 其他更改 3-10
3.10 不提倡使用的功能 3-11
3.11 不再使用的功能 3-12

附录 A **使用 armenv 工具**
A.1 关于 armenv 工具 A-2
A.2 使用 armenv 工具 A-3

附录 B **关于早期版本**
B.1 RVDS v3.1 Professional 版与 RVDS v3.1 之间的差异 B-3
B.2 RVDS v3.1 与 RVDS v3.0 SP1 之间的差异 B-4
B.3 RVDS v3.0 SP1 与 RVDS v3.0 之间的差异 B-8
B.4 RVDS v3.0 与 RealView Developer Suite v2.2 SP1 之间的差异 B-9
B.5 RealView Developer Suite v2.2 SP1 与 RealView Developer Suite v2.2 之间的差异 B-13
B.6 RealView Developer Suite v2.2 与 RealView Developer Suite v2.1 之间的差异 B-14
B.7 RealView Developer Suite v2.1 与 RealView Developer Suite v2.0 之间的差异 B-16
B.8 RealView Developer Suite v2.2 与 ADS v1.2.1 之间的差异 B-18

术语表

前言

本前言对 ARM® 《RealView® Development Suite 入门指南》进行了介绍，说明如何开始使用 ARM *RealView Development Suite* (RVDS) 管理软件项目和调试应用程序。本章分为以下几节：

- 第viii 页的*关于本手册*
- 第xiii 页的*反馈*

关于本手册

RVDS 提供了一些工具，用于对以基于 ARM 体系结构的处理器为目标的软件开发项目进行生成、调试和管理。本手册包含以下内容：

- 介绍构成 RVDS 的软件组件
- 总结 RVDS v4.0 与早期版本 RVDS 之间的差异
- 提供适用于 RVDS 新用户的术语表

适用对象

本手册专为使用 RVDS 管理基于 ARM 体系结构的处理器开发项目的开发人员而编写。本手册假定您是一位有经验的软件开发人员，但可能不熟悉 ARM 开发工具。

使用本手册

本手册由以下章节组成：

第 1 章 ***简介***

本章介绍了 RVDS 组件、授权和文档。

第 2 章 ***RealView Development Suite 入门***

本章概述可以使用 RVDS 工具执行的主要任务。本章还介绍了随 RVDS 提供的示例项目。

第 3 章 ***RealView Development Suite 的变化***

本章介绍 RVDS v4.0 与 RVDS v3.1 之间的差异。

附录 A ***使用 armenv 工具***

本附录介绍如何使用 `armenv` 工具。

附录 B ***关于早期版本***

本章介绍 RVDS 的早期版本。

Glossary RVDS 文档中所用术语的定义 （按字母排序）。

印刷约定

本手册使用以下印刷约定：

斜体	突出显示重要注释，介绍特殊术语，表示内部交叉参考和引用。
粗体	突出显示界面组件，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。
`mono`space	表示允许的命令或选项缩写。可只输入下划线标记的文本，无需输入命令或选项的全名。
`monospace italic`	表示此处的命令和函数的变量可用特定值代替。
`等宽粗体`	表示使用示例代码以外的语言关键字。
`...`	路径名末尾的 ... 表示所需目录在上次指定的目录名称的下面。未指定的路径名通常是那些操作系统不同的目录名称。例如： *`install_directory`*`\ARM\RVDS\Examples\...` 路径名中间的 ... 表示在指定的目录名称之间存在附加目录。未指名的路径名通常是版本号和编号以及特定平台的目录名称。例如： *`install_directory`*`\ARM\RVD\Core\...\etc`

更多参考读物

本节列出了 ARM Limited 和第三方的出版物。

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://infocenter.arm.com/help/index.jsp`。

ARM 出版物

有关控制 ARM 应用程序使用的 FLEX*net* 许可证管理系统的信息，请参阅以下文档：

- 《ARM 工具 FLEXnet 许可证管理指南 4.2 版》(ARM DUI 0209)。

注意

若要获得有关 RVDS v4.0 中的许可证管理的信息，请务必使用此文档的 4.2 版。

注意

FLEX*net* 许可证管理系统由 Acresso Software Inc.（前身是 Software Business Unit of Macrovision Corporation）所有。

本书是 RVDS 文档的一部分。此文档套件中的其他手册包括：

- 《ARM Workbench IDE 用户指南》 (ARM DUI 0330)
- 《RealView 编译工具要点指南》 (ARM DUI 0202)
- 《RealView 编译工具开发指南》 (*RealView Compilation Tools Developer Guide*) (ARM DUI 0203)
- 《RealView 编译工具汇编程序指南》(*RealView Compilation Tools Assembler Guide*) (ARM DUI 0204)
- 《RealView 编译工具编译器用户指南》 (*RealView Compilation Tools Compiler User Guide*) (ARM DUI 0205)
- 《RealView 编译工具编译器参考指南》 (*RealView Compilation Tools Compiler Reference Guide*) (ARM DUI 0348)
- 《RealView 编译工具库和浮点支持指南》 (ARM DUI 0349)
- 《RealView 编译工具链接器用户指南》 (ARM DUI 0206)
- 《RealView 编译工具链接器参考指南》 (ARM DUI 0381)
- 《RealView 编译工具实用程序指南》 (ARM DUI 0382)
- 《RealView Debugger 要点指南》 (*RealView Debugger Essentials Guide*) (ARM DUI 0181)
- 《RealView Debugger 用户指南》(*RealView Debugger User Guide*) (ARM DUI 0153)
- 《RealView Debugger 目标配置指南》 (*RealView Debugger Target Configuration Guide*) (ARM DUI 0182)
- 《RealView Debugger 跟踪用户指南》(*RealView Debugger Trace User Guide*) (ARM DUI 0322)

- 《RealView Debugger RTOS 指南》 (*RealView Debugger RTOS Guide*) (ARM DUI 0323)
- 《RealView Debugger 命令行参考指南》 (*RealView Debugger Command Line Reference Guide*) (ARM DUI 0175)
- 《RealView Emulation Baseboard 实时系统模型用户指南》 (*RealView Emulation Baseboard Real-Time System Model User Guide*) (ARM DUI 0424A)
- 《RealView ARMulator ISS 用户指南》(*RealView ARMulator ISS User Guide*) (ARM DUI 0207)

有关基本标准、软件接口以及 ARM 支持的标准的完整信息，请参阅 *install_directory*\Documentation\Specifications\...。

此外，有关与 ARM 产品相关的特定信息，请参阅下列文档：

- 《RealView ICE 和 RealView Trace 用户指南》 (*RealView ICE and RealView Trace User Guide*) (ARM DUI 0155)
- 《ARM 体系结构参考手册， ARMv7-A™ 和 ARMv7-R™ 版》 (ARM DDI 0406)
- 《ARM7-M™ 体系结构参考手册》 (ARM DDI 0403)
- 《ARM6-M™ 体系结构参考手册》 (ARM DDI 0419)
- 《ARM 参考外围设备规范》(*ARM Reference Peripheral Specification*) (ARM DDI 0062)
- 您的硬件设备的 ARM 数据表或技术参考手册。

其他出版物

有关对 ARM 体系结构的介绍，请参阅 Andrew N. Sloss、Dominic Symes 和 Chris Wright 合著的 《ARM 系统开发人员指南：设计和优化系统软件》 (*ARM System Developer's Guide: Designing and Optimizing System Software*) （2004 年出版）。Morgan Kaufmann， ISBN 1-558-60874-5。

有关面向使用 ARM 处理器的片上系统设计人员和使用 ARM 体系结构的工程师的要点参考手册，请参阅 Steve Furber 编著的《ARM 片上系统体系结构》(*ARM system-on-chip architecture*) （2000 年的第二版）。Addison Wesley， ISBN 0-201-67519-6。

有关 CEVA, Inc. 出品的 CEVA-Oak、 CEVA-TeakLite 和 CEVA-Teak 处理器的详细信息，请访问 http://www.ceva-dsp.com。

反馈

ARM Limited 欢迎用户就 RVDS 及其文档提供反馈意见。

关于 RealView Development Suite 的反馈

如果您有关于 RVDS 的任何问题，请与您的供应商联系。为便于供应商快速提供有用的答复，请提供：

- 您的姓名和公司
- 产品序列号
- 所使用的版本的详细信息，包括版本号和内部版本号
- 您运行的平台的详细信息，如硬件平台、操作系统类型和版本
- 能重现问题的一小段独立的程序
- 您预期发生和实际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

还可以使用 ARM 支持向导向 ARM 支持部门报告有关 RVDS v4.0 的任何问题。有关详细信息，请参阅第1-8 页的*ARM 支持向导*。

——注意——

如果您有关于 RealView Debugger 的任何问题，建议您创建一份软件问题报告。要完成此操作，请在 RealView Debugger 的 **Help/帮助**菜单中选择 **Send a Problem Report/发送问题报告**。有关详细信息，请参阅 RealView Debugger 在线帮助。

关于本手册的反馈

如果您对本书有任何意见，请发送电子邮件至 errata@arm.com，并提供：

- 文档标题
- 文档号
- 您要对其发表意见的页码
- 您的意见的简单说明

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

本章介绍 ARM® RealView® Development Suite (RVDS) v4.0。文中描述了组件应用程序以及您可以购买的用于扩展 RVDS 功能的附加许可证，此外还概述了文档套件。

本章分为以下几节：

- 第1-2 页的*RealView Development Suite 组件*
- 第1-10 页的*RealView Development Suite 授权*
- 第1-12 页的*RealView Development Suite 文档*
- 第1-14 页的*RealView Development Suite 示例*
- 第1-16 页的*ARM Profiler 示例（仅适用于 RVDS Professional 版）*
- 第1-17 页的*RealView Debugger 中的调试接口支持*
- 第1-18 页的*解决 RVDS 环境的问题*

1.1 RealView Development Suite 组件

RVDS 为 ARM 系列处理器上运行的嵌入式系统应用程序提供了协调的开发环境。RVDS 由一系列工具、支持文档和示例组成。这些工具使您可以在目标硬件或软件模拟器上编写、生成和调试应用程序。

1.1.1 RVDS 的安装目录、示例目录和文档目录

随 RVDS 会安装各种目录，其中包含示例代码和文档。RVDS 文档在需要时会引用这些目录。

表 1-1 中列出了主安装目录、示例目录和文档目录。其中显示的 *install_directory* 是缺省安装目录。如果您指定不同的安装目录，则路径名称相对于您选择的目录。

表 1-1 RealView Development Suite 目录

目录	Windows 缺省路径	Red Hat Linux 缺省路径
install_directory	C:\Program Files\ARM	~/arm
示例	*install_directory*\RVDS\Examples\...	*install_directory*/RVDS/Examples/...
ARM Profiler 示例 （仅适用于 RVDS Professional 版）	*install_directory*\Profiler\...\examples\...	*install_directory*/Profiler/.../examples/...
项目模板	*install_directory*\project_templates\...	*install_directory*/project_templates/...
RealView Debugger Flash 示例	*install_directory*\RVD\Flash\...	*install_directory*/RVD/Flash/...
文档	*install_directory*\Documentation\... 和 *install_directory*\Documentation_*component_N.n*\...	*install_directory*/Documentation/... 和 *install_directory*/Documentation_*component_N.n*/...

另请参阅

- 有关访问文档的详细信息，请参阅第1-12 页的*RealView Development Suite 文档*。
- 有关所提供示例的摘要说明，请参阅以下内容：
  - 第1-14 页的*RealView Development Suite 示例*
  - 第1-16 页的*ARM Profiler 示例 （仅适用于 RVDS Professional 版）*

1.1.2 ARM Workbench IDE

ARM Workbench 是一种集成开发环境 (IDE)，将软件开发与 RealView 工具的编译和性能分析技术结合在一起。可以将其作为项目管理器，针对 ARM 目标创建、生成和管理项目。它使用一个称为 “工作区”的文件夹来存储与特定项目相关的文件和文件夹。

另请参阅

- 《ARM Workbench IDE 用户指南》

1.1.3 RealView 编译工具

可以使用 RealView 编译工具从 C、 C++ 或 ARM 汇编语言源代码生成程序。RealView 编译工具 (RVCT) 有以下组成部分：

- ARM 和 Thumb™ C 和 C++ 编译器 (armcc)
- 使用 `armcc --vectorize` 命令调用的 NEON™ 向量化编译器，另请参阅第1-10 页的*RealView Development Suite 授权*。
- ARM 和 Thumb 汇编器 (armasm)
- ARM 链接器 (armlink)
- ARM 库管理程序 (armar)
- ARM 映像转换实用程序 (fromelf)
- 支持库

注意

随 RVDS Professional 版提供了一个许可证，使您可以使用 NEON 向量化编译器。

另请参阅

- 有关访问文档的详细信息，请参阅第1-12 页的*RealView Development Suite 文档*。
- 有关 RVCT 中的可用工具和功能的详细信息，请参阅 《RealView 编译工具要点指南》。
- 有关 RVCT 的可用更新和补丁，请访问 ARM 网站。

1.1.4 RealView Debugger

结合使用 RealView Debugger 和受支持的调试目标，您可以调试应用程序并完全控制程序的执行流，从而快速找出并纠正错误。有关详细信息，请参阅第1-17 页的*RealView Debugger 中的调试接口支持*。

——注意——

有关在 Red Hat Linux 中使用 RealView Debugger 的特定信息，请参阅介绍 Red Hat Linux 中的 RealView Debugger 的附录。您可在 《RealView Debugger 用户指南》 (*RealView Debugger User Guide*) 中找到该附录。

RealView Debugger 支持：

- 多处理器调试
- 数字信号处理器 (DSP) 调试
- 跟踪、分析和性能分析
- 操作系统 (OS) 感知

RealView Debugger 的缺省许可证允许您调试在单个或多个基于 ARM 体系结构的处理器上运行的应用程序。但是，您必须另外购买许可证才能扩展 RealView Debugger 的功能，以支持在 DSP 上调试。

RealView Debugger 下载

ARM 网站上提供了下载，使您可以使用受支持的插件调试 OS 感知应用程序，并获取软件更新和实用程序。

要访问 RealView Debugger 下载，请在 RealView Debugger 中选择：

Help/帮助 → ARM on the Web/Web 上的 ARM → Goto RTOS Awareness Downloads/转至 RTOS 感知下载

这会在 ARM 网站上显示 *OS-Aware and Middleware Debug* （OS 感知和中间件调试）网页。从此处您可找到并下载 OS 插件。

Help/帮助 → ARM on the Web/Web 上的 ARM → Goto Update and Utility Downloads/转至更新和实用程序下载

这会在 ARM 网站上显示 *ARM Technical Support - Downloads* （ARM 技术支持 - 下载）网页。从此处您可以找到并下载任何 ARM 软件更新和实用程序。

另请参阅

- 有关授权的详细信息，请参阅第1-10 页的*RealView Development Suite 授权*。
- 有关访问文档的详细信息，请参阅第1-12 页的*RealView Development Suite 文档*。
- 有关 RealView Debugger 中的可用功能的详细信息，请参阅 《RealView Debugger 要点指南》 (*RealView Debugger Essentials Guide*)。

1.1.5 ARM Profiler

ARM Profiler 是 ARM Workbench IDE 的一个插件。使用 ARM Profiler 可以通过以下两种方式查看代码在目标系统上的执行情况：使用 RealView ICE 和 RealView Trace 2 在目标硬件上观察代码，或是针对 ARM 实时系统模型 (RTSM) 测试代码。当应用程序停止执行时， ARM Profiler 会生成一个分析文件，其中包含有关已执行代码的详细信息 （如各种函数的调用序列、计时特征、周期计数和指令计数）。

如果您有 RVDS Professional 版，则选择安装产品时使用 **Full/全部**选项即会安装 ARM Profiler。这样还提供一个 ARM Profiler 使用许可证。

如果您有 RVDS Standard 版，则可以单独购买 ARM Profiler。此时还必须获取 ARM Profiler 许可证。

另请参阅

- *RealView ICE 主机软件*
- *《ARM Workbench IDE 用户指南》*
- *《ARM Profiler 用户指南》*
- 《RealView ICE 和 RealView Trace 用户指南》 (*RealView ICE and RealView Trace User Guide*)

1.1.6 RealView ICE 主机软件

RealView ICE 运行控制单元在目标硬件与随 RVDS 提供的调试和分析工具之间提供了接口：

- 使用 RealView ICE 和 RealView Debugger 可以调试在目标硬件上运行的应用程序，或是分析在嵌入式跟踪缓冲区™ (ETB™) （如果存在）中捕获的跟踪。如果添加了 RealView Trace 或 RealView Trace 2 数据捕获单元，还可以直接从嵌入式跟踪宏单元™ (ETM™) 捕获并分析跟踪。

- 要使用 ARM Profiler 执行硬件性能分析，您需要以下项目：
  - 使用 TCP/IP 或 USB 连接到主机的 RealView ICE 运行控制单元
  - 使用 USB 连接到主机的 RealView Trace 2 数据捕获单元

注意

RealView Trace 和 RealView Trace 2 仅适用于 Windows 平台。

若在选择安装产品时使用 **Full/全部**选项，即会安装此次 RVDS 发布中提供的 RealView ICE 主机软件版本。

还为 ARM Workbench IDE 提供了一个 RealView ICE 插件。使用该插件可以设置 RealView Debugger 和 ARM Profiler 所使用的硬件目标。

另请参阅

- 第1-4 页的*RealView Debugger*
- 第1-5 页的*ARM Profiler*
- *《ARM Workbench IDE 用户指南》*
- *《RealView Debugger 用户指南》 (RealView Debugger User Guide)*
- *《RealView Debugger 目标配置指南》 (RealView Debugger Target Configuration Guide)*
- *《ARM Profiler 用户指南》*
- 《RealView ICE 和 RealView Trace 用户指南》 (*RealView ICE and RealView Trace User Guide*)

1.1.7 处理器支持

RVDS 支持以下处理器：

- ARM7™、 ARM9™、 ARM10™ 和 ARM11™ 处理器系列
- ARM11 MPCore™ 多核处理器
- Cortex™ 系列处理器

  注意

  随 RVDS Professional 版提供了一个许可证，用于支持 Cortex-A9 处理器。

- RealView Debugger 中的 SecurCore® SC100™ 和 SC200™ 处理器
- RVCT 中的 SecurCore SC300™ 处理器

- RealView Debugger 中支持 Faraday FA526、 FA626 和 FA626TE 处理器
- Marvell Feroceon 88FR101 和 88FR111 处理器 （除了 RVCT 之外，还添加到 RealView Debugger 中）

1.1.8 模拟器支持

RVDS 支持以下模拟器：

- RealView ARMulator® 指令集模拟器 (RVISS)
- 指令集系统模型 (ISSM)
- RTSM
- SoC Designer

RealView ARMulator 指令集模拟器

RealView 可模拟 ARM7、 ARM9、 ARM10 和 ARM11 处理器系列的指令集和体系结构，以及内存系统和外围设备。

RVISS 使您可以在没有目标硬件的情况下开始开发和调试嵌入式应用程序。当硬件仍处于开发阶段或可用的开发板数量有限时，此工具非常有用。

另请参阅：

- *《RealView ARMulator ISS 用户指南》(RealView ARMulator ISS User Guide)*
- 《RealView Debugger 目标配置指南》 (*RealView Debugger Target Configuration Guide*)

指令集系统模型

ISSM 模拟 Cortex 系列 ARM 处理器的指令集和体系结构。

有关详细信息，请参阅 《RealView Debugger 目标配置指南》 (*RealView Debugger Target Configuration Guide*)。

实时系统模型

随 RVDS Professional 版提供了以下 RTSM：

- 具有 ARM926EJ-S™ 的 Versatile *Emulation Baseboard* (EB)
- 具有 ARM1136JF-S™ 的 Versatile EB
- 具有 ARM1176JZF-S™ 的 Versatile EB
- 具有 Cortex-A8 的 Versatile EB

- 具有 Cortex-A9 的 Versatile EB
- 具有 Cortex-R4F 的 Versatile EB

这些模型可以用于 ARM Profiler 和 RealView Debugger。

注意

请注意，*Emulation Baseboard* (EB) RTSM 不适合作为 EB 硬件的特定修订版的软件实现。

注意

要创建自己的 RTSM，您必须购买 RealView System Generator 软件。

另请参阅：

- 《RealView Emulation Baseboard 实时系统模型用户指南》 (*RealView Emulation Baseboard Real-Time System Model User Guide*)
- *《ARM Workbench IDE 用户指南》*
- *《ARM Profiler 用户指南》*
- 《RealView Debugger 目标配置指南》 (*RealView Debugger Target Configuration Guide*)

SoC Designer

RealView Debugger 提供了一个 SoC Designer 调试接口。当在 SoC Designer Simulator 中打开 SoC Designer 模型时，可以使用此接口配置到这些模型的连接。当尝试连接到 SoC Designer 模型中的目标处理器时， RealView Debugger 可以通过包含目标处理器的模型自动打开 SoC Designer Simulator。

没有随 RVDS 提供任何 SoC Designer 模型。

注意

必须单独购买 RealView SoC Designer 软件。

1.1.9 ARM 支持向导

可通过以下方式访问 ARM 支持向导：

- 在 Windows 上，通过**开始 → 所有程序 → ARM → Support Wizard/支持向导**菜单。

- 在 Red Hat Linux 上，通过**开始菜单 → 程序 → ARM → Support Wizard/支持向导**菜单。

该向导收集有关已安装的 ARM 产品的信息，并使您可以将报告保存到文件或通过电子邮件将报告发送给 ARM 支持部门。在将报告发送给 ARM 支持部门之前，必须在其中包含对问题的说明。

1.2 RealView Development Suite 授权

所有 RVDS 授权均由 FLEXnet 许可证管理系统控制。使用 FLEXnet 服务器软件可跟踪并控制您的 RVDS 许可证。您可使用 http://license.arm.com 上的 ARM Web 授权网页请求许可证。有关详细信息，请参见 《ARM 工具 FLEXnet 许可证管理指南》。

本节介绍为 RealView 工具和关联功能单独提供的 RVDS 许可证。

1.2.1 ARM Profiler 许可证

ARM Profiler 许可证使您可以使用 ARM Profiler 通过运行时性能分析对代码的性能进行分析。有关详细信息，请参阅 《ARM Profiler 用户指南》。

注意

随 RVDS Professional 版提供了一个许可证。

1.2.2 NEON 向量化编译器许可证

有了 NEON 向量化编译器许可证，只要对具有 NEON 单元的目标 ARM 处理器（如 Cortex-A8 或 Cortex-A9）合适，编译器便可生成 NEON 指令。

注意

随 RVDS Professional 版提供了一个许可证。

另请参阅

- *《ARM 工具 FLEXnet 许可证管理指南》*
- *《RealView 编译工具要点指南》*
- *《RealView 编译工具编译器用户指南》*
- *《RealView 编译工具编译器参考指南》*

1.2.3 CEVA-Oak 和 CEVA-Teaklite DSP 调试许可证

CEVA-Oak 和 CEVA-Teaklite DSP 支持许可证使您能够调试在 CEVA-Oak 和 CEVA-Teaklite DSP 上运行的应用程序。

有关详细信息，请参阅：

- *《ARM 工具 FLEXnet 许可证管理指南》*
- *《RealView Debugger 要点指南》* (*RealView Debugger Essentials Guide*)

1.2.4 CEVA-Teak DSP 调试许可证

CEVA-Teak DSP 支持许可证使您能够调试在 CEVA-Teak DSP 上运行的应用程序。

有关详细信息，请参阅：

- *《ARM 工具 FLEXnet 许可证管理指南》*
- *《RealView Debugger 要点指南》(RealView Debugger Essentials Guide)*

1.2.5 StarCore SC1200 DSP 调试许可证

StarCore SC1200 DSP 支持许可证使您能够调试在 SC1200 DSP 上运行的应用程序。

有关详细信息，请参阅：

- *《ARM 工具 FLEXnet 许可证管理指南》*
- *《RealView Debugger 要点指南》(RealView Debugger Essentials Guide)*

1.3 RealView Development Suite 文档

第ix 页的*ARM 出版物*中列出了 RVDS 随附的文档。本节介绍如何在线获取更多信息。

注意

提供了一个术语表，其中包含在 RVDS 文档中使用的术语。请参阅本手册中的第Glossary-1 页的*术语表*。

另请参阅每本手册中的 “其他参考读物”部分，了解相关出版物。

1.3.1 在线获得更多信息

根据安装的不同，以 HTML 和 PDF 格式提供了完整的文档套件：

- 根据平台的不同，若要查看文档套件，请执行以下操作：
  - 在 Windows 上，请选择：
    开始 → 所有程序 → ARM → Help viewer v1.0
  - 在 Red Hat Linux 上，请选择：
    开始菜单 → 程序 → ARM → Help viewer v1.0

  这将显示一个单独的浏览器，您可在其中：
  - 查看 HTML 格式的 RVDS 文档
  - 对所有文档或部分文档执行文本搜索
  - 访问每个文档的对应 PDF 文件

  注意

  在从帮助查看器查看 PDF 文档时，无法搜索所有 PDF 文档。

- 根据平台的不同，若要查看 PDF 文档，请执行以下操作：
  - 在 Windows 上，请选择：
    开始 → 所有程序 → ARM → RealView Development Suite v4.0 → RVDS v4.0 Documentation Suite/RVDS v4.0 文档套件
  - 在 Red Hat Linux 上，请选择：
    开始菜单 → 程序 → ARM → RealView Development Suite v4.0 → RVDS v4.0 Documentation Suite/RVDS v4.0 文档套件

  这将显示一个 PDF 文档，其中包含指向 PDF 格式 RVDS 文档的链接。您还可以对该 PDF 文档执行文本搜索。

另请参阅

- 第1-2 页的*RVDS 的安装目录、示例目录和文档目录*

1.4 RealView Development Suite 示例

RVDS 文档中许多示例的代码都位于主示例目录中。有关详细信息，请参阅第1-2 页的*RVDS 的安装目录、示例目录和文档目录*。

此外，该目录还包含文档中没有介绍的示例代码。请参阅每个示例目录中的 `readme.txt` 文件获得更多信息。示例安装在以下几个子目录中：

`asm` ARM 汇编语言编程中的一些示例。《RealView 编译工具汇编器指南》中使用了这些示例。

`cached_dhry` 用于初始化各种 ARM 处理器上的高速缓存和紧耦合内存的例程的示例，这些示例是基于 Dhrystone 示例生成的。支持的处理器包括：

- ARM9xx 处理器
- ARM11xx 处理器
- Cortex-A8
- Cortex-A9
- Cortex-R4

`Cortex-M1` ARM Cortex-M1 处理器的示例，包括分散文件示例和生成脚本示例。

`Cortex-M3` ARM Cortex-M3 处理器的示例，包括分散加载文件示例和构建脚本示例。

`cpp` 一些基本的 C++ 示例。

`databort` 标准的 Data Abort 处理程序的设计说明文档和示例代码。

`dcc` 介绍如何使用调试通信通道的代码示例。该示例在 《RealView 编译工具开发指南》 (*RealView Compilation Tools Developer Guide*) 中进行了介绍。

`dhrystone` Dhrystone 基准程序。RealView Debugger 文档中使用了该示例。

`dsp` 此示例演示 `dspfns.h` 中提供的欧洲电信标准协会 (ETSI) 基本操作的用法。

`emb_sw_dev` 《RealView 编译工具开发指南》中介绍嵌入式软件开发的一章中引用的示例项目。

`fft_v5te` 针对 ARM 体系结构 v5TE 优化的快速傅里叶变换 (FFT) 代码。

`interwork` 说明如何在 ARM 代码和 Thumb 代码之间进行交互操作的示例。有关详细信息，请参阅《RealView 编译工具开发指南》中介绍 ARM 和 Thumb 交互操作的一章。

`linux_apps` 提供一些示例，演示 RVCT 与 GNU 工具链和 GNU 库之间的交互操作，以便生成在 Linux 上运行的应用程序和共享库。还提供一个 PDF 格式的 《应用程序注释 212 ： 使用 RVCT 4.0 以及 GNU 工具和库生成 Linux 应用程序》 (*Application Note 212 Building Linux Applications Using RVCT 4.0 and the GNU Tools and Libraries*)。

`mandelbrot` Mandelbrot 示例 `brot.c`，《RealView Emulation Baseboard 实时系统模型用户指南》 (*RealView Emulation Baseboard Real-Time System Model User Guide*) 中引用了该示例。

`mmugen` MMUgen 实用程序的源代码和文档。这个实用程序可以根据规则文件生成 MMU 页表数据，规则文件描述了所需的虚拟地址到物理地址的转换。

`picpid` 如何编写位置无关代码的示例。

`sorts` 比较 ARM C 库中所使用的插入排序法、希尔排序法和快速排序法的示例代码。

`svc` *超级用户调用* (SVC) 处理程序的示例。

`trace` 《RealView Debugger 跟踪用户指南》(*RealView Debugger Trace User Guide*) 中介绍的跟踪指导中使用的示例应用程序 `trace.c`。应用程序：

- 模拟一个读取一组输入数据样本并计算出样本平均值的小系统
- 针对常见的指令和数据跟踪情况提供了一个框架

`unicode` 能够评估多字节字符支持的示例代码。

`vfpsupport` 启用并执行 VFP 操作的示例代码。还提供了在使用 VFP 时用于配置调试系统的各种实用程序文件，以及 PDF 格式的 《应用程序注释 133：在 RVDS 中使用 VFP》 (*Application Note 133 Using VFP in RVDS*)。

1.5 ARM Profiler 示例 （仅适用于 RVDS Professional 版）

ARM Profiler 作为 RVDS Professional 版安装过程中的选项提供。如果安装了 ARM Profiler，则还会安装以下示例：

`doom` 运行 Doom 的示例代码。

注意

必须下载外部共享软件文件，才可以成功编译和运行此示例。有关详细信息，请参阅 《ARM Profiler 用户指南》。

`fireworks` 产生焰火燃放模拟效果的示例代码。

`fft` 运行 FFT 的示例代码。

`xvid` 显示 MPEG4 编码视频的示例代码。

有关 ARM Profiler 示例目录的位置，请参阅第1-2 页的*RVDS 的安装目录、示例目录和文档目录*。

1.6 RealView Debugger 中的调试接口支持

表 1-2 中显示了 RVDS 中 RealView Debugger 所支持的调试接口。

表 1-2 RVDS v4.0 中支持的 RealView Debugger 调试接口

调试接口	说明	Windows	Red Hat Linux
Instruction Set System Model (ISSM)（不提倡使用）	到模拟 Cortex 目标的连接。	是	是
Real-Time System Model (RTSM)	到选择的部分模拟 ARM Versatile EB 目标的连接。	是	是
RealView ARMulator ISS (RVISS)（不提倡使用）	到模拟 ARM7、 ARM9 和 ARM11 目标的连接。	是	是
RealView ICE	包括以下支持：		
	• 通过 RealView ICE 运行控制单元连接到目标硬件	是	是
	• 使用 RealView Trace 或 RealView Trace 2 数据捕获单元进行跟踪。	是	否
SoC Designer（不提倡使用）	到使用 RealView SoC Designer 软件创建的单处理器或多处理器系统中的模拟目标的连接。	是	是

有以下几点需要注意：

- 要创建 SoC Designer 连接，您必须购买并安装 RealView SoC Designer。
- 要使用 RealView Trace 或 RealView Trace 2 进行跟踪，您必须购买相应的产品。
- 在 RealView Debugger 中，您可以结合使用 RealMonitor 和 RealView ICE。

另请参阅：

- 有关结合使用 RealMonitor 和 RealView ICE 的详细信息，请参阅《RealView Debugger 目标配置指南》 (*RealView Debugger Target Configuration Guide*)。

1.7 解决 RVDS 环境的问题

如果在 RVDS 上运行组件应用软件出现了问题，那么请确保 RVDS 环境配置正确：

- 在 Red Hat Linux 上运行 RVDS40env.posh 脚本。这是在 Red Hat Linux 上设置 RVDS 环境的首选方法。有关运行此脚本的详细信息，请参阅 《RealView Development Suite 安装指南》。
- 在 Windows 上，您可以在安装后使用 armenv 实用程序修改 RVDS 环境。有关详细信息，请参阅附录 A *使用 armenv 工具*。

注意

您不能对自定义安装使用 armenv 实用程序。如果在 Windows 上执行了自定义安装，您必须自己设置环境变量，请参阅*RVDS 环境变量*。在 Red Hat Linux 上请使用 RVDS40env.posh 脚本。

1.7.1 RVDS 环境变量

表 1-3 显示了必须在 Windows 上设置的 RVDS 主要环境变量。用安装路径元素来替换 ...。如果可能的话，使用 *解决 RVDS 环境的问题* 介绍的首选方法来进行设置。此外，确保 PATH 环境变量包括可执行的各种 RVDS 组件应用程序的位置。

表 1-3 在 Windows 中的 RVDS 主要环境变量

环境变量	设置
ARMROOT	安装根目录 (*install_directory*)。缺省值为 C:\Program Files\ARM。
ARMLMD_LICENSE_FILE	ARM RealView 许可证文件的位置。有关此环境变量的信息，请参阅 《ARM 工具 FLEXnet 许可证管理指南》。
ARM_RVI_GDBEXEC	CodeSourcery GNU 调试器可执行文件的位置，其中 *N_n* 是所安装的 RealView ICE 的版本。 *install_directory*\RVI\GDB\N.n\...\arm-none-eabi-gdb.exe
ARM_RVI_HELP_*N_n*	RealView ICE 实用程序的在线帮助文件，其中 *N_n* 是所安装的 RealView ICE 的版本: *install_directory*\Documentation\RVI*N.n*\...
ARM_RVI_ROOT	RealView ICE 的安装根目录 *install_directory*\RVI

表 1-3 在 Windows 中的 RVDS 主要环境变量 （续）

环境变量	设置
ARM_RVI_TOOLS	RealView ICE 实用程序的可执行文件的位置，其中 N_n 是所安装的 RealView ICE 的版本： *install_directory*\RVI\Tools\N.n\...\win_32-pentium\rel
RVCT 环境变量	有关 RVCT 所用环境变量的列表，请参阅 《RealView 编译工具要点指南》。
RealView Debugger 环境变量	有关 RealView Debugger 所用环境变量的列表，请参阅 《RealView Debugger 要点指南》 (*RealView Debugger Essentials Guide*)。
RVDS_PROJECT	标识项目模板目录。
RVDS_PROJECT_WORKDIR	标识项目工作目录。

另请参阅

- *《ARM Workbench IDE 用户指南》*
- *《ARM 工具 FLEXnet 许可证管理指南》*
- *《RealView ARMulator® ISS 用户指南》(RealView ARMulator ISS User Guide)*
- *《RealView 编译工具要点指南》*
- *《RealView 编译工具汇编器指南》*
- *《RealView 编译工具编译器参考指南》*
- *《RealView 编译工具库和浮点支持指南》*
- *《RealView 编译工具链接器参考指南》*
- *《RealView 编译工具实用程序指南》*
- *《RealView Debugger 要点指南》(RealView Debugger Essentials Guide)*
- *《RealView Debugger 目标配置指南》(RealView Debugger Target Configuration Guide)*
- *《RealView ICE 和 RealView Trace 用户指南》(RealView ICE and RealView Trace User Guide*)

第 2 章
RealView Development Suite 入门

本章向您介绍使用 *ARM® RealView® Development Suite (RVDS)* 工具进行生成和调试的基本任务。本章分为以下几节：

- 第2-2 页的*生成和调试任务概述*
- 第2-5 页的*使用示例项目*
- 第2-6 页的*ARM Profiler 入门（仅适用于 RVDS Professional 版）*

2.1 生成和调试任务概述

表 2-1 提供了一个较高层次的过程，显示使用 RVDS 工具生成和调试应用程序的主要任务，并给出了相关参考信息来源。

引用的文档中提及的任务不一定按照表 2-1 中所示的顺序进行介绍。如果您是第一次使用 RealView 工具，建议您按照引用文档中叙述的顺序执行这些任务。表 2-1 中的顺序反映了通常执行任务的顺序。

表 2-1 主要的生成和调试任务

步骤	说明	参考
1	要按哪些步骤操作取决于要调试的映像是否已存在： • 要调试现有映像 （如 RVDS 示例中的 `dhrystone.axf`），请继续步骤 9。 • 要调试尚未生成的映像，且需要使用 RVDS 工具生成该映像，请继续步骤 2。 或者，使用自己选择的生成工具生成映像，然后继续步骤 9 以调试该映像。	第2-5 页的 *使用示例项目*
2	选择要用于管理和生成项目的 RVDS 应用程序： • 要使用 ARM Workbench IDE，请继续步骤 4。 • 要使用 RealView 编译工具 (RVCT) 从系统命令行生成，请继续步骤 3。	
3	如果要直接使用 RVCT 生成工具，则为您的平台创建包含所需生成命令的 makefile 或命令文件。 继续步骤 9 以在 RealView Debugger 中加载和调试映像。	*《RealView 编译工具要点指南》* *《RealView 编译工具开发指南》* *《RealView 编译工具汇编程序指南》 (RealView Compilation Tools Assembler Guide)* *《RealView 编译工具编译器用户指南》 (RealView Compilation Tools Compiler User Guide)* *《RealView 编译工具编译器参考指南》* *《RealView 编译工具库和浮点支持指南》* *《RealView 编译工具链接器用户指南》* *《RealView 编译工具链接器参考指南》* *《RealView 编译工具实用程序指南》*
4	启动 ARM Workbench IDE。	*《ARM Workbench IDE 用户指南》*

表 2-1 主要的生成和调试任务 （续）

步骤	说明	参考
5	如果已存在 ARM Workbench IDE 项目，则继续步骤 7。否则，请为应用程序创建一个 ARM Workbench IDE 项目。	*《ARM Workbench IDE 用户指南》*
6	设置生成应用程序的映像所需的生成配置设置。继续步骤 8。	*《ARM Workbench IDE 用户指南》*
7	打开现有的 ARM Workbench IDE 项目。	*《ARM Workbench IDE 用户指南》*
8	生成 ARM Workbench IDE 项目的映像。	*《ARM Workbench IDE 用户指南》*
9	启动 RealView Debugger。	*《RealView Debugger 要点指南》(RealView Debugger Essentials Guide)*
10	根据需要配置调试目标和连接。	*《RealView Debugger 用户指南》(RealView Debugger User Guide)* *《RealView Debugger 目标配置指南》(RealView Debugger Target Configuration Guide)*
11	连接至您的调试目标。	*《RealView Debugger 要点指南》(RealView Debugger Essentials Guide)* *《RealView Debugger 用户指南》(RealView Debugger User Guide)*
12	加载映像以便调试。	*《RealView Debugger 要点指南》(RealView Debugger Essentials Guide)* *《RealView Debugger 用户指南》(RealView Debugger User Guide)*
13	做好调试的准备，例如断点和跟踪点。	*《RealView Debugger 要点指南》(RealView Debugger Essentials Guide)* *《RealView Debugger 用户指南》(RealView Debugger User Guide)* *《RealView Debugger 跟踪用户指南》(RealView Debugger Trace User Guide)* *《RealView Debugger RTOS 指南》(RealView Debugger RTOS Guide)*
14	运行映像。	*《RealView Debugger 要点指南》(RealView Debugger Essentials Guide)* *《RealView Debugger 用户指南》(RealView Debugger User Guide)*

表 2-1 主要的生成和调试任务 （续）

步骤	说明	参考
15	执行必需的调试和监控任务，例如，步进以及显示变量和存储器的内容。使用跟踪点时，使用 RealView Debugger 的跟踪分析功能来分析跟踪输出。	*《RealView Debugger 要点指南》(RealView Debugger Essentials Guide)* *《RealView Debugger 用户指南》(RealView Debugger User Guide)* *《RealView Debugger 跟踪用户指南》(RealView Debugger Trace User Guide)* *《RealView Debugger RTOS 指南》(RealView Debugger RTOS Guide)*
16	调试会话的结果如何？ • 如果有问题，请继续执行步骤 17。 • 如果没有问题，请重新生成映像以得到最终版本。	*《ARM Workbench IDE 用户指南》* *《RealView 编译工具要点指南》*
17	确定如何修复源代码中的任何问题： • 使用 ARM Workbench IDE。 • 使用您选择的其他源代码编辑器。	*《ARM Workbench IDE 用户指南》*
18	问题解决后，必须重新生成、重新载入并调试映像： • 如果使用的是 ARM Workbench IDE，请返回步骤 8。 • 如果在直接使用 RVCT，请返回步骤 3。	

2.2 使用示例项目

RVDS 文档中描述的任务使用了随 RVDS 提供的一些示例项目。

请以第2-2 页的*生成和调试任务概述* 中所述操作步骤为指导生成和调试您的应用程序，直到您熟悉了这些操作中所涉及的步骤。但是，用户文档中描述的许多任务都要求您修改示例中的文件。在您修改这些文件之前，请备份示例项目文件和目录。

有关随 RVDS 提供的示例项目的详细信息，另请参阅第1-14 页的*RealView Development Suite 示例*。

2.3 ARM Profiler 入门（仅适用于 RVDS Professional 版）

使用 ARM Profiler 可以通过以下两种方式查看代码在目标系统上的执行情况：使用 RealView ICE 和 RealView Trace 2 在实际目标硬件上观察代码，或是针对 ARM 实时系统模型 (RTSM) 测试代码。当应用程序停止执行时， ARM Profiler 会生成一个分析文件，其中包含有关已执行代码的详细信息 （如各种函数的调用序列、计时特征、周期计数和指令计数）。

表 2-2 以较高层次显示了主要任务的执行过程。表 2-2 中的顺序反映了任务的惯常执行顺序。

表 2-2 主要性能分析任务

步骤	说明	参考
1	生成要分析的映像。	第2-2 页的*生成和调试任务概述*
2	启动 ARM Workbench IDE。	*《ARM Workbench IDE 用户指南》*
3	如果已存在 ARM Workbench 项目，请继续步骤 5。	
4	否则，为应用程序创建一个 ARM Workbench 项目并添加映像文件。	*《ARM Workbench IDE 用户指南》*
5	选择要分析的映像。	*《ARM Workbench IDE 用户指南》*
6	确定要使用的收集方法： • 如果要使用硬件或创建自己的运行配置，请继续步骤 7。 • 如果要使用预配置的 RTSM，请继续步骤 8。	
7	在 ARM Workbench 中配置目标连接。	*《ARM Profiler 用户指南》* *《RealView ICE 和 RealView Trace 用户指南》* *(RealView ICE and RealView Trace User Guide)* *《ARM Workbench IDE 用户指南》*
8	运行映像。	*《ARM Profiler 用户指南》* *《ARM Workbench IDE 用户指南》*

表 2-2 主要性能分析任务 （续）

步骤	说明	参考
9	执行所需性能分析任务，如分析摘要报告、代码视图、图表和图形。 如果无需优化，则重新生成映像以获得最终版本。	*《ARM Profiler 用户指南》* *《ARM Workbench IDE 用户指南》* *《RealView 编译工具要点指南》*
10	优化源代码： • 使用 ARM Workbench IDE。 • 使用您选择的其他源代码编辑器。	*《ARM Workbench IDE 用户指南》*
11	返回步骤 1。	

第 3 章
RealView Development Suite 的变化

本章介绍 ARM® RealView® Development Suite (RVDS) v4.0 与上一版本 RVDS v3.1 之间的主要差异。本章分为以下几节：

- 第3-2 页的*处理器支持*
- 第3-3 页的*模拟器支持*
- 第3-4 页的*RealView 编译工具*
- 第3-5 页的*RealView Debugger*
- 第3-6 页的*RealView ICE*、 *RealView Trace 和 RealView Trace 2*
- 第3-7 页的*ARM Profiler*
- 第3-8 页的*IDE 支持*
- 第3-9 页的*文档*
- 第3-10 页的*其他更改*
- 第3-11 页的*不提倡使用的功能*
- 第3-12 页的*不再使用的功能*

3.1 处理器支持

支持以下更多的处理器：

- Cortex™-A9 处理器
- Faraday FA526、 FA626 和 FA626TE 处理器
- Marvell Feroceon 88FR101 和 88FR111 处理器

注意

随 RVDS Professional 版提供了一个许可证，用于支持 Cortex-A9 处理器。

3.2 模拟器支持

对模拟器支持所做的更改包括以下实时系统模型 (RTSM)：

- 具有 ARM926EJ-S™ 的 Versatile Emulation Baseboard (EB)
- 具有 ARM1136JF-S™ 的 Versatile EB
- 具有 ARM1176JZF-S™ 的 Versatile EB
- 具有 Cortex-A8 的 Versatile EB
- 具有 Cortex-A9 的 Versatile EB
- 具有 Cortex-R4F 的 Versatile EB

这些模型取代了 RVDS v3.1 中支持的基于 Integrator/CP 的模型。

注意

随 RVDS Professional 版提供了一个许可证，用于支持 Cortex-A9 模拟处理器。

3.3 RealView 编译工具

《RealView 编译工具要点指南》中介绍了对 RealView 编译工具 (RVCT) 所做的更改。

3.4 RealView Debugger

《RealView Debugger 要点指南》中介绍了对 *RealView Debugger 所做的更改*。

3.5 RealView ICE、 RealView Trace 和 RealView Trace 2

RVDS v4.0 所有版本都随附提供了 RealView ICE 主机软件。

另请参阅：

- 《RealView ICE 和 RealView Trace 用户指南》 (*RealView ICE and RealView Trace User Guide*)。

3.6 ARM Profiler

随 RVDS Professional 版提供了 ARM Profiler 插件及使用许可证。

对 ARM Profiler 的更改包括：

- 支持对 Symbian OS 上运行的应用程序进行性能分析。
- 支持对以下处理器进行数据捕获：
  - ARM7TDMI®
  - ARM946E-S™
  - ARM966E-S™
  - ARM11 MPCore™
  - Cortex-A8
  - Cortex-M3
  - Cortex-R4(F)
- 支持对以下 RTSM 进行数据捕获：
  - Cortex-A9
  - Cortex-R4F
- 对以 10MHz 运行的处理器的准确周期性能分析。
- 支持基于调用链报告进行数据筛选。
- 实时更新，用于实时查看性能分析数据的捕获。
- 堆栈深度跟踪，用于提供调用链在执行过程中的堆栈深度使用情况摘要。

另请参阅：

- 《ARM Profiler 用户指南》。

3.7 IDE 支持

ARM Workbench IDE 面向所有支持的平台提供，并随 RVDS 一起安装。对 ARM Workbench IDE 进行了以下更改：

- Eclipse IDE 及关联插件更新到最新版本。
- 提供了以下编辑器：
  - 用于 ARM 汇编器和 C/C++ 源文件的属性编辑器
  - 用于创建和编辑分散加载描述文件的分散文件编辑器
  - ELF 内容查看器
- 可以导出 IP-XACT 设计文件，供以 RealView Debugger 板/芯片定义 (BCD) 文件格式创建内存映射和外围设备定义。
- ARM 网站上提供了对 ARM Workbench IDE 及插件的更新，网址为：`http://www.arm.com/eclipse`。

另请参阅：

- *《ARM Workbench IDE 用户指南》*
- *《RealView Debugger 目标配置指南》*

3.8 文档

对 RVDS 文档套件的更改包括：

- 提供了新的独立文档浏览器。这使您可以查看 HTML 格式的文档，而不必运行 ARM Workbench IDE。
- 《RealView Development Suite 词汇表》不再是独立的文档。该词汇表包含在 《RealView Development Suite 入门指南》文档中。
- 《RealView 编译工具要点指南》中介绍了对 RVCT 文档所做的更改。
- 《RealView Debugger 要点指南》中介绍了对 RealView Debugger 文档所做的更改。
- 所有文档都反映了组件工具中的功能更改。

有关 RVDS 文档中所用 ARM 术语的列表，另请参阅第Glossary-1 页的*术语表*。

3.9 其他更改

在此版本中还另外进行了以下更改：

- 提供了 ARM 支持向导。有关详细信息，请参阅第1-8 页的*ARM 支持向导*。
- 提供了新示例项目：
  - `linux_apps`
  - `mandelbrot`

  有关详细信息，请参阅第1-14 页的*RealView Development Suite 示例*。

3.10 不提倡使用的功能

以下功能不提倡使用，将在 RVDS 今后的版本中删除：

- *RealView ARMulator® ISS* (RVISS) 支持。
- 指令集系统模型 (ISSM) 支持。
- SoC Designer 支持。
- RVCT 的一些功能。有关详细信息，请参阅 《RealView 编译工具要点指南》。
- RealView Debugger 的一些功能。有关详细信息，请参阅 《RealView Debugger 要点指南》。

3.11 不再使用的功能

以下功能已不再使用：

- Windows 2000 支持。
- CodeWarrior IDE 支持。
- ARM Ltd. 到 Versatile 开发板的直接连接。
- 从以下位置删除了 **Uninstall Wizard/卸载向导**选项：
  - Windows 上的**开始 → 所有程序 → ARM** 菜单。
  - Red Hat Linux 上的**开始菜单 → 程序 → ARM** 菜单。

  特定于产品的 **Modify or Uninstall RVDS/修改或卸载 RVDS *N.n*** 选项仍然可用。
- RVCT 的一些功能。有关详细信息，请参阅 《RealView 编译工具要点指南》。
- RealView Debugger 的一些功能。有关详细信息，请参阅 《RealView Debugger 要点指南》。

附录 A
使用 armenv 工具

本附录介绍 armenv 工具，您可以使用该工具管理 ARM® RealView® 产品安装。

本章分为以下几节：

- 第A-2 页的*关于 armenv 工具*
- 第A-3 页的*使用 armenv 工具*

A.1 关于 armenv 工具

使用 armenv 工具可以：

- 设置 ARM RealView 产品的环境变量
- 删除 ARM RealView 产品的环境变量
- 检查特定主机上安装的 ARM RealView 产品之间是否存在冲突
- 设置同一产品的不同版本

注意

不能使用 armenv 工具在本次发行的 *RealView Development Suite* (RVDS) 中进行自定义安装。

您可以在以下位置找到 armenv 工具：

install_directory/bin/*platform*

A.2 使用 armenv 工具

本节介绍 armenv 命令的句法，并显示如何使用该命令的一些示例。

A.2.1 armenv 命令语法

armenv 工具的命令语法为：

```
armenv [-r root] [-u] -p product [[--and] -p product]...
[--user|--system|--proc] [--bat|--sh|--csh|--posh|--exec program [args]]
```

表 A-1 显示了所有平台上均可用的命令行参数。

表 A-1 一般 armenv 参数

参数	说明
--help	显示有关命令行参数的帮助。
-r *root*	产品安装的根目录的绝对路径， *install_directory*。例如，在 Windows 上，缺省根目录为： C:\Program Files\ARM
-p *product*	ARM RealView 产品。有关详细信息，请参阅第 A-5 页的*产品语法*。
--and	计算该参数之前所有产品的设置，然后计算该参数后面的产品设置。第二组中的设置会覆盖第一组中的设置。
--proc	仅更改当前过程的环境。 在 Windows 上，该参数不能与 --system 或 --user 一起使用。
--exec	在新环境中运行程序。 在 Windows 上，该参数不能与 --bat 一起使用；在 Red Hat Linux 上，该参数不能与 --sh、--csh 或 --posh 一起使用。
-u	在安装产品时，尝试撤消对环境所做的更改。

表 A-2 显示了特定于 Windows 系统的命令行参数。

表 A-2 特定于 Windows 的 armenv 参数

参数	说明
`--system`	更新注册表的 Windows SYSTEM 区域。此为缺省值。
`--user`	更新注册的 Windows USER 区域。
`--bat`	更改当前命令提示符窗口的环境。此为缺省值。

表 A-3 显示了特定于 Red Hat Linux 系统的命令行参数。您只能指定这些参数中的一个。

表 A-3 特定于 Red Hat Linux 的 armenv 参数

参数	说明
`--csh`	生成 csh 句法命令行解释器脚本。
`--sh`	生成 sh 句法命令行解释器脚本。
`--posh`	生成可移植的命令行解释器脚本。此为缺省值。

产品语法

用于指定产品的语法是：

```
armenv -p category [name] [version [revision]] [-v name value]...
```

其中：

category 为产品标识符，例如 RVDS。

name armenv 使用缺省名称 Contents。请不要为此选项使用任何其他名称。

version 为产品的版本号，例如 3.1。如果未指定版本，则使用安装产品的最新版本。

revision 为产品的特定内部版本号。如果未指定内部版本号，则使用安装产品的最新内部版本号。

`-v name value`

标识同一产品的变化版本：

name 变化版本的类型，例如 platform。建议您仅使用 platform 变化版本。

value 特定的变化版本，例如 linux-pentium。

例如，您可能已安装 RVDS v3.1 的 Red Hat Linux 变化版本。

示例 A-1 如何使用 armenv

- 要设置用于 csh shell 和 RVDS v4.0 的最新内部版本的 Red Hat Linux 环境变量，请输入：

  ```
  armenv -r ~/ -p RVDS 4.0 -v platform linux-pentium --csh
  ```

- 要检查 RVCT v4.0 与 RVCT v3.1 之间是否存在冲突，请输入：

  ```
  armenv -p RVCT 4.0 -p RVCT 3.1
  ```

- 要使用 RVCT v4.0 设置覆盖 RVCT v3.1 设置，请输入：

  ```
  armenv -p RVCT 3.1 --and -p RVCT 4.0
  ```

附录 B
关于早期版本

本章概括介绍 ARM® *RealView® Development Suite* (RVDS) 各早期版本之间的主要差异。本章分为以下几节：

- 第B-3 页的*RVDS v3.1 Professional 版与 RVDS v3.1 之间的差异*
- 第B-4 页的*RVDS v3.1 与 RVDS v3.0 SP1 之间的差异*
- 第B-8 页的*RVDS v3.0 SP1 与 RVDS v3.0 之间的差异*
- 第B-9 页的*RVDS v3.0 与 RealView Developer Suite v2.2 SP1 之间的差异*
- 第B-13 页的*RealView Developer Suite v2.2 SP1 与 RealView Developer Suite v2.2 之间的差异*
- 第B-14 页的*RealView Developer Suite v2.2 与 RealView Developer Suite v2.1 之间的差异*
- 第B-16 页的*RealView Developer Suite v2.1 与 RealView Developer Suite v2.0 之间的差异*
- 第B-18 页的*RealView Developer Suite v2.2 与 ADS v1.2.1 之间的差异*

有关 RVDS v4.0 与 RVDS v3.1 之间的差异，请参阅第 3 章 *RealView Development Suite 的变化*。

B.1 RVDS v3.1 Professional 版与 RVDS v3.1 之间的差异

本节介绍 RVDS v3.1 Professional 版与 RVDS v3.1 之间的主要差异。

B.1.1 RVDS v3.1 Professional 版中的新功能

RVDS v3.1 Professional 版中提供以下新功能：

- RealView Profiler，其中包括示例、文档和以下实时系统模型：
  - ARM926 Emulation Board (EB)
  - ARM1136 EB
  - ARM1176 EB
  - Cortex™-A8 EB
- RealView ICE v3.2 主机软件。
- 用于 NEON™ 向量化编译器和 RealView Profiler 的许可证。
- Eclipse 插件：
  - ARM Flash 编程器
  - ARM 汇编器编辑器
  - CodeWarrior 导入程序

B.2 RVDS v3.1 与 RVDS v3.0 SP1 之间的差异

本节介绍 RVDS v3.1 与 RVDS v3.0 SP1 之间的主要差异。

B.2.1 处理器支持

处理器支持包括：

- ARM Cortex-M1
- ARM Cortex-M3 修订版 1
- ARM Cortex-R4
- StarCore SC1200 DSP （仅支持调试）

B.2.2 模拟器支持

RVDS 提供以下模拟器支持：

- 指令集系统模型 (ISSM) 模拟以下附加处理器：
  - Cortex-M1
  - Cortex-M3 修订版 1，该版本支持周期计数
  - Cortex-R4
- RealView SoC Designer，用于支持到 SoC Designer 目标的连接。必须单独购买 RealView SoC Designer。
- 实时系统模型 (RTSM)，用于支持到 RTSM 目标的连接。

对上述各项的支持随 RealView Debugger 一起安装。

B.2.3 项目模板支持

Eclipse 新项目向导允许您根据应用程序要求为 RVDS 组件工具创建新项目。这些项目可以基于与 RVDS 一起提供的项目模板。

提供了其他 RealView Debugger 和 RealView 编译工具 (RVCT) 命令行选项，以支持 RVDS 项目模板的使用：

- `--no_project`
- `--project` *`filename`*
- `--reinitialize_workdir`
- `--workdir` *`pathname`*

另外，以下目录中还提供了预配置的项目模板：

install_directory\project_templates

这些项目模板分在以下子目录中：

ARM RealView 开发板

这些项目模板包含 RealView Debugger 配置，使您可以通过 RealView ICE 和关联的连接接口连接到目标。

裸 ARM 内核

这些项目模板包含 RealView Debugger 配置，使您能够根据具体需要通过 ISSM 和 RealView ARMulator ISS 接口连接到目标。

您还可以使用以下环境变量来设置项目模板和工作目录值：

- RVDS_PROJECT
- RVDS_PROJECT_WORKDIR

另请参阅

- *《RealView 编译工具编译器参考指南》*
- *《RealView Debugger 用户指南》 (RealView Debugger User Guide)*

B.2.4 RealView 编译工具

《RealView 编译工具要点指南》中介绍了对 RVCT 所做的更改。

B.2.5 RealView Debugger

《RealView Debugger 要点指南》 (*RealView Debugger Essentials Guide*) 中介绍了对 RealView Debugger 所做的更改。

B.2.6 IDE 支持

Eclipse 和 RVDS Eclipse 插件作为 RVDS 安装的一部分安装在支持的所有平台上。有关使用 RVDS Eclipse 插件的详细信息，请参阅 《RealView Development Suite Eclipse 插件用户指南》。

B.2.7 文档的更改

RVDS 文档的主要更改如下：

- 所有 RVDS 文档都以 HTML 格式提供。Eclipse 查看器支持在所有文档中进行搜索。尽管可以在另外一个 Web 浏览器中查看文档，但不能对所有文档进行搜索。
- 《RealView Debugger 要点指南》 (*RealView Debugger Essentials Guide*) 中介绍了对 RealView Debugger 文档所做的更改。
- 《RealView 编译工具要点指南》中介绍了对 RVCT 文档所做的更改。

B.2.8 不提倡使用的功能

以下是 RVDS v3.1 中不提倡使用的功能：

- Windows 2000 支持。
- CodeWarrior IDE 支持。
- RVCT 的一些功能。有关详细信息，请参阅 《RealView 编译工具要点指南》。
- RealView Debugger 的一些功能。有关详细信息，请参阅 《RealView Debugger 要点指南》 (*RealView Debugger Essentials Guide*)。

B.2.9 不再使用的功能

以下功能在 RVDS v3.1 中已不再使用：

- 对 *ARM eXtended Debugger* (AXD) 和 *ARM Symbolic Debugger* (armsd) 的支持。
- 不再提供 ARM Developer Suite™ v1.2.1 CD-ROM。
- 对 Solaris 平台的支持。
- 对 Red Hat Enterprise Linux v3 平台的支持。
- 不再提供 Dynatext 文档。
- RVCT 的某些功能已不再使用。
- RealView Debugger 的某些功能已不再使用。

另请参阅

- *《RealView 编译工具要点指南》*
- *《RealView Debugger 要点指南》* (*RealView Debugger Essentials Guide*)

B.3 RVDS v3.0 SP1 与 RVDS v3.0 之间的差异

RVDS v3.0 Service Pack 1 在原始 RVDS v3.0 版本的基础上，还合并了 RVCT 和 RealView Debugger 中的增强功能，其中包括：

- 对 Cortex-R4 的初步支持，包括编译器支持、调试器支持以及一个新的指令集系统模型 (ISSM)
- 通过 RVDS v3.0 缩短编译时间并改进 DWARF3 调试数据的大小
- SIMD NEON 汇编器扩展至包含编程器的表示法
- 多处理器 MPCore™ 目标的改进的调试用户界面
- 附加的 Cortex-M3 示例
- Marvell Feroceon 88FRxxx 处理器

“RealView Debugger Synchronization Control/RealView Debugger 同步控制”窗口已重新设计，其中也包含多种操作的同步。提供了相应的 `SYNCHACTION` CLI 命令。

有关详细信息，请参阅 《RealView Development Suite v3.0 SP1 版本声明》(*RealView Development Suite v3.0 SP1 Release Notes*)。

B.4 RVDS v3.0 与 RealView Developer Suite v2.2 SP1 之间的差异

本节介绍 RVDS v3.0 与 RealView Developer Suite v2.2 SP1 之间的差异。

B.4.1 RVDS v3.0 中的新功能

RVDS v3.0 提供以下新功能：

- 对 TrustZone® 技术的支持。
- 支持 Thumb®-2 执行环境 (Thumb-2EE)。
- 支持 ARM Cortex 处理器系列产品：
  - Cortex-A8
  - Cortex-M3
- 提供了 Cortex-A8 和 Cortex-M3 处理器的模拟器模型。这些模型可通过 RealView Debugger 中的新 ISSM 目标访问来访问。

B.4.2 调试器支持

对 RealView Debugger v3.0 的主要更改如下：

- RealView Debugger 作为单进程运行。目标载体服务器 (TVS) 不再作为单独实体。
- “Connection Control/连接控制”窗口已重新设计。有关详细信息，请参阅《RealView Debugger 用户指南》 (*RealView Debugger User Guide*)。
- **Synch/同步**选项卡上的功能可在单独的 “Synchronization Control/同步控制”窗口中使用。有关详细信息，请参阅 《RealView Debugger 用户指南》 (*RealView Debugger User Guide*)。
- “Register/寄存器”窗格已重新设计。您可以通过将选中的寄存器复制到“User/用户”选项卡来创建一个用户专用视图。有关详细信息，请参阅《RealView Debugger 用户指南》 (*RealView Debugger User Guide*)。
- RealView Debugger 项目管理器及相关功能已删除，因此您不能再在 RealView Debugger 内创建项目和生成映像。但是，源代码编辑和搜索功能仍然可用。

注意

要在 RVDS v3.0 中创建并生成项目，请使用 RVDS CodeWarrior （请参阅第B-11 页的*RVDS CodeWarrior 的更改*）。

- 模拟器支持已更改。有关详细信息，请参阅第B-11 页的*模拟器支持*。
- RealView Broker (RVBroker) 已重新设计。虽然 RealView Debugger 仍自动运行 RVBroker 以进行本地主机 RealView ARMulator® ISS 连接，但用于远程模拟器连接的 RealView Broker 的启动方式已更改。在远程工作站上启动 RealView Broker 时必须指定用户名。有关详细信息，请参阅《RealView Debugger 目标配置指南》 (*RealView Debugger Target Configuration Guide*)。

注意

RVDS v3.0 中已删除对 Multi-ICE® 直接连接的支持。

有关对 RealView Debugger 的更改的详细信息，请参阅 《RealView Debugger 要点指南》 (*RealView Debugger Essentials Guide*)。

B.4.3 编译工具支持

对 RVCT v3.0 的主要更改如下：

- RVCT v3.0 支持 Thumb-2EE。
- ARM 汇编器可用于汇编 Intel 无线 MMX 技术指令，从而针对 PXA270 处理器开发代码。
- RVCT v3.0 为 DWARF 3 （标准草案 9.6）调试表提供完全支持，如《ARM 体系结构的 ABI （基本标准）》 (*ABI for the ARM Architecture (base standard)*) [BSABI] 中所述。
- ARM 编译器和链接器支持*线程局部存储* (TLS) 使程序可以使用多个线程。
- ARM 编译器支持改进的循环优化。

有关对 RVCT 的更改的详细信息，请参阅 《RealView 编译工具要点指南》。

B.4.4 模拟器支持

RVDS 提供以下模拟器支持：

- ISSM，模拟 Cortex-A8 和 Cortex-M3 处理器。
- RealView ARMulator ISS 中提供了 MPCore 模拟目标。但是，这并不对多个处理器建模，因此与此模型的连接将仅连接至一个处理器。

RDI ARMulator 模拟目标不再可用。请使用以下方法之一：

- `localhost` 目标访问上的 `new_arm` 连接，用于使用 RealView ARMulator ISS 连接到模拟的 ARM 处理器。
- `ISSM` 目标访问，用于连接到一种 Cortex 模型。

这些均随 RealView Debugger 一起安装。

B.4.5 RVDS CodeWarrior 的更改

对 RVDS CodeWarrior 的主要更改如下：

- 支持外部生成向导。该向导旨在取代已不提倡使用的 makefile 导入程序和 Batch File Runner 功能。
- 增加了对 `.cc` 文件扩展名的支持。
- 如果使用了无法识别的源文件扩展名 （如 `.cmd`）， CodeWarrior 会发出警告。
- 根据对编译工具的更改增减了面板设置。有关详细信息，请参阅《RealView 编译工具要点指南》。

有关详细信息，请参阅 《RealView Development Suite CodeWarrior IDE 指南》(*RealView Development Suite CodeWarrior IDE Guide*)。

B.4.6 文档的更改

除了新增介绍 RVDS 新功能的文档以外，对 RVDS 文档的更改主要是对 RealView Debugger 文档的更改。对 RealView Debugger 文档的重组如下所示：

- 《RealView Debugger 扩展用户指南》 (*RealView Debugger Extensions User Guide*) 中的信息移入以下文档：
  - 介绍 DSP 支持的章节包含在 《RealView Debugger 用户指南》(*RealView Debugger User Guide*) 中

  - 介绍多目标调试的章节包含在 《RealView Debugger 用户指南》 (*RealView Debugger User Guide*) 中
  - 介绍 RealView Debugger 中的跟踪的章节包含在《RealView Debugger 跟踪用户指南》 (*RealView Debugger Trace User Guide*) 中
  - 介绍 OS 支持的章节包含在 《RealView Debugger RTOS 指南》 (*RealView Debugger RTOS Guide*) 中

- 《RealView Debugger 目标配置指南》 (*RealView Debugger Target Configuration Guide*) 中介绍的有关连接目标的章节包含在 《RealView Debugger 用户指南》 (*RealView Debugger User Guide*) 中。
- 《RealView Debugger 用户指南》 (*RealView Debugger User Guide*) 的结构已更改，现在更多以任务为主线。
- 由于 RealView Debugger 项目管理器已删除，因此未提供 《RealView Debugger 项目管理指南》 (*RealView Debugger Project Management Guide*)。

有关对 RVDS 文档套件所做的其他详细更改，请参阅：

- *RealView Debugger 要点指南*
- *《RealView 编译工具要点指南》*

B.4.7 已弃用和已删除的功能

以下功能在 RVDS v3.0 中已弃用或已删除：

- 已弃用对 *ARM eXtended 调试器* (AXD) 和 *ARM Symbolic 调试器* (armsd) 的支持。
- 已弃用 CodeWarrior 中的 makefile 导入程序和 Batch File Runner 功能。
- 已删除对通过 Multi-ICE 直接连接进行的远程 RealView Debugger 连接的支持。这意味着与 DSP 处理器的连接只可随 RealView ICE 一起使用，而您必须单独购买 RealView ICE。
- 已删除 RealView Debugger 项目管理器及相关功能。

有关其他已不提倡使用的功能的详细信息，请参阅：

- *《RealView 编译工具要点指南》*
- *《RealView Debugger 要点指南》 (RealView Debugger Essentials Guide)*

B.5 RealView Developer Suite v2.2 SP1 与 RealView Developer Suite v2.2 之间的差异

本节介绍 RealView Developer Suite v2.2 SP1 与 RealView Developer Suite v2.2 之间的差异。

B.5.1 文档的更改

对文档的更改包括：

- 提供介绍如何使用 CodeWarrior 的 ARM 功能的《RealView Developer Suite CodeWarrior IDE 指南》(*RealView Developer Suite CodeWarrior IDE Guide*)。
- 介绍 CodeWarrior 使用入门的章节已从 《RealView Developer Suite 使用入门指南》(*RealView Developer Suite Getting Started Guide*) 中删除，并合并至《RealView Developer Suite CodeWarrior IDE 指南》 (*RealView Developer Suite CodeWarrior IDE Guide*)。
- 对受支持的 DSP 的 RealView Debugger 文档的更改。

B.5.2 调试器支持

RealView Developer Suite v2.2 SP1 和 RealView Developer Suite v2.2 中的调试工具之间的主要差异在于前者采用了 RealView Debugger，它支持 CEVA-Oak、CEVA-TeakLite、 CEVA-Teak、 ZSP400 和 ZSP500 DSP。

B.5.3 编译工具支持

RealView Developer Suite v2.2 SP1 与 RealView Developer Suite v2.2 的编译工具之间略微存在一些差异。有关详细信息，请参阅 《RealView 编译工具要点指南》。

B.6 RealView Developer Suite v2.2 与 RealView Developer Suite v2.1 之间的差异

本节介绍 RealView Developer Suite v2.2 与 RealView Developer Suite v2.1 之间的差异。

B.6.1 IDE 支持

提供了 CodeWarrior IDE 以取代 RealView Debugger IDE。RealView Developer Suite v2.2 中的 CodeWarrior IDE 基于 Metrowerks CodeWarrior v5.6。

——注意——

在 RealView Developer Suite v2.2 中，仅在 Windows XP 和 Windows 2000 系统中支持 RealView Developer Suite CodeWarrior，没有为 Red Hat Linux 提供该组件。

B.6.2 调试器工具支持

RealView Developer Suite v2.2 与 RealView Developer Suite v2.1 中的调试工具之间的主要差异在于前者采用了 RealView Debugger， RealView Debugger 具有：

- 改进的菜单结构
- 改进的窗格处理机制
- 改进的采用新的 “Data Navigator/数据浏览器”窗格的数据浏览
- 国际化支持
- 改进的源代码着色
- 跟踪、分析和配置增强功能
- 增强的 RTOS 支持
- 支持 gcc 构建映像
- 附加 CLI 命令， `PRINTDSM` 和 `TRACEEXTCOND`

同时，已从 RealView Debugger 中删除对独立编辑器和 Vi 编辑模式的支持。

有关更改的详细列表，请参阅 《RealView Debugger 要点指南》 (*RealView Debugger Essentials Guide*)。

B.6.3 编译工具支持

RealView Developer Suite v2.2 与 RealView Developer Suite v2.1 中的编译工具之间的主要差异在于：

- RVCT v2.2 包含对新的 ARMv6 内核（例如采用了 ARM TrustZone 技术的 ARM1176JZF-S™ 以及 ARM968EJ-S™、 ARM1156T2F-S™ 和 ARM MPCore™）的支持。
- RVCT v2.2 中支持的新 Thumb-2 指令集引入了许多新的 32 位指令和一些新的 16 位指令。

  Thumb-2 指令集包含旧版本的 16 位 Thumb 指令作为子集。
- RVCT v2.2 完全符合《ARM 体系结构的基础平台 ABI》(*Base Platform ABI for the ARM Architecture*) [BPABI] （未公布的草案）。
- RVCT v2.2 提供了对 DWARF3 （草案标准 9）调试表的初级支持，如《ARM 体系结构的 ABI（基本标准）》(*ABI for the ARM Architecture (base standard)*) [BSABI] 中所述。
- 命令行选项 `-g` 启用生成的当前编译的调试表。优化选项由 `-O0`、 `-O1`、 `-O2` 或 `-O3` 指定。缺省情况下，使用 `-g` 选项并不会影响优化设置。

  这是对 RVCT v2.2 行为的更改。
- RVCT v2.2 支持命令行选项 `--apcs /fpic` 以编译与 System V 共享库兼容的代码。
- ARM 链接器支持生成共享库以及对共享库进行链接。新的命令行选项可用于构建 SVr4 执行文件和共享对象，以及指定如何生成代码。
- ARM 链接器支持 GNU 扩展的符号版本模型。
- 浮点计算的 ARM 执行已更改为为 C99 功能提供改进的支持。在此更改显著改变行为的情形中，引入了兼容模式，来辅助开发人员迁移代码以使用新的功能。
- RVCT v2.2 支持 Linux 应用程序和共享库的构建。

有关更改的详细列表，请参阅 《RealView 编译工具要点指南》。

B.6.4 安捷伦探测器支持

RealView Developer Suite v2.2 中以自定义安装选项的方式提供安捷伦探测器支持。

B.7 RealView Developer Suite v2.1 与 RealView Developer Suite v2.0 之间的差异

本节介绍 RealView Developer Suite v2.1 与 RealView Developer Suite v2.0 之间的差异。

B.7.1 调试器工具支持

RealView Developer Suite v2.1 与 RealView Developer Suite v2.0 中的调试工具之间的主要差异在于：

- 包含了 *ARM eXtended Debugger* (AXD)
- 包含了 *ARM Symbolic Debugger* (armsd)
- RealView Debugger 拥有：
  - 跟踪和配置的增强功能
  - 增强的 RTOS 支持
  - 对工具栏按钮和菜单所做的新更改更加方便对常用功能的访问

B.7.2 编译工具支持

RealView Developer Suite v2.1 与 RealView Developer Suite v2.0 中的编译工具之间的主要差异在于：

- 更为符合 《ARM 体系结构的应用程序二进制接口 （基本标准）》 (*Application Binary Interface for the ARM Architecture (Base Standard)*) （ARM 体系结构的 ABI （基本标准））。请参阅 `http://www.arm.com/` 上的“ARM 体系结构的 ABI”一页。
- 支持 C++ 异常处理。因此，根据 *ISO/IEC 14822 :1998 International Standard for C++* 的定义，支持 ISO C++ 的除导出模板以外的剩余部分。
- 包含更多优化功能，如多文件编译和链接器反馈。
- 提供了可读写数据区压缩以进一步缩小映像大小。
- 支持某些 GNU C 和 C++ 扩展。
- 许多新的命令行选项添加至构建工具。
- 不提倡使用单短线关键字和某些命令行选项。

注意

编译工具更严格地检查 8 字节堆栈对齐的要求。编译器通过 PRESERVE8 和 REQUIRE8 生成代码。链接器检查要求 8 字节对齐的代码仅调用保持 8 字节对齐的代码。因此，这与您的旧版本的汇编程序代码、目标文件和库有关。您必须检查您现有的汇编文件、目标文件或库是否符合 8 字节对齐要求，并在需要时纠正它们。有关详细信息，请参阅 《RealView 编译工具汇编器指南》和《RealView 编译工具链接器和实用程序指南》。

B.8 RealView Developer Suite v2.2 与 ADS v1.2.1 之间的差异

本节介绍 RealView Developer Suite v2.2 与 ADS v1.2.1 之间的差异。

B.8.1 CodeWarrior IDE 的更改

RealView Developer Suite CodeWarrior 与 ADS CodeWarrior 之间的差异为：

- ADS CodeWarrior 以 CodeWarrior v4.2 为基础。RealView Developer Suite CodeWarrior 基于 Metrowerks CodeWarrior v5.6。
- CodeWarrior v5.6 中已删除 CodeWarrior v4.2 中为处理 Perl 脚本提供支持的 CodeWarrior Perl 插件 MWPerl。Metrowerks 不再对其进行支持。
- ARM 工具特定的配置面板在 RealView Developer Suite v2.2 中根据需要进行了改造。
- 各个单独的 ARM 编译器在 RealView Developer Suite v2.2 中合并成一个编译器，因此在 RealView Developer Suite v2.2 中只有一个编译器配置面板。
- 除 AXD 和 armsd 之外，您还可以通过 RealView Debugger 运行和调试映像。
- 您可以将各个库连接起来。
- 您可以将 ADS 项目的 CodeWarrior 导入 RealView Developer Suite CodeWarrior 中。
- RealView Developer Suite CodeWarrior 的缺省 ARM stationery 不包含 `DebugRel` 生成目标。但是，如果您导入 ADS 项目的 CodeWarrior，则会创建一个 `DebugRel` 生成目标以保存您可能为该生成目标配置的所有设置。
- 不同于 ADS 编译器， RVCT 编译器不生成浏览器信息。此功能由 CodeWarrior 的内置语言分析器提供。
- 代码格式设置。
- 代码完成，包含 C++ 模板类的代码完成。
- 转到下一/上一功能。
- 打印时换行。
- 支持与源文件相关的 `#includes`。
- 在注释内/外查找。

- 改进的语言分析器速度和反馈。
- 新建编辑器绑定。
- 在项目窗口中显示和隐藏 “Code/代码”列和 “Data/数据”列的功能。
- 支持工作空间。

——注意——

RealView Developer Suite CodeWarrior 中未提供 CodeWarrior IDE 中的所有目标连接和调试功能。您必须运行一个 ARM 调试器才能执行这些功能。

B.8.2 调试器的更改

RealView Developer Suite v2.2 与 ADS v1.2.1 中的调试工具之间的主要差异在于：

- RealView Debugger 是最新的 ARM 调试器，使您能够执行高级调试功能，例如：
  - 多处理器调试
  - OS 感知调试
  - 扩展的目标可见度
  - 跟踪、分析和配置
  - 通过以太网和 USB 访问 RealView ICE JTAG 控制单元
- 增强的 AXD 可以调试使用随 RealView Developer Suite v2.2 提供的新 RVCT 生成的 C 和 C++ 程序。

B.8.3 编译工具的更改

RealView Developer Suite v2.2 与 ADS v1.2.1 中的生成工具之间的主要差异如下：

- 符合 ARM 体系结构的新 ABI（基本标准）。请参阅 `http://www.arm.com/` 上的 “ARM 体系结构的 ABI”一页。这与旧版本的 ADS ABI 不同。某些兼容性随 `--apcs /adsabi` 命令行选项一起提供。
- 根据 *ISO/IEC 14822 :1998 International Standard for C++* 的定义，现经由 EDG (Edison Design Group) 前端完全支持 ISO C++。这包括例外、名称空间、模板和*运行时类型信息* (RTTI) 的智能实现，但是导出模板除外。
- 支持某些 GNU 语言扩展。
- 在每个函数基础上的 ARM 和 Thumb 编译。

- 重新设计的内联汇编程序和新的嵌入式汇编程序使您可以包含外联的汇编代码。
- 链接器反馈以删除未使用的功能。
- 添加了对 ARM 体系结构第 6 版指令的完全支持。
- 压缩读/写数据可以优化 ROM 的大小。
- 删除未使用的 C++ 虚函数。
- 多文件编译，可在多个文件之间实现优化。
- 您可以指定一个库搜索路径以指示在何处搜索您的用户库。
- 您可以将 RO 代码和数据分割至不同的执行区。
- 现有新的分散加载属性。
- 支持 Unicode 字符和多字节字符。
- 可使用编译器内在函数访问函数的返回地址、当前堆栈指针值和当前程序计数器值。附加内在函数使您能够在 C 或 C++ 代码中插入 `BKPT` 指令。
- 您可以定义未返回的函数，这样编译器便可生成更多有效代码。
- C++ 名称重整方案已更改。
- 不随 RVCT 一起提供 ARM Profiler (`armprof`)。

  注意

  这与随 RVDS v3.1 Professional 版及更高版本提供的 ARM Profiler 不同。

- 不随 RVCT 一起提供 ARM 应用程序库。
- 不同于 ADS 编译器， RVCT 编译器不生成浏览器信息。
- 汇编程序、编译器和链接器命令行选项均有变更。

  支持双短线 -- 指示命令行关键字 （例如， `--cpp`），并且支持单短线 - 指示带或不带参数的命令行单字母选项 （例如 `-S`）。

  注意

  为了向后兼容，仍支持在早期版本 ADS 和 RVCT 中使用的单短线命令行选项。

- 删除了 fromelf 选项 `-ihf`。

注意

编译工具更严格地检查 8 字节堆栈对齐的要求。编译器通过 PRESERVE8 和 REQUIRE8 生成代码。链接器检查要求 8 字节对齐的代码仅调用保持 8 字节对齐的代码。因此，这与您的旧版本的汇编程序代码、目标文件和库有关。您必须检查您现有的汇编文件、目标文件或库是否符合 8 字节对齐要求，并在需要时纠正它们。有关详细信息，请参阅 《RealView 编译工具汇编器指南》和《RealView 编译工具链接器和实用程序指南》。

B.8.4 ARM 模拟器的更改

RealView ARMulator ISS 是 ARM 模拟器的最新版本。它通过 RealView Connection Broker 和 RDI 支持连接。在 RealView Debugger 下通过 RealView Connection Broker 连接到模拟器时，有多种连接。您可以使用 RealView Debugger、 AXD v1.3 和 armsd 连接到 RealView ARMulator ISS 的 RDI 接口。

注意

虽然可以同时安装 RealView Developer Suite v2.2 和 ADS，但在同时使用 RealView ARMulator ISS 和 ADS ARMulator 时务必小心谨慎。有关详细信息，请参阅《RealView Developer Suite v2.2 版本声明》(*RealView Developer Suite v2.2 Release Notes*)。

术语表

随 ARM® *RealView®* *Development Suite* (RVDS) 提供的文档中使用了以下术语：

AAPCS *请参阅* 《ARM 体系结构的过程调用标准》。

ARM 体系结构的 ABI （基本标准） (BSABI)

ARM 体系结构的 ABI 是一个规范集；这些规范有些是开放式的，有些是 ARM 体系结构特有的，它们在基于 ARM 体系结构执行环境的范围内管理二进制代码的内部运算。基本标准规定了代码生成中那些必须标准化以支持内部运算的方面；基本标准是面向 C 和 C++ 编译器、链接器和运行时库的作者和供应商制定的。

适应性时钟 RealView ICE 采用的一种技术， RealView ICE 用这种技术发出时钟信号，然后等到返回时钟后再产生下一个时钟脉冲。该技术使得 RealView ICE 运行控制单元能够适应不同的信号驱动功能和不同的电缆长度。

高级可扩展接口 (AXI)

一种总线协议，它支持单独的地址/控件和数据段、使用字节选通的非线性数据传输、仅发布起始地址的基于脉冲的事务、启用低成本 DMA 的独立读写数据渠道、发布多个未决地址的能力、无序事务完成以及轻松地添加寄存器级以提供定时闭合。

AXI 协议还包括可选的扩展，以覆盖低电源操作的信号发送功能。

AXI 的目标是高性能、高时钟频率的系统设计，并且还包括大量使其非常适合高速亚微米内部连接的功能。

高级微控制器总线体系结构 (AMBA®)

高性能的 32 位和 16 位嵌入式微控制器的芯片上通信标准。

高级高性能总线 (AHB)

一种总线协议，地址/控件与数据段之间有固定的管道。它只支持由 AMBA AXI 协议提供的功能的一个子集。完整的 AMBA AHB 协议规范包括许多在进行主从 IP 开发时不常需要的功能， ARM Limited 建议通常只用该协议的一个子集。我们将该子集定义为 AMBA AHB-Lite 协议。

另请参阅 高级微控制器总线体系结构 *和* AHB-Lite。

AHB

请参阅 高级高性能总线。

AHB 访问端口 (AHB-AP)

CoreSight™ 支持使用 *调试访问端口* (DAP) 中的 *AHB 访问端口* (AHB-AP) 来访问系统总线基础架构。AHB-AP 提供了直接访问系统内存的 AHB 主端口。如果执行了另一种总线协议，则可以使用 AHB 桥来映射事务。例如，您可以使用 AHB-AXI 桥来启用对 AXI 总线矩阵的访问。

CoreSight 还支持使用 *AHB 跟踪宏单元* (HTM) 来支持 AHB 总线跟踪。

另请参阅 高级可扩展接口、 AHB 跟踪宏单元、 CoreSight *和* 调试访问端口。

AHB-AP

请参阅 AHB 访问端口。

AHB-Lite

完整的 AMBA AHB 协议规范的子集。它提供了多数 AMBA AHB 主从设计所需的所有基本功能，尤其是与多层 AMBA 内连一起使用时，它更能够提供这些基本功能。在大多数情况下，使用 AMBA AXI 协议接口可以更有效地实现由完整的 AMBA AHB 接口提供的额外实用工具。

AHB 跟踪宏单元 (HTM)

AHB 跟踪宏单元是一个跟踪源，它可使用户看到使用 ETM 无法从处理器跟踪推断出的总线信息：

- 对多层总线利用的了解。
- 软件调试。例如，内存区域的访问和数据访问的显示。
- 针对跟踪触发器或过滤器以及总线配置进行的总线事件检测。

另请参阅 高级高性能总线。

AMBA

请参阅 高级微控制器总线体系结构。

AMBA 跟踪总线 (ATB)

AMBA 跟踪总线通过 SoC 中的 CoreSight 基础架构传输跟踪数据。跟踪源是 ATB 主，接收器是 ATB 从属。链接组件同时提供了主接口和从属接口。

另请参阅 CoreSight。

APB-AP *请参阅* 调试访问端口。

armar ARM 库管理程序，可用于创建文件 （如目标文件）的库。

另请参阅 RealView 编译工具 (RVCT)。

armasm ARM 汇编器。

另请参阅 RealView 编译工具 (RVCT)。

armcc C 和 C++ 代码的 ARM 编译器。

另请参阅 RealView 编译工具 (RVCT)。

armlink ARM 链接器。

另请参阅 RealView 编译工具 (RVCT)。

ARM 的高级 SIMD 扩展

ARM 的高级 SIMD 扩展是 ARMv7 体系结构的一个可选组件。NEON 是面向高级媒体和信号处理应用程序以及嵌入式处理器的 64/128 位混合 SIMD 技术。它是作为 ARM 内核的一部分实现的，但有自己的执行管道和寄存器组，该寄存器组不同于 ARM 内核寄存器组。

ARM 高级 SIMD 扩展支持整数、定点和单精度浮点 SIMD 运算。这些指令在 ARM 和 Thumb®-2 中都可用。

ARM 高级 SIMD 扩展又称为 *ARM NEON 技术* (NEON™)。

ARM 指令 为 ARM 状态下运行的 ARM 处理器编码运算的字。ARM 指令必须字对齐。

另请参阅 Thumb 指令、 Thumb-2 指令 *和* Thumb-2EE 指令。

ARM Profiler ARM Workbench IDE 的一个插件，可在以最高 250 MHz 工作频率运行的目标上对嵌入式软件进行长时间的非侵入型分析。目标可以为 *实时系统模型* (RTSM) 和硬件目标。ARM Profiler 随 RVDS Professional 版提供，也可作为独立的产品提供。

该插件与旧版本 ARM Profiler 工具 `armprof` 不同。

另请参阅 实时系统模型 (RTSM) *和* RealView Development Suite Professional 版。

ARM 状态

执行 ARM 指令的处理器即是在 ARM 状态下运行。处理器可在 BX 或 BLX 等状态更改指令的引导下切换到 Thumb 状态 （以识别 Thumb 指令）。

另请参阅 Jazelle® 状态、 Thumb 状态*和* ThumbEE 状态。

ARM TrustZone 技术

将安全功能集成到整个 SoC 设备中的硬件和软件。

ARM Workbench IDE

ARM Workbench IDE 以 Eclipse IDE 为基础，另外提供了一些功能以支持在 RVDS 中提供的 ARM 开发工具。

另请参阅 RealView Development Suite (RVDS)。

ATB

请参阅 AMBA 跟踪总线。

AXI

请参阅 高级可扩展接口。

BCD 文件

请参阅 板/芯片定义 (BCD) 文件。

大端

在 ARM 体系结构的上下文中，大端被定义为字的最低效字节位于较高地址、最高效字节位于较低地址的内存组织。

另请参阅 小端。

板文件

RealView Debugger 使用此术语来表示顶级配置文件 （通常称为 rvdebug.brd），该文件引用一个或多个其他配置文件。板文件包含：

- 调试配置 （连接级）设置；
- 指向调试接口配置文件的引用，该文件在开发平台上标识目标；
- 指向分配给调试配置的任何*板/芯片定义* (BCD) 文件的引用。

另请参阅 板/芯片定义 (BCD) 文件、调试配置、调试接口、开发平台*和* 目标。

板/芯片定义 (BCD) 文件

在 RealView Debugger 的上下文中， BCD 文件使您能够为目标开发板或处理器定义内存映射和内存映射寄存器。随 RVDS 为 ARM 开发板和处理器内核模块提供了各种 BCD 文件，例如为 Integrator®/CP 开发板提供的 CP.bcd 和为 ARM940T™ 处理器提供的 CM940T.bcd。

另请参阅 板文件*和* 调试配置。

断点单元 在 RealView Debugger 上下文中，指连锁的断点内的单元；它与其他断点单元相结合创建复杂的硬件断点。

*另请参阅*连锁的断点*和*硬件断点。

BSABI *请参阅* ARM 体系结构的 ABI （基本标准）。

规范帧地址 (CFA) 在 DWARF 中，这是堆栈上的一个地址，它指定被中断函数的调用帧的位置。

俘获线程 俘获线程是指所有可置于 RVDS 控制之下的线程。称为非俘获线程的特殊线程对*运行系统调试* (RSD) 的操作的至关重要，因此它们不受调试器的控制。在 GUI 中，非俘获线程显示为灰色。

*另请参阅*运行系统调试。

CFA *请参阅*规范帧地址。

连锁的断点 在 RealView Debugger 的上下文中，指由多个硬件断点单元构成的复杂断点。

*另请参阅*断点单元、条件断点、数据断点*和*硬件断点。

连锁的跟踪点 在 RealView Debugger 的上下文中，指由多个跟踪点单元构成的复杂跟踪点。

*另请参阅*跟踪点*和*跟踪点单元。

条件断点 分配了一个或多个条件限定符的断点。在满足所有已分配的条件时该断点将被激活，并将停止执行或继续执行 （取决于已分配的操作限定符）。条件通常会引用断点位置作用域内的程序变量值。

*另请参阅*连锁的断点、数据断点、硬件断点、指令断点、软件断点*和*无条件断点。

内核模块 ARM Integrator 开发板上下文中，指包含基于 ARM 体系结构的处理器和本地存储器的附加开发板。内核模块可以独立运行，也可以与 Integrator 开发板实现堆栈。

另请参阅 Integrator。

CoreSight CoreSight 是一种基础架构，它可对完整的*芯片上系统* (SoC) 设计的性能进行调试、监视和优化。

另请参阅 CoreSight ECT、 CoreSight ETB、 CoreSight ETM、跟踪通道*和*跟踪端口接口单元。

CoreSight ECT CoreSight ECT 是一个控制和访问组件，支持 SoC 内的多个触发事件的交互和同步：

另请参阅 CoreSight、交叉触发接口、交叉触发矩阵*和*嵌入式交叉触发。

CoreSight ETB CoreSight ETB 是一个跟踪接收器，它可使用可配置大小的 RAM 为跟踪数据提供芯片上存储。

另请参阅 CoreSight、 CoreSight ETB、嵌入式跟踪缓冲区 *和* 嵌入式跟踪宏单元。

CoreSight ETM CoreSight ETM 是一个跟踪源，它可通过符合 ATB 标准的跟踪端口提供处理器驱动的跟踪。

另请参阅 AMBA 跟踪总线、 CoreSight、 CoreSight ETB *和* 嵌入式跟踪宏单元。

CPSR *请参阅* 当前程序状态寄存器。

交叉触发接口 (CTI) 交叉触发接口提供组件或子系统与交叉触发矩阵之间的接口。系统要求每个子系统的 CTI 都支持交叉触发。

另请参阅 CoreSight、 CoreSight ECT、交叉触发矩阵 *和* 嵌入式交叉触发。

交叉触发矩阵 (CTM) 交叉触发矩阵将从 CTI 生成的触发请求结合在一起，并将它们作为通道触发器广播到所有 CTI。这使得各子系统之间能够相互交互、相互交叉触发。CTM 可以连接在一起来增加 CTI 的数量。

另请参阅 CoreSight、 CoreSight ECT、交叉触发接口 *和* 嵌入式交叉触发。

CTI *请参阅* 交叉触发接口。

CTM *请参阅* 交叉触发矩阵。

当前程序状态寄存器 (CPSR) 包含控制位和标记的当前状态的寄存器。

另请参阅 程序状态寄存器 *和* 保存的程序状态寄存器。

DAP *请参阅* 调试访问端口。

数据断点 一种硬件断点，它可在以特定的方式访问给定的位置时激活。根据需要，该断点还可以查找正在给定位置访问的特定数据值。

另请参阅 连锁的断点、条件断点、硬件断点、指令断点、软件断点 *和* 无条件断点。

DCC *请参阅* 调试通信通道。

调试代理 (DA) 调试代理驻留在目标上，为 RealView Debugger 中的 *运行系统调试* (RSD) 提供目标端支持。调试代理可以是线程，也可以构建于 RTOS 中。调试代理和 RealView Debugger 可使用 *调试通信通道* (DCC) 相互通信。这样便可以使用 ICE 接口在调试器和目标之间传递数据，无需停止程序或进入调试状态。

另请参阅 运行系统调试 *和* 调试通信通道。

调试访问端口 (DAP) 调试访问端口是一个控制和访问组件，它允许通过系统主端口访问整个 SoC。

对内部接口的外部读/写访问由 *JTAG 调试端口* (JTAG-DP) 提供。JTAG-DP 是进行调试访问的标准 JTAG 接口，它通过 DAP 提供对 SoC 的标准 JTAG 访问。它与 DAP 内部总线接合。

对芯片上总线和其他接口的内部访问由 *访问端口* (AP) 提供。三个 AP 是：

- *AHB 访问端口* (AHB-AP)，它提供 ABH-Lite 主，用于访问系统 AHB 总线；
- *APB 访问端口* (APB-AP)，它提供 AMBA 3 APB 主，用于访问配置所有 CoreSight 组件的调试 APB；
- *JTAG 访问端口* (JTAG-AP)，提供对芯片组件的 JTAG 访问，并可用作 JTAG 主端口驱动整个 SoC 中的 JTAG 链。

另请参阅 CoreSight。

Debug Communications Channel (DCC)
调试通信通道可使数据通过 JTAG 接口在 RealView Debugger 与目标上的 EmbeddedICE 逻辑之间传递，无需停止程序流或进入调试状态。

调试配置 在 RealView Debugger 上下文中，调试配置为开发平台定义一个通过特定调试接口访问的调试环境。可以为一个调试接口创建多个调试配置，每个配置为不同的开发平台分别提供不同的调试环境，或是为同一开发平台提供不同的调试环境。

所有调试配置都存储在主 RealView Debugger 板文件中。每个配置可以引用一个或多个 BCD 文件。

另请参阅 板文件、板/芯片定义 (BCD) 文件、调试接口 *和* 目标。

调试效应 软件开发人员所感受的由调试器创建的体验。调试效应的重要功能包括：

- 混合源代码和反汇编代码
- 一个函数调用堆栈，显示了符号函数原型以及名称和参数类型
- 使用变量的源代码名称显示变量
- 源代码级步进和调试

调试器使用正在调试的系统中的数据以及代码生成工具链中的符号调试信息创建此效应。

调试接口

在 RealView Debugger 上下文中，调试接口在您的开发平台上标识目标，并提供让 RealView Debugger 与这些目标通信的机制。调试接口直接对应于一个硬件或软件模拟器。

另请参阅 调试配置 *和* 目标。

开发平台

包含用于开发应用程序的组件，可以是硬件，也可以是模拟软件。它可能包括：

- 开发板，如 Integrator/CP
- 外围设备
- 一个或多个基于 ARM 体系结构的处理器
- CoreSight 组件
- 一个或多个 DSP

另请参阅 CoreSight *和* 目标。

双字

在 ARM 体系结构的上下文中，指 64 位单位信息。除非另外声明，否则其内容将被视为无符号整数。

ECT

请参阅 嵌入式交叉触发。

嵌入式汇编器

嵌入式汇编器是包含在 C 或 C++ 文件中的汇编器代码，且独立于其他 C 或 C++ 函数。

嵌入式交叉触发 (ECT)

嵌入式交叉触发提供了围绕 SoC 传递调试或配置报告事件的标准内连机制。它由以下两部分构成：

- *交叉触发接口* (CTI)
- *交叉触发矩阵* (CTM)

另请参阅 CoreSight *和* CoreSight ECT。

嵌入式跟踪缓冲区™ (ETB™)

嵌入式跟踪缓冲区提供了内核内部的逻辑，它可扩展嵌入式跟踪宏单元的信息捕获功能。

另请参阅 CoreSight ETB *和* 嵌入式跟踪宏单元。

嵌入式跟踪宏单元™ (ETM)

嵌入硬件中的逻辑块，已连接至处理器的地址、数据和状态信号。它通过跟踪端口，以压缩协议广播分支地址、数据和状态信息。它包含用于触发和筛选跟踪输出的资源。

另请参阅 CoreSight ETM *和* 嵌入式跟踪缓冲区。

EmbeddedICE® 逻辑

EmbeddedICE 逻辑是芯片上的一个逻辑块，它为基于 ARM 体系结构的处理器提供基于 TAP 的调试支持。可以使用 JTAG 接口并通过基于 ARM 体系结构的处理器上的 TAP 控制器来对其进行访问。

另请参阅 IEEE1149.1。

仿真器 在目标连接硬件的上下文中，仿真器提供了与实际内核的引脚间的接口 （将引脚仿真至外部领域），并使您可以控制或操作这些引脚上的信号。

ETB *请参阅* 嵌入式跟踪缓冲区。

ETM *请参阅* 嵌入式跟踪宏单元。

ETV *请参阅* 扩展的目标可见度。

执行载体 调试目标接口的一部分，处理从客户机工具到目标的请求。

另请参阅 调试接口。

执行视图 在映像载入存储器并开始执行后的区和节的地址。

扩展的目标可见度 (ETV)

扩展的目标可见度使 RealView Debugger 可以访问基础目标的功能，例如由硬件制造商或 SoC 设计者提供的芯片级信息。

筛选 在 RealView Debugger Trace 上下文中，指一种使您进一步限定在 RealView Debugger 中执行的跟踪捕获的结果的工具。如果您要进一步限定显示内容中关注的方面，此功能将很有帮助。

FIQ 快速中断。

fromelf ARM 映像转换实用程序。此实用程序接受 ELF 格式的输入文件，并将其转换成多种输出格式。`fromelf` 还可以生成关于输入映像的文本信息，如代码和数据大小。

另请参阅 RealView 编译工具 (RVCT)。

GCC GNU 编译器集合。

GDB GNU 调试器。

半字 在 ARM 体系结构上下文中，半字被定义为 16 位信息单元。除非另外声明，否则其内容将被视为无符号整数。

暂停系统调试 (HSD) *暂停系统调试* (HSD) 意味着只能调试没有运行的目标，通常用于 OS 感知调试。必须停止所有的目标，然后才能执行系统分析。目标停止后， RealView Debugger 会通过读取和解释目标内存来展示 OS 感知信息。

另请参阅 运行系统调试 (RSD)。

硬件断点 使用非侵入型附加硬件实现的断点。位置在*只读内存* (ROM) 或闪存中时，硬件断点是停止执行的唯一方法。使用硬件断点通常会导致处理器完全停止运行。对于实时系统而言，这种方法是不可取的。

另请参阅 连锁的断点、条件断点、数据断点、指令断点、软件断点*和*无条件断点。

暗示指令 暗示指令提供硬件可以利用的硬件的信息。实现可以选择是否实现暗示指令。如果不实现，则它们的执行方式将与 `NOP` 类似。

HSD *请参阅* 暂停系统调试。

HTM *请参阅* AHB 跟踪宏单元。

ICE 扩展单元 EmbeddedICE 逻辑的硬件扩展，它提供了更多断点单元。

IEEE 1149.1 用于定义 TAP 的 IEEE 标准。通常称为 JTAG，但并不恰当。

即时值 直接在指令中编码并且在执行指令时作为数字数据使用的值。许多 ARM 和 Thumb 指令允许将小的数值在对它们执行操作的指令中编码为即时值。

已定义的实现 在 ARM 体系结构的上下文中，这意味着不会以体系结构方式定义实现，而必须按各个实现来定义和记录。

线路中仿真器 在电路运行时允许访问电路信号并对其进行修改的设备。

输入节 输入节包含代码或初始化数据，或者说明应用程序启动前必须设为 0 的内存空间。

指令断点 映像中的一个位置，该位置包含一个指令，该指令在执行时会激活一个断点。通过分配条件限定符可延迟断点激活，对映像的后续执行将由分配给断点的操作确定。

另请参阅 条件断点、数据断点、硬件断点、软件断点*和*无条件断点。

指令寄存器 (IR) 当谈到 TAP 控制器时，指一个控制 TAP 运行的寄存器。

指令集系统模型 (ISSM)

在 RVDS 上下文中，指一组用于模拟 ARM Cortex™ 系列处理器的模型。这些模型随 RVDS 提供。

*另请参阅*实时系统模型 (RTSM)、 RealView ARMulator ISS、 Simulator *和* SoC Designer Simulator。

Integrator

一系列 ARM 硬件开发平台。提供的内核模块包含处理器和本地存储器。

*另请参阅*内核模块。

交互操作

允许在 ARM 和 Thumb 代码之间进行跳转的一种工作方法。

IRQ

中断请求。

ISSM

*请参阅*指令集系统模型 (ISSM)。

IT 块

位于 16 位 Thumb-2 *If-Then* (IT) 指令之后、最多由四个指令组成的指令块。该块中的每条指令都是有条件的。这些指令的条件可以都相同，也可以是其中一些互为反面情况。

Jazelle

Jazelle 体系结构可扩展现有的 ARM 体系结构，以便直接执行所选的 *Java 虚拟机* (JVM) 操作代码指令。

Jazelle 状态

执行 Jazelle 字节代码指令的处理器即在 Jazelle 状态下运行。

另请参阅 ARM 状态、 Thumb 状态 *和* ThumbEE 状态。

JTAG-AP

*请参阅*调试访问端口。

JTAG-DP

*请参阅*调试访问端口。

JTAG 接口单元

在 ARM RealView 工具上下文中，指一种协议转换器，可以将来自 RVDS 调试器的低级命令转换为发送给处理器的 JTAG 信号 （例如转换为 EmbeddedICE 逻辑和 ETM）。

另请参阅 RealView Debugger *和* Workbench Debugger。

小端

在 ARM 体系结构上下文中，小端指一种内存组织方式，在这种方式中字的最高有效字节位于较高地址、最低有效字节位于较低地址。

*另请参阅*大端。

载入视图

在映像载入存储器但还未开始执行时的区和节的地址。

内存提示

在 ARM 体系结构上下文中，内存提示指令可使编程人员为内存系统提供有关未来内存访问的高级信息，但不会实际加载或存储任何数据。

MPCore 一种作为传统的单处理器内核提供的集成式*对称多处理器系统* (SMP)。该芯片包含最多四个基于 ARM1136J-S™ 的具有高速缓存一致性的 CPU。

MPU 多处理器单元。

NEON *请参阅* ARM 高级 SIMD 扩展。

正常领域和安全领域 实际上是单个物理处理器上的两个虚拟处理器。正常领域处理不注重安全性的操作，并将注重安全性的操作委托给安全领域。客户端应用程序驻留在正常领域中，并在其中执行。本机服务驻留在安全领域中，并在其中执行。TrustZone 软件的安全部分在安全领域中运行。

另请参阅 安全监控。

正常领域 *请参阅* 正常领域和安全领域。

输出节 具有相同 RO、RW 或 ZI 属性的连续输入节序列。各节组合在一起形成一大片称为区的空间。多个区组合在一起，成为最终的可执行映像。

另请参阅 区。

OS 感知 OS 感知是由让您执行以下操作的 RealView Debugger 提供的功能：

- 调试在嵌入式 OS 开发平台 （如*实时操作系统* (RTOS)）上运行的应用程序。
- 提供线程信息，并将某些调试操作限定到特定的线程范围。

PCH *请参阅* 预编译头文件。

预编译头文件 (PCH) 预编译的头文件。这可避免每次在头文件被源文件包含时编译器都对其进行编译。

ARM 体系结构的过程调用标准 (AAPCS)

ARM 体系结构的过程调用标准 定义在子程序调用过程中如何使用寄存器和堆栈。

性能分析 在 RealView Debugger Trace 的上下文中，这是在执行被调试的程序期间累加的统计资料，用于测量性能或确定关键的代码区域。

程序计数器 (PC) 在 ARM 体系结构上下文中，指整数寄存器 R15。

程序状态寄存器 (PSR)

包含一些有关当前程序和当前处理器的信息。也被称为 “当前处理器状态寄存器 (CPSR)”，以着重强调它与 “保存的处理器状态寄存器 (SPSR)” 之间的区别。SPSR 保留了调用当前函数时 PSR 所具有的值，并在返回控制时恢复该值。

增强程序状态寄存器 (EPSR) 包含一些基于 ARM 的处理器 （ARM9E）所使用的附加位 （Q 位，符号化饱和）。

另请参阅 当前的程序状态寄存器 *和* 保存的程序状态寄存器。

项目模板 一个针对特定目标开发平台的 RealView Debugger 和 RVCT 配置文件的集合。使用这些模板可以在 ARM Workbench IDE 中创建目标特定的开发项目。

另请参阅 RealView 编译工具 (RVCT)、 RealView Debugger、 ARM Workbench IDE *和* Workbench Debugger。

PSR *请参阅* 程序状态寄存器。

PU 保护单元。

只读位置无关 (ROPI) 在 ARM 体系结构的上下文中，这是指可以放在任意地址的代码或只读数据。

读写位置无关 (RWPI) 在 ARM 体系结构的上下文中，这是指可以在运行时更改的读/写数据地址。

RealMonitor 一个小程序，当集成到目标应用程序或 *实时操作系统* (RTOS) 中时，它可让您在应用程序的某些部分继续运行的同时观察和调试您的目标。

实时系统模型 (RTSM)

RTSM 模型包含一个硬编码的系统，该系统具有一个或多个特定模拟处理器和一个仿真基板。有些 RTSM 随 RVDS Professional 版一起提供。

另请参阅 指令集系统模型 (ISSM)、RealView ARMulator ISS (RVISS)、RealView Development Suite Professional 版、 ARM Profiler、模拟器 *和* SoC Designer Simulator。

RealView ARMulator® ISS (RVISS)

随 RVDS 提供的一种 ARM 模拟器。

RVISS 收集了模拟各种 ARM 处理器的指令集和体系结构的程序。可提供指令精确的模拟功能，并可让 ARM 和 Thumb 可执行程序在非本机硬件上运行。

RVISS 提供以下模型的模块：

- ARM 处理器内核
- 处理器所用的内存

每个部件都有备选的预定模型。但是，如果所提供的模型不能满足您的要求，您可以创建自己的模型。

另请参阅 指令集系统模型 (ISSM)、实时系统模型 (RTSM)、模拟器 *和* SoC Designer Simulator。

RealView 编译工具 (RVCT)

RVCT 是一套工具以及支持文档和示例，可用于编写和生成适用于 ARM 系列处理器的应用程序。

另请参阅 armar、 armasm、 armcc、 armlink *和* fromelf。

RealView Debugger ARM Limited 提供的最新调试器软件，它使您能够利用调试代理来检查和控制软件在调试目标上的执行。RealView Debugger 同时以 Windows 和 Red Hat Linux 版本提供。

RealView Debugger Trace

RVDS 产品的一部分，通过新增的实时程序和数据跟踪扩展了调试功能。在“RealView Debugger Code/RealView Debugger 代码”窗口中可以访问此功能。

另请参阅 RealView ICE *以及* RealView Trace 和 RealView Trace 2。

RealView Development Suite (RVDS)

一套附带支持文档和示例的软件开发应用程序，可用于编写和调试适用于 ARM 系列处理器的应用程序。在 Professional 和 Standard 版中都可获得 RVDS v3.1 及更高版本。RVDS 取代了 ARM Developer Suite™。

另请参阅 实时系统模型 (RTSM)、 RealView Development Suite Professional 版*和* RealView Development Suite Standard 版。

RealView Development Suite Professional 版

RVDS Professional 版包含：

- RVDS Standard 版中的所有工具
- ARM Profiler 及其使用许可证
- RTSM
- 一个用于调试最新 ARM 处理器的附加许可证

另请参阅 实时系统模型 (RTSM)、 RealView Development Suite Standard 版*和* ARM Profiler。

RealView Development Suite Standard 版

RVDS Standard 版包含：

- ARM Workbench IDE
- RealView 编译工具
- RealView Debugger
- RealView ICE 和 RealView Trace 主机软件
- 对 ISSM、 RVISS、 RTSM 和 SoC Designer 模拟器的支持

- ISSM 和 RVISS 模型

另请参阅 ARM Workbench IDE、指令集系统模型 (ISSM)、实时系统模型 (RTSM)、 RealView ARMulator ISS、 RealView 编译工具、 RealView Debugger、 RealView Development Suite Professional 版、 RealView ICE、 RealView Trace 和 RealView Trace 2 *以及* SoC Designer Simulator。

RealView ICE 一种基于 JTAG 的调试解决方案，用于调试运行在基于 ARM 体系结构的处理器上的软件。RealView ICE 主机软件随 RVDS 提供。RealView ICE 运行控制单元只能作为独立的产品购买。

另请参阅 RealView Debugger Trace *以及* RealView Trace 和 RealView Trace 2。

RealView Trace 和 RealView Trace 2
与 RealView ICE 结合使用，为那些在具有深层嵌入式处理器内核的前沿芯片上系统设备中运行的软件提供实时跟踪功能。RealView Trace 2 还可使数据直接流向 ARM Profiler，以执行实时硬件平台性能分析。RealView Trace 和 RealView Trace 2 硬件单元只能作为独立的产品购买。

另请参阅 RealView Debugger Trace、 ARM Profiler *和* RealView ICE。

区 在映像中，区是指一至三个输出节（RO、 RW 和 ZI）的相邻序列。区通常映射到物理存储器设备，如 ROM、 RAM 或外围设备。

另请参阅 根区。

ROPI *请参阅* 只读位置无关。

根区 映像中，区的载入地址和执行地址相同。非根区是指必须从其加载地址复制到其执行地址的区。

RSD *请参阅* 运行系统调试。

RTSM *请参阅* 实时系统模型 (RTSM)。

运行系统调试 (RSD) *运行系统调试* (RSD) 用于 OS 感知调试，意味着可对正在运行的目标进行调试。这意味着在执行任何系统分析之前不必停止调试目标。RSD 可以访问使用驻留在目标上的 *调试代理* (DA) 的应用程序。调试代理是与系统中的其他任务一起计划的。

另请参阅 调试代理 *和* 暂停系统调试 (HSD)。

RVCT *请参阅* RealView 编译工具 (RVCT)。

RVDS *请参阅* RealView Development Suite (RVDS)。

RWPI *请参阅* 读写位置无关。

保存的程序状态寄存器 (SPSR)
一种寄存器，保存最近发生异常之前，当前程序状态寄存器中最新信息的副本。每个异常模式都有其自身的 SPSR。

分散加载
单独分配地址和对代码和数据段进行分组，而不使用单个大块。

节
在面向基于 ARM 体系结构的处理器的应用程序的上下文中，指映像的一个软件代码块或数据块。

另请参阅 输入节 *和* 输出节。

安全监控
使 ARM 处理器在正常领域与安全领域执行环境之间进行可靠切换。安全监控对 TrustZone 软件开发人员是透明的。

安全领域
请参阅 正常领域和安全领域。

半主机
一种通信机制，目标使用这种机制将应用程序代码发出的 I/O 请求传送给主机系统，而不是尝试自己支持 I/O。

简单跟踪点
一种跟踪点，可使您设置触发点、跟踪起点和终点，以及存储器和数据访问的跟踪范围。

另请参阅 跟踪点。

模拟器
在 ARM 工具上下文中，模拟器执行软件中的非本机指令 （模拟内核）。

另请参阅 指令集系统模型 (ISSM)、实时系统模型 (RTSM)、 RealView ARMulator ISS *和* SoC Designer Simulator。

SoC Designer Simulator
SoC Designer Simulator 是 RealView SoC Designer 工具集的一部分，可用于快速执行对复杂*芯片上系统* (SoC) 设计的建模、模拟和调试。将 SoC Designer Simulator 与 RealView Debugger 结合使用可调试用 RealView SoC Designer 工具创建的系统和处理器模型。

另请参阅 指令集系统模型 (ISSM)、实时系统模型 (RTSM)、 RealView ARMulator ISS *和* 模拟器。

软件断点
通过将内存中的指令替换为导致处理器执行异常操作的指令而实现的一种断点。如果指令存储在只读内存中，则不能使用软件断点，因为这会改变指令内存。使用软件断点可以在断点期间继续执行中断处理，因此更适于在实时系统中使用。

另请参阅 连锁的断点、条件断点、数据断点、硬件断点、指令断点、软件断点 *和* 无条件断点。

SPSR 保存的程序状态寄存器。

*另请参阅*程序状态寄存器。

堆栈指针 (SP) 整数寄存器 R13。

超级用户调用 (SVC) 导致处理器调用程序员指定子程序的指令。ARM 标准 C 库使用它来处理半主机。该指令将取代*软件中断* (SWI)。

SVC *请参阅*超级用户调用。

SWI *请参阅*超级用户调用。

TAP 控制器 设备上的逻辑，使您可以访问部分或整个设备以便进行测试。IEEE1149.1 中定义了电路的功能。

*另请参阅*测试访问端口*和* IEEE1149.1。

目标 在 RealView Debugger 上下文中，目标是 RealView Debugger 可以连接到的开发平台部分，在该部分平台上可以执行调试操作。目标可以是：

- 可运行的目标，如基于 ARM 体系结构的处理器或 DSP。当连接到可运行的目标时，您可以执行针对该目标的执行相关调试操作，例如分步和跟踪。
- 非可运行 CoreSight 组件。CoreSight 组件为进行实时调试和跟踪提供了系统范围的解决方案。

另请参阅 CoreSight、调试配置*和*调试接口。

目标载体 目标载体为 RVDS 提供一个用于各种不同目标的标准接口，以便调试器可以轻松连接至新的目标类型，而不必对调试器核心软件进行更改。该接口可以为硬件或软件接口。

*另请参阅*指令集系统模型 (ISSM)、实时系统模型 (RTSM)、 RealView ARMulator ISS、 RealView ICE *和* SoC Designer Simulator。

TCM 紧偶合存储器。

TDI 测试数据输入。

TDO 测试数据输出。

Thumb 指令 为以 Thumb 状态运行的、基于 ARM 体系结构的处理器的操作进行编码的一个半字或两个半字。Thumb 指令必须为半字对齐。

另请参阅 ARM 指令、 Thumb-2 指令*和* Thumb-2EE 指令。

Thumb 状态
执行 Thumb 指令的处理器工作在 Thumb 状态。当直接通过 BX、 BLX 等指令完成时，处理器可切换到 ARM 状态 （以识别 ARM 指令）。

另请参阅 ARM 状态、 Jazelle 状态 *和* ThumbEE 状态。

Thumb-2 指令
Thumb-2 是 Thumb 指令集的一项主要增强功能，并且由 ARMv6T2 和 ARMv7M 体系结构定义。Thumb-2 提供了几乎与 ARM 指令集完全一样的功能。它兼有 16 位和 32 位指令，并可检索与 ARM 类似的性能，但其代码密度与 Thumb 代码类似。

另请参阅 ARM 指令、 Thumb 指令 *和* Thumb-2EE 指令。

Thumb-2EE 指令
Thumb-2 执行环境 (Thumb-2EE) 由 ARMv7 体系结构定义。Thumb-2EE 指令集基于 Thumb-2，前者进行了一些更改和添加，使得动态生成的代码具有更好的目标，也就是说，就在执行之前或在执行过程中即可在该设备上编译代码。

另请参阅 ARM 指令、 Thumb 指令 *和* Thumb-2 指令。

ThumbEE 状态
执行 Thumb-2EE 指令的处理器正在以 ThumbEE 状态运行。在此状态下，该指令集几乎与 Thumb 指令集相同。不过，有些指令已经修改了行为，有些原有的指令已不再提供，另外还新添了一些指令。

另请参阅 ARM 状态、 Jazelle 状态 *和* Thumb 状态。

TPA
跟踪端口分析器。

TPIU
请参阅 跟踪端口接口单元。

跟踪通道
跟踪通道在单个通道中结合了多达 8 个跟踪源 （ETM 或 HTM）。但是，在此发行版本中，一次只能从一个 ETM 捕获跟踪数据。

另请参阅 AHB 跟踪宏单元、 CoreSight、 CoreSight ETM *和* 嵌入式跟踪宏单元。

跟踪端口接口单元 (TPIU)
跟踪端口接口单元是一个跟踪接收器，它可将芯片外的跟踪数据完全放入一个 TPA （例如 RealView Trace）中。

另请参阅 CoreSight、 CoreSight ETB、 CoreSight ETM *和* RealView Trace。

跟踪点
可以对一行源代码、一行汇编代码或一个内存地址设置跟踪点。在 RealView Debugger 中，您可以设置多种跟踪点，以便准确确定要跟踪的程序信息。

另请参阅 连锁跟踪点 *和* 跟踪点单元。

跟踪点单元
在 RealView Debugger 的上下文中，这是连锁跟踪点内的一个单元，它可将其他跟踪点结合起来创建一个复杂的跟踪点。

另请参阅 连锁跟踪点 *和* 跟踪点。

触发
在断点的上下文中，触发是一种操作，用于通知目标已到达断点且符合任何相关条件。

涉及到跟踪时，触发是一种事件，可以指示调试器在不停止处理器运行的情况下停止收集跟踪数据并显示触发位置周围的跟踪信息。实际显示的信息取决于触发操作在缓冲区内所处的位置。

TrustZone 软件
一种安全软件框架，可充分利用内置于 ARM 体系结构中的安全扩展。用在可作为两个虚拟 CPU 的单处理器 ARM 内核中。

无条件断点
未分配条件限定符的断点。该断点轻轻一点即可激活，但后续的映像执行由分配给该断点的任何操作确定。

另请参阅 条件断点、数据断点、硬件断点、指令断点、软件断点 *和* 无条件断点。

未定义
在 ARM 体系结构的上下文中，尝试执行未定义的指令会导致未定义的指令异常。

不可预测的
在 ARM 体系结构的上下文中，不可预测的指令的结果不能依赖。不可预测的指令或结果不应表示安全漏洞。不可预测的指令不应暂停或挂起处理器，或系统的任何部分。

胶合代码
在 ARM 体系结构的上下文中，这是指一个小代码块，在当前的处理器状态下更改处理器状态或跳转到一个无法达到的地址时与子程序调用配合使用。

VFP
浮点协处理器的一种标准，可由单条指令处理多个数据值。

观察
在 RealView Debugger 中，一个观察就是一个您要求调试器在每一步或每一个断点显示的变量或表达式，目的是查看该变量或表达式的值的变化情况。“Watch/观察”窗格是 “RealView Debugger Code/RealView Debugger 代码”窗口的一部分。它显示已定义的观察点。

观察点
在 RVDS 中，指一个硬件断点。

字
在 ARM 体系结构上下文中，一个字具有一个由四个连续字节表示的值。32 位的信息单元。除非另外声明，否则其内容将被视为无符号整数。