

RVCT 内部版本 R471~R902的变更说明

aozime 整理
www.rt-thread.org

RVCT 4.0: 针对 Windows 的内部版本 902 补丁

说明

RVCT 4.0 版 Web 补丁 9 (内部版本 902)适用于 ARM RVDS 4.0 版产品。它可用于更新 RVDS 4.0 标准版、专业版或评估版安装,而不论其以前安装过补丁还是原始安装。它与其他版本的 RVCT/RVDS 不兼容。

此补丁包含更新的 RVCT 4.0 编译器、链接器、汇编器、fromelf 和 armar 程序可执行文件、include 文件以及 C/C++ 库。

操作系统平台

此补丁在以下支持的平台上进行了测试:

- Windows 7
- Windows XP SP2, 32 位和 64 位
- Windows XP SP3, 32 位
- Windows Vista Business Edition SP1, 32 位和 64 位
- Windows Vista Enterprise Edition SP1, 32 位和 64 位
- Windows Server 2003, 32 位和 64 位
- Red Hat Linux Enterprise 4 for x86, 32 位和 64 位
- Red Hat Linux Enterprise 5 for x86, 32 位和 64 位
- Solaris for SPARC 10

此外,此补丁还在以下不支持的平台上进行了测试:

- Windows 2000 SP4
- SUSE Linux 9.2
- Ubuntu Linux 8.10

安装说明

要安装补丁,请执行以下步骤:

确保当前正在使用 RVCT 4.0。为此,请键入:

```
armcc --vsn
```

并确保它返回 RVCT 4.0 (内部版本 400)或更高版本。

将 ZIP 文件中的所有文件解压缩到一个临时目录中。

复制临时目录中的 include 目录以替换现有 include 目录。可通过 RVCT40INC 环境变量定位现有目录,例如:

```
C:\> set RVCT40INC
```

RVCT40INC=C:\Program Files\ARM\RVCT\Data\4.0\400\include\

复制临时目录中的 lib 目录以替换现有 lib 目录。可通过 RVCT40LIB 环境变量定位现有目录，例如：

```
C:\> set RVCT40LIB
```

RVCT40LIB=C:\Program Files\ARM\RVCT\Data\4.0\400\lib\

复制临时目录中的 win_32-pentium 目录以替换现有 win_32-pentium 目录。可通过 RVCT40BIN 环境变量定位现有目录，例如：

```
C:\> set RVCT40BIN
```

RVCT40BIN=C:\Program Files\ARM\RVCT\Programs\4.0\400\win_32-pentium\

- 1 现在更新后的工具应已安装完毕。要确认这一点，请打开一个 DOS 命令提示符窗口并输入：

```
armcc --vsn
```

```
armlink --vsn
```

```
armasm --vsn
```

```
fromelf --vsn
```

```
armar --vsn
```

- 2 返回的版本应当是 RVCT 4.0（内部版本 902）。

自 RVCT 4.0（内部版本 870）以来的增强功能

- 已添加了一个新选项 `--[no_]intermediate_object_files`。在未指定 `-c` 的合并的编译并链接步骤中，如果指定了 `--no_intermediate_object_files`，则编译器将使用临时文件用于已编译的对象文件，并且在链接后删除这些临时文件。对象文件将不会正常在当前目录中创建。常规 `armcc` 调用的默认值为 `--intermediate_object_files`（保留之前的行为），而在使用 GCC 命令行转换时，默认值为 `--no_intermediate_object_files`。[SDCOMP-10711]

- `symdefs` 文件允许在一个链接步骤的输出中使用的全局符号用作另一个链接步骤的输入。在 `symdefs` 文件包含作为 `comdat` 组的成员的节所定义的全局符号，并且 `symdefs` 文件的用户将生成相同的 `comdat` 组时，可能会发生多个符号定义错误 L6200E。此类符号的一个示例就是 `symdef` 文件和应用程序均定义的编译器帮助器函数。

链接器现在将通过选择这些符号定义中的一个并拒绝其他定义，解决这一冲突。选择哪一定义由命令行选项 `--comdat_symdefs_preference=preference` 控制，其中，是 `symdefs` 或 `comdat` 之一。在为 `symdefs` 时，链接器将从 `symdefs` 文件中选择定义；在为 `comdat` 时，链接器将倾向于从 `comdat` 组中选择定义。的默认值为 `symdefs`。

[SDCOMP-11078]

`armlink` 现在支持 `ANY_SIZE` 分散文件关键字。这允许用户指定可分配给该执行区域中的 `.ANY` 选择器的内容量的限制。语法为：

```
exec_region ANY_SIZE
```

- 对某个区域指定 `ANY_SIZE` 将取代针对该区域的自动相邻。`armlink` 将不会分配超出此用户指定的限制的任何内容。

`armlink --info=any` 诊断已得到增强，可显示每个区域的相邻区域。

`armlink` 错误消息 L6220E 已得到改进，可报告区域中链接器生成的超出其限制的内容量。

以前，armlink 有时候会从区域中错误地清除 .ANY 相邻区域。此问题已得以更正。
[SDCOMP-10588]

- 新的排序算法可用于链接器的 `--sort` 命令行选项。可使用 `--sort=objectcode` 对此进行选择。此算法按中断器（tiebreaker）对代码节进行排序，所有其他节按语法排序。在 `--tiebreaker=cmdline` 时此算法最有用，因为这往往会将内存映射中来自相同对象的所有节组合在一起。[SDCOMP-10898]

自 RVCT 4.0（内部版本 870）以来进行的更正

编译器（armcc）

- 在优化级别 `-O3 -Otime`，对于具有固定迭代计数的 `while` 循环，如果循环条件涉及将无符号整数与非零的常量进行比较，并且循环迭代的数目很小，则编译器可能会生成错误的代码。[SDCOMP-11082]
- 在优化级别 `-O3 -Otime`，如果函数引用作为声明块中的初始化出现，并且解除引用的指针作为相同声明块的初始化出现，则在该函数更改该指针所指向的值时，编译器可能会生成错误的代码。[SDCOMP-10940]
- 在某些情况下，在自身中存储递归数据结构的地址（或者使用实现类似结果的指针算法）的所有优化级别上都可能导致生成 UNPREDICTABLE 存储结构。现在，此问题已得以解决。[SDCOMP-10900]
- 使用 `--gnu_instrument` 时内部错误 0x413ad2 的某些实例已修复。[SDCOMP-10801]
- armcc 有时会在 `-O3 -Otime` 时生成不正确的代码。如果已展开通过在末尾变量递减导出的 `while` 循环，将会跳过首次迭代。[SDCOMP-10789]
- armcc 有时会在 `-O3 -Otime` 时生成不正确的代码。某些从指针访问指针的循环在展开时将会引用过期值。[SDCOMP-10787]
- 在 `-O1` 和更高时在某些情况下，如果使用包含 `?:` 或内联函数的数组索引表达式写入短值的本地数组，则在以后读取该数组时可能会具有错误值。现在，此问题已得以解决。[SDCOMP-7494]
- 在使用 NEON“乘以 16 位标量”内部函数时在某些情况下会发生的内部错误已得以解决。0xa95fee [SDCOMP-7352]
- 在 `-O3 -Otime` 上，对于具有在返回时隐式转换的内联函数，armcc 有时候可能会错误地删除必需的转换。[SDCOMP-7346]
- 在 Thumb 模式下使用链接时代码生成时，可能会由于跨模块内联所作出的决策引发内部错误 0x2732f5。此情况不会再次发生。[SDCOMP-7333]
- 使用 NEON 内部函数，在第一个参数为静态常量时，armcc 在过去可能会将 VBIC 和 VORN 错误地视为累积运算。此问题已得以解决。[SDCOMP-7318]
- 使用 `--remove_unneeded_entities` 进行编译时内部错误 0xaac48d 的某些实例已修复。

链接器（armlink）

- 在 Cygwin 下（CYGPATH 环境变量设置为 `cygpath.exe` 的位置）运行时，RVCT 工具将调用 `cygpath.exe` 以便在 Cygwin 和本机 Windows 路径之间进行转换。对于 1.7.7 版的 Cygwin，链接器无法使用 `cygpath` 转换包括 Cygwin 路径的 `--libpath`

或 RVCT40LIB。现在，此问题已得以解决。 [SDCOMP-10621]

- `--veneer_inject_type=pool` 中间代码生成模型要求分支到池中的中间代码，以便能够达到池限值，这是中间代码可采用的可能的最高地址。如果以后找到了无法达到池限值的分支，则 `armlink` 可能会发出错误消息 L6828。如果分支能够将池中的所有中间代码放入较低的池限值中，则 `armlink` 现在将减小该池限值以便适合分支。只有在链接器无法减小该池限值的情况下，才会提供错误消息 L6828。 [SDCOMP-10620]
- `armlink --info=any` 输出已得到了增强，包含节索引和大小，并且在适当时可以打印存档名称。 [SDCOMP-10604]
- 在通过 `RVCT40_LINKOPT` 环境变量传递对象文件时，`armlink` 不再崩溃。 [SDCOMP-10589]
- Unix 和 Windows 工具有时候会无缘无故地以不同顺序输出 `.ARM.attributes` 节的子块，很难确定某个图像是否发生了任何重要更改。 [SDCOMP-1836]
- 以前，链接器会错误地设置版本定义节的 `entry size` 属性。现在，此属性被正确地设置为 0。 [SDCOMP-10616]

汇编器 (armasm)

- 在直接指定即时编码的 ARM 数据处理即时指令（例如 `ADD r0,r1,#3,#4`）中，汇编器现在在轮换前接受可选的 `#`。 [SDCOMP-8966]

C/C++ 库

- 以前，Cortex-M3 C 库只有 `strcmp` 的一个变体，此变体依赖于非对齐访问。现在，它们还提供一个不依赖于非对齐访问的变体。 [SDCOMP-7954]
- 在 `errno` 在输入时为非零时，库中的复杂数字函数 `clog()` 和 `cexp()` 会出错。`cexp` 有时候会将 `errno` 清除到零，`clog` 甚至可能返回错误结果。 [SDCOMP-1933]

ELF 格式转换器 (fromelf)

- 以前，如果对于不是压缩文件（如“`archive.a(*.o)`”）或不存在的文件使用了用于选择压缩文件成员的语法，`fromelf` 无法报告错误消息。现在，它将报告错误消息。 [SDCOMP-2381]
- 在对包含比较是否相等的元素的数组进行排序时，`microlib qsort()` 函数可能会永远挂起。此外，即使该函数不挂起，所表现出的性能也很差。 [SDCOMP-7957]

内部版本 870

自 RVCT 4.0 内部版本 821 以来进行的更正

编译器 (armcc)

- 以前，如果在启用 C++ 异常 (`--exceptions`) 的情况下进行编译，包含引用类型的函数异常规格（例如 `void g(void) throw(int&) {...}`）就会导致编译器内部错误。现在，此问题已得以解决。如果函数异常规格类型引用类指针 (`C * &`)，则无法得到正确处理，并会产生可降级的编译时错误 #2818 作为警报。降级该错误会在处理这些类

型时省略引用 (&)，这会导致运行时语义与源表示的语义略有不同（将允许引发派生类指针）。请参见“C++ 标准”（包含 TC1）的第 15.4/7 15.3 节。[727197]

- 以前，如果在 GNU 模式下尝试使用“aligned”属性减少变量的对齐字节数，将会产生错误并显示消息 #1039。现在允许这样做，但数组变量除外，因为这些变量在上述情况下会被忽略（与 GCC 相同）。[727773]
- 内部故障 0x84aef9 和 0x45464 的根源已得以解决。[740654]
- -O3 -Otime 时的内部故障 0x44784c 的根源已得以解决。[741821]
- 包含 --exceptions 的内部故障 0x3e9d63 的根源已得以解决。[730514]
- 以前在使用 --dllimport_runtime 和 LTCG 时，编译器有时无法导入某些从编译器生成的帮助器函数中引用的运行时函数。现在，此问题已得以解决。[735414]
- -O3 -Otime 时的内部故障 0xfe1945 的根源已得以解决。[737716]
- 已改进针对不正确使用位段的错误处理。如果将位段对象放置在位段区域之外（使用 at 属性或整数文本指针），编译器将执行以下操作：
 - 如果已向对象类型应用位段属性，而且未在命令行中指定 --bitband，编译器将显示错误。
 - 如果已向对象类型应用位段属性，而且在命令行中指定了 --bitband，编译器将显示警告并忽略位段请求。
 - 如果尚未向对象类型应用位段属性，而且在命令行中指定了 --bitband，编译器将忽略位段请求。
- 此外，在声明不完整的类型时，不再接受 at 属性。[740150]
- 在 GNU C++ 模式下，现在会预先将 std::type_info 声明为不完整的类型。[740973]
- 以前，如果在编译大量 switch 语句时（生成超过 128 KB 的代码）使用了许多 case 语句（超过 256 个 case），编译器就可能因内部错误而停止。现在，此问题已得以解决。[741084]
- 以前，对于包含涉及同一数组的重复表达式的源代码，如果指定 -O3 -Otime --vectorize 并转换不同赋值表达式的左侧部分，编译器就可能生成错误代码。现在，此问题已得以解决。[742024]
- 以前，在对某个对象调用方法时，如果该对象是由另一个方法调用中的引用返回的，则有时会在访问该对象时生成不正确的堆栈偏移。现在，此问题已得以解决。[742819]
- 以前，在对架构 v5TE 目标进行编译时，编译器可能对属于字对齐但非双字对齐的地址生成 LDRD。现在，此问题已得以解决。[743326]
- 编译器有时会在 -O3 -Otime 时生成不正确的代码，无符号 int 倒数索引的迭代计数有时会出现计算不正确的情况；如果循环索引没有符号，并且按 1 之外的增量进行倒数就会出现上述情况。[744069]

汇编器 (armasm)

- 以前，如果 LDR/STR pc, [pc, #n] 中的 n 不是 4 的倍数，汇编器将诊断为错误 A1322E。此消息只能对结果为 UNPREDICTABLE 的 LDR 指令发出。现在，此问题已得以解决。[731801]
 - 以前，对于协处理器 0、1、10 和 11 上的不可预测的协处理器指令，汇编器不会进行诊断。现在此问题已得到解决，汇编器将在上述情况下报告错误 A1477。[734903]
- 以前，汇编器允许忽略对从 NEON 通道到 ARM 寄存器的传输 (VMOV.
Rd, Dm[n]) 编制索引。此语法无效，现在将引发错误。[734910]

- 以前，如果在 Thumb 中对将 PC 作为目标的 ANDS、EORS、ADDS 和 SUBS 指令进行汇编，汇编器将静默接受这一无效语法；现在，汇编器会报告错误 A1912E。[741070]
- 以前，汇编器会允许在 Thumb 中使用带 .W 限定符的 IT 指令，但这一用法并不正确。现在，此问题已得以解决。[741275]
- 以前，如果在并行内部版本中使用 --cpreproc 选项，将导致错误 A1023E；现在此问题已得到解决。[741370]
- 以前，在汇编器中按未对齐偏移编写分支会在诊断后导致内部错误。现在，此问题已得以解决。[742922]
- 以前，在“MOVS pc, lr,”指令中指定的任何移位都会被汇编器忽略。由于这种情况不可预测，因此汇编器现在会将其诊断为错误 A1477。[743220]
- 以前，汇编器允许在寻址模式下对 PLD、PLDW 和 PLI 指令进行寄存器移位，但这一用法并不正确。现在，此问题已得以解决。[743221]
- 以前，汇编器错误地将 r15 作为 MRC 指令中的目标寄存器。这是 UAL 之前的协处理器指令语法，该语法将 CPSR 调整为结果。现在，此问题已得以解决。[744774]

链接器 (armlink)

- 链接器现在可以在 --sysv 映像中正确地对 .fini_array 节排序，以确保以正确的顺序执行析构函数。[737347]
- 当使用 --ltdc 进行链接时，链接器不再试图签出两个许可证。[737667]
- --any_contingency 选项已得到改进，现在允许为可能的中间代码添加临时空间。该开关的用法与以前一样，但在以前，需要大量中间代码的映像有时可能会溢出具有 .ANY 的区域。[743027]
- 以前，在合并 comdat 组时，链接器可能会错误地将针对被拒绝组中弱定义的引用转化为弱引用。这可能会导致删除未使用节的操作清除应保留的节。现在，此问题已得以解决。[743820]
- 以前，在启用大区域支持的情况下，--sort=LexicalState 有时可能会错误地对节进行排序（按平均调用深度）。现在，这种情况将不再出现。[744630]
- 以前，--any_sort_order=cmdline 并未按说明运行。现在，它将按 .ANY 在命令行索引顺序中的位置对要处理的节进行排序。[744636]

C 库

- posix_memalign 函数在多线程环境中无法正确运行：它要么不锁定任何互斥函数，要么就锁定不存在的互斥函数，具体视情况而定。[741380]
- 从内部版本 762 起，在多线程程序中，如果 fgetc()、gets()、fputs() 和 puts() 函数因错误或 EOF 而返回，则可能无法对流的互斥函数解锁。现在，它们能够可靠地对互斥函数解锁。[742769]
- 以前，如果 _sys_seek() 返回了正值，microlib 中的最小 stdio 实现就会表现出不正确的行为。内部状态不会由 fseek 更新，而且 fseek 和 fsetpos 会返回不正确的值。现在，此问题已得以解决。[743383]
- 现在，当堆用完内存时，库中的 stdio 实现应该不会崩溃。[743634]

ELF 格式转换器 (fromelf)

- 以前，通过重定位对与 16 位 Thumb PC 相关的 LDR 指令进行反汇编时，通常会在

目标符号名称中附加一个虚假的“+1024”。现在此问题已得以解决。[738998]

- 在 Windows 中,如果对包含 `.debug_str` 节的映像应用“`fromelf -v -g`”或“`fromelf -y -g`”,则可能导致 `fromelf` 因为非法存储访问而崩溃。[743339]

多个工具

- 到目前为止,在硬件浮点环境中,C 库中的 C99 复杂数字函数(在 `<math.h>` 中声明)一直使用错误的过程调用标准。按标准名称定义的库函数(`csin`、`ccos`、`cexp` 等)依靠 VFP 寄存器提供参数,编译器按相同方式生成对它们的调用。结果,调用复杂库函数的 `hardfp` 和 `softfp` 编译代码无法链接在一起(因为其中之一会错误地尝试调用这些库函数)。

现在此问题已得到解决,这样,`csin`、`ccos` 等将其参数传递到整数寄存器中,库还提供了使用 VFP 寄存器的其他入口点(其名称为 `__hardfp_csin`、`__hardfp_ccos` 等)。

遗憾的是,这种处理会带来以下影响:在对固定库进行链接时,调用复杂库函数的现有硬 FP 对象代码不会正常运行。对于 `.comment` 节包含来自受影响的 `armcc` 内部版本的标识字符串且 `.ARM.attributes` 节指示其使用 AAPCS 的 VFP 变体(`Tag_ABI_VFP_args = 1`)的对象,如果从中解析对任何函数的引用,链接器都会发出警告。[741936]

内部版本 821

自 RVCT 4.0 (内部版本 771) 以来的增强功能

一般增强功能

- 这些工具现在支持 `--cpu=7E-M`,以便为在 Cortex-M4 处理器中提供的 DSP (饱和 32 位 SIMD) 指令的作用下得以增强的 7-M 架构生成代码。[736675]
- `armlink` 现在支持一种新的排序类型: `--sort=LexicalState`. 这种方法先对所有 Thumb 代码排序,接着对 ARM 代码排序,然后按语法排序。[738147]
- `armlink` 现在支持新的开关 `--[no_]crosser_veneershare`。默认值为 `--crosser_veneershare`,此值保留现有行为,即允许跨执行区域共享中间代码。`--no_crosser_veneershare` 禁止跨执行区域共享中间代码。[737516]
- `armlink` 现在支持选项 `--emit-non-debug-relocs`。此选项的工作方式类似于 `--emit-relocs`,但仅从非调试节输出重定位。[737515]
- 添加了新的链接器选项 `--vneer_inject_type=`,其中可以是 `individual` 或 `pool`。默认值为 `individual`。此选项控制 `--largeregions` 模式为打开时的中间代码布局。当 `--vneer_inject_type=individual` 时,链接器将确定中间代码的位置,以便最大量的使用中间代码的节能够找到这些中间代码。在执行区域之间重用中间代码是允许的行为。此类型可最大限度地减少所需的中间代码数量,但会最大程度地损坏映像的结构。`--vneer_inject_type=pool` 时,链接器将从执行区域的邻近范围收集中间代码,并将该范围生成的所有中间代码放入位于该范围末尾的池中。大型执行区域可能具有多个范围,因此具有多个池。这种类型对于映像结构的影响要小得多,但重用的

机会较小，因为代码范围无法在另一个池中重用中间代码。链接器基于所存在的跳转指令来计算链接器预测可能需要中间代码的范围。当需要状态更改时，或当从源到目标的距离（加上相邻区域）大于分支范围时，则预测此分支需要中间代码。可以使用 `--veneer_pool_size=` 来设置相邻区域的大小。默认情况下，此值设置为 102400 字节。`--info=veneerpools` 选项提供有关链接器如何放置中间代码池的信息。[737514]

- 添加了新的链接器选项 `--tiebreaker=`，其中是 `creation` 或 `cmdline`。当排序算法需要各节的汇总排序时，将使用一个中断器（`tiebreaker`）。默认中断器为 `creation`，它是在链接器内创建节数据结构的顺序。而 `--tiebreaker=cmdline` 则是该节出现在命令行上的顺序。命令行顺序定义为 `File.Object.Section`，其中“Section”是对象内该节的 `sh_idx`。“Object”是该对象出现在该文件中的顺序。“File”是该文件出现在命令行上的顺序。仅当文件为 `ar` 压缩文件时，对象在文件中出现的顺序才有意义。要修改映射文件的格式（由 `--map` 生成）以显示命令行顺序，请使用 `--section_index_display=cmdline` 开关。[737517]
- 链接器现在支持其他用于控制各个 `.ANY` 节的位置的选项。提供了两个新的命令行选项：第一个选项是 `--any_sort_order=`，其中是 `descending_size`（按大小降序排序）或 `cmdline`（按命令行索引排序）。默认值为 `descending_size`。第二个选项是 `--any_placement=`，其中可以为 `first_fit`（将该节放在第一个具有空间的 ER 中）、`best_fit`（将该节放在第一个具有最小可用空间的 ER 中）、`worst_fit`（将该节放在第一个具有最大可用空间的 ER 中）或 `next_fit`（如果当前 ER 具有足够空间，则将该节放入此 ER 中；或者如果此 ER 空间不足，则将该节移到下一个 ER。决不要回溯到前一个 ER）。默认值为 `worst_fit`。此外，还添加了一个新的 `info` 选项：`--info any`。此时将输出排序顺序、分配算法以及按照放置算法处理的顺序将哪些节分配给哪些执行区域等信息。[737513]

自 RVCT 4.0（内部版本 771）以来进行的更正

编译器（armcc）

- 现在已解决了一个代码生成问题，该问题对未初始化的指针生成解除参考。[734706]
- 警告和错误在系统头文件中始终处于禁用状态。现在，这些警告和错误在相应的隐式包含的模板实现文件中也处于禁用状态。[736105]
- 当使用 `--remarks` 时，将宽字符串（`L"abc"`）传递给 `printf` 函数可能生成有关不兼容的 `printf` 参数的 #181 注释，即使对应的格式字符串为 `%ls` 也不例外。现在，在这种情况下不会发生此注释。[735496]
- 当在 GNU 模式下使用 `--multibyte_chars` 时，如果某行以一个多字节字符结尾（例如在注释中）且后跟一个或多个空白字符，则编译器可能检测到虚假的行联结字符。当使用 C++ 样式的尾随注释时，这可能导致不正确地解释源代码。[737748]
- 在 GNU C++ 模式下，使用复合文字（C99/GNU 扩展）初始化文件范围的变量可导致运行时故障。现在，此问题已得以解决。[736266]
- 已对 GCC 命令行转换进行了一些小的更正。[735554]
- 与 `-g --remove_unneeded_entities` 有关的内部故障 0xe70a0c 的根源已得以解决。[738296]
- `-O3 -Otime` 时的 0xfe1945 和 0xd9189 内部故障的根源已得以解决。[737716,

735852]

汇编器 (armasm)

- 新增的 `--branchpatch=cortex-a8_q109` 选项用于解决 Cortex-A8 错误，方法是：插入 NOP 以避免出现跨越页面边界的宽分支。[736360]

链接器 (armlink)

- armlink 可为 `.dynstr` 节分配错误类型，这将导致 `fromelf` 生成错误 Q0454E。现在，此问题已得以解决。[732898]
- armlink 现在将正确地处理旧样式区域表。以前在某些特定情况下，armlink 无法链接采用旧样式区域表实现其自己的分散加载机制的映像，并在退出时显示“ARM Linker: Execution interrupted due to an illegal storage access”。[734549]
- 现在，当 armlink 无法写入大小超过最大大小 2 GB 的连续区块时，将报告错误消息 L6065E。[728400]
- armlink 现在可以在 `--sysv` 映像中正确地对 `.fini_array` 节排序，以确保以正确的顺序执行析构函数。[737347]
- armlink 调用图可提供增强的函数指针信息，并综合显示同一节中对同一函数指针的多个调用。[736849]
- 当使用 `--ltcg` 进行链接时，链接器不再试图签出两个许可证。[737667]
- armlink 将从“PI”执行区域中独立于位置的代码到“绝对”执行区域中的目标，为分支生成分支中间代码。默认情况下，armlink 不重用这些导致代码大小大幅度增加的中间代码。现在，此问题已得以解决，并且 armlink 在确保安全的情况下将重用这些中间代码。[738802]

C/C++ 库

- 如果使用 `#pragma import __use_utf8_ctype` 和 `#pragma import __use_full_wctype` 启用完整的 Unicode 宽字符 ctype 函数 `toupper()` 和 `tolower()`，并且使用 `--wchar32` 编译这两个函数，则会针对字符对 U+1D79 和 U+A77D 返回错误答案，而正确答案应是彼此的大小写版本。[736950]
- 在严格的 C++ 模式下，`FILE` 不再声明 `FILE` 类型。此外，在此版本中也可以定义 `__ARM_WCHAR_NO_IO` 宏，以使这些头不声明 `FILE` 或宽 I/O 函数原型。[733711]

ELF 格式转换器 (fromelf)

- 有时，`fromelf --bin` 可能用不相关的 `SHT_NOBITS` 节来对输出文件命名，此节将同一个文件偏移量共享为文件中包含的段。[735346]

多个工具

- 当生成 DWARF 调试信息时，汇编器和编译器可能输出 CIE 和 FDE 记录，而这些记录的长度不是四的倍数，这可能导致第三方调试工具出现问题。这一问题现已解决，可在适当的位置生成填充。[733101]
- `--cpu=Cortex-M4.fp` 现在设置预定义的宏 `__TARGET_CPU_CORTEX_M4`，而以前则设置 `__TARGET_CPU_CORTEX_M4_FP`。可以使用 `__TARGET_FPU_VFP` 宏测试 FP 是否存在。[736497]

- 在日语消息区域设置中，当使用 `--diag_style=ide` 时，文件的行和列之间的逗号现在将变为 ASCII 逗号，而不是日文汉字逗号。[736125]
-

内部版本 771

自 RVCT 4.0（内部版本 697）以来的增强功能

为改进与其他工具链之间的兼容性所做的更改

- AEABI 规定根据易失性位字段的容器类型大小来存取易失性位字段，而某些 GCC 版本将使用包含整个位字段的最小存取大小。添加的编译器开关 `--narrow_volatile_bitfields` 用于模拟这种不符合 AEABI 的行为。[731902]
- 添加的编译器命令行选项 (`--configure_gcc_version`) 采用 `x.y.z` 格式的 GCC 版本号，并使用此版本号作为要报告的 GCC 版本。该选项也可在根据 GCC 安装进行配置以便替代报告的版本时使用，但是，如果用户替代版本早于查询的 GCC 所报告的版本，该选项则会发出警告。[729761]
- 编译器现在采用 `--[no_]execstack` 选项来生成 `.note.GNU-stack` 节，该节将堆栈标记为可执行或不可执行。如果未使用此选项，则不会生成 `note` 节。选项 `--arm_linux` 隐含 `--no_execstack`。[725814]
- 如果指定了 `--gnu`，编译器便可使用无符号数进行枚举（不包含负值）。[728146]
- C 库现在提供 `posix_memalign()` 函数来分配对齐的存储区。可使用 `free()` 来释放该存储区。[731812]

一般增强功能

- RVCT 现在支持 VFPv4 浮点架构 (`--fpu=VFPv4`)。[731258]
- 现在，如果当前目录和/或源文件目录不同于预编译头 (PCH) 文件的创建目录，编译器可重用 PCH。[375431]
- 现在，IMPORT 和 EXPORT 指令支持可选的 SIZE 属性，以便允许在程序集中指定符号大小（例如，`EXPORT dataSymbol[DATA,SIZE=8]`）。[730839]
- 添加了一个新的编译器选项 `--import_all_vtbl`。这会导致将类障碍变量 (vtables、RTTI 等) 的外部引用标记为具有动态链接。该选项不会导致类障碍定义具有动态链接。[727750]
- 现在，当在 OVERLAY 区域中生成动态重定位时，链接器将生成警告消息 L6491W。当动态重定位目标位于 OVERLAY 区域时，将生成消息 L6492W。[728049]
- 新增的选项 `--[no_]implicit_key_function` 用于控制可否将隐式实例化的模板成员函数选择为关键函数。（通常，某个类的关键函数（或决定函数）是该类的第一个非内联虚拟函数。但是，对于隐式实例化的模板函数，该函数将具有模糊链接，即，可能被多次定义。）默认值未发生更改，允许关键函数为隐式函数。此外，当关键函数为隐式函数时，还将生成新的 #2819-D 注释。使用 `--remarks` 或 `--diag_warning=2819` 可查看此注释。[732323]
- 对于多线程用户，库现在提供不同格式的高级 stdio 函数 (`fputs`、`fgets`、`printf` 等)，这些函数一次性为整个调用锁定目标流，而不是基于每个字符进行单独锁定。这

应当能够提高性能，同时还意味着对 `printf` 的调用在输出中可靠地显示为一个整体（而不可能与其他线程的输出逐字符相互交错）。[727119]

- `--library_interface` 开关现在支持的选项允许 `armcc` 使用早期 RVCT 库版本中的特定于版本的优化函数。这些选项包括 `rvct30`、`rvct30_c90`、`rvct31` 和 `rvct31_c90`。此外，还提供了 `rvct40`、`rvct40_c90` 选项。[717109]
- 在严格的 C++ 模式下，`FILE` 不再声明 `FILE` 类型。此外，在此版本中也可以定义 `__ARM_WCHAR_NO_IO` 宏，以使这些头不声明 `FILE` 或宽 I/O 函数原型。[733711]
- `armlink` 现在支持在映射文件中打印执行区域的加载地址。必须使用 `--load_addr_map_info` 开关启用此功能（同时必须指定 `--map`）。默认情况下会禁用此功能。

自 RVCT 4.0（内部版本 697）以来进行的更正

编译器（`armcc`）

- 以下内部故障的某些根源已得以解决：0x87ecef、0x6bca8b、0x6bca8b、0x413ad2、0xd9827b、0xcc3958、0x2732f5、0xafc2db、0xa05965、0xf7584e、0xb36758、0x2d4b56、0xa718eb 和 0x4553df。
- 以前，对易失性变量的访问或测试可能会在 `-O3 -Otime` 时错误地退出循环。此情况不会再次发生。[733001, 729956]
- 在 Windows 上，编译器可能无法正确处理 BOM（字节顺序标记）引入的 UTF-8 源代码。现在，此问题已得以解决。[732849]
- 以前，当在某个类模板成员函数的声明中将该函数标记为 `dllimport`，并在类外内联定义中标记为 `dllexport` 时，实例化该类可能不会生成成员函数的外联副本，这与将始终为 `dllexport` 函数（即使标记为 `inline` 的函数）生成外联副本的原理相违背。现在，此问题已得以解决。[732208]
- 利用优化选项 `-O3 -Otime`，当内联多次使用了形参的某个函数时，如果相应实参包含一个增量，此增量则会多次出现。现在，此问题已得以解决。[732499]
- 在优化级别 `-O1` 或更高级别中，对于使用带有空体的默认构造函数构造的类类型的局部变量，可生成警告 #177-D（已声明但从未引用变量）和 #550-D（已设置但未使用变量）。现在已禁用这些警告，这与构造函数不为空时的通常情况一样。[732465]
- 当使用 `--message_locale` 时，未准确地本地化有关创建和重用预编译头（PCH）文件的诸多消息。现在，此问题已得以解决。[732406]
- `-O3 -Otime` 时的代码生成错误已得以解决，当在倒数 `do-while` 循环中在每次迭代中写入 3 个字节时，该错误导致写入操作超过缓冲区末尾。[732346]
- 对类型对齐超过 8 个字节（即，根据“aligned”属性设置时）的对象数组应用“new”和“delete”时，曾出现过各种问题。现在，该用法可正常运行。[731813]
- 异常表生成故障可导致将从析构函数的引发定向到错误的 `catch` 处理程序，该析构函数作为定位到嵌入 `try` 块或从嵌入 `try` 块返回的一部分调用。现在，此问题已得以解决。[730117]
- 当使用 `--visibility_inlines_hidden` 时，既不会生成某些标记为 `dllimport` 的类内联函数，也不会导入这些函数。现在将生成这些函数。[731207]
- 编译器可错误地优化掉对结构的易失性域的访问，而这些结构本身并非是易失性结构。现在，此问题已得以解决。[727881]

- 嵌入式汇编源代码中的错误可导致将输出对象文件保留为非完整状态。现在，此问题已得以解决。[726941]
- 如果对某个函数的所有直接调用都传递一个具有相同常数值参数，该值可替代为例行程序正文。在可间接调用此函数的某些情况下，编译器错误地执行此操作。现在，此问题已得以解决。[730281]
- 影响 RVDS 随附的 XVID 示例的代码生成故障已得以解决。[727352]
- 现在，不带 `-E` 的 `--list_macros` 选项可正确地只生成宏声明列表，因为这些宏是在预处理的源代码末尾定义的。如果指定了 `-E`（或类似选项，如 `-P`），在生成的预处理的源代码中，宏 `#define` 和 `#undef` 指令将相互交错。
- 已使用“alias”属性创建别名的静态例程永远保留在映像中，即使未使用这些例程也是如此。现在，此问题已得以解决。[729567]
- 现在，如果 FP 模型（由 `--fpmode` 设置）完全符合 C 标准的 F 附录中的 IEEE 浮点模型要求，编译器则仅定义 `__STDC_IEC_559__`。这与源语言（C90、C99 或 C++）无关。[724114]
- 现在，当显式或隐式使用 `--multifile` 选项时，编译器将定义 `__MULTIFILE` 宏。对于当前通过测试 `__OPTIMISE_LEVEL` 是否至少为 3 来检测此功能的任何源级别条件化体系，建议将这些体系更改为测试 `__MULTIFILE`。[729445]
- 当为标记为 `__forceinline` 的多态继承类的虚拟函数生成调试信息时，可发生内部故障 0x04cb7b。现在，此问题已得以解决。[732688]
- 编译器可能无法注意到循环计数器中已清除的最低有效位，并在 `-O3 -Otime` 时，在存在 `#pragma unroll` 的情况下展开之后生成不必要的循环清除代码。现在，此问题已得以解决。[626066]
- 组合 LTCG 和 `--dllimport_runtime` 选项可导致对某些异常运行时例行程序的引用并非由动态链接指定。现在，此问题已得以解决。[733308]
- 在某些情况下，编译器可能试图将以“-”开头的文件名作为选项进行分析，即使位于“--”选项终止符参数之后也是如此。现在，此问题已得以解决。[733245]
- 如果 `catch` 子句使用了对类类型的引用（如“`catch (C &c)`”），编译器可能无法为基础类 C 生成类型信息。现在，此问题已得以解决。[732867]
- 在某些情况下，编译器中包括 `__ssat()`、`__smuad()` 和 `__smlad()` 在内的某些内部函数被错误地视为不会影响 Q 标志，从而导致对 Q 标志的某些访问作为冗余访问而被删除。现在，此问题已得以解决。[729899]
- 当生成包含大量节的对象时（例如，当对大型源使用 `--split_sections` 时），编译器运行非常缓慢。现在，此问题已得以解决。[733748]
- 在 Windows 上，当通过 Perl、GNU make 或类似程序中的相对路径进行调用时（例如，当处理嵌入式汇编器时），编译器有时可能找不到汇编器和链接器。现在，此问题已得以解决。[733795]
- 在 `-O1` 及更高级别中，可能无法正确合成复制构造函数。如果类 A 派生自类 B，类 B 派生自多个（可能为多态）基类，并且所有这些基类的域都可通过逐位复制进行复制构造，则可能会发生此种情况。此种情况下，为 A 合成的复制构造函数可能会仅设置 `vtable` 指针，而不会复制基类内容。现在，此问题已得以解决。[733659]
- `-O3 -Otime` 时，如果具有递增式下标的对象表达式具有内联成员函数调用，下标增量将发生重复。现在，此问题已得以解决。[734246]

汇编器 (armasm)

- 汇编器现在允许 REQUIRE8 或 PRESERVE8 替代早期实例。[732197]
- 使用 VMSR 写入到系统寄存器 MVFR0 和 MVFR1 被诊断为不可预测。[720647]
- 汇编器以前允许您使用表达式格式为 {ABSOLUTE} - {RELATIVE} 的 EQU 指令来定义符号，但无法正确解析该指令。现在，如果您尝试执行此操作，汇编器将引发 A1109E 错误。[647017]
- 如果在使用 REQUIRE 指令不久后便使用 RELOC，则将忽略 REQUIRE 指令，此问题已得以解决。[731252]
- 当为不带 CMN 的 16 位 Thumb 指令集生成代码时，汇编器会将 CMN Rn, #0 替代为 CMP Rn, #0，这会产生错误的标志设置行为。现在，该指令为错误指令。[728052]
- 已解决当使用 ALIGN 指令时偏移量不是 4 字节的倍数所带来的内存不足的问题。[729860]
- 汇编器不接受仅指定了源操作数数据类型的 N 和 L 形指令(例如, VADDHN, I16, I16 d0, q1, q2)。现在，此问题已得以解决。[719906]

链接器 (armlink)

- 链接器内联可能错误地仅内联含有 3 个半字指令的 Thumb 函数的两个指令。现在，此问题已得以解决。[732427]
- 现在，与位置无关的 R0 节可与 RW 节位于同一 PI 执行区域中，而不会导致虚假 L6265E 错误。[730622]
- 分散加载选择器 (+R0) 可能根据名称错误地匹配名称为 R0 的节。RW 和 ZI 也存在类似情况。现在，这些问题已得以解决。[730269]
- 当对含有多种损坏的 C++ 符号的大型项目使用 armlink --symbols 选项时，如果试图显示未损坏的符号名称，链接器可能会生成消息“Fatal Error L6000U: Out of memory”。现在，此问题已得以解决；现在当使用 --symbols 选项时，armlink 使用的内存要少得多。[730197]
- 根据使用 --enum_is_int 编译的代码链接 microlib 将不再导致链接错误 L6242E。[729408]

C/C++ 库

- 现在，C 库包含函数 _srand_r()、_rand_r() 和 _setlocale_r()，这些函数与标准 C 函数 srand()、rand() 和 setlocale() 相似，但前者使用用户提供的缓冲区，而不是 C 库中的静态数据。[721560]
- 使用 C++ 异常时选择的 __cxa_vec_ctor 版本无法正确处理传递的构造函数指针为 NULL 的情况。现在，此问题已得以解决。[730012]
- 如果通过 --remarks 或其他方式将 armcc 配置为报告 printf 格式字符串及其参数之间的不匹配情况，armcc 将报告 inttypes.h 宏 PRIu8 及其应专门匹配的参数类型 (uint8_t) 之间的不匹配情况。对于 PRIu16、PRIuLEAST8 和 PRIuLEAST16 也同样如此。而且，缺少 inttypes.h 宏 SCNd8、SCNi8、SCNo8、SCNu8 和 SCNx8。[733080]

ELF 格式转换器 (fromelf)

- 以前在某些情况下，当对象文件的调试信息描述含有匿名子结构的结构类型或描述继承自无名基类的 C++ 类时，“fromelf --fieldoffsets” 往往会发生崩溃。[733770]

多个工具

- 在 队列模式下签出许可证会迅速超时,并且有时会导致出现 FLEX 错误:“Licensed number of users already reached. (-4,342)”,而不是显示正确的消息:“Warning:C9933W:Waiting for license...”,即使在使用 `--licretry` 选项时也是如此。当许可证签出频率非常高时便可能出现此问题,例如在使用并行编译 (`make -j`) 时。现在,此问题已通过增加等待时间得以解决。 [721403]
 - 有时,可以尝试使用 `--version_number` 选项获取许可证。现在,此问题已得以解决。 [730056]
-

内部版本 697

此版本中的新增增强功能 - 关键功能

为改进与其他工具链之间的兼容性所做的更改:

- 已对 GCC 命令行转换功能进行了相应修改,可提高兼容性。特别是,当您现在使用 GCC 命令行转换时,默认优化级别将设置为 `-O0`,而不是 `armcc` 常规默认级别 `-O2`。 [727559]
- 现在,编译器定义了 `__builtin_bswap32` 和 `__builtin_bswap64` 内部函数,以便与 GCC 兼容。这些内部函数分别还原 32 位和 64 位字词中的字节。 [714828]
- 现在,如果 `#elif` 指令中缺少表达式,编译器将通过诊断 `#29-D` 来发出警告,即使因以前已成功执行 `#if` 或 `#elif` 而导致通常不会对 `#elif` 求值时也是如此。当模拟 GCC 4.4 以前的版本时,不会输出此警告(请注意,GCC 模拟模式下的默认值为模拟 GCC 4.2)。 [728705]
- 当使用 `--sysv` (由 `--arm_linux` 表示) 进行链接时,链接器会将 `.ARM.exidx` 节的 `sh_link` 域设置为 `.text` 节的索引。这将提高与 `objcopy` 的互操作性。 [723436]
- 现在,在 GNU 模式下,对表达式上下文中的 `C::C` 的引用(其中 `C` 表示类名)被视为对构造函数的调用。这更加贴切地反映了 GCC 行为。 [726327]
- 默认情况下,`--gnu_defaults` 选项现在还会设置 `--no_implicit_include`。 [726255]

对数学库的改进:

- 以前,对数学库 (`mathlib`) 中错误情况的处理都是通过 RVCT 实现的,从某种程度上说,这不符合 C99 标准附录 F 中的规定。现在,支持将符合 C99 的错误处理作为一种备用方法,但默认情况下仍采用以前的 RVCT 行为。在程序中导入符号 `__use_c99_matherr` 可启用 C99 行为。明确受到影响的函数包括:
 - `atan2`
 - `atanh`
 - `hypot`
 - `lgamma`
 - `log`
 - `log10`
 - `log2`
 - `log1p`

- pow
- tgamma
- 及其对应的单精度函数。 [720506]

一般增强功能:

- Cortex-A5 现在已成为公认的支持编译器的 CPU。 [726103]
- 现在, 链接器可在构建 DLL 和共享库时删除未使用的节。 [720889]
- 仅当对含有 32 位 Thumb 指令的 CPU 进行汇编时, 汇编器才会发出针对指令中已弃用的寄存器的诊断。要在其他架构启用这些诊断, 请使用选项 `--diag_warning=1745,1786,1788,1789`。 [728201]
- 要跟踪《ARM 架构参考手册》中的变更内容, 汇编器允许省略立即操作数前面显示的“#”字符。当使用此语法以允许检测可能不会使用其他(或早期)汇编器汇编的源代码时, 将发出警告 A1865W。 [365430]
- 新增的汇编器选项 `--[no_]execstack` 可用于构建描述代码是否需要可执行堆栈的 `.note.GNU-stack` 节。 [725813]
- 编译器现在可忽略源文件开头的含有 3 个字节的字节顺序标记 (BOM)。 [725395]
- 自 RVCT 3.0 起, 编译器已实现 C++ 核心问题 #446 的解决方案, 该解决方案要求使用临时存储保存其结果为类 rvalue 的条件表达式的结果。由于这可以在极少情况下更改程序行为, 因此, 现在可通过使用新的 #2816 注释来诊断引入了此类临时存储的情况。使用 `--remarks` 或 `--diag_warning=2816` 可查看此注释。 [726389]
- 新增的编译器选项用于指定输出文件的目录。`--output_dir` 通常指定输出目录, 包括 `.o` 文件的输出目录。汇编器输出目录可使用 `--asm_dir` 指定, 对于相关性输出, 可使用 `--depend_dir`, 对于 `--list` 输出, 可使用 `--list_dir`。如果未使用这些选项, 则将相应输出放置到 `--output_dir` 指定的目录中。如果未指定该目录, 则放置到默认位置(如当前目录)。 [724352]
- 现在, 通过使用组合 `--md -E` (或 `--md -P` 以便禁用行号生成), 可以在一个步骤中生成 makefile 相关性和预处理的源输出。 [606117]
- 现在, 当执行链接步骤时, 可以从标准输入流中提取源代码, 并将其与其他文件组合。例如, 现在允许以下命令: “`armcc -o output.axf - object.o mylibrary.a`”, “`armcc -o output.axf --c90 source.c -`”。此外, 还会更改 “`armcc -o test.axf -`” 的行为以便与上述内容保持一致, 因此, 这将编译在标准输入中提供的源代码, 并将其链接到 `test.axf` 中。仍禁止在未链接时组合标准输入和其他源文件, 并且这仍将生成错误 C3078E。 [723225]
- 链接器具有一个新选项 `--[no_]import_unresolved`。默认选项为 `--import_unresolved`。当使用 `--sysv - shared` 链接共享对象时, 通常将导入未解析的符号, 而不会导致生成错误。当指定 `--no_import_unresolved` 时, 在链接器命令行上明确列出的对象文件中的未解析的引用将导致未定义的符号错误, 而不会导入这些引用。 [727609]
- 新增的链接器选项 `--no_eager_load_debug` 指示链接器在加载对象之后不要将调试节数据保留在内存中。这将降低链接器的内存使用峰值。较低的内存使用还有利于部分链接。 [722351]
- 现在, 为了支持不可执行的堆栈, 当选择 `--sysv` (由 `--arm_linux` 表示) 时, 链接器会生成相应的 `PT_GNU_STACK` 程序头。链接器将根据输入对象中的 `.note.GNU-stack` 节的状态派生堆栈的可执行状态。新增的相关选项为

--[no_]execstack。[725212]

- C 库宽字符 ctype 函数 (isw*, towupper、towlower) 现在已使用 Unicode 5.0 更新。[726104]
- 现在, fromelf 可正确保留 DT_RELCOUNT。[727579]

已弃用的功能

- 汇编器已不再使用 | 运算符作为 :OR: 的同义词, 因为这会导致在源行中替换变量时出现问题。[727348]

此版本包含自 RVCT 4.0 版本 (内部版本 697) 以来所做的以下修正:

编译器 (armcc):

- 以前, 如果在启用 C++ 异常 (--exceptions) 的情况下进行编译, 包含引用类型的函数异常规格 (例如 void g(void) throw(int&) {...}) 就会导致编译器内部错误。现在, 此问题已得以解决。[727197]
- --list 文档错误地声明: 当用于多个源文件时, --list 将为第一个文件生成一个列表。实际上, 它将为最后一个文件生成一个列表, 但该列表的文件名将根据第一个源文件名生成。现在, 编译器可为所有源文件生成列表, 并将每个列表写入到单独的列表文件中, 列表的文件名将根据相应的源文件名而生成。[728254]
- #pragma unroll(1) 现在可对展开程度产生正确影响。[727935]
- 现在, #pragma push/pop 对根据 #pragma pack 设置的当前结构对齐进行正确压栈和出栈。[727532]
- 当使用的 va_list 自定义声明的类型与 stdarg.h 中的类型的标准声明 (根据 ARM 过程调用标准指定) 不兼容时, 编译器将生成错误 #256, 指示当与 --gnu 选项配合使用时, 重新声明的类型不兼容。现在, 此问题已得以解决。[727506]
- 在某些情况下, 错误地删除了对带有参数 (用于设置 errno) 的 fmod 的调用。现在, 此问题已得以解决。[726236]
- 现在, 使用 __attribute__((used)) 标记的函数和数据已在对象中标记出来, 以免被链接器的未使用节删除功能删除。[724019]
- 对于所有浮点模型 (由 --fpmode 设置), 编译器将求反浮点比较优化为相反比较 (例如, 将 !(a < b) 转换为 (a >= b))。这种情形已有所变化, 这样, 仅当 FP 模型指示我们并不关注 NaN (默认为 --fpmode std、--fpmode fast) 时, 才会应用优化。[725146]
- 在某些情况下 (即在 -O3 -Otime 时, 循环测试中从无符号整数到有符号整数的转换), 无提示代码生成故障会被丢弃, 现在此问题已得以解决。[726234]
- 当使用编译器选项 --locale 时, 生成警告消息有时可能会重置当前区域设置, 从而导致错误地解释源代码中的多字节字符序列。现在, 此问题已得以解决。[725074]
- 当使用某些多字节编码 (具体而言, 即普通 ASCII 字符可显示为多字节字符的后续字节的多字节编码, 因此包括 Shift-JIS, 但不包括 UTF-8) 对 C/C++ 源代码进行编码时, 在宏扩展中使用 “#” 字符串化操作符可产生意外结果。现在, 此问题已得以解决。[725874]
- 已使用 __attribute__((section("name"))) 指定显式节属性的内联函数的外联实例可生成链接时错误。现在, 此问题已得以解决。[723920]
- 链接时代码生成有时无法为 “C2” 构造函数或 “D2” 析构函数发出符号。现在, 此问题已得以解决。[728099]

- 使用 `makefile` 相关性生成选项可能会不必要地阻止编译器重用预编译头 (PCH) 文件, 并显示警告 #629-D。现在可重用 PCH 文件。 [728897]
- 如果使用了 `--remarks`, 当模板中的 `friend` 定义使用的变量与从中调用 (即首次从中实例化) 变量的函数中的变量同名, 编译器可生成虚假注释。现在, 此问题已得以解决。 [728548]
- 以前, 编译器可能无法为同时具有虚拟函数的 `dllimport` 类的已导入内联函数提供动态链接。现在, 此问题已得以解决。 [727752]
- 以下内部故障的某些根源已得以解决:
 - 0x87ecef, 0xcc3958, 0x87ecef
 - 0xa05965, 0xbfdfa3, 0x9be513
 - 0xa85545, 0x56e87b, 0x63faa5
 - 0xafee20, 0xfe1945
- [728236, 685024, 727558, 589466, 726944, 726750, 726300, 725422, 728252, 728260]

汇编器 (armasm):

- 现在, 导致错误报告 A1201E 的问题已得以解决。 [722855]

链接器 (armlink):

- 链接器通过 `OVERLAY` 属性可能会算错两个或多个执行区域之后的执行区域的基址。当上述 `OVERLAY` 区域的基址不相同, 则会出现此问题。现在, 此问题已得以解决。 [725446]
- 当链接 `test_mla` Cortex-M3 示例时, 链接器不会再报告 L6967E。 [725351]
- 现在, 为了更好地与 GNU 工具的行为保持一致, 在选择了 `--arm_linux` 模式的情况下, 如果建立了从已用节到某个节的弱引用, 链接器将保留后一个节。 [726778]
- 当读取含有 Thumb 符号的 `symdefs` 文件时, 如果 Thumb 代码分支到该符号, 链接器可能会生成虚假警告: L6307W。现在, 此问题已得以解决。 [726869]
- 当对象包含对 `$Sub$$Symbol` 的非法引用时, 链接器将生成错误消息 L62551; 当对象中存在 `$Super$$Symbol` 定义时, 链接器将生成错误消息 L6250。如果重定位到上述任一非法符号, 链接器将生成内部故障 0x3ae533。现在, 此内部故障已得以解决。 [726491]
- 如果某个对象文件包含 `$Sub$$Symbol` 的定义, 并且另一对象文件和某个库均包含符号定义, 链接器将错误地在该库中搜索符号, 加载成员, 并最终返回多个符号定义错误。现在, 此问题已得以解决。 [726487]
- 当使用 `--arm_linux` 调用时, `armlink` 始终会将诊断 L6029E (当输入对象文件中的重定位与无效符号或缺少的符号相关时) 报告为警告, 并且禁止将其严重性降级为注释或完全禁用。现在, 此问题已得以解决。 [725951]

C/C++ 库:

- 以前, 当要求 `scanf` 系列函数分析表示零的 C99 十六进制浮点文字 (如 "0x0.0p0") 时, 这些函数往往报告分析故障。现在, 这些函数可正确分析浮点文字。 [726414]
- 早期 RVCT 4.0 补丁版本中的变更引入了一个新问题, 即 C 库 `stdio` 清理函数 `_terminateio()` 将在试图两次锁定同一互斥函数时出现死锁。现在, 此问题已得以解决。 [726232]
- 根据 C++ 标准库问题 #109 的解决方案, 中的 `binder1st` 和 `binder2nd` 现在将

具有采用非常量引用的 `operator()` 的额外过载。[724709]

- 有关非常见用例中的错误运行时行为的问题已得以解决。当重新引发异常时，如果在堆栈展开时调用的析构函数再次重新引发，并捕获同一异常对象，则会错误地继续执行第一次重新引发操作（例如，错误地还原寄存器）。[728231]

库管理程序 (armar):

- 当使用 `armar -d` 删除不存在的成员时，现在，错误消息 L6831E 将变为警告。[727938]

ELF 格式转换器 (fromelf):

- `fromelf--elf` 复制选项将符号定义的“需要的版本”条目错误地转换为全局版本。这可能会影响使用版本定义访问共享库中的全局数据的 ARM Linux 应用程序。现在，此问题已得以解决。[727621]
- 当某个 ELF 段中的未初始化部分与另一 ELF 段的初始化部分重叠时，`fromelf` 选项“`--bincombined`”在过去往往会错误地报告错误。现在，此问题已得以解决。[725745]

内部版本 650

此版本中的新增增强功能

新增核心支持

- 支持 ARM 提供的新型 [Cortex A5 处理器](#)

为改进与其他工具链之间的兼容性所做的更改

- 利用链接时代码生成 (`--ltcg`)， 如果不符合“常量”条件的外部引用将某个转换单元中的常量数据导入到其他转换单元中，则可能会将该数据放置到可写节中。现在，该数据始终转入到只读节中。 请注意，此类程序并不会严格遵循 C，C 要求带有外部链接的符号在其声明所在的所有位置都具有一致的类型。[721646]
- 现在，编译器对文件扩展名的处理方式更符合现有约定。当使用 `--gnu_defaults` 选项时，扩展名 `.ac`、`.tc` 和类似扩展名不再视为 C 源代码；而是将这些扩展名传递到链接器。此外，`.sx`（及 Unix 主机上的 `.S`）现在指示需要预处理的汇编器源代码。[720541, 644030]
- 编译器现在支持 `#pragma weak`。这是已弃用的语言扩展，用于将符号标记为弱符号或定义符号的弱别名，它是使用现有 `__weak` 关键字或 GCC “weak” 和 “alias” 属性的一种备用方法。[720667]
- 编译器现在支持 GNU C “constructor” 和 “destructor” 属性上的整数优先级指示。先调用具有较低优先级的构造函数，然后调用具有较高优先级的构造函数。[722671]
- 编译器现在支持新增的 GCC “pcs” 属性，该属性用于指定带有硬件浮点的目标上的调用约定，以便作为现有 `__softfp` 关键字的一种备用方法。至于 `__softfp`，`__attribute__((pcs("aapcs-vfp")))` 使用浮点寄存器；`__attribute__((pcs("aapcs")))` 使用整数寄存器。[721049]
- 在 GNU 模式下，编译器现在实现 “`#pragma GCC system_header`”。这会导致将当前

文件中的后续声明标记为出现在系统头文件中。这可能会对某些诊断的严重性产生影响。 [723702]

- 在 GNU C++ 模式下，编译器现在可在特定情况下根据 C++ 标准要求允许省略“template”关键字。 [722869]
- 在 GNU 模式下，在函数过载处理中，指向函数的指针现在将视为与对函数的引用兼容，以便实现部分排序目的。某些在标准模式下诊断为不明确的源代码现在可在 GNU 模式下编译。 [715526]
- 选择 --gnu_defaults 模式（由 --arm_linux 表示）将启用 --allow_null_this，并在 null 对象指针上调用非虚拟成员函数时指定明确定义的行为。 [723911]
- 编译器命令行选项“--gnu_defaults”不再表示 --enum_is_int，以便与 GCC 的典型默认值保持一致。此开关仍启用 --gnu、--no_debug_macros、--signed_bitfields、--wchar32 和 --no_hide_all。 [719559]
- 已添加某些针对罕见的旧版 GCC 选项的 GCC 命令行转换。 [722785]
- 当为 makefile 生成依赖行（例如，--md 或 -M），并从标准输入中提取源代码时，对于在“”上提供相关性的输出，编译器将不会继续在这样的输出中生成行。 [723556]
- 当使用多文件编译并包括 glibc math.h 头时，可能会报告 __fabs 或 __fabsf 函数存在冲突。现在，此问题已得以解决，并且当使用 --gnu（或表示 --gnu 的其他选项）时，将不会报告任何冲突。 [722614]
- 现在，RVCT 头文件中的 arm_linux 子目录已拆分为 arm_linux 和 arm_linux_compat 子目录。arm_linux_compat 子目录现在包含与 arm_linux 子目录以前包含的 glibc 头文件保持兼容所需的头文件。在面向 ARM Linux 的模式（--arm_linux、--arm_linux_paths 以及采用 GCC 命令行转换）下，编译器将在系统包含搜索路径前面预置 arm_linux_compat 子目录，并且，编译器现在还会将 arm_linux 子目录附加到该搜索路径中。对于依赖于这些文件的确切包含顺序的特定应用程序，当从 glibc 头文件引入这些文件时，这将提高与这些应用程序的兼容性。 [723765]
- 现在，当处理具有优先级的构造函数/析构函数节时，armlink 将遵循 GNU 工具链行为。与未按优先级排列的构造函数相比，现在，按优先级排列的构造函数的排序方式与 GNU 相同，并且这些构造函数将使用 --arm_linux 开关保存。 [722673]
- 现在，使用 --arm_linux 进行链接可以抑制错误 L6654E 的出现，以避免在使用某些 GCC 编译代码进行链接时出错。 [723142]
- GNU 链接器 LD 具有一个控制脚本，用于指示链接器在 ELF 映像的各个点定义符号。以前，armlink 只定义 ARM linux 启动代码需要的符号。现在，它可定义默认控制脚本中存在的所有符号，前提条件是引用这些符号。支持的符号包括：__executable_start（仅限可执行文件）、etext、_etext、__etext、__data_start、edata、_edata、__bss_start、__bss_start__、__bss_end__、end、_end、__end、_bss_end__。此外，当存在 .dynamic 节时，链接器将始终定义符号 _DYNAMIC。 [720112]
- 向链接器中新增了“--undefined=”开关，该开关创建对指定符号的引用。例如，该开关可用于迫使链接器从库加载特定定义，并确保该定义不会被未使用节删除功能删除。创建的引用具有 STV_DEFAULT 可见性。这与相应的 GNU ld 开关的行为一致。 [719570]
- 新增了一个“--undefined_and_export=”开关。这将创建对 的 STV_DEFAULT 可见性引用，其创建方式与 --undefined= 相同。如果 定义具有 STV_DEFAULT 或

STV_PROTECTED 可见性，则还会将该定义放置到动态符号表中，就像使用了“EXPORT symbol”操作文件命令一样。如果该符号具有 STV_HIDDEN 可见性，则不会将其放置在动态符号表中，除非同时指定了“--override_visibility”。[719570]

为提高输入文件中的容错性所做的更改

- 链接器的 --[no_]muldefweak 选项的默认值已更改为 --muldefweak。如果弱符号存在多个定义，则会导致 armlink 选择其中的第一个定义，而不会生成错误。[720148]
- 现在可以降级由错误字符串合并节引起的链接器错误 6682 和 6683。降级这些错误将会禁用字符串合并。[719900]
- 现在，在链接器输入文件中进行错误重定位将生成可降级的错误，而不是致命错误。

降级以下错误将导致跳过重定位：L6217E、L6286E、L6738E、L6769E、L6781E、L6923E、L6924E。

降级以下错误将导致处理重定位，而忽略错误条件：L6237E、L6241E、L6802E、L6898E

- 现在，对于非关键输出，链接器的文件写入 (L6002U) 错误可降级为警告。非关键输出是 --feedback、--symdef、--callgraph 和 --list 选项指定的文件。[719898]
- 当对象声称支持 VFPv2 或 VFPv3 但不支持 ARM VFP 库所需的 VMOV d, r, r 指令时，可能出现链接器错误消息“L6242E: Cannot link object as its attributes are incompatible with the image attributes”。现在，在这种情况下，armlink 将引发错误“L6482E: Image Attributes have no support for VMOV d,r,r and VMOV r,r,d instructions used by C-Libraries, please use the --cpu and --fpu options to select a core that supports these instructions”。可禁用此错误消息，这将导致链接器选择使用 VFPv2 指令的浮点库。[720433]

对浮点支持的改进

- ARM Linux 现在支持硬件浮点链接。但是，默认情况下将使用软件浮点链接，在指定了隐式支持硬件浮点的 --cpu 选项的情况下也是如此。为了在面向 ARM Linux 时使用硬件浮点链接，必须使用相应的 --fpu 选项（如 --fpu=vfpv3）或 --apcs=/hardfp 明确指定此功能。请注意，ARM Linux ABI 补充 (http://www.codesourcery.com/sgpp/lite/arm/arm_gnu_linux_abi.pdf) 不支持硬件浮点链接，并且编译器将发出警告以指示这一点。[720519]
- --library_interface 命令行开关已添加以下选项：
 - aeabi_glibc_hardfp
 - aeabi_clib_hardfp
 - aeabi_clib90_hardfp
 - aeabi_clib99_hardfp
- 这些选项将导致对 C 库（包括数学库）的调用采用硬件浮点调用约定。当使用 --fpu=softvfp 编译时，使用这些库接口会发生错误。[721565]
- microlib 中的某些浮点例程的性能已得以显著提高。[678021]

一般增强功能

- 支持 Windows 7
- 编译器已添加 __smladx、__smlaldx、__smlsdx、__smlsldx、__smuadx 和 __smusdx

内部函数，并采用与现有内部函数（如 `__smlad`）相同的签名。这些内部函数生成指令的“X”变量，并交换相关操作数中的半字。[721831]

- 矢量化已得以改进，从而可以矢量化某些以前无法量化的循环。[670517]
 - 现在，当在汇编器源或嵌入式汇编器上调用选项 `--[no_]hide_all` 时，编译器会将该选项一直传递到汇编器。该选项设置使用动态链接时符号的默认可见性。[512423]
 - 现在，可以将嵌入式汇编器函数声明为静态函数。以前，将 `__asm` 函数声明为 `static` 将导致错误 #1122 或内部错误 0xafc2db（当“static”出现在单独的原型上时）。[720153]
 - 新增的编译器选项 `--[no_]rtti_data` 用于控制 C++ RTTI 数据的生成。选项 `--no_rtti` 仅禁用源级别 RTTI 功能，例如，`dynamic_cast`；而 `--no_rtti_data` 禁用源级别功能和 RTTI 数据的生成。当使用 GCC 转换时，`-fno-rtti` 表示 `--no_rtti_data`。[721182]
 - 编译器具有一个新选项 `-P`。该选项对输入进行预处理，而不会进行编译，这与 `-E` 相同，但前者不会在预处理输出中生成行标记。当预处理输出将由其他脚本或实用程序分析时，可使用此选项。[719509]
 - 编译器现在接受来自 `/dev/null` 等字符设备的输入。以前，这些设备将作为无效设备出错。请注意，通常需要 `--compile_all_input` 选项才能将此类名称视为源代码。[723224]
 - 此版本提供了一个新的链接器开关“`--export-dynamic`”，仅当映像中已存在动态段时，该开关才会导出非隐藏符号。这是对“`--export-all`”开关的补充，后者不能用于生成静态链接的映像，因为它始终会导出非隐藏符号，并强制创建动态段。[720121]
 - 现在，如果因未在命令行上指定 `--override_visibility` 开关而导致无法应用操作文件中的 `IMPORT` 或 `EXPORT` 命令，链接器则会发出警告。[677320]
 - 在创建共享库时，`--symbolic` 选项用于防止动态加载程序覆盖共享库中的全局定义。现在，`armlink` 将利用此消息，并且不会为来自同一库中的多个引用创建 PLT 条目。[721954]
 - 现在，在 `symver` 脚本和分散文件中，`armlink` 接受 C 样式的多行注释。[720472]
- RVCT 4.0 补丁（内部版本 650）包含自 RVCT 4.0（内部版本 591）以来的以下修正和改进：

编译器：

- 编译器现在包含 `--[no_]type_traits_helpers` 选项，可支持 `__is_union` 和 `__has_virtual_destructor` 等类型特征帮助器。默认情况下，这些帮助器在非 GNU C++ 模式下启用，也可在模拟 g++ 4.3 或更高版本时启用。唯一变更是：提供此选项目的是为了覆盖默认值。[721045]
- 当采用按配置优化时，为 `switch` 语句生成的代码的问题已得以解决。[720056]
- 在某些情况下，编译器将错误地重用加载基址寄存器，从而导致后续加载偏移量溢出，该问题现在已得以解决。[720891]
- 在 GNU C 模式下，“`&&a - &&b`”格式的相对标签偏移量的静态表可能包含错误数据。现在，此问题已得以解决。[721194]
- 合并嵌套循环可导致上限超出数据类型限制。在某些情况下，编译器无法避免这种情况，现在此问题已得以解决。[722205]
- 将字符数组作为 `printf` 函数的格式字符串参数传递，使用该函数的返回值将错误地导致引用符号 `_scanf_wstring`。现在，此问题已得以解决。[723780]
- 如果未使用节删除功能删除了包含对 `printf` 的调用的所有节，将不再继续链接

printf。 [667818]

- 以前, 尝试在嵌入式汇编器函数的原型上使用符号重命名关键字 `__asm("name")` 不起任何作用, 因为嵌入式汇编器将使用原始 C 名称。现在, 此问题已得以解决: 可使用提供的汇编器名称。 [721806]
- 如果在具有析构函数空体的类类型的局部变量生存期内出现了 `switch` 语句, 在这种情况下, 生成 C++ 异常可能会引发崩溃。现在, 此问题已得以解决。 [721019]
- 对于以隐式或显式方式声明内联且放置在指定 ELF 节中的 C++ 类方法, 当使用 DWARF2 调试信息 (`--debug --dwarf2`) 进行编译时, 可能会生成链接器故障。现在, 此问题已得以解决。 [723369]
- 以下内部故障的某些根源已得以解决: 0xe664d6、0xfc68b8、0x87ecef、0xafee20、0x9f28b9 和 0x0d9491 [720108、722435、722511、722766、723231 和 723245]

汇编器:

- 已提高采用 SP 作为目标寄存器对 MOV 和 MVN 指令的 A1546W 的检测准确性。 [720034]
- 现在, 带有无效模式编号的 SRS 已视为错误。可使用 `--diag_warning=1584` 获取以前的行为。 [720203]
- 对于立即移位等于数据类型大小的 NEON VSHL 和 VQSHL 指令, 汇编器不会再为这些指令接受和生成无效代码。 [719812]
- 根据诊断, 在 ARM1136-rev1 和 ARM1176 上, YIELD、WFE、WFI 和 SEV 的行为现在与 NOP 的行为相同 (诊断 1807)。 [719465]
- 已提高在反汇编时对不可预测的指令的架构诊断。 [720255]
- 根据诊断, 现在无法预测在 ARMv7-M 中的非字对齐偏移量上汇编 PC 相关 LDRD。 [720720]

链接器:

- 包含 `IMPORT *` 和 `EXPORT *` 的操作文件中的问题已得以解决。armlink 不会再生成错误消息 `Error: L6279E: Ambiguous selectors found for ('IMPORT *' and 'EXPORT *')`。 [719859]
- 链接器操作文件中的 `RENAME` 符号命令不会再重命名 `Lib$$Request` 符号条目, 因为这将中断库搜索。 [719736]
- 当选择了 `--no_veneershare` 时, armlink 不会再为分散文件错误地生成 `"Error: L6236E: No section matches selector - no section to be FIRST/LAST."`。 [720539]
- `armlink --show_cmdline` 现在将按照早期版本中的方式显示输入对象文件和库。 [720197]
- 对于同时包含 RW 和 ZI 节的执行区域, 当使用分配给这些执行区域的 `__AT` 节时, 链接器可能会丢失 ZI 节。现在, 此问题已得以解决。 [720096]
- 部分链接对象文件可能会丢失同时作为某个组的一部分的节的 SHF_STRINGS 标志。如果在后续链接中使用该部分对象, 并且选择包含该节的组, 链接器将会引发错误 L6883E。现在, 此问题已得以解决。 [720254]
- 在某些情况下, GCC 生成的 ELF 对象可重定位到索引 0 处的符号。此错误通常由对象生成器生成, 因此, 除非重定位为 `R_ARM_V4BX` 类型, 否则 armlink 将显示错误消息 L6029。如果降级此错误并进行部分链接 (或使用带有 `R_ARM_V4BX` 重定位的对象), 则会发生内部错误。现在, 此问题已得以解决。 [721112]

- 链接器现在支持选项 `--[no_]merge_partial_comdat`，该选项控制在部分链接时是否合并 SHF_COMDAT 组。[720884]
- 如果链接器因 ELF 节较大而无法将中间代码放置到距离调用站点足够近的位置，该链接器可能会无限循环，并尝试为该节生成中间代码。现在，此问题已得以解决。[720424]
- 如果输入对象包含无效 VFE 节，链接器可能发生崩溃。此问题已得以解决，现在可以禁用无效 VFE 信息的错误消息 (L6643、L6644、L6645、L6656)。如果链接器发生错误，并且已禁用此消息，链接器将关闭 VFE 优化，这与在命令行上使用 `--vfemode=off` 开关等效。[720128 和 719899]
- 如果映像中存在 TLS 重定位 R_ARM_TLS_LDM32，当使用 `--sysv`（由 `--arm_linux` 表示）进行链接时，链接器可能会低估 GOT 节的大小。当通过动态链接器解析重定位或在静态链接时通过 `armlink` 解析重定位时，这可能会引发问题。现在，此问题已得以解决。[723207]
- 如果根据包含全局数据符号的弱别名的共享库链接应用程序，并且库和应用程序均引用弱别名，则可能会出现这个问题。对于应用程序而言，链接器必须导入与弱别名位于同一地址的非弱符号，并且必须重新定义弱别名，并将其作为定义导出。这样，应用程序和库便可以使用相同的数据项副本。以前，弱别名并非从应用程序导出，因此库和应用程序使用不同的数据副本。现在，此问题已得以解决，并且 `armlink` 将导出弱别名。[720795]
- 当自动放置其基址与大小之和大于 `0xFFFFFFFF` 的 `.ARM.__AT` 节时，可能会生成 L6260E。现在，此问题已得以解决。[721272]
- 当在某些条件下使用 `microlib` 时，链接器可能会错误地生成错误“Fatal error: L6787U: Region table handler ‘__scatterload_null’ needed by entry for ER_DECOMPRESS was not found”。现在，此问题已得以解决。[723439]
- 当使用了大量输入时，由于子进程命令行长度的限制，Windows 主机上的链接时代码生成 (`--ltcg`) 有时可能会失败。现在，此问题已得以解决。
- 链接器现在可以合并其地址由 `MOVT` 和 `MOVW` 指令序列生成的字符串节。以前，链接器将生成警告 L6412W，并对包含不支持的重定位的节禁用字符串合并。[720759]

库:

- 传递给 `exit()` 或从 `main()` 返回的程序退出代码未可靠地传递给基础 `_sys_exit()` 函数。现在，此问题已得以解决。[722485]
- 当在多线程程序中使用实时堆时，如果 `malloc()` 返回失败，则不会释放堆互斥函数。现在，此问题已得以解决。[722557]
- 以前错误实现了 C++ 运行时库函数 `__aeabi_vec_ctor_cookie_nodtor`，现在，此问题已得到更正。[713271]

ELF 格式转换器 (fromelf):

- “`fromelf --elf`” 错误地删除了由链接器插入的所有文件偏移量。现在，它将保留 `p_align` ELF 域指定的文件偏移量。[720261]

此版本中的新增增强功能

新增目标支持

- Cortex-M0 处理器的核心支持 (`--cpu=Cortex-M0`)。
- i.MX51 和 OMAP4xxx 的设备支持。通过在命令行中键入 “`armcc --device list`” (无引号)，可以生成支持的设备的完整列表。

改进的浮点数学库

为了提高速度、准确性和单调性，并在需要时更有效地利用硬件浮点 (VFP)，已重新实现单精度 C 数学函数 (`sinf`、`logf`、`powf` 等)。在未采用 VFP 硬件的目标上，新实现的速度预期将提高 2-4 倍；而在采用 VFP 硬件的目标上，新实现的速度预期将提高 5-10 倍。

扩展的 gcc 兼容性

- 我们在此版本中添加了 “`--gnu_defaults`” 选项。该选项设置与 GCC 中的行为类似的基本默认行为，在 GCC 中，这些行为是大多数 ARM GCC 配置 (但不包括任何平台特定设置，例如，面向 ARM Linux 的设置) 的常见行为。具体而言，`--gnu_defaults` 选项将启用以下各项：`--gnu`、`--no_debug_macros`、`--enum_is_int`、`--signed_bitfields`、`--wchar32` 和 `--no_hide_all`。`--gnu` 选项行为保持不变；该选项启用 GCC 语言扩展，而不会更改其他设置的 `armcc` 默认值。`--arm_linux` 选项及其他面向 ARM Linux 的选项表示 `--gnu_defaults` 和适用于 Linux 的平台特定设置。
- 当针对 GCC 4.3.1 或更高版本进行配置，并使用适用于 ARM Linux 的 GCC 命令行转换时，编译器将在 `-O3` 中自动启用 NEON 矢量化 (仅限 RVDS 专业版)。您可以使用 GCC 选项 `-ftree-vectorize` 或 `-fno-tree-vectorize` 显式启用或禁用矢量化。
- 现在，头文件 `arm_linux/linux_rvct.h` 和 `arm_linux/limits.h` 将 `MB_LEN_MAX` 定义为 16 (glibc 需要的值) 而不是 6 (ABI 推荐值)。
- 使用 `--arm_linux` 时，链接器错误消息 “Error: L6029E: Relocation #REL:0 in object.o(section) is wrt invalid/missing symbol” 现在降级为警告。这可以使用含有到 NULL 的无效重定位的 GNU 对象来完成链接。

一般增强功能

ARM 编译器 (armcc):

- 数组对象析构已更新为使用更有效的 ABI 函数 `__aeabi_vec_dtor`，而不是 `__cxa_vec_dtor` 函数。
- 编译器现在将为空从属语句发出一条注释，这些语句通常由分号的错误放置触发。
- 改进了交错汇编输出中的长字符串和 64 位数据指令的格式。

链接器 (armlink):

- 增强了 `--ltdc` 选项以支持包含入口点的多个输入对象。注意：如果输入对象包含多个入口点，则必须使用 `--entry=entrypoint` 命令选择其中一个。注意，由于 `--ltdc` 选项会丢失对象与节之间的映射，因此必须使用一个全局符号定义作为您的入口点。例如：`--entry=__rt_entry`。
- 我们添加了一个新链接器选项 “`--[no_]implicit_branchpatch`” (默认启用)，用于

控制 `armlink` 是否会基于其他命令行选项自动添加补丁。例如，如果指定 `--cpu=cortex-a8`，可能隐含 `--branchpatch=cortex-a8_q109` 选项。

- 新选项 “`--[no_]keep_protected_symbols`”（默认启用）可防止删除包含具有 `STV_PROTECTED` 可见性的符号的节。
- 我们添加了禁用不匹配的 `PRES8/REQ8` 生成属性警告的功能。以前，当使用 `--arm_linux` 时，会通过有关不匹配的 `PRES8/REQ8` 生成属性的链接器降级错误进行警告，以保持与不含这些属性的 GNU 对象的兼容。现在，例如，用户可以使用 “`--diag_suppress=6238`” 禁止这些行为。
- 对于客户使其应用程序导出函数共享对象的情况，我们提供了一个选项，可防止“删除未使用的节”链接器功能删除这些所需函数。

多个工具：

- 在 Windows 上，现在这些工具可以接受 Cygwin 路径。来自 Cygwin 命令集的 “`cygpath`” 程序将作为一个过滤器予以调用，以便在必要时转换路径字符串。要启用此转换，`CYGPATH` 环境变量必须包含适当 `cygpath.exe` 的完整绝对 Windows 路径和名称，例如 “`CYGPATH=C:/cygwin/bin/cygpath.exe`”。
- 一个新开关（`--version_number`）能以一种易于解析的格式输出版本和内部版本号。
- 现在，`--apcs=/softfp` 和 `--apcs=/hardfp` 选项为请求过程调用标准的整型或浮点变体提供了另一种替代方式。这使得我们可以独立于可通过 `--fpu` 选项获得的浮点硬件版本来指定过程调用标准。我们仍可以像以前一样在 `--fpu` 选项中指定过程调用标准，但建议使用 `--apcs`。
- 这些工具现在允许 `--fpmode=none`，作为 `--fpu=none` 的一种替代选择；它表示不允许源使用任何类型的浮点运算。

此版本中所做的修正

RVCT 4.0 补丁（内部版本 591）包含自 RVCT 4.0 版本（内部版本 529）以来所做的以下修正：

编译器（`armcc`）：

- 对于 `--wchar32`，从结构初始化器中为引用生成的宽字符串文本现在放在具有 4 字节对齐的节中，从而可以确保正确对齐。
- 使用 GCC 命令行转换时，命令行上的浮点选项现在可以正确替代早期选项。
- 在某些情况下，当加载一个 64 位非对齐值，然后仅使用它的一部分时，编译器会错误地生成一个非对齐单词加载指令。现在，此问题已得以解决。
- 当使用 “`--cpu QSP`” 进行编译时，有时会错误地对 `SWP` 指令重新排序。现在，此问题已得以解决。
- 内联汇编器会错误地将具有移位操作数的 `MVN` 优化为不可求反的纯移位。现在，此问题已得以解决。
- 编译器无法为可变参数函数维护 8 字节堆栈对齐的问题已得以解决。
- 使用 `--sys_include` 与 `--preinclude` 的组合会导致编译器无法找到所包括的文件的问题已得以解决。
- 现在对于将函数的返回内容赋予某个变量的情况已经可以正确设置函数参数。

- 现在对于派生类的 RTTI 需要引用基类的 RTTI 且基类标记为 `dllimport` 的情况，除非编译器确定会从其他转换单元中导出基类 RTTI，否则它将定义基类 RTTI。
- 以下内部故障的某些根源已得以解决：0x5ac887, 0x87ecef, 0xafee20, #2409, 0xc31b6d, 0x6bca8b, 0x19c380
- 现在与循环展开优化相关的一个可能的崩溃问题已得以解决。
- 当派生类中的析构函数包含 `try` 块且基类析构函数定义为空函数时的一个可能的崩溃问题已得以解决。

C 库:

- 在用户代码中定义 `__user_heap_extend` 会导致因库中存在重复的符号定义而出现链接错误。现在，此问题已得以解决。

汇编器 (armasm):

- 现在,当使用 `--arm_only` 时,{ARCHITECTURE} 内置变量将具有所选目标架构的值。
- 现在,在 NEON 指令中可接受 `.F16` 数据类型,作为 `.16` 数据类型的替代选择。
- 现在,汇编器可正确处理具有 16 个以上单精度寄存器的 `VLDM/VSTM` 指令。
- 用于从寄存器列表中包含 LR 和 PC 的 ARM 异常状态中返回的 `LDM` 指令(例如 `LDMFD sp!, {r0-r12, lr, pc}^`)不会再生成“Warning: A1745W: This register combination is DEPRECATED”。

ELF 格式转换器 (fromelf)

- 在处理包含 `INIT_ARRAY`、`FINI_ARRAY` 或 `PREINIT_ARRAY` 节的 Linux 映像时, `fromelf --elf` 不会再报告内部故障。
- `fromelf` 现在可以正确反汇编 Thumb2EE 代码。

库管理程序 (armar):

- 现在,以替换 (r) 模式为 `armar` 提供两个同名的输入文件时,第二个文件将不会替换第一个文件。

链接器 (armlink):

- 链接器会生成额外的链接器定义符号:
 - `Load$$region_name$$ZI$$Base` Load- 此执行区域中的 ZI 输出节的地址。
 - `Load$$region_name$$ZI$$Length` Load- ZI 输出节的长度 (以字节为单位)。
 - `Load$$region_name$$ZI$$Limit`- 执行区域中超出 ZI 输出节末尾的字节的加载地址。
- 除非 `region_name` 设置了 `ZEROPAD` 分散加载执行区域属性,否则 ZI 的加载长度将为 0。如果设置了 `ZEROPAD`,则加载长度 = `Image$$region_name$$ZI$$Length`
- 当目标架构为仅 ARM 或仅 Thumb 时,如果一个交互操作节获取了一个非交互操作节的地址, `armlink` 将不会再报告错误。
- 不再为 ARM 状态符号生成“Warning L6765W : Shared object entry points must be ARM-state when linking architecture 4T objects.”。
- 现在,当同时使用 `--arm_linux`、`--ltdc` 和 `--library=` 命令行选项时, `armlink` 可以正确工作。

- 现在，在编写从执行区域中派生的第一个输出节的 `sh_addralign` 域时，可以将分散加载区域属性 `ALIGN` 考虑进去。加载区域上的 `ALIGN` 现在也将设置相关程序头的 `p_align` 域。
-

内部版本 529

RVCT 4.0 补丁（内部版本 529）包含自 RVCT 4.0（内部版本 471）以来所做的以下修正和改进：

编译器（armcc）：

- 新增了一个“`--version_number`”开关。与 `--vsu` 相比，`--version_number` 仅报告简单的 6 位编译器版本字符串（例如 400529）并且不会签出许可证。它旨在供需要检查所使用的 RVCT 版本的编译系统使用。
- 新增了一个“`--visibility_inlines_hidden`”开关。此开关可停止内联成员函数从 `__declspec(dllexport)` 类、类可见性属性、`#pragma GCC visibility` 或 `--no_hide_all` 获取动态链接（“默认”可见性）。它不会影响非成员函数。
- 当且仅当导出对应的 `vtable` 时，“`--export_all_vtbl`”开关现在会导致导出构造 `vtable` 表（`_ZTT*`）。
- 如果用户未使用 RVCT 库接口，则编译器不会再为用户代码中定义的数学函数（如 `sin()`）添加 `__hardfp_` 前缀。
- 现在可以在命令行上重新定义宏 `__FILE__`。例如，“`-D__FILE__=__MODULE__`”会导致 `__FILE__` 提供文件名但不含路径信息。
- 在完全 IEEE 模式（`--fpmode=ieee_full`）下，编译器现在设置 `Tag_ABI_FP_exceptions`。
- 在 GNU C 模式下，现在编译器将未初始化的全局常量对象放在可写存储中，并假设它们将通过其他方式（动态）初始化。这适用于诸如“`int const n;`”等定义。注意，在 `--relaxed_ref_def mode`（这是 GCC 命令行转换模式中的默认模式）中，这种定义很常见。
- 包含 `-MD` 的 GCC 命令行转换（不带由 `-MF` 开关指定的明确相关性文件位置）总是会被错误地放在当前工作目录中。
现在，相关性文件将在相同的目录中作为输出目标文件生成。
- 编译器现在支持 GCC “`__attribute__((destructor))`”。“`__attribute__((destructor))`”将对例程的引用放到 `SHT_FINI_ARRAY` 节中。“`__attribute__((constructor))`”早已得到支持。
- 现在可以正确实现变量和函数的 GCC “`weakref`”属性。
- 在针对现有 GNU 工具链为 GCC 命令行转换配置 `armcc` 方面进行了一些改进，尤其是支持为基于 Google Android 的平台配置 GCC。
- 类上的 `__declspec(dllexport)` 现在会导致类型信息名称变量（`_ZTS...`）随其他 RTTI 信息一起导出。
- 如果不使用“`dllexport`”，当在类的外部定义一个标记为“`dllexport`”的类的成员时，以前的 RVCT 4.0 补丁（内部版本 471）会导致显示一个诊断。由于在这种情况下“`dllexport`”会隐式应用于成员函数，因此进行诊断并不适当，以后不会再生成诊断。

- 现在，绝对地址 ZI 变量（AT 扩展）可以正确放在 ZI 数据节而不是 RW 数据节中。
 - 此版本进行了一些优化改进，例如：
 - 通过用 if/else 条件分支取代指向函数的指针，改进了具有长管道的处理器的性能。
 - 在 -O3 -Otime 时改进了 USAT 指令的生成。
- 修正了 __builtin_parity 内部函数的输出。以前该奇偶校验位于结果的最低有效位。现在该内部函数只返回 0 或 1 奇偶校验位。
- 通过 goto 或开关进入循环体时所发生的某些虚假编译器警告 #128-D “loop unreachable from preceding code” 现在不会再出现。
- 在极少数情况下，当一个整型子表达式将最小负数（即 INT32_MIN）除以 -1 时，会产生溢出而使编译器崩溃。现在，此问题已得以解决。
- 内联一个可返回易失性对象而返回结果又不予使用的函数会导致删除该易失性对象的加载。现在，此问题已得以解决。
- 在 Windows 上运行时，编译器会为缺失 C++ 头文件路径的 ARM Linux 生成一个配置文件。现在，此问题已得以解决。
- 在 GCC 命令行转换模式下使用大小为零的数据项（GCC 扩展），或对于更一般的 --bss_threshold=0 的情况，会导致内部故障 0x6e3f3b。现在，此问题已得以解决。这些数据项现在可以保留一个字节的零初始化存储（BSS）。
- 以下内部故障的某些根源已得以解决：0x265580, 0x6e3f3b, 0x87ecef, 0xb36758, 0xc31b6d, 0xc18024, 0xd24575, 0xdb2e1f, 0xeea2c1
- 在 Windows 上，当编译器将链接器作为子程序进行调用时，如果 armcc 的路径和传递给链接器的文件的路径都包含空格，则调用将失败并显示诸如 ‘C:\Program’ is not recognized...’ 之类的消息。现在，此问题已得以解决。

C/C++ 库：

- 为帮助链接某些应用程序和库，C 库现在将 __errno 定义为 __aeabi_errno_addr 的别名，从而提供了一种获取 errno 变量地址的方式。C 原型为 ‘int *__errno(void)’。
- 堆栈指针（SP）现在由 Microlib 在使用 RW 数据解压缩器之前进行初始化。
- atan2f() 已得到修正，当其输入参数为 +0/-0 时可返回 NaN。
- modf() 已得到修正，当第一个输入参数为 NaN 时可返回 NaN。

汇编器（armasm）：

- 使用 EXTERN 指令会导致对象文件损坏的问题已得以解决。
- 现在，在空 AREA 中定义的标签的类型将始终派生自 AREA 的类型。

链接器（armlink）：

2 armlink 能够解决 Cortex-A8 错误 657417，该错误会影响对两个 4K 区域进行编码的 32 位 Thumb-2 跳转指令，其中目标地址位于第一个 4K 区域。在以下任一情况下，将激活该解决方法：

使用 armlink 命令行开关 --branchpatch=cortex-a8_q109，或者

2 使用 armlink 命令行开关 --cpu=cortex-a8 并且正在执行一个完整链接。

2 仅对于要在 Cortex-A8 版本 r3p0 或最早期部件上运行的 Thumb-2 代码才需要使用此解决方法。有关此项错误的详细信息，请与您的芯片供应商联系。

激活后，链接器将扫描可能触发该错误行为的指令序列。如果发现这样的序列，链接器将插入一个补丁以禁止该错误行为。可以使用“--info patches”开关以获得有关应用了补丁的节中的对象、节和偏移信息。

链接器所查找的触发器条件为：

一个 4 字节直接 Thumb-2 跳转指令对区域边界进行编码，即 $(\text{address} \& 0\text{xffe}) == 0\text{xffe}$ 。

2 该跳转指令是向后的且其范围小于 4KB。

2 在该分支之前的指令是一个 4 字节非跳转指令。

2 链接器补丁将后向分支转换为指向该节末尾的补丁的前向分支。该补丁由指向原始目标的后向无条件分支组成。该补丁采取这样一种对齐方式，使得后向分支永远不会对区域边界进行编码。

这里的限制条件为：

当部分链接时，链接器不会始终获得地址信息以了解地址永远不会放在区域边界上。这意味着与在完整链接中应用补丁相比，链接器可能会在节中放入更多补丁指令。

2 需要补丁的分支必须能够抵达节末尾的补丁。如果链接器无法将补丁放在分支范围之内，它将给出一条警告消息。Thumb-2 无条件分支的范围是 16MB，条件分支的范围是 512KB，因此在实践中不可能发生这种情况。

2 链接器将增加包含 4KB 补丁的任何程序段的对齐字节数。

- 新增了一个“--use_definition_visibility”开关。默认情况下，当链接器组合全局符号时，符号的可见性将设置为所组合符号的最严格的可见性。因此，可见性为 STV_HIDDEN 的符号引用与可见性为 STV_DEFAULT 的定义的组合将获得可见性为 STV_HIDDEN 的定义。如果设置了 --use_definition_visibility，链接器在组合符号时将优先于引用的可见性使用定义的可见性。因此，可见性为 STV_HIDDEN 的符号引用与可见性为 STV_DEFAULT 的定义的组合将获得可见性为 STV_DEFAULT 的定义。如果不希望引用与共享库匹配，但希望导出定义，则这会很有用。请注意，

--use_definition_visibility 与 ELF 标准冲突，因此默认为禁用。

- 新增的“--remarks”开关可启用某些默认为禁用的诊断消息。这些额外的诊断消息将对以下情况提出警告：

使用帮助器库；完全从 4.0 版本的 armcc 构建的对象已不再需要此库。

- 在尚未使用 --apcs=/fpic 或 --apcs=/ropi 编译对象时使用 --fpic。

- 已禁用 RW 压缩的某些情况。

- 具有位于节之外的地址的映射符号（\$a、\$t、\$d...）。

- 无法通过操作文件中的 IMPORT 或 EXPORT 命令从动态符号表导入或向其导出符号。

- 使用 -sysv 运行时，armlink 在以下各项的输出 ELF 文件中添加了额外的节：

.init

- .fini

- .ARM.exidx

- .preinit_array

- .init_array

- .fini_array

- 在以前的版本中，这些节分配给 .text 或 .data 节。

- 根据相关说明，现在链接器会将未分配的节分配给 `.ANY` 执行区域，即首先将最大的节分配给具有最大可用空间的执行区域，以此类推。
 - 与弱引用和动态链接相关的链接器行为已经更改。新的行为是：
 - 对于 `--sysv`：未解析的弱引用现在可以通过操作文件中的 `IMPORT` 指令放在动态符号表中。
 - 对于 `--bpabi`：仅当通过命令行上提供的 `DLL` 定义符号时，链接器才允许将弱引用放入动态符号表中。未解析的弱引用无法通过 `IMPORT` 指令放在动态符号表中。
- 在存在匿名全局符号定义时 `ARM/Thumb` 同义词处理的问题已得以解决。
- 导致 `4.0 armlink --info=debug` 开关所报告的重复调试信息少于 `RVCT 3.1` 的问题已得以修正。
- 现在，链接器可以正确诊断恰好包含两个带有 `+FIRST` 标记的节的执行区域。
- 当映像中没有具有 `C++` 损坏名称的符号时，`armlink --match=crossmangled` 不会再崩溃。
- `armlink` 现在可以正确处理在被拒绝的 `comdat` 组中的节中不含目标符号的 `R_ARM_V4BX` 重定位。

ELF 格式转换器 (fromelf)

- “`fromelf --strip debug, symbols`” 现在可以完全剥离符号表 (`.symtab`) 和字符串表 (`.strtab`)。

内部版本 471

RVCT 4.0 补丁 (内部版本 471) 包含自 RVCT 4.0 (内部版本 400) 以来所做的以下修正和改进：

编译器 (armcc)：

- 编译器现在可以解决当基址寄存器也是第一个加载的寄存器时涉及 `LDRD` 的 Cortex-M3 错误 602117。C/C++ 库也进行了重新构建，以避免这些指令。
- 此版本对按配置优化机制进行了一些代码大小和性能方面的改进。
- 此版本对 NEON 编译进行了一些代码生成和优化改进。
- 对于 `--gnu` 和 `--arm-linux` 的编译进行了一些兼容性改进，例如：
 - 编译器现在可以在 `gcc` 转换模式下正确转换 `-mfpu=neon` 和 `-fno-inline` 选项。
 - 此版本为 GCC 浮点优化添加了更好的近似转换。
 - 以前 GCC 命令行选项的某些别名无法正确转换
 - 编译器现在可以在使用 `--library_interface=aeabi_glibc` 时正确识别 `glibc setjmp` 变体并在存在这些调用时生成正确的代码。
 - 以前，编译器不会正确允许在 `ARM Linux` 模式下取代默认符号可见性。
 - 现在实现了 `gnu __builtin_va_...` 函数
 - 基于现有 GNU 工具链为 `ARM Linux` 配置编译器时，现在可以在所有路径上检查只按名称指定的库。

- 现在可以正确转换 GNU 内置函数 `__builtin_powi`、`__builtin_powif` 和 `__builtin_powil`。
- 对编译器生成的 DWARF 调试信息进行了一些修正。
- 向 `c55x.h` 中进一步添加了一些 TI C55x 内部函数。
- 现在, 编译器可以生成代码以捕捉由使用类型 ID 运算符引发的 `bad_typeid` 异常。
- 对于标记为 `__weak` 的嵌入式汇编器 (`__asm`) 函数, 编译器不再从 C 代码中生成弱引用, 以确保不会丢失定义。在汇编器输出中仍会生成弱符号。
- 除了将具有 `.i` 扩展名的文件识别为已经预处理的 C 源代码(即应禁止进行预处理的源代码)外, 编译器现在还可以将 `.ii` 识别为表示已经过预处理的 C++ 源代码。
- 某些代码生成问题已得以解决。
- 编译器现在能够识别更多机会以使用带有 ASR 移位的 `PKHTB` 指令和带有 LSL 移位的 `PKHBT` 指令。
- 编译器现在可为 64 位数据类型的易失性访问生成 `LDRD/STRD`。
- 对类使用 `__declspec` 不会再被忽略。
- 以下内部故障的某些可能根源已得以解决:
0x15f950、0x298170、0x3dd550、0x44784c、0x4ccb3b、0x9c3725、0x974f57、0xafee20、0xeeed90、0xf28d9c。

C/C++ 库:

- `$RVCT40INC/arm_linux` 目录中为用于 ARM Linux 而提供的 `float.h` 的变体已经更改。默认情况下, 它将 `FLT_ROUNDS` 定义为 1, 表示舍入到最近的值, 这与 GCC 行为相对应。但是, 这并不严格符合 ISO C。要遵循该标准, `FLT_ROUNDS` 必须扩展到一个值随浮点环境变化的表达式。如果需要这种严格的 ISO C 行为, 我们还提供了 `FLT_ROUNDS` 的一种替代定义。通过在编译源代码时定义宏 `__ARMCC_STRICT_FLT_ROUNDS` 可启用此定义。该宏的这一变体将使用对 `fegetround()` 的调用, 因此如果使用此变体, 则必须链接 `glibc` 数学库 (`libm.so` 或 `libm.a`)。
- 以前, 为用于 ARM Linux 而提供的 `float.h` 的变体会错误地引用 `__ieee_status()`。
- `arm_neon.h` 不再将 `__const` 定义为宏, 定义为宏会禁止将其在 `--gnu` 模式下用作常量的同义词。
- 在 `$RVCT40INC/arm_linux` 中为 `glibc` 头文件 `sys/cdefs.h` 添加了一个包装器。这是为了避免使用 GCC 4.3 内置函数 `__builtin_va_arg_pack` 和 `__builtin_va_arg_pack_len`, `armcc` 不支持这两个函数。
- 在 RVCT 附带的 STL 实现中, 和 有时无法传播由分配器引发的异常。现在, 此问题已得以解决。
- 在 C++ 模式下已不再预定义 `_Bool`, 因为这会破坏使用 “`typedef bool _Bool;`” 以提供帮助的 C++ 代码。根据相关说明, 在 C 模式下仍会对其进行定义。

汇编器 (armasm):

- 为 v7-A + 安全扩展添加了一个新的 `--cpu` 选项 (“`--cpu=7-A.security`”), 在对 v7-A 架构进行汇编时, 它允许使用 SMC (以前称为 SMI) 指令。
- 在对 Thumb-2 进行汇编时, 为 IT 块中的标签错误进行降级的 `armasm --diag_warning=1722` 的行为退化问题已得以解决。
- 现在, 条件 BKPT 指令在 Thumb IT 块内会报错, 因为 BKPT 始终只能是无条件的。

- 汇编器不会再错误地将某些有效 ARM 数据处理指令诊断为不可预测或已弃用。
- 如果尚未应用 ATTR 指令，汇编器现在能够发出警告。
- 将 RELOC 用于由 EQU 定义的符号的问题已得以解决。

链接器 (armlink):

- 以前，“链接器反馈”失败且 main() 报告为未使用，因为未检测到 C 库初始化代码对 main() 的调用。现在，此问题已得以解决。
- 当节中存在非分支重定位且有多个符号位于与目标符号相同的地址时，链接器的 callgraph 输出会将某个调用的函数显示为“未使用”。现在，此问题已得以解决。
- 链接器 callgraph 现在可以正确检测出已由链接器转换为 nop 的尾部调用。以前，这些调用会被遗漏，从而导致尾部调用的函数标记为未使用。
- 在分散文件中，现在可以正确处理包含“.”的对象文件中的节，例如，“foo.bar.o”不再与“foo.o”相匹配。
- 当数据扩展到指定节以外时，链接器现在可以生成警告 L6784E。
- 链接器现在能够为分散文件中与其他没有区域表项的区域（如 OVERLAY 区域）重叠的执行区域报告错误 L6221E。
- 当输入对象中所存在的 ARM 或 Thumb 代码与派生自映像的编译属性的架构不匹配时，不会再发生 armlink 内部故障 0xb4652d。
- 在构建 Linux 应用程序时，对象中的 .note.ABI-tag SHT_NOTE 节现在按照与共享对象中的 .note.ABI-tag 节相同的方式进行处理。
- 添加了一个新的命令行选项 --gnu_linker_defined_syms。使用 --arm_linux 时，将启用此选项；否则，将默认禁用此选项。

启用此选项时，链接器将支持输入节符号的 GNU 等效项：

__start_SectionName (RVCT 等效项为 SectionName\$\$Base) - 名为 SectionName 的合并节的起始地址。

__stop_SectionName (RVCT 等效项为 SectionName\$\$Limit) - 在名为 SectionName 的合并节的末尾之后的字节地址。

请注意，一个 GNU 输入节符号对 SectionName 的引用已足以防止 armlink 将其作为未使用的节删除。而由一个 RVCT 输入节符号进行的引用则不足以防止将其作为未使用的节删除。

- 添加了一个新的命令行选项 --[no_]add_shared_references。使用 --arm_linux 时，将启用此选项；否则，将默认禁用此选项。

此选项会影响 --sysv mode 的行为。如果设置了 --add_shared_references，则在链接应用程序时，链接器将添加来自共享库的引用。如果这些引用未由应用程序或某个其他共享库定义，链接器将给出一个未定义的符号错误消息。这些引用可由静态“ar”格式库满足。

注意：来自共享库的引用只能由具有受保护的可见性或默认可见性的符号定义满足，因为只有这些符号能够导入到动态符号表中。如果符号引用由具有隐藏或内部可见性的符号解析，链接器将给出一个错误消息。

- 除非要求其重定位，否则由符号版本脚本给出其局部符号版本的默认可见性符号将不会再导出到动态符号表中。在以前的 armlink 版本中，该符号将使用本地版本导出。
- 对于程序所依赖的每个共享库，链接器都将向动态段中添加一个 DT_NEEDED 标记。程序加载器（如 ld.so）将读取这些相关性并自动加载这些共享库。库的加载顺序有时很重要，因此链接器会使 DT_NEEDED 标记的顺序与库在命令行上的显示顺序相匹配。

在 4.0 版本中，此顺序有时会出错，使得本应处于列表中位置 2 的库出现在位置 3 处。仅当位置 2 处的库需要优先于位置 3 处的库中的符号时，才需要关注这一点。此问题现在已得以解决。

ELF 格式转换器 (fromelf):

- 在比较两个具有不同长度的零初始化数据节的文件时，“fromelf --compare sections”不会再因内部故障 0x0ad3f4 而失败。
- 在处理库（包含多个对象）时会导致崩溃的 fromelf 中问题已得以解决。
- 可执行映像上的 fromelf -ec 现在可显示正确的从异常表到代码的引用偏移。