

To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
 - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.



用户手册

CubeSuite Ver. 1.00

综合开发环境

78K0R Build

目标设备

78K0R 微控制器

文档编号. U19385CA1V0UM00 (第 1 版)
发行日期 2009 年 1 月

© NEC Electronics Corporation 2008

日本印刷

[备注]

总目录

| | |
|--------------------|-----|
| 第 1 章 通用 | 20 |
| 第 2 章 功能 | 22 |
| 第 3 章 构建输出列表 | 93 |
| 第 4 章 示例程序 | 118 |
| 第 5 章 注意事项 | 122 |
| 附录 A 窗口参考 | 128 |
| 附录 B 参考命令 | 332 |
| 附录 C 索引 | 568 |

本档中所有商標或注册商標拥有各自所有权。

- 本档所登载的内容有效期截止至 2009 年 01 月，信息先于产品的生产周期发布。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本档所登载内容引发的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：

“标准等级”、“专业等级”以及“特殊等级”三种质量等级。

“特殊等级”仅适用于为特定用途而根据用户指定的质量保证程序所开发的日电电子产品。另外，各种日电电子产品的推荐用途取决于其质量等级，详见如下。用户在选用本公司的产品时，请事先确认产品的质量等级。

“标准等级”：计算机，办公自动化设备，通信设备，测试和测量设备，音频·视频设备，家电，加工机械以及产业用机器人。

“专业等级”：运输设备（汽车、火车、船舶等），交通信号控制设备，防灾装置，防止犯罪装置，各种安全装置以及医疗设备（不包括专门为维持生命而设计的设备）。

“特殊等级”：航空器械，宇航设备，海底中继设备，原子能控制系统，为了维持生命的医疗设备、用于维持生命的装置或系统等。

除在本公司半导体产品的数据表或数据手册等资料中另有特别规定以外，本公司半导体产品的质量等级均为“标准等级”。如果用户希望在本公司设计意图以外使用本公司半导体产品，务必事先与本公司销售代表联系以确认本公司是否同意为该项应用提供支持。

（注）

- （1）本声明中的“本公司”是指日本电气电子株式会社（NEC Electronics Corporation）及其控股公司。
- （2）本声明中的“本公司产品”是指所有由日本电气电子株式会社开发或制造的产品或为日本电气电子株式会社（定义如上）开发或制造的产品。

前言

本手册描述了在 78K0R 微控制器的开发应用和系统中所使用的 CubeSuite 综合开发环境，并介绍该软件的特点。CubeSuite 是 78K0R 微控制器的综合开发环境(IDE)，它集合了在软件开发阶段(如设计，执行和调试)所必须的工具在一个平台上。

通过这个综合环境，使得通过它来实现所有开发成为可能，而不需要再分开使用各种不同工具。

| | | | | | | | | | | | | | |
|----------------|--|--------|--------------------|---------|----------------------------------|-----------|---------------------|------------|---------------|------------|------|---------------|--------------------------------|
| 读者 | 本手册适用于想了解 CubeSuite 功能和设计软件及硬件应用系统的用户。 | | | | | | | | | | | | |
| 目的 | 本手册旨在硬件或软件系统开发使用这些设备，给用户在使用 Cubesuite 功能作为参考。 | | | | | | | | | | | | |
| 组成 | 本手册由如下几部分组成 第 1 章 通用 第 2 章 功能 第 3 章 构建输出列表 第 4 章 示例程序 第 5 章 注意事项 附录 A 参考窗口 附录 B 参考命令 附录 C 索引 | | | | | | | | | | | | |
| 怎样阅读本手册 | 在阅读本手册前，读者应掌握电子工程、逻辑电路和微控制器等电子工程方面的基础知识。 | | | | | | | | | | | | |
| 规定 | <table><tr><td>数据的意义:</td><td>数据的高位部分在左边，地位部分在右边</td></tr><tr><td>低态有效表示:</td><td>\overline{XXX} (在引脚或信号名称上划线表示)</td></tr><tr><td>注:</td><td>用脚标“注”来表示手册中需要注解的条目</td></tr><tr><td>注意:</td><td>表示需要特别注意的信息提示</td></tr><tr><td>备注:</td><td>补充信息</td></tr><tr><td>数字表示法:</td><td>十进制... XXXX 十六进制 ... 0xXXXX</td></tr></table> | 数据的意义: | 数据的高位部分在左边，地位部分在右边 | 低态有效表示: | \overline{XXX} (在引脚或信号名称上划线表示) | 注: | 用脚标“注”来表示手册中需要注解的条目 | 注意: | 表示需要特别注意的信息提示 | 备注: | 补充信息 | 数字表示法: | 十进制... XXXX 十六进制 ... 0xXXXX |
| 数据的意义: | 数据的高位部分在左边，地位部分在右边 | | | | | | | | | | | | |
| 低态有效表示: | \overline{XXX} (在引脚或信号名称上划线表示) | | | | | | | | | | | | |
| 注: | 用脚标“注”来表示手册中需要注解的条目 | | | | | | | | | | | | |
| 注意: | 表示需要特别注意的信息提示 | | | | | | | | | | | | |
| 备注: | 补充信息 | | | | | | | | | | | | |
| 数字表示法: | 十进制... XXXX 十六进制 ... 0xXXXX | | | | | | | | | | | | |

相关文档

本手册中提到的相关文档可能包括有初稿版本。但是，初稿版本没有特别注明。

| 文档名称 | | 文档编号 |
|--------------------------------------|----------|---------|
| CubeSuite Ver.1.00 综合开发环境 用户手册 | 启动 | U19377E |
| | 编程 | U19390E |
| | 消息 | U19391E |
| | 78K0R 编码 | U19382E |
| | 78K0R 构建 | 本手册 |
| | 78K0R 调试 | U19388E |
| | 78K0R 设计 | U19379E |

注意事项 上面所列相关文档可能会有新的版本，请确认使用最新版本的文档进行设计、开发等。

[备注]

目录

| | |
|-------------------------------------|----|
| 第 1 章 通用 | 20 |
| 1.1 概要 | 20 |
| 1.2 特征 | 21 |
| 第 2 章 功能 | 22 |
| 2.1 概要 | 22 |
| 2.1.1 创建载入模块 | 22 |
| 2.1.2 建立用户库 | 23 |
| 2.2 改变 Build tool 版本 | 24 |
| 2.3 设置构建目标文件 | 25 |
| 2.3.1 设置启动程序 | 25 |
| 2.3.2 添加文件到工程中 | 27 |
| 2.3.3 从工程中移除文件 | 31 |
| 2.3.4 从构建目标中移除文件 | 31 |
| 2.3.5 将文件归类到分项中 | 31 |
| 2.3.6 改变文件显示顺序 | 32 |
| 2.3.7 升级文件依赖关系 | 33 |
| 2.4 设置输出文件的类型 | 36 |
| 2.4.1 改变输出文件名称 | 36 |
| 2.4.2 输出汇编列表 | 37 |
| 2.4.3 输出映射信息 | 38 |
| 2.4.4 输出符号信息 | 38 |
| 2.5 设置编译选项 | 40 |
| 2.5.1 进行优化时代码大小优先。 | 40 |
| 2.5.2 进行优化时运行速度优先。 | 41 |
| 2.5.3 添加 include 路径 | 41 |
| 2.5.4 设置宏定义 | 42 |
| 2.5.5 使用 C++ 注释 | 43 |
| 2.5.6 使用兼容浮点的标准输入 / 输出函数 | 43 |
| 2.5.7 改变设置使用乘法器和除法器 / 乘法器 | 43 |
| 2.6 设置汇编选项 | 45 |
| 2.6.1 添加 include path | 45 |
| 2.6.2 设置宏定义 | 46 |
| 2.7 设置连接选项 | 48 |
| 2.7.1 添加用户库 | 48 |
| 2.8 为使用片上调试器做准备 | 50 |
| 2.9 准备执行 Boot-Flash Relink 功能 | 52 |
| 2.9.1 准备构建的目标文件 | 52 |
| 2.9.2 设置引导区工程 | 52 |
| 2.9.3 设置闪存区域工程 | 55 |
| 2.10 设置目标转换选项 | 59 |
| 2.10.1 设置 hex 文件的输出。 | 59 |
| 2.11 设置创建库选项 | 61 |
| 2.11.1 设置库文件的输出。 | 61 |

| | | |
|---------------------------|-----------------------|------------|
| 2.12 | 变量 / 函数重定位选项 | 62 |
| 2.12.1 | 有效分配的变量和函数 | 62 |
| 2.12.2 | 显示 ROM/RAM 用法 | 66 |
| 2.13 | 单独设置构建选项 | 67 |
| 2.13.1 | 在工程层设置构建选项 | 67 |
| 2.13.2 | 在文件层设置构建选项 | 67 |
| 2.14 | 设置构建操作 | 70 |
| 2.14.1 | 设置文件的链接顺序 | 70 |
| 2.14.2 | 改变子工程的文件构建顺序 | 71 |
| 2.14.3 | 显示构建选项列表 | 71 |
| 2.14.4 | 改变文件构建目标工程 | 72 |
| 2.14.5 | 添加构建模式 | 73 |
| 2.14.6 | 改变构建模式 | 74 |
| 2.14.7 | 删除构建模式 | 75 |
| 2.14.8 | 设置当前构建选项作为工程的标准 | 77 |
| 2.15 | 执行创建 | 78 |
| 2.15.1 | 运行升级文件的构建 | 80 |
| 2.15.2 | 执行所有文件的构建 | 81 |
| 2.15.3 | 运行与其他操作平行的构建。 | 81 |
| 2.15.4 | 运行构建模式中的批量构建。 | 83 |
| 2.15.5 | 编译 / 汇编 独立的文件 | 84 |
| 2.15.6 | 停止运行构建 | 85 |
| 2.15.7 | 保存构建结果到文件中 | 85 |
| 2.15.8 | 删除中间件文件和生成的文件 | 86 |
| 2.16 | 堆栈使用跟踪程序 | 87 |
| 2.16.1 | 开始和退出 | 87 |
| 2.16.2 | 检查调用关系 | 88 |
| 2.16.3 | 检查堆栈信息 | 89 |
| 2.16.4 | 检查未知函数 | 90 |
| 2.16.5 | 改变帧大小 | 91 |
| 第 3 章 构建输出列表 | | 93 |
| 3.1 | C 编译器 | 93 |
| 3.1.1 | 汇编程序源文件 | 93 |
| 3.1.2 | 出错列表文件 | 96 |
| 3.1.3 | 预处理列表文件 | 98 |
| 3.1.4 | 交叉引用列表文件 | 100 |
| 3.2 | 汇编程序 | 102 |
| 3.2.1 | 汇编列表文件头 | 102 |
| 3.2.2 | 汇编列表 | 103 |
| 3.2.3 | 符号列表 | 104 |
| 3.2.4 | 交叉引用列表 | 105 |
| 3.2.5 | 错误列表 | 107 |
| 3.3 | 链接器 | 107 |
| 3.3.1 | 链接列表文件头 | 108 |
| 3.3.2 | Map 列表 | 109 |
| 3.3.3 | 公用符号列表 | 110 |
| 3.3.4 | 局部符号列表 | 111 |
| 3.3.5 | 错误列表 | 112 |
| 3.4 | 目标转换器 | 112 |
| 3.4.1 | 错误列表 | 113 |

| | |
|----------------------------------|------------|
| 3.5 库管理程序 | 113 |
| 3.5.1 库信息输出列表 | 113 |
| 3.6 列表转换器 | 114 |
| 3.6.1 完全汇编列表 | 114 |
| 3.6.2 错误列表 | 114 |
| 3.7 变量 / 函数信息文件生成器 | 114 |
| 3.7.1 变量 / 函数信息文件说明 | 114 |
| | |
| 第 4 章 示例程序 | 118 |
| | |
| 4.1 C 编译器 | 118 |
| 4.1.1 C 源文件 | 118 |
| 4.2 汇编程序 | 120 |
| 4.2.1 k0rmain.asm | 120 |
| 4.2.2 k0rsub.asm | 121 |
| | |
| 第 5 章 注意事项 | 122 |
| | |
| 附录 A 窗口参考 | 128 |
| | |
| A.1 说明 | 128 |
| | |
| 附录 B 参考命令 | 332 |
| | |
| B.1 C 编译器 | 332 |
| B.1.1 I/O 文件 | 332 |
| B.1.2 功能 | 334 |
| B.1.3 操作方法 | 336 |
| B.1.4 选项 | 340 |
| B.2 汇编程序 | 385 |
| B.2.1 I/O 文件 | 385 |
| B.2.2 功能 | 386 |
| B.2.3 操作方法 | 386 |
| B.2.4 选项 | 389 |
| B.3 链接器 | 430 |
| B.3.1 I/O 文件 | 430 |
| B.3.2 功能 | 431 |
| B.3.3 操作方法 | 431 |
| B.3.4 选项 | 434 |
| B.3.5 Boot-flash relink 功能 | 476 |
| B.4 目标转换器 | 489 |
| B.4.1 I/O 文件 | 489 |
| B.4.2 功能 | 490 |
| B.4.3 操作方法 | 502 |
| B.4.4 选项 | 505 |
| B.5 库管理程序 | 520 |
| B.5.1 I/O 文件 | 520 |
| B.5.2 功能 | 520 |
| B.5.3 操作方法 | 522 |
| B.5.4 选项 | 525 |
| B.5.5 子命令 | 533 |

| | | |
|--------------|-----------------------|------------|
| B.6 | 列表转换器 | 542 |
| B.6.1 | I/O 文件 | 542 |
| B.6.2 | 功能 | 543 |
| B.6.3 | 操作方法 | 546 |
| B.6.4 | 选项 | 547 |
| B.7 | 变量 / 函数信息文件生成器 | 555 |
| B.7.1 | I/O 文件 | 555 |
| B.7.2 | 功能 | 556 |
| B.7.3 | 变量 / 函数信息 | 556 |
| B.7.4 | 操作方法 | 560 |
| B.7.5 | 选项 | 563 |
| | | |
| 附录 C | 索引 | 568 |

插图列表

| 插图编号 | 标题 | 页码 |
|------|---|----|
| 1-1 | Build tool 处理流程 | 20 |
| 2-1 | [Version Select] 分项 | 24 |
| 2-2 | [Use standard startup routine] 属性 | 25 |
| 2-3 | [Use standard startup routine] 属性 | 25 |
| 2-4 | 工程树面板 (在添加启动文件之后) | 25 |
| 2-5 | Project Tree Panel (文件放入位置) | 27 |
| 2-6 | 添加文件夹和文件对话框 | 28 |
| 2-7 | 添加文件对话框 | 29 |
| 2-8 | 工程树面板 (在添加文件 "main.c" 之后) | 29 |
| 2-9 | 工程树面板 (在添加文件夹 "src" 之后) | 30 |
| 2-10 | [Remove from Project] 条目 | 31 |
| 2-11 | [Set as build-target] 属性 | 31 |
| 2-12 | [Add New Category] 条目 (用于 File 节点) | 32 |
| 2-13 | 工程树面板 (在 Adding Category 节点之后) | 32 |
| 2-14 | 工具栏 (工程树面板) | 32 |
| 2-15 | [Update Dependencies] 条目 | 33 |
| 2-16 | [Update Dependencies of active project] 条目 | 34 |
| 2-17 | [Output file type] 属性 | 36 |
| 2-18 | [Output file name] 属性 (用于载入模块文件) | 36 |
| 2-19 | [Hex file name] 属性 | 37 |
| 2-20 | [Output file name] 属性 (库文件) | 37 |
| 2-21 | [Output assemble list file] 属性 | 37 |
| 2-22 | [Link List] 分项 (用于映射信息) | 38 |
| 2-23 | [Link List] 分项 (用于本地符号信息) | 38 |
| 2-24 | [Link List] 分项 (用于公用符号信息) | 39 |
| 2-25 | 属性面板: [Compile Options] 标签 | 40 |
| 2-26 | [Perform optimization] 属性 (代码大小优先) | 40 |
| 2-27 | [Perform optimization] 属性 (执行速度优先) | 41 |
| 2-28 | [Additional include paths] 属性 | 41 |
| 2-29 | 路径编辑对话框 | 41 |
| 2-30 | [Additional include paths] 属性 (在添加 include path 之后) | 42 |
| 2-31 | [Macro definition] 属性 | 42 |
| 2-32 | 文本编辑对话框 | 42 |
| 2-33 | [Macro definition] 属性 (在设置宏之后) | 42 |
| 2-34 | [Allow C++ format comments] 属性 | 43 |
| 2-35 | [Use standard I/O library supported floating-point data] 属性 | 43 |
| 2-36 | [Use multiplier] 属性 | 44 |
| 2-37 | 属性面板: [Assemble Options] 标签 | 45 |
| 2-38 | [Additional include paths] 属性 | 45 |
| 2-39 | 路径编辑对话框 | 46 |

| | | |
|------|---|----|
| 2-40 | [Additional include paths] 属性 (在添加 include path 之后) | 46 |
| 2-41 | [Macro definition] 属性 | 46 |
| 2-42 | 文本编辑对话框 | 47 |
| 2-43 | [Macro definition] 属性 (在设置宏之后) | 47 |
| 2-44 | 属性面板 : [Link Options] 标签 | 48 |
| 2-45 | [Using libraries] 属性 | 48 |
| 2-46 | 文本编辑对话框 | 49 |
| 2-47 | [Using libraries] 属性 (在设置库文件之后) | 49 |
| 2-48 | [Device] 分项 | 50 |
| 2-49 | [Device] 分项 | 50 |
| 2-50 | [Security ID] 属性 | 51 |
| 2-51 | 引导区域工程 | 52 |
| 2-52 | 在引导区域的 [Output folder for variables/functions information file] 属性 | 53 |
| 2-53 | 在引导区域中的 [Memory Model] 分项 | 53 |
| 2-54 | 在引导区域的 [Use standard startup routine] 属性 | 53 |
| 2-55 | 在引导区域中的 [Device] 分项 | 54 |
| 2-56 | 在引导区域的 [Split hex file] 属性 | 54 |
| 2-57 | 为引导区域创建文件 | 55 |
| 2-58 | 闪存区工程 | 55 |
| 2-59 | 在 Flash 区域的 [Output folder for variables/functions information file] 属性 | 56 |
| 2-60 | 在 Flash 区域中的 [Memory Model] 分项 | 56 |
| 2-61 | 在闪存区域的 [Use standard startup routine] 属性 | 56 |
| 2-62 | 在 Flash 区域的 [Variables/functions information file for boot area] 属性 | 56 |
| 2-63 | 在闪存区的 [Boot area load module file name] 属性 | 57 |
| 2-64 | 在闪存区域的 [Split hex file] 属性 | 57 |
| 2-65 | 为闪存区域创建文件 | 58 |
| 2-66 | 属性面板 : [Object Convert Options] 标签 | 59 |
| 2-67 | [Hex File] 分项 | 59 |
| 2-68 | 属性面板 : [Create Library Options] 标签 | 61 |
| 2-69 | [Output File] 分项 | 61 |
| 2-70 | 属性面板 : [Memory Bank Relocation Options] 标签 | 62 |
| 2-71 | [Output variables/functions information file] 属性 | 62 |
| 2-72 | 工程树面板 (在生成变量 / 函数信息文件之后) | 63 |
| 2-73 | 工程树面板 (在生成载入模块文件之后) | 64 |
| 2-74 | [Output File] 分项 | 65 |
| 2-75 | [Output ROM/RAM usage] 属性 | 66 |
| 2-76 | ROM/RAM 用法显示 | 66 |
| 2-77 | [Set individual compile option] 属性 | 67 |
| 2-78 | 消息对话框 | 67 |
| 2-79 | 属性面板 : [Individual Compile Options] 标签 | 68 |
| 2-80 | [Set individual assemble option] 属性 | 68 |
| 2-81 | 消息对话框 | 68 |
| 2-82 | 属性面板 : [Individual Assemble Options] 标签 | 69 |

| | |
|---|-----|
| 2-83 链接顺序对话框 | 70 |
| 2-84 [Format of build option list] 属性 | 71 |
| 2-85 [Set selected project as Active Project] 条目 | 72 |
| 2-86 活动工程 | 72 |
| 2-87 构建模式设置对话框 | 73 |
| 2-88 字符串输入对话框 | 73 |
| 2-89 构建模式设置对话框（在添加构建模式之后） | 74 |
| 2-90 [Build Mode] 属性 | 74 |
| 2-91 构建模式设置对话框 | 75 |
| 2-92 构建模式设置对话框 | 75 |
| 2-93 消息对话框 | 76 |
| 2-94 属性面板（在改变标准构建选项之后） | 77 |
| 2-95 [Set to Default Build Option for Project] 条目 | 77 |
| 2-96 属性面板（在设置标准构建选项之后） | 77 |
| 2-97 构建执行结果（构建、重新构建或增量创建） | 78 |
| 2-98 构建执行结果（快速构建） | 78 |
| 2-99 Build tool 生成文件 | 79 |
| 2-100 [Build active project] 条目 | 80 |
| 2-101 [Rebuild active project] 条目 | 81 |
| 2-102 [Rapid Build] 条目（当快速构建有效时） | 82 |
| 2-103 [Rapid Build] 条目（当快速构建无效时） | 82 |
| 2-104 批量构建对话框 | 83 |
| 2-105 [Compile] 条目 | 84 |
| 2-106 [Assemble] 条目 | 84 |
| 2-107 另存为对话框 | 85 |
| 2-108 [Clean Project] 条目 | 86 |
| 2-109 [Clean active project] 条目 | 86 |
| 2-110 启动堆栈使用跟踪程序 | 87 |
| 2-111 树型显示区 | 88 |
| 2-112 列表显示区 | 89 |
| 2-113 堆栈大小未知 / 校准函数列表对话框 | 90 |
| 2-114 调整栈堆大小对话框 | 91 |
| 2-115 堆栈大小设定文件举例 | 92 |
| A-1 主窗口 | 130 |
| A-2 工程树面板 | 134 |
| A-3 属性面板 | 145 |
| A-4 属性面板:[Common Options] 标签 | 149 |
| A-5 属性面板:[Compile Options] 标签 | 159 |
| A-6 属性面板:[Assemble Options] 标签 | 178 |
| A-7 属性面板:[Link Options] 标签 | 184 |
| A-8 属性面板:[Object Convert Options] 标签 | 193 |
| A-9 属性面板:[Create Library Options] 标签 | 198 |
| A-10 属性面板:[Memory Bank Relocation Options] 标签 | 201 |

| | | |
|------|--|-----|
| A-11 | 属性面板 : [Build Settings] 标签 (在选定 C 源文件时) | 203 |
| A-12 | 属性面板 : [Build Settings] 标签 (在选定汇编源文件时) | 203 |
| A-13 | 属性面板 : [Build Settings] 标签 (在选定链接指令文件时) | 203 |
| A-14 | 属性面板 : [Build Settings] 标签 (在选定变量 / 函数信息文件时) | 204 |
| A-15 | 属性面板 : [Build Settings] 标签 (在选定功目标文件时) | 204 |
| A-16 | 属性面板 : [Build Settings] 标签 (在选定功库文件时) | 204 |
| A-17 | 属性面板 : [Individual Compile Options] 标签 | 206 |
| A-18 | 属性面板 : [Individual Assemble Options] 标签 | 220 |
| A-19 | 属性面板 : [File Information] 标签 | 227 |
| A-20 | 属性面板 : [Category Information] 标签 | 229 |
| A-21 | 编辑器面板 | 230 |
| A-22 | 输出面板 | 233 |
| A-23 | 添加文件对话框 | 236 |
| A-24 | 添加文件夹和文件对话框 | 238 |
| A-25 | 字符串输入对话框 | 240 |
| A-26 | 文本编辑对话框 | 242 |
| A-27 | 路径编辑对话框 | 244 |
| A-28 | 链接顺序对话框 | 246 |
| A-29 | 构建模式设置对话框 | 248 |
| A-30 | 批量构建对话框 | 250 |
| A-31 | 查找和替换对话框 | 252 |
| A-32 | 查找和替换对话框: [Quick Search] 标签 | 253 |
| A-33 | 查找和替换对话框: [Whole Search] 标签 | 256 |
| A-34 | 查找和替换对话框: [Quick Search] 标签 | 259 |
| A-35 | 查找和替换对话框: [Whole Replace] 标签 | 262 |
| A-36 | 跳转定位对话框 | 265 |
| A-37 | 进度状态对话框 | 266 |
| A-38 | 选项对话框 | 267 |
| A-39 | 选项对话框 ([General - Startup and Exit] 分项) | 269 |
| A-40 | 选项对话框 ([General - Display] 分项) | 271 |
| A-41 | 选项对话框 ([General - Text Editor] 分项) | 273 |
| A-42 | 选项对话框 ([General - Font and Color] 分项) | 275 |
| A-43 | 颜色编辑对话框 | 278 |
| A-44 | 字体对话框 | 278 |
| A-45 | 选项对话框 ([General - External Tools] 分项)。 | 280 |
| A-46 | 选项对话框 ([General - Build/Debug] 分项) | 284 |
| A-47 | 选项对话框 ([General - Update] 分项) | 286 |
| A-48 | 选项对话框 ([Other - User Information] 分项) | 288 |
| A-49 | 用户设置对话框 | 290 |
| A-50 | 用户设置对话框 : [Toolbars] 标签 | 292 |
| A-51 | 用户设置对话框 : [Commands] 标签 | 294 |
| A-52 | 新建工具栏对话框 | 296 |
| A-53 | 重命名工具栏对话框 | 297 |

| | | |
|------|---|-----|
| A-54 | 自定义键盘对话框 | 298 |
| A-55 | 重排命令对话框 | 300 |
| A-56 | Add Command Dialog Box | 301 |
| A-57 | 添加现有的文件对话框 | 303 |
| A-58 | 文件夹浏览对话框 | 305 |
| A-59 | 设定启动区变量 / 函数信息文件对话框 | 307 |
| A-60 | 设定引导区载入模块文件对话框 | 309 |
| A-61 | 另存为对话框 | 311 |
| A-62 | 打开程序对话框 | 313 |
| A-63 | 选择程序对话框 | 315 |
| A-64 | 选择外部文本编辑器对话框 | 317 |
| A-65 | 堆栈用法跟踪窗口 | 319 |
| A-66 | 堆栈大小未知 / 校准函数列表对话框 | 326 |
| A-67 | 调整栈堆大小对话框 | 328 |
| A-68 | 打开对话框 | 331 |
| B-1 | C 编译器的 I/O 文件 | 332 |
| B-2 | 属性面板: [Compile Option] 标签 | 339 |
| B-3 | 汇编程序的 I/O 文件 | 385 |
| B-4 | 属性面板: [Assemble Option] 标签 | 389 |
| B-5 | 链接器的 I/O 文件 | 430 |
| B-6 | 属性面板: [Link Option] 标签 | 434 |
| B-7 | 在引导区域 | 477 |
| B-8 | 在闪存区域 | 477 |
| B-9 | 从引导区域到闪存区域 | 478 |
| B-10 | 从闪存区域到引导区域 | 479 |
| B-11 | 引导区域工程 | 482 |
| B-12 | 在引导区域的 [Output folder for variables/functions information file] 属性 | 482 |
| B-13 | 在引导区域中的 [Memory Model] 分项 | 483 |
| B-14 | 在引导区域的 [Use standard startup routine] 属性 | 483 |
| B-15 | 在引导区域中的 [Device] 分项 | 483 |
| B-16 | 在引导区域的 [Split hex file] 属性 | 483 |
| B-17 | 为引导区域创建文件 | 484 |
| B-18 | 闪存区工程 | 484 |
| B-19 | 在 Flash 区域的 [Output folder for variables/functions information file] 属性 | 485 |
| B-20 | 在 Flash 区域中的 [Memory Model] 分项 | 485 |
| B-21 | 在闪存区域的 [Use standard startup routine] 属性 | 485 |
| B-22 | 在 Flash 区域的 [Variables/functions information file for boot area] 属性 | 486 |
| B-23 | 在闪存区的 [Boot area load module file name] 属性 | 486 |
| B-24 | 在闪存区域的 [Split hex file] 属性 | 486 |
| B-25 | 为闪存区域创建文件 | 487 |
| B-26 | 目标转换器的 I/O 文件 | 489 |
| B-27 | Intel 标准格式 | 490 |
| B-28 | Intel 扩展格式 | 491 |

| | |
|--|-----|
| B-29 Motorola S- 类型格式 | 497 |
| B-30 符号表格文件格式 | 501 |
| B-31 符号值的格式 | 502 |
| B-32 属性面板: [Object Convert Option] 标签 | 505 |
| B-33 库管理程序的 I/O 文件 | 520 |
| B-34 属性面板: [Create Library Options] 标签 | 524 |
| B-35 列表转换器的 I/O 文件 | 542 |
| B-36 变量 / 函数信息文件生成器的 I/O 文件 | 555 |
| B-37 78K0R 的存储映像 | 557 |
| B-38 属性面板: [Memory Bank Relocation Options] 标签 | 562 |

表格列表

| 表格编号 | 标题 | 页码 |
|------|----------------------------------|-----|
| 2-1 | 构建类型 | 78 |
| A-1 | 窗口 / 面板 / 对话框的列表 | 128 |
| B-1 | C 编译器的 I/O 文件 | 332 |
| B-2 | 优化方法 | 334 |
| B-3 | 编译选项 | 340 |
| B-4 | 编译选项的优先级 | 341 |
| B-5 | 汇编程序的 I/O 文件 | 385 |
| B-6 | 汇编选项 | 389 |
| B-7 | 汇编选项的优先级 | 390 |
| B-8 | 链接器的 I/O 文件 | 430 |
| B-9 | 链接选项 | 434 |
| B-10 | 编译选项的优先级 | 436 |
| B-11 | 目标转换器的 I/O 文件 | 489 |
| B-12 | 当 -zf 选项设定时的文件类型 | 490 |
| B-13 | 扩展 Tektronix 头文件 | 493 |
| B-14 | 扩展 Tektronix 的数据块格式 | 494 |
| B-15 | 扩展 Tektronix 的结束块格式 | 495 |
| B-16 | Symbol 扩展 Tektronix 的符号块特征 | 496 |
| B-17 | 扩展 Tektronix 的符号块格式 | 496 |
| B-18 | 扩展 Tektronix 的符号块片断定义字符段 | 496 |
| B-19 | 扩展 Tektronix 的符号块符号定义字符段 | 497 |
| B-20 | Motorola Hex 文件的记录类型 | 497 |
| B-21 | 每个记录的通用格式 | 497 |
| B-22 | 字符段的意义 | 498 |
| B-23 | 目标转换选项 | 505 |
| B-24 | 库管理程序的 I/O 文件 | 520 |
| B-25 | 创建库选项 | 525 |
| B-26 | 子命令 | 533 |
| B-27 | 列表转换器的 I/O 文件 | 542 |
| B-28 | 列表转换选项 | 548 |
| B-29 | 变量 / 函数信息文件生成器的 I/O 文件 | 555 |
| B-30 | 变量 / 功能重定位选项 | 563 |

第 1 章 通用

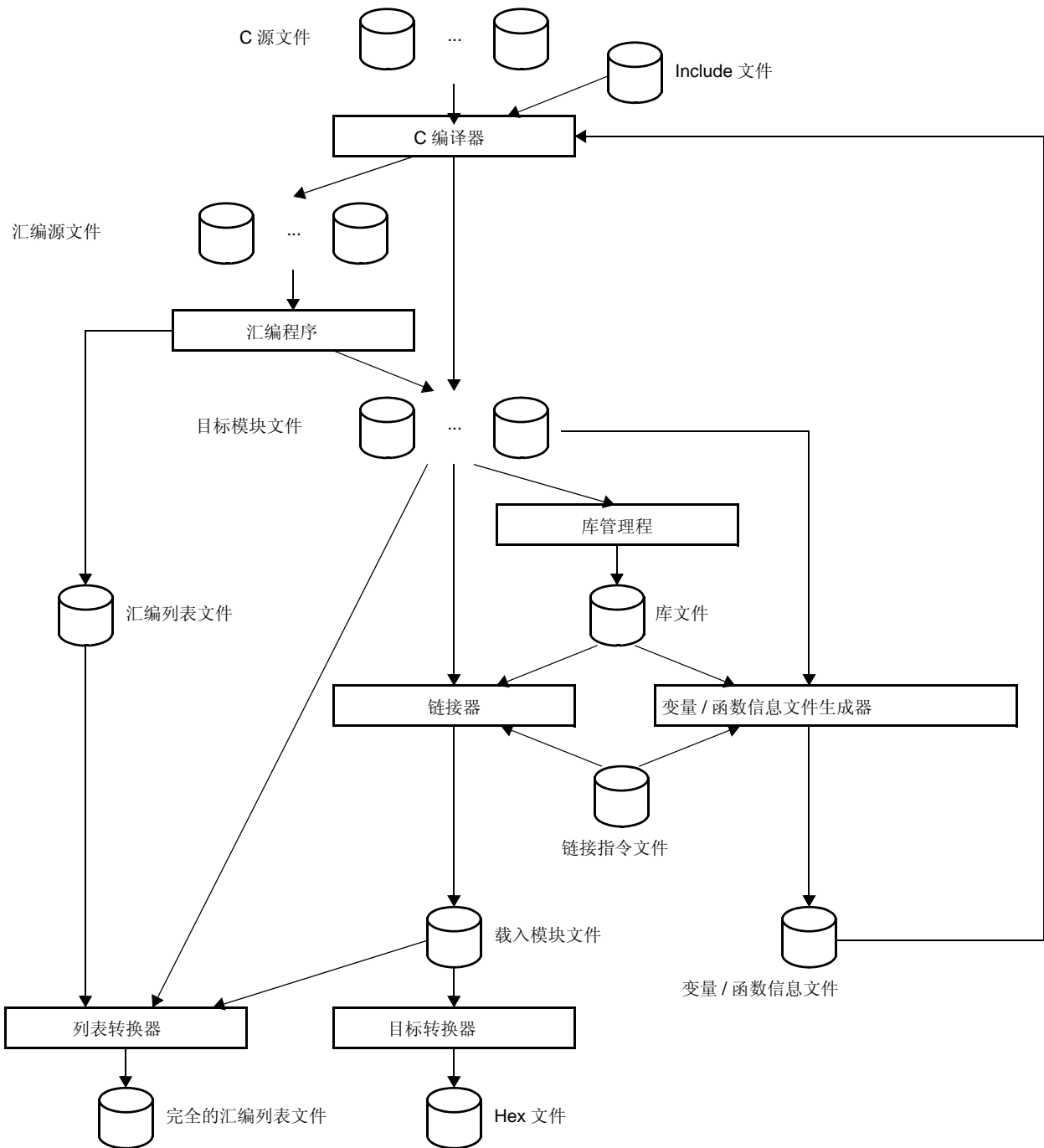
本章节介绍 build tool 的产品概要。

1.1 概要

Build tool 包括由 CubeSuite 提供的各部件。它通过 GUI 工具来配置不同类型的信息根据您的目标文件，能够从源文件生成载入模块文件，hex 文件或者库文件。

Build tool 的处理流程如下所示。

图 1-1. Build tool 处理流程



1.2 特征

Build tool 的特征如下所示。

- 优化功能
汇编时可通过执行优化生成高效的对象模块文件，例如优化代码大小或执行速度。

- ROM 化功能
ROM 化是指在系统执行时，复制存储在 ROM 中的外部变量的初始值到 RAM 中的处理过程。
C 编译器为启动程序提供了 ROM 化处理过程，您不需要在启动时书写这一部分代码。

- 宏功能
当在汇编源文件中多次写相同的指令时，可以将指令定义为一个具有单一名称的宏。

第 2 章 功能

本章节介绍使用 CubeSuit 构建的步骤以及主构建功能。

2.1 概要

本节介绍如何创建加载模块和用户库。

2.1.1 创建载入模块

从建立工程到创建载入模块的流程如下所示。

(1) 创建及载入工程

创建新工程，或载入已存在的工程。

备注 关于创建新工程或载入已存在的工程的详细说明，请参阅 CubeSuite 启动用户手册。

(2) 设置构建目标工程

当设定或执行构建时，要首先设置活动工程（见“[2.14 设置构建操作](#)”）。如果没有子工程，则当前工程总是处于活动中。

备注 当设置构建模式时，需要改变构建模式（见“[2.14.6 改变构建模式](#)”）。

(3) 设置构建目标文件

为工程添加或移除构建文件且升级附属文件（见“[2.3 设置构建目标文件](#)”）。

备注 1. 关于添加用户库到工程中的方法见“[2.7.1 添加用户库](#)”。

2. 另外，可以设置目标模块文件的和库文件的连接顺序（见“[2.14.1 设置文件的链接顺序](#)”）。

(4) 设定载入模块文件的输出。

设置载入模块文件的输出作为构建的生成物（见“[2.4 设置输出文件的类型](#)”）。

(5) 设置构建选项

设置编译器，汇编程序，连接器等选项（见“[2.5 设置编译选项](#)”，“[2.6 设置汇编选项](#)”，“[2.7 设置连接选项](#)”）。

(6) 执行创建

运行构建（见“[2.15 执行创建](#)”）。

有下列构建类型。

- 构建（参见“[2.15.1 运行升级文件的构建](#)”）
- 重新构建（参见“[2.15.2 执行所有文件的构建](#)”）
- 快速构建（参见“[2.15.3 运行与其他操作平行的构建](#)。”）
- 批量构建（参见“[2.15.4 运行构建模式中的批量构建](#)。”）

备注 如果在构建过程之前或之后要运行其它命令，在属性面板上，[\[Common Options\]](#) 标签的 [\[Others\]](#) 分
项中，设置 [\[Commands executed before build processing\]](#) 和 [\[Commands executed after build
processing\]](#) 属性。

以文件为单位，如果在构建过程之前或之后运行其它命令，可以从 [\[Individual Compile Options\]](#) 标签（
用于 C 源文件）和 [\[Individual Assemble Options\]](#) 标签（用于汇编源文件）中进行设置。

(7) 保存工程

保存工程的设置信息到工程文件中。

备注 关于保存工程的详细内容可以参阅 **CubeSuite** 启用用户手册。

2.1.2 建立用户库

从设置工程到创建用户库的流程如下所示。

(1) 创建及载入工程

创建新工程，或载入已存在的工程。

当建立新的工程时，设置库工程。

备注 关于创建新工程或载入已存在的工程的详细说明，请参阅 **CubeSuite** 启动用户手册。

(2) 设置构建目标工程

当设定或执行构建时，要首先设置活动工程（见“[2.14 设置构建操作](#)”）。

如果没有子工程，则当前工程总是处于活动中。

备注 当设置构建模式时，需要改变构建模式（见“[2.14.6 改变构建模式](#)”）。

(3) 设置构建目标文件

为工程添加或移除构建文件且升级附属文件（见“[2.3 设置构建目标文件](#)”）。

(4) 设置构建选项

设置编译器，汇编程序，库管理程序等选项（见“[2.5 设置编译选项](#)”，“[2.6 设置汇编选项](#)”，“[2.11 设置创建库选项](#)”）。

(5) 运行构建

运行构建（见“[2.15 执行创建](#)”）。

有下列构建类型。

- 构建（参见“[2.15.1 运行升级文件的构建](#)”）
- 重新构建（参见“[2.15.2 执行所有文件的构建](#)”）
- 快速构建（参见“[2.15.3 运行与其他操作平行的构建](#)。”）
- 批量构建（参见“[2.15.4 运行构建模式中的批量构建](#)。”）

备注 如果在构建过程之前或之后要运行其它命令，在属性面板上，[\[Common Options\]](#) 标签的 [\[Others\]](#) 分项中，设置 [\[Commands executed before build processing\]](#) 和 [\[Commands executed after build processing\]](#) 属性。

以文件为单位，如果在构建过程之前或之后运行其它命令，可以从 [\[Individual Compile Options\]](#) 标签（用于 C 源文件）和 [\[Individual Assemble Options\]](#) 标签（用于汇编源文件）中进行设置。

(6) 保存工程

保存工程的设置信息到工程文件中。

备注 关于保存工程的详细内容可以参阅 **CubeSuite** 启用用户手册。

2.2 改变 Build tool 版本

可以改变在工程（主工程或子工程）中使用的 build tool 的版本（编译包）。

在工程树上选择 build tool 节点并在属性面板上选择 [Common Options] 标签。选择 [Always latest version which was installed] 或在 [Version Select] 分项中的 [Using compiler package version] 属性上的版本。

图 2-1. [Version Select] 分项



| Version Select | |
|---|---|
| Using compiler package install folder | C:\Program Files\NSEC Electronics |
| Using compiler package version | Always latest version which was installed |
| Latest compiler package version which was installed | V1.00 |

备注 当在主工程和子工程中使用的构建工具相同时，可以通过选择所有的 Build tool 节点和设置属性来改变 Build tool 的版本。

2.3 设置构建目标文件

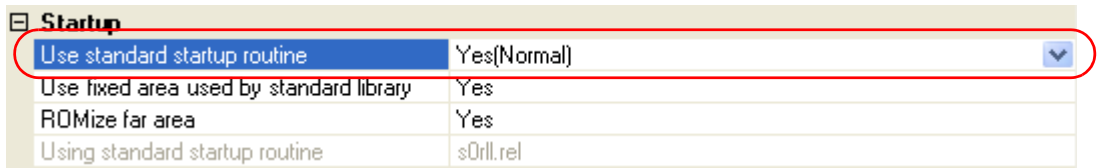
在运行构建之前，必须在工程中添加构建目标文件（例如 C 源文件，汇编源文件）。
本章节介绍在工程中的设置文件的操作。

2.3.1 设置启动程序

(1) 使用标准启动程序

在工程树上选择 build tool 节点并在属性面板上选择 [Compile Options] 标签。
使用标准的启动程序，在 [Startup] 分项中的 [Use standard startup routine] 属性上选择 [Yes(Normal)]/[Yes(For boot area)]/[Yes(For flash area)]。

图 2-2. [Use standard startup routine] 属性

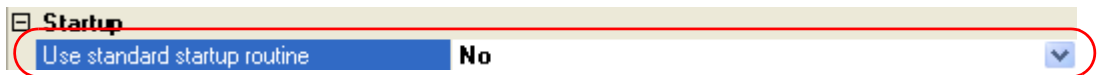


已使用标准启动程序的目标文件名称将显示在 [Using standard startup routine] 属性上。

(2) 不使用标准启动程序

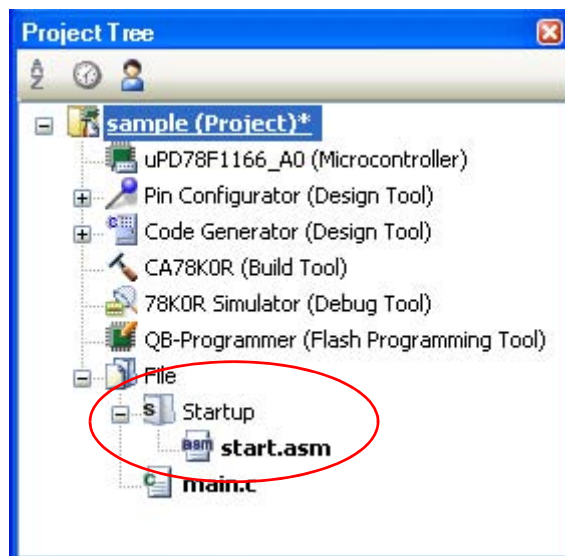
在工程树上选择 build tool 节点并在属性面板上选择 [Compile Options] 标签。
使用除标准启动程序以外的程序，在 [Startup] 分项中的 [Use standard startup routine] 属性上选择 [No]([Yes(Normal)] 被选择为默认)。

图 2-3. [Use standard startup routine] 属性



接下来，添加启动文件（描述启动程序的文件）到工程树上的 Startup 节点。关于添加文件到工程树中的方法见 "2.3.2 添加文件到工程中"。

图 2-4. 工程树面板（在添加启动文件之后）



注意事项 构建目标文件作为启动文件直接添加到工程树上的 **Startup** 节点之下。如果添加到 **Startup** 节点下的分项中时，则不能当作启动文件。
当添加启动文件到 **Startup** 节点时，如果之前启动文件已经添加则构建仅指向最后添加的启动文件；任何之前添加的文件将不能指向。
设置启动文件时，如其没有作为构建目标被指向，而其它启动文件已添加，则构建指向该文件，且不指向其它文件。

备注 见 CubeSuite 78K0 编码用户手册有关建立启动程序的方法。

2.3.2 添加文件到工程中

文件可以通过以下方法添加到工程中。

- 添加存在的文件
- 建立和添加空文件

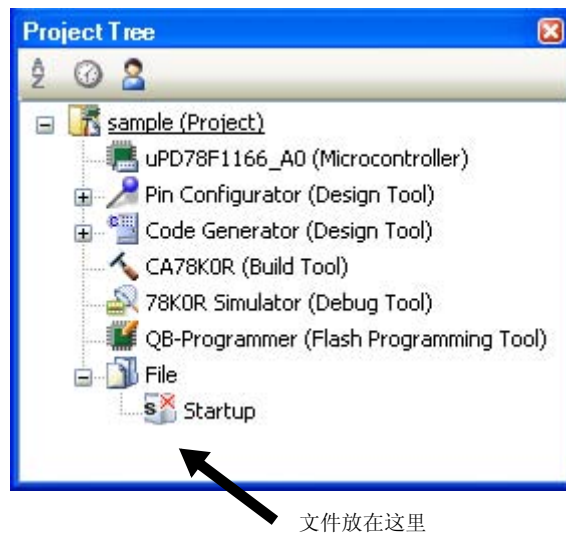
(1) 添加存在的文件

(a) 添加独立的文件

从浏览器中拖动一个文件夹，并放置于工程树下的空白区域。

文件添加到 File 节点下。

图 2-5. Project Tree Panel (文件放入位置)



注意事项 添加启动程序，放入文件在所 **Startup** 节点中。见 "[2.3.1 设置启动程序](#)" 有关使用启动程序的详细介绍。

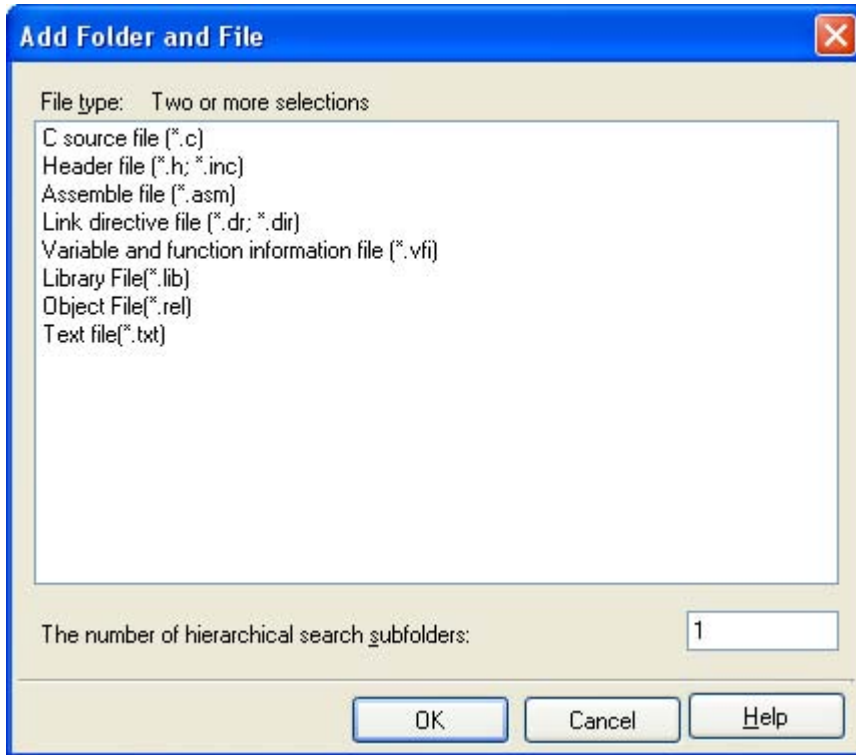
(b) 添加文件夹

从浏览器中拖动一个文件夹，并放置于工程树下的空白区域。打开[添加文件夹和文件对话框](#)。

备注 也可以通过在同一时间拖动多个文件并放置它们到工程树上的方式在同一时间添加多个文件夹到工程中。

注意事项 当放置名称超过 **200** 个字符的文件夹时，此文件夹作为一个分项添加到工程树中，且第 **201** 个以后的字符将被删除。

图 2-6. 添加文件夹和文件对话框



在对话框中，选择文件类型并添加到工程中，设定所需添加的子文件夹层数，然后点击 [OK] 按钮。

备注 点击左键并保持 [Ctrl 或 [Shift] 键按下的状态下选取多个文件类型，可以实现多重文件类型的选择。
如果没有选择内容，则假定选择所有类型。

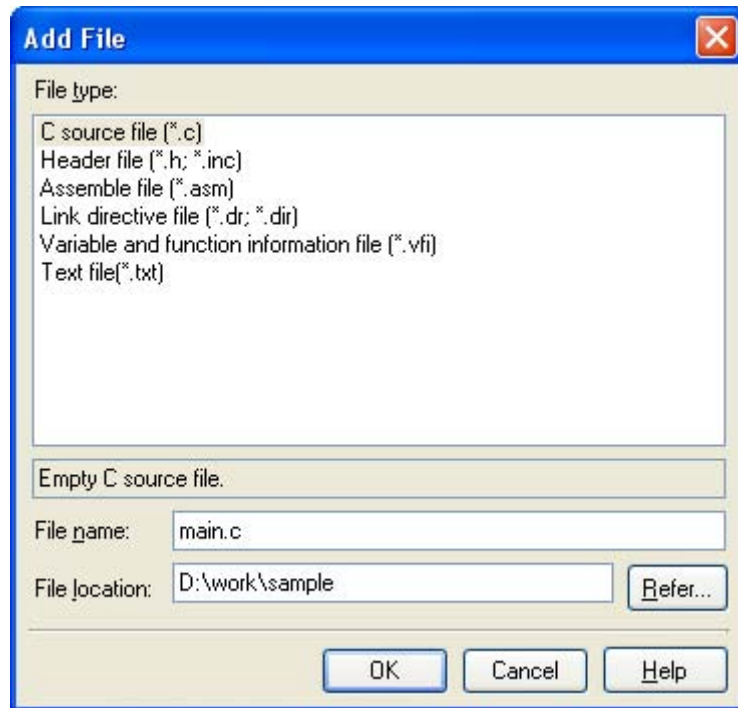
文件夹添加到 **File** 节点下。注意：在工程树上，文件夹就是一个分项。

备注 用户创建 **Category** 节点后，可以通过拖动文件至该节点来添加文件（见类型节点的“[2.3.5 将文件归类到分项中](#)”）。

(2) 建立和添加空文件

在工程树上，选择任意一个 **Project** 节点，**Subproject** 节点，或 **File** 节点，并从快捷菜单 [Add New File...] 中选择 [Add]>>。打开添加文件对话框。

图 2-7. 添加文件对话框



在对话框中，指定要被创建的文件并点击 [OK] 按钮。

文件添加到 **File** 节点下。

在添加文件之后工程树如下所示。

图 2-8. 工程树面板（在添加文件 "main.c" 之后）

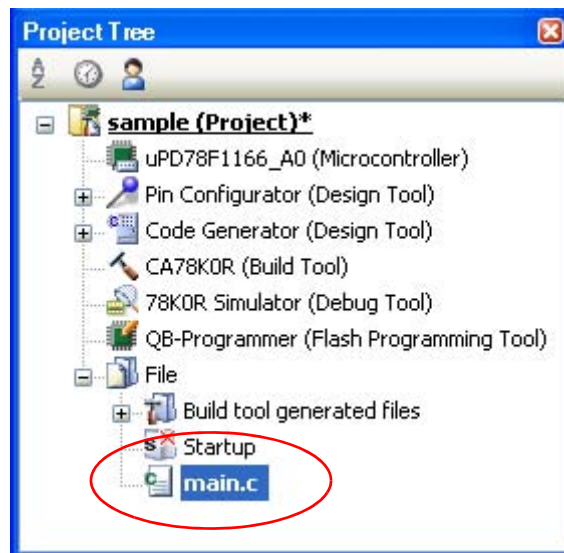
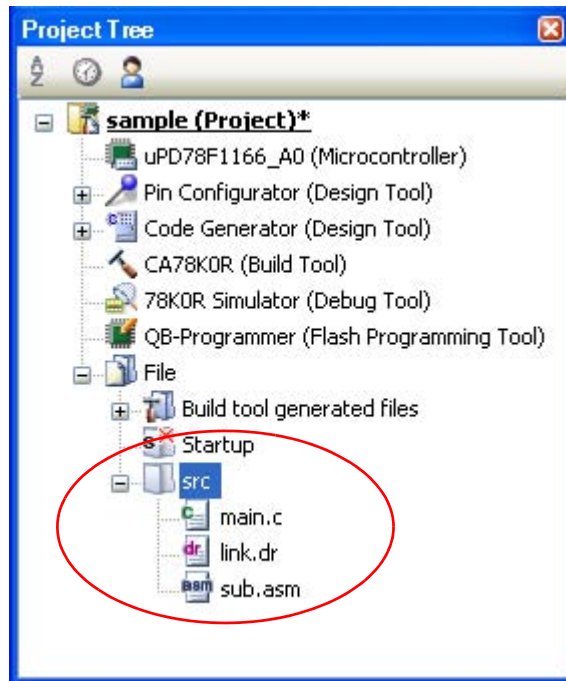


图 2-9. 工程树面板（在添加文件夹 "src" 之后）



备注 在 File 节点下添加的文件位置依赖于当前文件显示顺序的设置。见“2.3.6 改变文件显示顺序”中关于改变文件显示顺序的方法。

- 注意事项 1.** 如果路径有区别，可以添加相同名称的源文件。注意：如果输出文件名作为默认状态靠左边设置，输出文件会有同样的名称，这样将造成此构建不能正确运行（例如，当添加 D:\sample1\func.c 和 D:\sample2\func.c，这些文件的输出文件名的默认名称都是 func.rel）。为正确运行构建，通过独立的构建选项为每个源文件设置不同地输出文件名称。要改变 C 源文件的名称，通过 [\[Individual Compile Options\]](#) 标签的 [Output File] 分项中的 [Object file name] 属性来设置。改变汇编源文件的名称，通过 [\[Individual Assemble Options\]](#) 标签的 [Output File] 分项中的 [Object file name] 属性来设置。见“2.13.2 在文件层设置构建选项”中关于如何设置独立的构建选项。
- 2.** 扩展名为 ".dr" 或 ".dir" 的文件添加到工程中时，则作为链接指令文件。如果添加到 Startup 节点中时，则作为链接指令文件。当添加链接指令文件到工程中时，如果之前链接指令文件已经添加，则构建仅指向最后添加的链接指令文件；之前添加的文件将不能指向。当设置一个链接指令文件，如其没有作为构建目标被指向，而其它链接指令文件已添加，则构建指向该文件，且不会指向其它文件。
- 3.** 可将多达 5000 个文件添加到主工程或子工程中。

添加新文件时，在[添加文件对话框](#)中的指定位置上创建空文件。通过在工程树上双击文件名，可以打开 [编辑器面板](#)并编辑此文件。文件可以在[编辑器面板](#)中打开，所示如下。

- C 源文件 (*.c)
- 汇编源文件 (.asm)
- 头文件 (.h, .inc)
- 链接指令文件 (.dr, .dir)
- 变量 / 函数信息文件 (*.vfi)
- Map 文件 (.map)
- 符号表格文件 (.sym)
- Hex 文件 (.hex, .hxb, .hxf)
- 文本文件 (.txt)

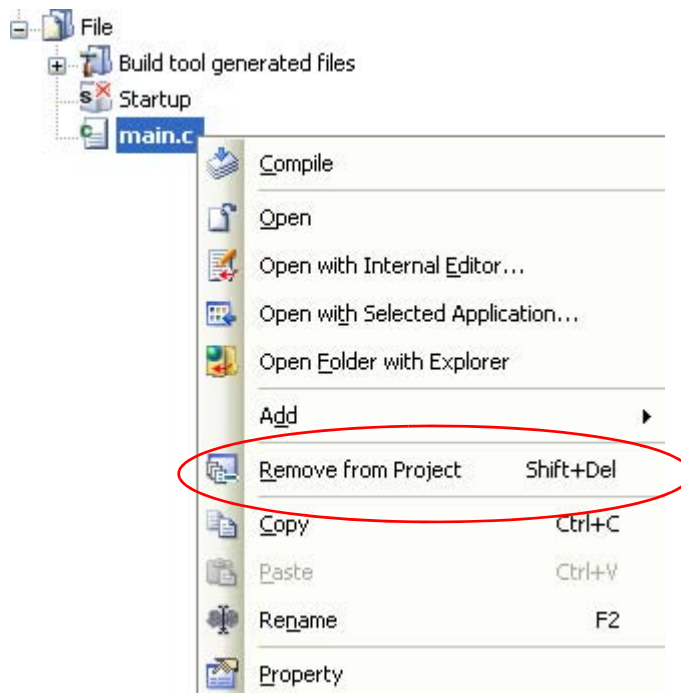
- 备注 1.** 用户可以使用在打开文件下方而不是那些被列在**编辑器面板**上方的方法。
- 拖动文件并将其放进**编辑器面板**。
 - 选择文件然后在快捷菜单中选择 [Open with Internal Editor...]
- 2.** 当在设定使用外部编辑器时**选项对话框**上，文件使用已设置好的外部编辑器打开。其他文件使用主机操作系统中与之相对应的应用程序打开。

2.3.3 从工程中移除文件

从工程中移除已添加的文件，在工程树中选择需要移除的文件并且从快捷菜单中选择 [Remove from Project]。另外，文件本身并没有从文件系统中删除。

注意事项 如果在工程树上选择文件并按 [Delete] 键，可将此文件从系统中删除。

图 2-10. [Remove from Project] 条目



2.3.4 从构建目标中移除文件

可以从构建目标以外的所有被添加到工程的文件中移除指定的文件。

工程树上选择在构建目标中要被移除的文件并在**属性面板**上选择 [Build Settings] 标签。在 [Build] 分项中的 [Set as build-target] 属性上选择 [No]。

图 2-11. [Set as build-target] 属性



备注 能够应用此功能的文件是 C 源文件、汇编源文件、链接指令文件、变量 / 函数信息文件、目标文件和库文件。

2.3.5 将文件归类到分项中

可以在 File 节点下创建分项并通过分项来给文件分类。这可以使得工程树上添加的文件更容易察看且根据其功能更容易管理文件。

如要在工程树上创建 **category** 节点, 选择 **Project** 节点、**Subproject** 节点或 **File** 节点, 并在菜单中选择 **[Add]>>[Add New File...]**。

图 2-12. **[Add New Category]** 条目 (用于 **File** 节点)

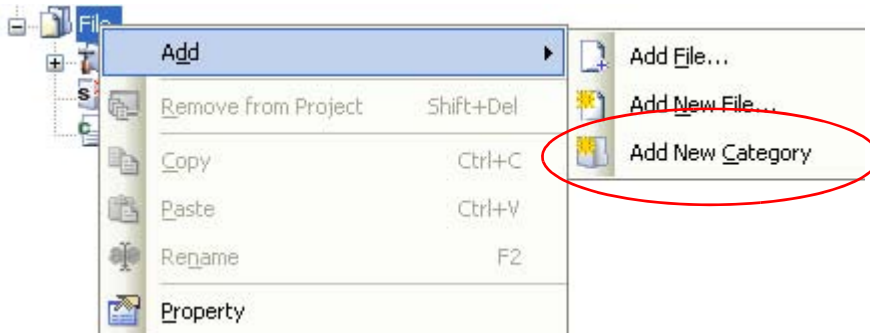
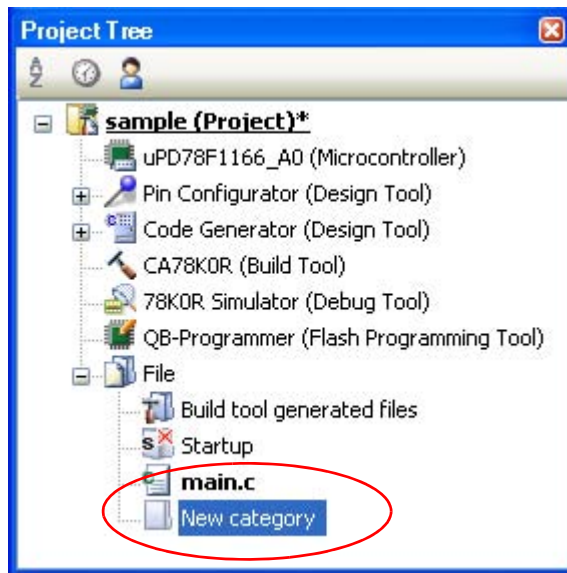


图 2-13. 工程树面板 (在 **Adding Category** 节点之后)



- 备注 1.** 默认的分项名称是 "New category"。
要改变分项名称, 可以从 **category** 节点中的快捷菜单选择 **[Rename]**。
2. 也可以添加 **Category** 节点以同名的方式添加到已存在的 **Category** 节点中。
 3. 分项可以支持多达 20 层嵌套。

可以通过拖放文件的方式从已建立的 **Category** 节点中对文件进行分类。

2.3.6 改变文件显示顺序

可以在工程树上改变文件及 **Category** 节点的显示顺序。
选择在工程树面板的工具栏上任意按钮。

图 2-14. 工具栏 (工程树面板)



| 按钮 | 说明 |
|---|--|
|  | 根据名称排序 Category 节点和文件。  : 升序  : 降序  : 升序 |
|  | 根据时间标签排序 Category 节点和文件。  : 降序  : 升序  : 降序 |
|  | 通过用户（默认）指定的顺序显示 Category 节点和文件。 可以通过拖动的方式随意地改变 Category 节点和文件的显示顺序。 |

2.3.7 升级文件依赖关系

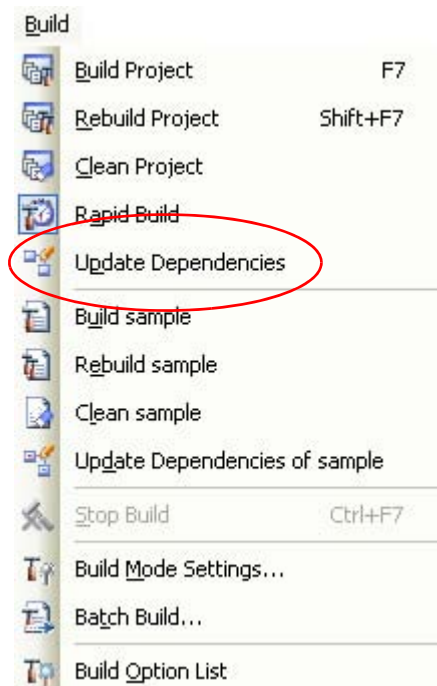
当修改程序（改变包含文件路径，添加包含头文件状态到 C 源文件中）时，会影响在编译选项和汇编选项的文件依赖关系。

在整个工程（主工程和子工程）或活动工程中可以进行文件依赖关系的升级。

(1) 为整个工程

在 [Build] 菜单中，选择 [Update Dependencies]。

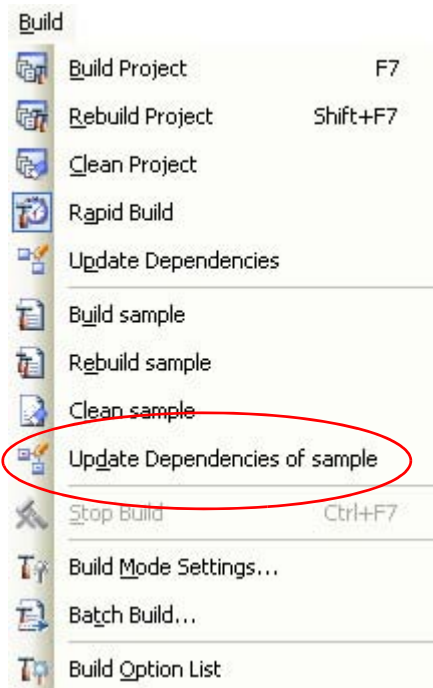
图 2-15. [Update Dependencies] 条目



(2) 用于活动工程

在 [Build] 菜单中，选择 [Update Dependencies of active project]。

图 2-16. [Update Dependencies of active project] 条目



备注 如果有文件在编辑器面板中正在编辑，当升级文件依赖关系时，那么所有这些文件被保存。

注意事项 1. 在 CubeSuite 中检查包含文件相关度时，忽略条件声明例如 #if 和注释。因此，无需构建的包含文件被错误认为需要构建的文件（在下述的例子中，判断 header1.h 和 header5.h 是需要构建的文件）。

```
#if      0
#include  "header1.h"      /* Dependence relationship judged to exist */
#else
/* ! zero */
#include  "header2.h"      /* Dependence relationship to exist */
#endif

#define   AAA
#ifdef   AAA
#include  "header3.h"      /* Dependence relationship to exist */
#else
#include  "header4.h"      /* Dependence relationship to exist */
#endif

/*
#include  "header5.h"      /* Dependence relationship judged to exist */
*/
```

2. 在 CubeSuite 中检查包含文件相关度时，忽略条件声明例如 #if 和注释。因此，需构建的包含文件被错误当作无需构建的文件（在下述的例子中，判断 header1.h 和 header5.h 为无需构建的文件）。

```
/* Dependence relationship judged not to exist */
/* comment */ #include "header6.h"

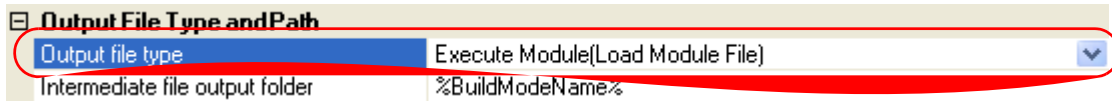
/* Dependence relationship judged not to exist */
/*
comment
*/ #include "header7.h"
```

2.4 设置输出文件的类型

设置构建产物的输出文件的类型。

在工程树上选择 build tool 节点并在属性面板上选择 [Common Options] 标签。在 [Output File Type and Path] 分项中的 [Output file type] 属性上选择文件类型。

图 2-17. [Output file type] 属性



(1) 当 [Execute Module(Load Module File)] 被选中时 (默认)

创建载入模块文件。

在 [Link Options] 标签上 [Output File] 分项中的文件设置是调试目标。

(2) 当选中 [Execute Module(Hex File)] 时

也创建 hex 文件。

在 [Object Convert Options] 标签上 [Hex File] 分项中的文件设置是调试目标。

注意事项 在库工程中，这些属性总是 [Library] 且不能改变。

2.4.1 改变输出文件名称

Build tool 输出载入模块文件，hex 文件及库文件的名称默认名称如下。

"%ProjectName%" 是嵌入式宏。它替换为工程名称。

载入模块文件名称：%ProjectName%.lmf

Hex 文件名称：%ProjectName%.hex

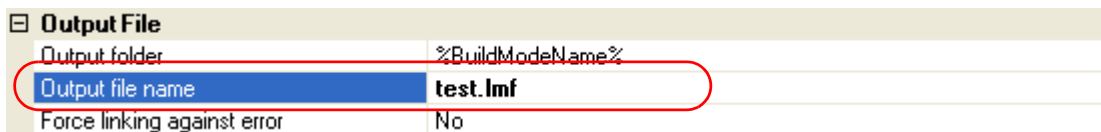
库文件名称：%ProjectName%.lib

改变这些文件的方法所示如下。

(1) 当改变载入模块文件名称时

在工程树上选择 build tool 节点并在属性面板上选择 [Link Options] 标签。在 [Output File] 分项中的 [Output File] 属性上输入被改变的文件名称。

图 2-18. [Output file name] 属性 (用于载入模块文件)



备注 也可以用相同的方法通过在 [Common Options] 标签上的 [Frequently Used Options(for Link)] 分项中的 [Output file name] 属性改变选项。

(2) 当改变载入 hex 文件名称时

在工程树上选择 build tool 节点并在属性面板上选择 [Object Convert Options] 标签。在 [Hex File] 分项中的 [Hex file name] 属性上输入要改变的文件名称。

图 2-19. [Hex file name] 属性

| | |
|----------------------------|---------------------------------|
| Hex File | |
| Output hex file | Yes |
| Output folder for hex file | %BuildModeName% |
| Hex file name | test.hex |
| Hex file format | Intel expanded hex format(-kie) |
| Split hex file | No |

注意事项 在 [Split hex file] 属性上选中 [Yes(-zf)]，则 hex 文件被分为独立的文件：.hxb 和 .hxf。
参阅 "B.4.2 功能" 获取详细信息。

备注 也可以用相同的方法在 [Common Options] 标签上的 [Frequently Used Options(for Object Convert)] 分项中的 [Hex file name] 属性改变选项。

(3) 当改变库文件名称时

在工程树上选择 build tool 节点并在属性面板上选择 [Create Library Options] 标签。在 [Output File] 分项中的 [Output File] 属性上输入被改变的文件名称。

图 2-20. [Output file name] 属性 (库文件)


| | |
|------------------|-----------------|
| Output File | |
| Output folder | %BuildModeName% |
| Output file name | test.lib |

2.4.2 输出汇编列表

汇编的结果输出到汇编列表文件中。

在工程树上选择 build tool 节点并在属性面板上选择 [Assemble Options] 标签。输出汇编列表，在 [Assemble List] 类别中的 [Output assemble list file] 属性上选择 [Yes(-p)] (默认)。

图 2-21. [Output assemble list file] 属性

| | |
|------------------------------------|---|
| Assemble List | |
| Output assemble list file | Yes(-p)  |
| Execute list converter | No |
| Output with assemble list info | Yes |
| Output with symbol list | No |
| Output with cross reference list | No |
| Output with form feed control code | No |
| Number of characters in 1 line | 132 |
| Number of lines on 1 page | 0 |
| Tab width | 8 |
| Header title | |

- 备注 1.** 参见 "3.2.2 汇编列表" 有关汇编列表详细信息。
- 如果在 [Output assemble list file] 属性上选择 [No(-np)]，当进行汇编时只输出目标模块文件，可以减少汇编的时间。

2.4.3 输出映射信息

映射信息 (是有关段位置的信息) 输出到连接列表文件中。

在工程树上选择 **build tool** 节点并在 **属性面板** 上选择 **[Link Options]** 标签。通过 **[Link List]** 类别进行的设置输出连接列表文件。

图 2-22. [Link List] 分项 (用于映射信息)

| Link List | |
|------------------------------------|-----|
| Output link list file | Yes |
| Output with link directive info | Yes |
| Output with local symbol list | No |
| Output with public symbol list | No |
| Output with map list | Yes |
| Output with form feed control code | No |
| Number of lines on 1 page | 0 |

如果在 **[Output link list file]** 属性上选择 **[Yes]** (默认), 则显示 **[Output with map list]** 属性。选择 **[Yes]** (默认) 输出映射信息到连接列表文件中来。

备注 参阅 "3.3.2 Map 列表" 中有关映射信息部分。

2.4.4 输出符号信息

在输入模块中定义的符号信息 (本地符号和公用符号) 输出到链接列表文件中。在工程树上选择 **build tool** 节点并在 **属性面板** 上选择 **[Link Options]** 标签。

通过 **[Link List]** 分项进行的设置输出符号信息。

(1) 当输出本地符号列表时

图 2-23. [Link List] 分项 (用于本地符号信息)


| Link List | |
|------------------------------------|----------|
| Output link list file | Yes |
| Output with link directive info | Yes |
| Output with local symbol list | Yes(-kl) |
| Output with public symbol list | No |
| Output with map list | Yes |
| Output with form feed control code | No |
| Number of lines on 1 page | 0 |

如果在 **[Output link list file]** 属性上选择 **[Yes]** (默认), 则显示 **[Output with local symbol list]** 属性。选择 **[Yes(-kl)]** (**[No]** 选择为默认) 输出本地符号列表到链接列表文件中。

备注 见 "3.3.4 局部符号列表" 中有关本地符号列表的更详细说明。

(2) 当输出公用符号列表时

图 2-24. [Link List] 分项 (用于公用符号信息)

| Link List | |
|------------------------------------|--|
| Output link list file | Yes |
| Output with link directive info | Yes |
| Output with local symbol list | No |
| Output with public symbol list | Yes(-kp)  |
| Output with map list | Yes |
| Output with form feed control code | No |
| Number of lines on 1 page | 0 |

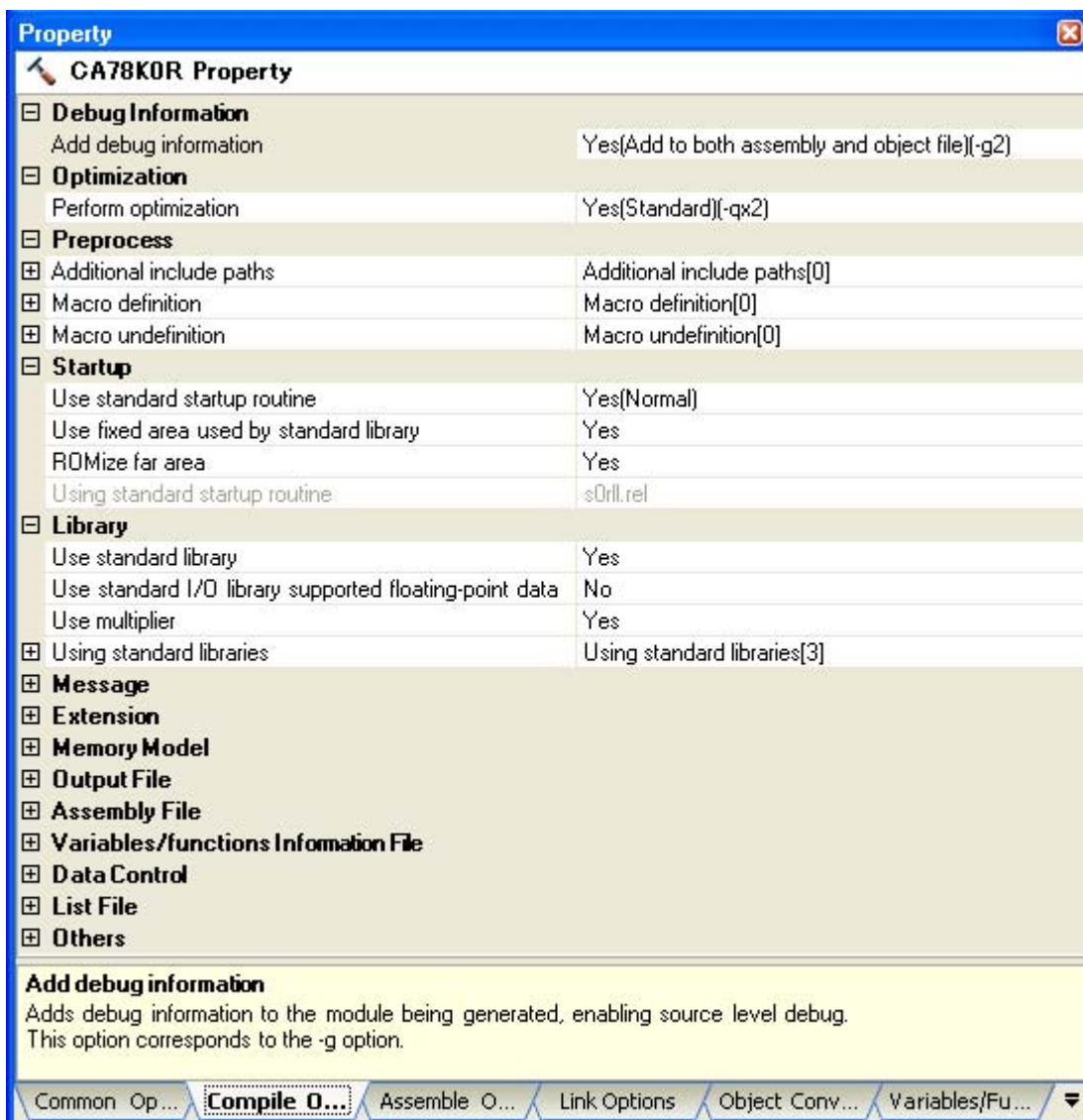
如果在 [Output link list file] 属性上选择 [Yes] (默认), 则显示 [Output with public symbol list] 属性。选择 [Yes(-kp)] ([No] 选择为默认) 输出公用符号列表到链接列表文件中。

备注 见 "3.3.3 公用符号列表" 中有关公用符号列表的更详细说明。

2.5 设置编译选项

设置编译器选项时，在工程树上选择 **Build tool** 节点并在属性面板上选择 **[Compile Options]** 标签。
可以通过在该标签中设定需要的属性来设置不同的编译选项。

图 2-25. 属性面板 : **[Compile Options]** 标签



备注 经常使用的选项可在 **[Common Options]** 标签上的 **[Frequently Used Options(for Compile)]** 分项下取得。

2.5.1 进行优化时代码大小优先。

在工程树上选择 **build tool** 节点并在属性面板上选择 **[Compile Options]** 标签。

要选择进行优化时代码大小优先，在 **[Optimization]** 类别中设置 **[Perform optimization]** 属性为 **[Yes(Code size precedence)(-qx3)]** ([No] 为默认选项)。

图 2-26. **[Perform optimization]** 属性 (代码大小优先)



备注 也可以用相同的方法通过在 **[Common Options]** 标签上的 **[Frequently Used Options(for Compile)]** 分项中的 **[Perform optimization]** 属性设置选项。

2.5.2 进行优化时运行速度优先。

在工程树上选择 build tool 节点并在属性面板上选择 [Compile Options] 标签。

要选择进行优化时执行速度优先，在 [Optimization] 类别中设置 [Perform optimization] 属性为 [Yes(Code size precedence)(-qx1)]([No] 为默认选项)。

图 2-27. [Perform optimization] 属性 (执行速度优先)



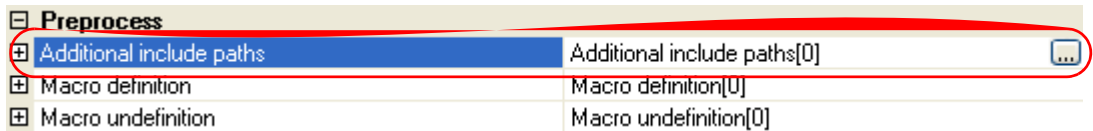
备注 也可以用相同的方法通过在 [Common Options] 标签上的 [Frequently Used Options(for Compile)] 分项中的 [Perform optimization] 属性设置选项。

2.5.3 添加 include 路径

在工程树上选择 build tool 节点并在属性面板上选择 [Compile Options] 标签。

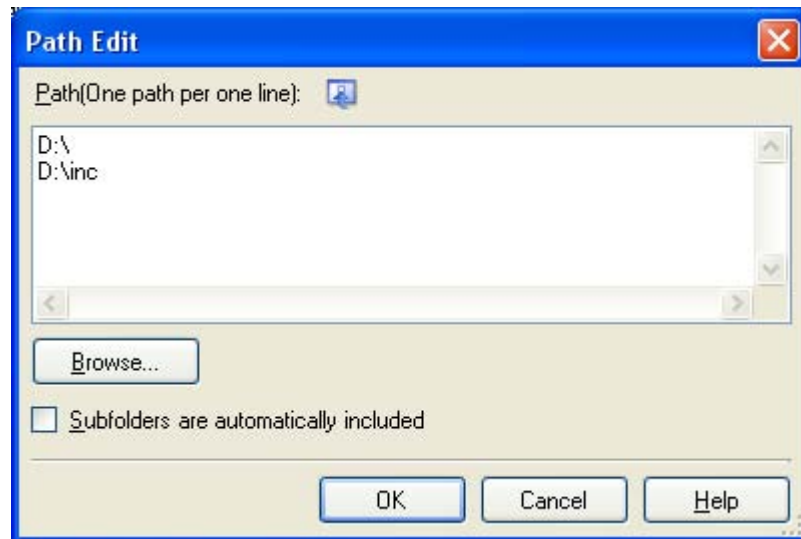
通过在 [Preprocess] 分项中设置 [Additional include paths] 属性来设置 include 路径。

图 2-28. [Additional include paths] 属性



单击 [...] 按钮后，路径编辑对话框将会打开。

图 2-29. 路径编辑对话框

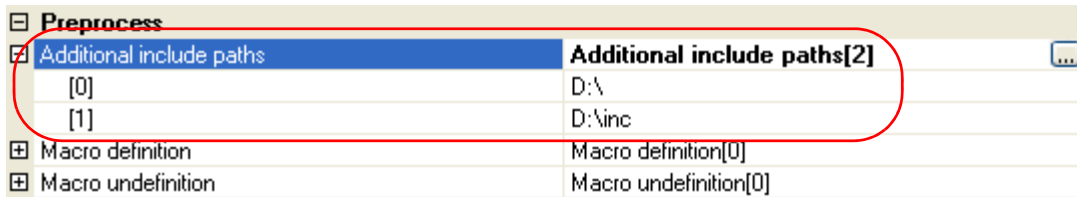


在 [Path(One path per one line)] 的每行输入 include path。每行可以设定多达 259 个字符，最多 64 行。

备注 可以通过 [Browse...] 按钮指定 include path。在点击 [Browse...] 按钮后选择 [Subfolders are automatically included] 复选框，添加所有路径在指定的（每页最多显示 5 层）到 [Path(One path per one line)] 上。

单击 [OK] 按钮后，输入的 include path 作为子属性显示出来。

图 2-30. [Additional include paths] 属性（在添加 include path 之后）



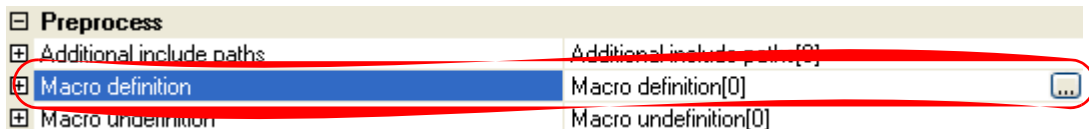
可以使用 [...] 按钮或直接在子属性的文本框中输入路径，改变 include 路径

备注 也可以用相同的方法通过在 [Common Options] 标签上的 [Frequently Used Options(for Compile)] 类别中的 [Additional include paths] 属性设置此选项。

2.5.4 设置宏定义

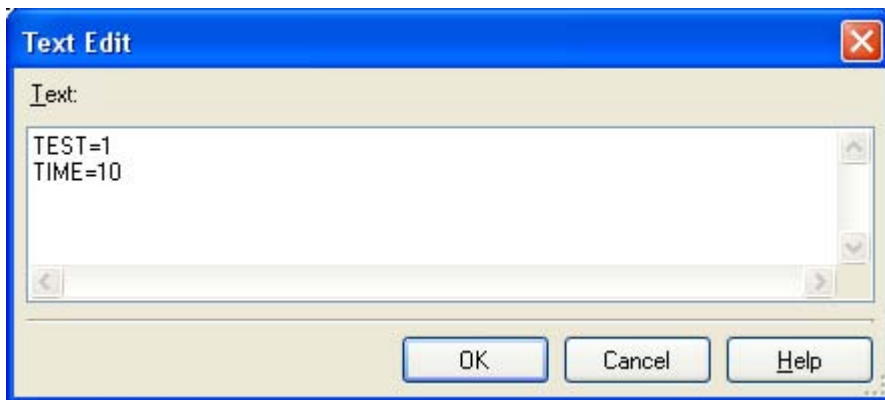
在工程树上选择 build tool 节点并在属性面板上选择 [Compile Options] 标签。宏定义通过在 [Preprocess] 分项中设置 [Macro definition] 属性来设置。

图 2-31. [Macro definition] 属性



单击 [...] 按钮后，文本编辑对话框将会打开。

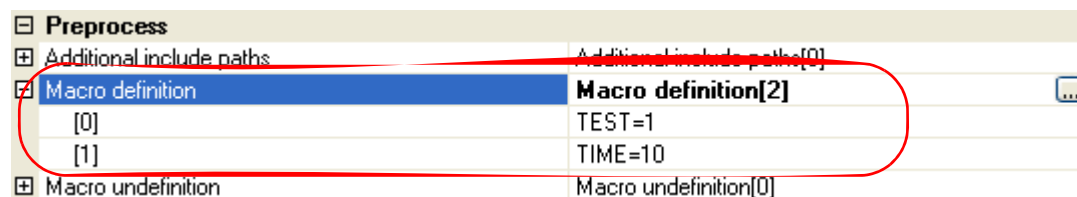
图 2-32. 文本编辑对话框



通过 "宏名称 = 定义值" 这种格式来输入宏定义，每行一个宏名称。每行可以设定多达 256 个字符，最多 30 行。"= 定义值" 部分可以省略，此时默认值为 1。

如果单击 [OK] 按钮，输入的宏定义作为子属性显示出来。

图 2-33. [Macro definition] 属性（在设置宏之后）



改变宏定义，可以使用 [...] 按钮或直接在子属性的文本框中输入路径。

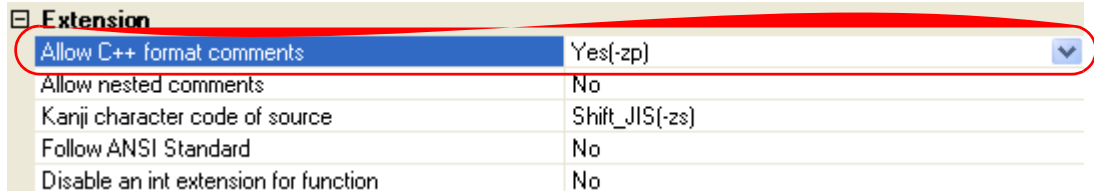
备注 也可以用相同的方法通过在 [\[Common Options\]](#) 标签上的 [\[Frequently Used Options\(for Compile\)\]](#) 分项中的 [\[Macro definition\]](#) 属性设置此选项。

2.5.5 使用 C++ 注释

在工程树上选择 build tool 节点并在属性面板上选择 [\[Compile Options\]](#) 标签。

要选择使用 C++ 注释，在 [\[Extension\]](#) 分项中 [\[Allow C++ format comments\]](#) 属性上选择 [\[Yes\(-zp\)\]](#)(默认)。

图 2-34. [\[Allow C++ format comments\]](#) 属性



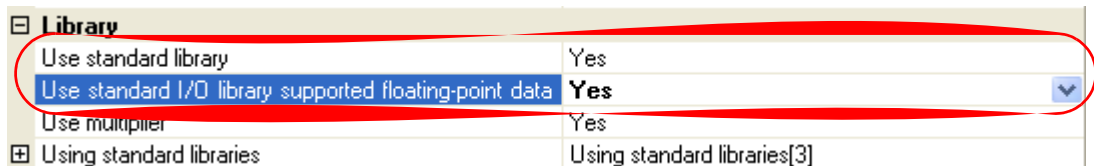
| Extension | |
|---------------------------------------|----------------|
| Allow C++ format comments | Yes(-zp) |
| Allow nested comments | No |
| Kanji character code of source | Shift_JIS(-zs) |
| Follow ANSI Standard | No |
| Disable an int extension for function | No |

2.5.6 使用兼容浮点的标准输入 / 输出函数

在工程树上选择 build tool 节点并在属性面板上选择 [\[Compile Options\]](#) 标签。

在 [\[Library\]](#) 分项中，如果在 [\[Use standard library\]](#) 属性上选择 [\[Yes\]](#)，则显示 [\[Use standard I/O library supported floating-point data\]](#)。使用支持浮点数据的标准输入 / 输出函数 (sprintf, sscanf, printf, vprintf, and vsprintf)，选择 [\[Yes\]](#)。

图 2-35. [\[Use standard I/O library supported floating-point data\]](#) 属性



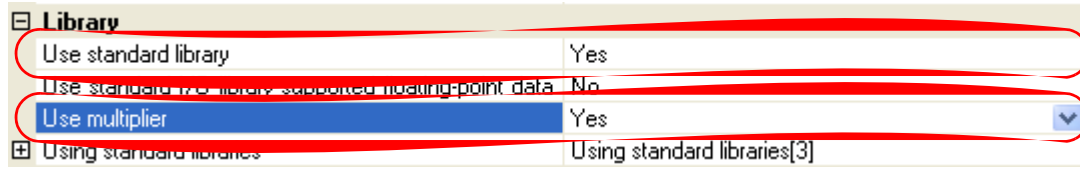
| Library | |
|--|-----------------------------|
| Use standard library | Yes |
| Use standard I/O library supported floating-point data | Yes |
| Use multiplier | Yes |
| Using standard libraries | Using standard libraries[3] |

2.5.7 改变设置使用乘法器和除法器 / 乘法器

在工程树上选择 build tool 节点并在属性面板上选择 [\[Compile Options\]](#) 标签。

在 [\[Library\]](#) 分项中，如果在 [\[Use standard library\]](#) 属性上选择 [\[Yes\]](#)，显示 [\[Use multiplier and divider\]](#) 属性或 [\[Use multiplier\]](#) 属性。当使用标准库其支持乘数和除数 / 乘数，选择 [\[Yes\]](#) (默认)，当不使用时选择 [\[No\]](#)。

图 2-36. [Use multiplier] 属性

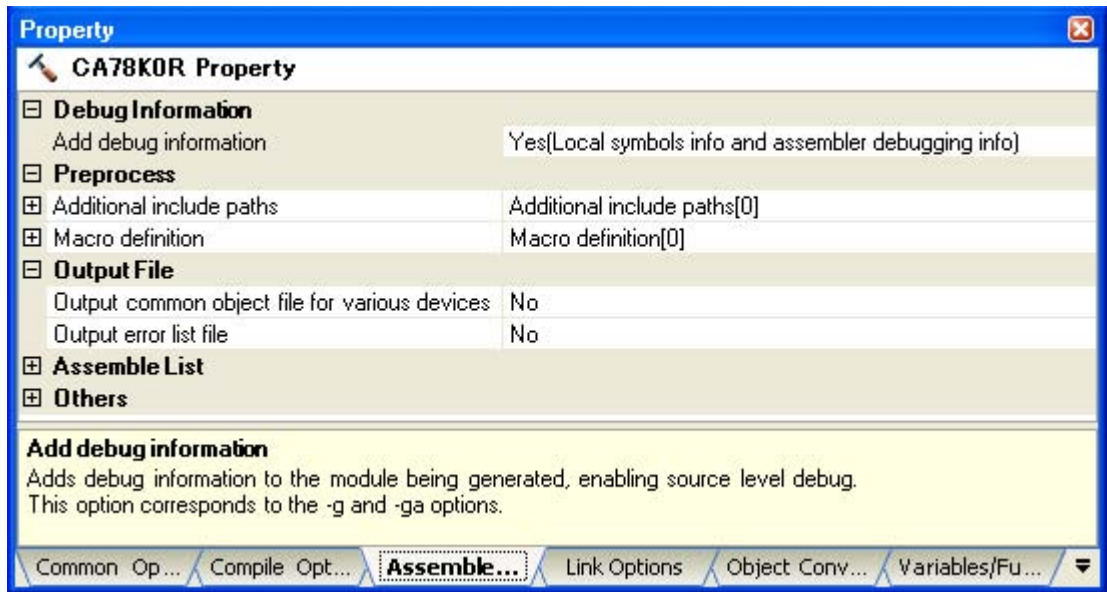


| Library | |
|--|-----------------------------|
| Use standard library | Yes |
| Use standard library supported floating-point data | No |
| Use multiplier | Yes |
| Using standard libraries | Using standard libraries[3] |

2.6 设置汇编选项

为汇编程序设置选项，在工程树上选择 **Build tool** 节点并在属性面板上选择 **[Assemble Options]** 标签。可以通过该属性页给各种汇编选项设置必要的值。

图 2-37. 属性面板 : **[Assemble Options]** 标签

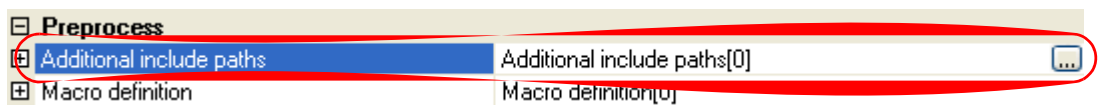


备注 经常使用的选项在 **[Common Options]** 标签上的 **[Frequently Used Options(for Assemble)]** 下取得。

2.6.1 添加 include path

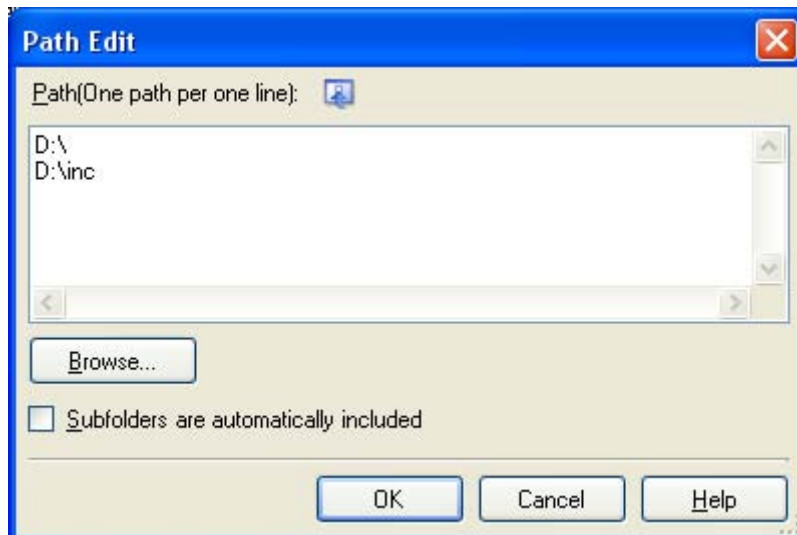
在工程树上选择 **build tool** 节点并在属性面板上选择 **[Assemble Options]** 标签。通过在 **[Preprocess]** 分项中设置 **[Additional include paths]** 属性来设置 include 路径。

图 2-38. **[Additional include paths]** 属性



单击 [...] 按钮后，**路径编辑对话框**将会打开。

图 2-39. 路径编辑对话框

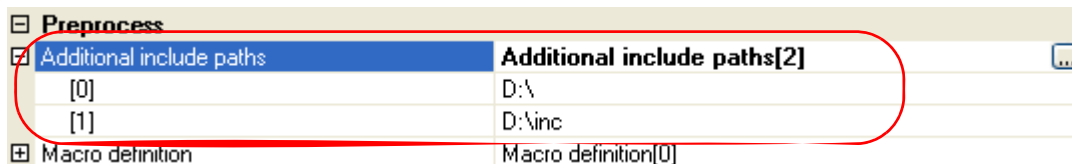


在 [Path(One path per one line)] 的每行输入 include path。每行可以设定多达 259 个字符，最多 64 行。

备注 可以通过 [Browse...] 按钮指定 include path。在点击 [Browse...] 按钮后选择 [Subfolders are automatically included] 复选框，添加所有路径在指定的（每页最多显示 5 层）到 [Path(One path per one line)] 上。

单击 [OK] 按钮后，输入的 include path 作为子属性显示出来。

图 2-40. [Additional include paths] 属性（在添加 include path 之后）



可以使用 [...] 按钮或直接在子属性的文本框中输入路径，改变 include 路径

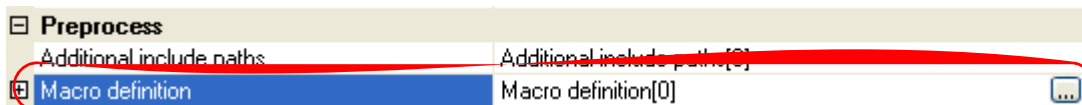
备注 也可以用相同的方法通过在 [Common Options] 标签上的 [Frequently Used Options(for Assemble)] 分项中的 [Additional include paths] 属性设置此选项。

2.6.2 设置宏定义

在工程树上选择 build tool 节点并在属性面板上选择 [Assemble Options] 标签。

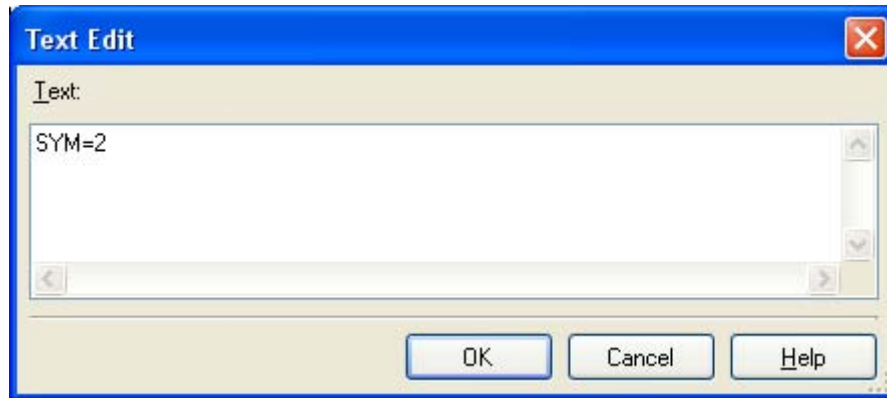
宏定义通过在 [Preprocess] 分项中设置 [Macro definition] 属性来设置。

图 2-41. [Macro definition] 属性



单击 [...] 按钮后，文本编辑对话框将会打开。

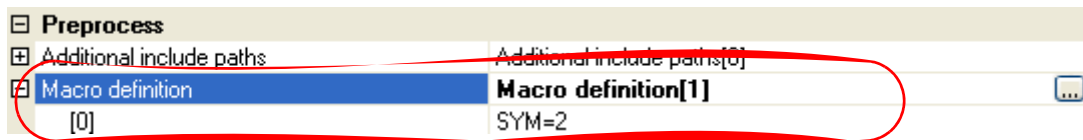
图 2-42. 文本编辑对话框



通过 "宏名称 = 定义值" 这种格式来输入宏定义, 每行一个宏名称。每行可以设定多达 31 个字符, 最多 30 行。" = 定义值" 部分可以省略, 此时默认值为 1。

如果单击 [OK] 按钮, 输入的宏定义作为子属性显示出来。

图 2-43. [Macro definition] 属性 (在设置宏之后)



改变宏定义, 可以使用 [...] 按钮或直接在子属性的文本框中输入路径。

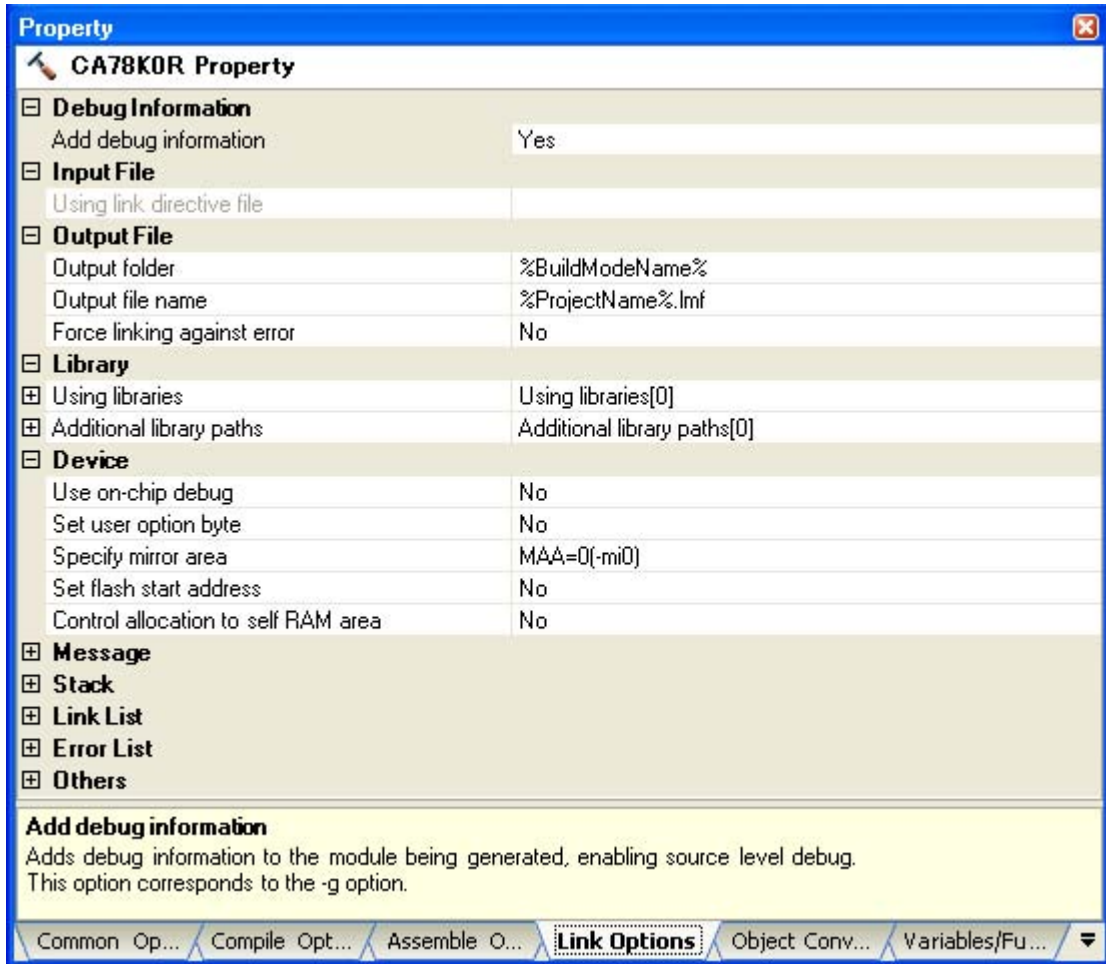
备注 也可以用相同的方法通过在 [Common Options] 标签上的 [Frequently Used Options(for Assemble)] 分项中的 [Macro definition] 属性设置此选项。

2.7 设置连接选项

为连接器设置选项，在工程树上选择 **Build tool** 节点并在属性面板上选择 **[Link Options]** 标签。
可以通过在该列表中设定需要的属性设置不同的链接选项。

注意事项 对库工程此标签不会显示。

图 2-44. 属性面板 : [Link Options] 标签

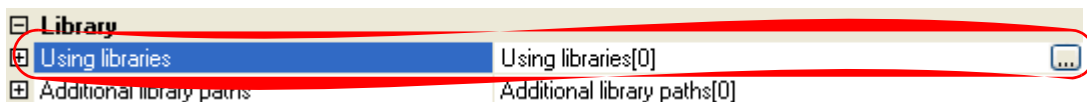


备注 经常使用的选项在 **[Common Options]** 标签上的 **[Frequently Used Options(for Link)]** 分项下取得。

2.7.1 添加用户库

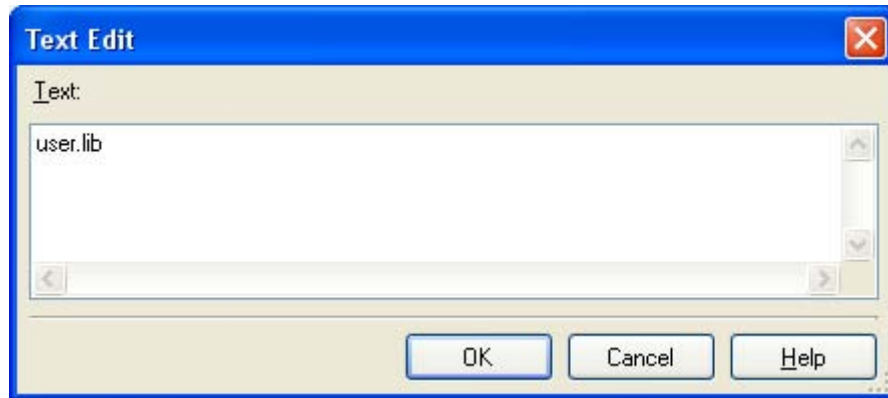
在工程树上选择 **build tool** 节点并在属性面板上选择 **[Link Options]** 标签。
通过设定在 **[Library]** 分项中的 **[Using libraries]** 属性添加用户库。

图 2-45. [Using libraries] 属性



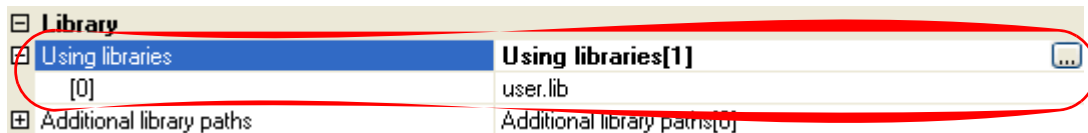
单击 [...] 按钮后，文本编辑对话框将会打开。

图 2-46. 文本编辑对话框



在 [Text] 中输入库文件名称，每行一个名称。每行可以设定多达 259 个字符，最多 64 行。
单击 [OK] 按钮后，输入的库文件作为子属性显示出来。

图 2-47. [Using libraries] 属性 (在设置库文件之后)



改变库文件，可以使用 [...] 按钮或直接在子属性的文本框中输入路径。

备注 也可以用相同的方法通过在 [Common Options] 标签上的 [Frequently Used Options(for Link)] 分项中的 [Using libraries] 属性设置此选项。

从库文件路径查找库文件。要添加库路径，需要设置 [Additional library paths] 属性。

注意事项 直接添加库文件到工程中也能够链接。在这种情况下，由于使用绝对路径链接库文件，所以不在库文件路径下查找该库文件。

2.8 为使用片上调试器做准备

使用片上调试器时，必须设置片上调试，用户选项字节和安全 ID。

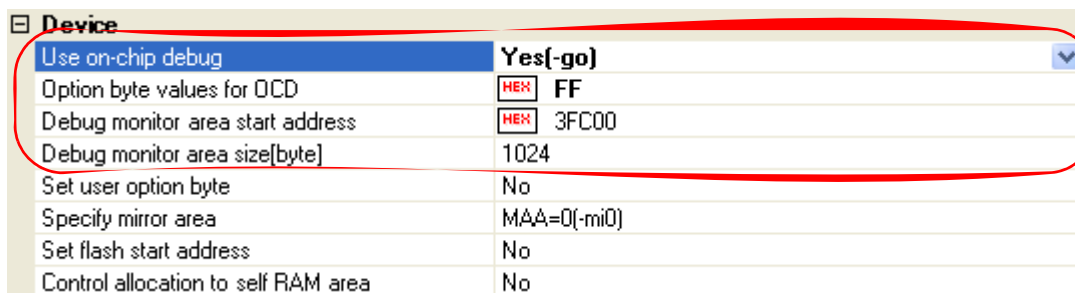
(1) 设置片上调试。

微型控制器的片上调试功能通过设置片上调试启用。

在工程树上选择 build tool 节点并在属性面板上选择 [Link Options] 标签。在 [Device] 分项中设置片上调试。

如果在显示 [Use on-chip debug] 属性，[Option byte values for OCD] 属性，[Debug monitor area start address] 属性，和 [Debug monitor area size[byte]] 属性上选择 [Yes(-go)]。

图 2-48. [Device] 分项



在 [Option byte values for OCD] 属性上，用不含 0x 的十六进制数来设定片上调试选项字节的控制值。可以用于设定值的范围是 0 ~ FFFF。

在 [Debug monitor area start address] 属性上，用不带 0x 的十六进制数来设定调试监视区域的起始地址。可以用于设定值的范围是 0 ~ FFFFF (默认：内部 ROM 结束地址 - 1024 + 1)。

在 [Debug monitor area size[byte]] 属性上，用十进制数来设定调试监视区域的大小。允许范围是 88 到 1024(默认：1024)。

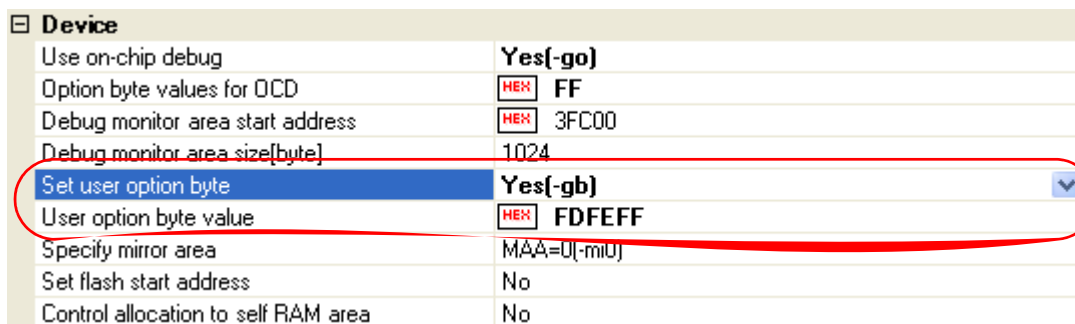
(2) 设置用户选项字节

通过设置用户选项字节，为监视计时器，低电压侦测电路，和系统保留区域的设置做准备。

也可在 [Link Options] 标签上的 [Device] 分项中做用户选项字节的设置。

如果在 [Set user option byte] 属性上选择 [Yes(-gb)], 则 [User option byte value] 属性显示。

图 2-49. [Device] 分项



在 [User option byte value] 属性上，用不带 0x 的十六进制数来设定用户选项字节的值。可以用于设定值的范围是 0 ~ FFFFFFFF。

如果做如上设置，设置以下的值：0xFD 到 0xC0, 0xFE 到 地址 0xC1, 0xFF 到地址 0xC2。

(3) 设置安全 ID

在调试器使能后，使用安全 ID 来进行验证。

在工程树上选择 build tool 节点并在属性面板上选择 [Common Options] 标签。

在 [Device] 分项中的 [Security ID] 属性上，用 20 位十六进制数来设定安全 ID。允许范围是 00000000000000000000 到 FFFFFFFFFFFFFFFFFFFF。

图 2-50. [Security ID] 属性



如果按以上设定，下列的值被设置为：0x11 到地址 0xC4, 0x22 到地址 0xC5, 0x33 到地址 0xC6, 0x44 到地址 0xC7, 0x55 到地址 0xC8, 0x66 到地址 0xC9, 0x77 到地址 0xCA, 0x88 到地址 0xCB, 0x99 到地址 0xCC, 0xAA 到地址 0xCD。

备注 有关连接调试工具内容，参阅 CubeSuite 78K0 调试用户手册。

2.9 准备执行 Boot-Flash Relink 功能

根据系统需要，除不能被重写 / 替换的区域（引导区域）以外，偶尔可以使用能被重写 / 替换的区域（闪存区域），例如闪存或外部 ROM。

在这些系统中，当改变在闪存区域中的程序时，"relink function" 可以正确地在引导区域和闪存区域中执行功能调用且在引导区域中不重建程序。

通过为引导区域和闪存区域建立载入模块文件，可以执行 relink 功能。执行 relink 功能方法如下所示。

备注 关于 relink 功能及如何执行的详细内容，参阅 "B.3.5 Boot-flash relink 功能"。

2.9.1 准备构建的目标文件

(1) 准备链接指令文件

在引导区域和闪存区域为工程准备连接指令文件。

备注 可以在引导区域和闪存区域使用相同的链接指令文件，但这样描述起来会比较复杂，建议每个区域使用一个单独的链接指令文件。

(2) #pragma ext_func directive

在 C 源文件中使用 #pragma ext_func directive。

通过 #ext_func 指令，为目标函数设定 ID 值（实际函数存在于闪存区域并从引导区域被调用）。

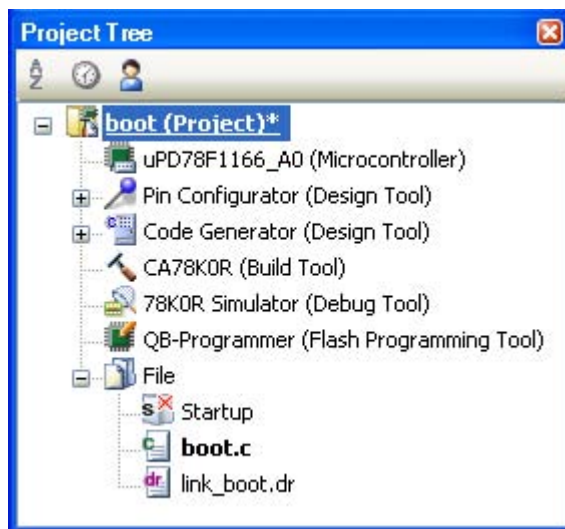
备注 为了防止标识错误和在源文件中的不一致，建议将 #ext_func directive 写在一个独立的文件中，且不管引导区域或闪存区域，在所有的 C 源文件包含该文件中。

2.9.2 设置引导区工程

(1) 创建引导区工程

在引导区创建工程并添加构建目标文件到工程中。

图 2-51. 引导区域工程



(2) 为引导区域工程设置构建选项。

在工程树上选择 build tool 节点并在属性面板上设置每个构建选项。

(a) 设置变量 / 函数重新配置选项

设置变量 / 函数重新配置选项，生成变量 / 函数信息文件并使用它来分配变量和函数。

选择 [Variables/Functions Relocation Options] 标签。

在 [Output File] 分项中，设置 [Output variables/functions information file] 属性为 [Yes]，生成空的变量 / 函数信息文件，并添加到工程中（它也将出现在工程树的 File 节点中）。输出目标通过在 [Output folder for variables/functions information file] 属性和 [Variables/functions information file name] 属性中进行设置。

备注 如果存在具有相同名称的变量 / 函数信息文件，则使用它配置构建。

图 2-52. 在引导区域的 [Output folder for variables/functions information file] 属性

| | |
|--|-------------------|
| Output File | |
| Output variables/functions information file | Yes |
| Output folder for variables/functions information file | %BuildModeName% |
| Variables/functions information file name | %ProjectName%.vfi |

设置 [Output folder for variables/functions information file] 属性和 [Variables/functions information file name] 属性来改变变量 / 函数信息文件的输出文件夹和文件名称。如果更改了 [Variables/functions information file] 属性，则生成空的变量 / 函数信息文件并添加到工程中（它也将出现在工程树的 File 节点中）。

(b) 设置编译选项

选择 [Compile Options] 标签。

在 [Memory Model] 分项中的 [Output objects for flash] 属性上选择 [No]。

另外，配置 [Start address of flash area] 属性和 [Start address of flash area branch table] 属性。

可以用于设定两个值的范围是 0 ~ FFFF。

备注 在 [Start address of flash area branch table] 属性中设定地址，它是在 flash 区域中的地址。

图 2-53. 在引导区域中的 [Memory Model] 分项

| | |
|--|--|
| Memory Model | |
| Memory model type | Medium model(Code:1M Byte,Data:512K Byte)(rom) |
| Output objects for flash | No |
| Start address of flash area | HEX 2000 |
| Start address of flash area branch table | HEX 2000 |
| Specify mirror area | M&A=0[-mi0] |

接下来，在 [Startup] 分项中的 [Use standard startup routine] 上选择 [Yes(For boot area)]。

图 2-54. 在引导区域的 [Use standard startup routine] 属性

| | |
|---|--------------------|
| Startup | |
| Use standard startup routine | Yes(For boot area) |
| Use fixed area used by standard library | Yes |
| ROMize far area | Yes |
| Using standard startup routine | sDrllb.rel |


(c) 设置链接选项

选择 [Link Options] 标签。

在 [Device] 分项中，如果在 [Set flash start address] 属性上选择 [Yes(-zb)],[Flash start address] 显示。

在属性中，与在 [Compile Options] 标签的 [Memory Model] 分项上的 [Start address of flash area] 属性值相同。

图 2-55. 在引导区域中的 [Device] 分项


| Device | |
|-------------------------------------|--|
| Use on-chip debug | No |
| Set user option byte | No |
| Specify mirror area | MΔΔ=0(-mi0) |
| Set flash start address | Yes(-zb)  |
| Flash start address | HEX 2000 |
| Boot area load module file name | |
| Control allocation to self RAM area | No |

(d) 设置目标转换选项

选择 [Object Convert Options] 标签。

在 [[Hex File] 分项中 (默认) 的 [Split hex file] 属性上选择 [No]。

图 2-56. 在引导区域的 [Split hex file] 属性

| Hex File | |
|----------------------------|--|
| Output hex file | Yes |
| Output folder for hex file | %BuildModeName% |
| Hex file name | %ProjectName%.hex |
| Hex file format | Intel expanded hex format(-hi) |
| Split hex file | No  |

(3) 运行引导区工程构建

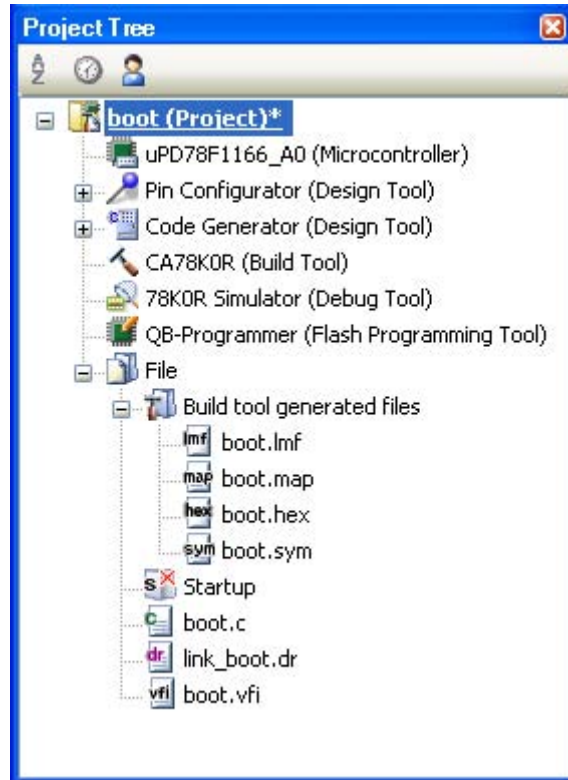
当执行引导区工程的构建时，创建载入模块文件。

同时也创建 hex 文件。

如果生成变量 / 函数信息文件，它将会自动输入到编译器中，并且会再次执行重建。

备注 通过运行重建，在 (1) 中会覆盖变量 / 函数信息文件。

图 2-57. 为引导区域创建文件

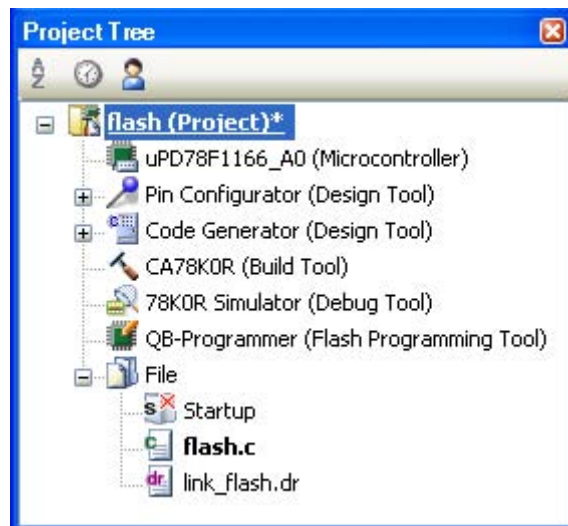


2.9.3 设置闪存区域工程

(1) 创建闪存区工程

在引导区创建工程并添加构建目标文件到工程中。

图 2-58. 闪存区工程



(2) 为闪存区工程设置构建选项

在工程树上选择 build tool 节点并在属性面板上设置每个构建选项。

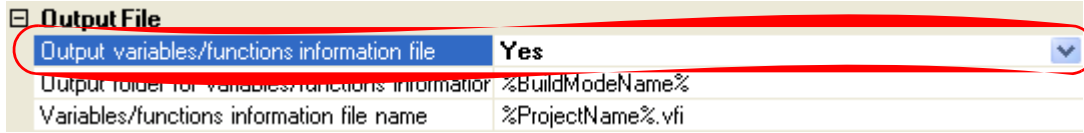
(a) 设置变量 / 函数重新配置选项

设置变量 / 函数重新配置选项，生成变量 / 函数信息文件并使用它来分配变量和函数。
选择 [\[Variables/Functions Relocation Options\]](#) 标签。

在 [Output File] 分项中，设置 [Output variables/functions information file] 属性为 [Yes]，生成空的变量 / 函数信息文件，并添加到工程中（它也将出现在工程树的 File 节点中）。输出目标通过在 [Output folder for variables/functions information file] 属性和 [Variables/functions information file name] 属性中进行设置。

备注 如果存在具有相同名称的变量 / 函数信息文件，则使用它配置构建。

图 2-59. 在 Flash 区域的 [Output folder for variables/functions information file] 属性



设置 [Output folder for variables/functions information file] 属性和 [Variables/functions information file name] 属性来改变变量 / 函数信息文件的输出文件夹和文件名称。如果更改了 [Variables/functions information file] 属性，则生成空的变量 / 函数信息文件并添加到工程中（它也将出现在工程树的 File 节点中）。

(b) 设置编译选项

选择 [Compile Options] 标签。

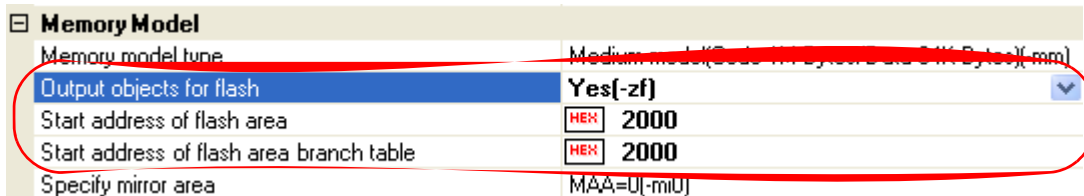
在 [Memory Model] 分项中的 [Output objects for flash] 属性上选择 Yes(-zf)。

另外，配置 [Start address of flash area] 属性和 [Start address of flash area branch table] 属性。

可以用于设定值的范围是 0C0 到 0EDFFF。

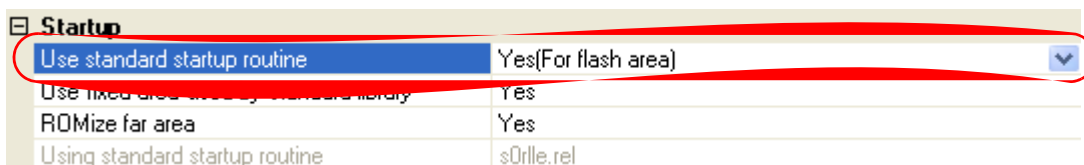
备注 在 [Start address of flash area branch table] 属性中设定的地址与在引导区域工程中设定的地址相同。

图 2-60. 在 Flash 区域中的 [Memory Model] 分项



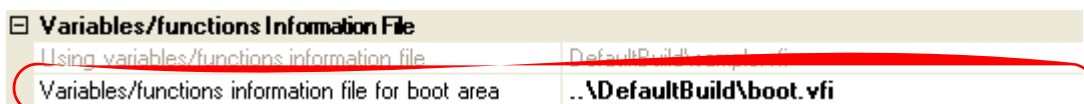
接下来，在 [Startup] 分项中的 [Use standard startup routine] 属性上选择 [Yes(For flash area)]。

图 2-61. 在闪存区域的 [Use standard startup routine] 属性



然后，添加由在 "2.9.2 设置引导区工程" 的启动区域创建的变量 / 函数信息文件到 flash 区域工程中。在 [Variable and Function Information File] 分项中的 [Variables/functions information file for boot area] 属性上设置启动区域的变量 / 函数信息文件。

图 2-62. 在 Flash 区域的 [Variables/functions information file for boot area] 属性



(c) 设置链接选项

在 "2.9.2 设置引导区工程" 中添加已创建引导区的载入模块文件到闪存区工程。选择 [Link Options] 标签。
在 [Device] 分项中的 [Boot area load module file name] 属性上设定引导区域载入模块文件。

图 2-63. 在闪存区的 [Boot area load module file name] 属性

| Device | |
|-------------------------------------|--------------------------|
| Use on-chip debug | No |
| Set user option byte | No |
| Specify mirror area | MAA=0(-mi0) |
| Set flash start address | No |
| Boot area load module file name | ..\DefaultBuild\boot.lmf |
| Control allocation to self RAM area | No |

(d) 设置目标转换选项

选择 [Object Convert Options] 标签。
在 [[Hex File] 分项中的 [Split hex file] 属性上选择 [Yes(-zf)]。

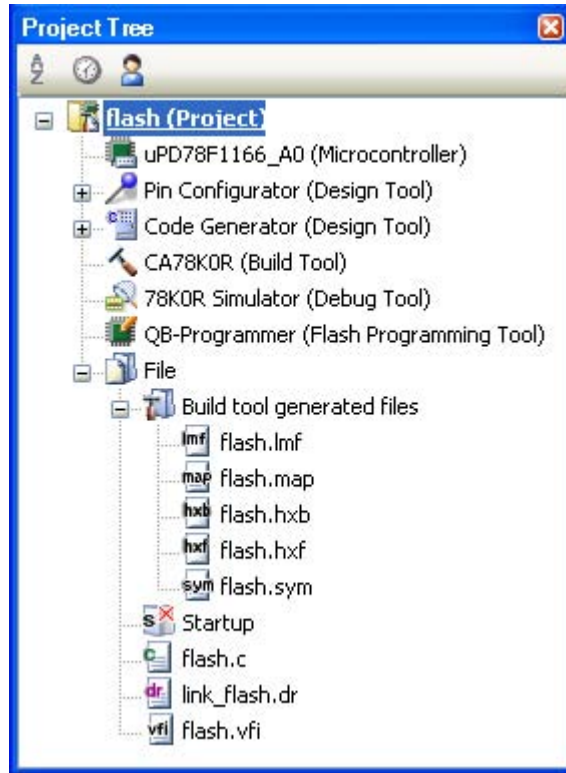
图 2-64. 在闪存区域的 [Split hex file] 属性

| Hex File | |
|----------------------------|---------------------------------|
| Output hex file | Yes |
| Output folder for hex file | %BuildModeName% |
| Hex file name | %ProjectName%.hex |
| Hex file format | Intel expanded hex format(.kie) |
| Split hex file | Yes(-zf) |

(3) 运行闪存区的工程构建

当运行闪存区工程的构建时，具有重链接功能接口的载入模块文件创建。
同时也创建引导区 hex 文件（通过在 "2.9.2 设置引导区工程" 中创建与其相同内容的文件）和闪存区 hex 文件。

图 2-65. 为闪存区域创建文件

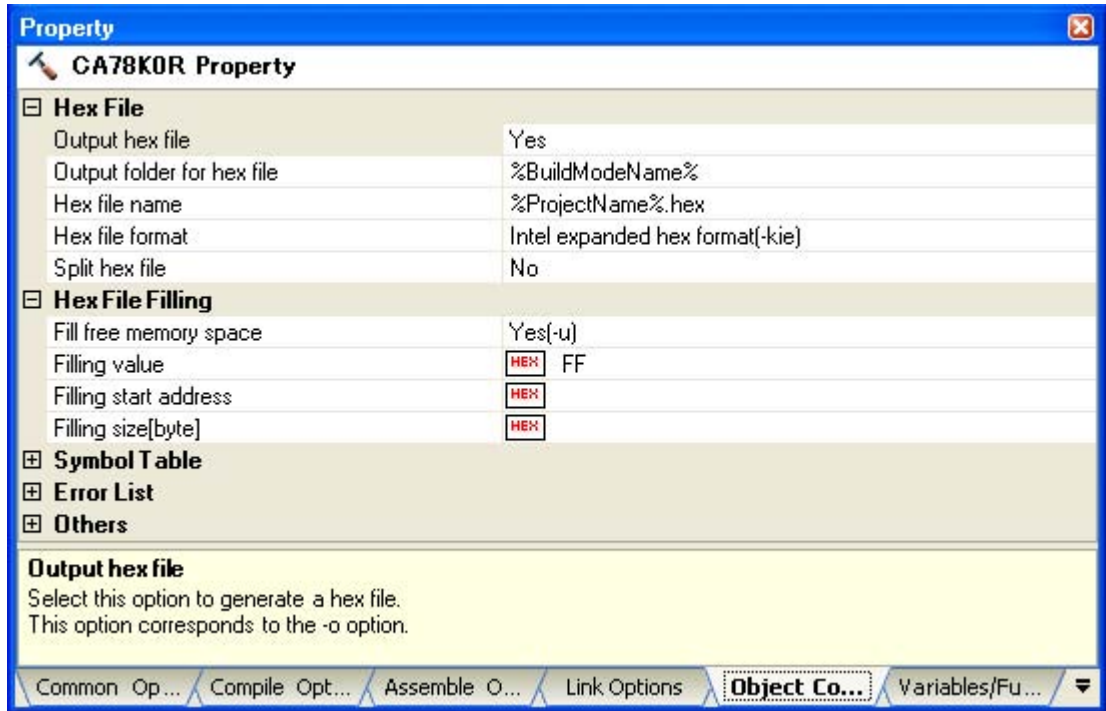


2.10 设置目标转换选项

为目标转换器设置选项时，在工程树上选择 Build tool 节点并在属性面板上选择 [Object Convert Options] 标签。可以通过在该列表中设定需要的属性设置不同的目标转换选项。

注意事项 对库工程此标签不会显示。

图 2-66. 属性面板 :[Object Convert Options] 标签



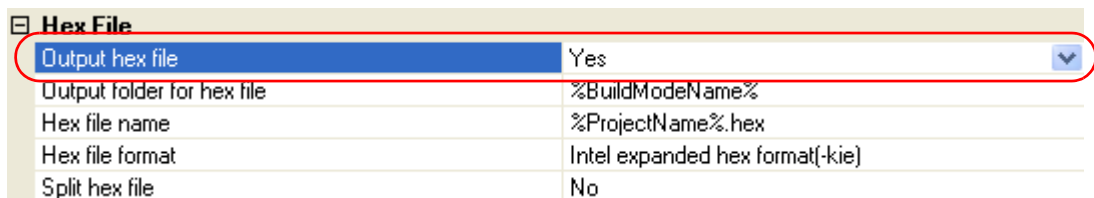
备注 经常使用的选项在 [Common Options] 标签上的 [Frequently Used Options(for Object Convert)] 分项下来取得。

2.10.1 设置 hex 文件的输出。

在工程树上选择 build tool 节点并在属性面板上选择 [Object Convert Options] 标签。

通过设置 [Hex File] 分项中的 [Output hex file] 属性输出 hex 文件 选择 [Yes](默认)，输出 hex 文件，选择 [No(-no)]，不输出 hex 文件。

图 2-67. [Hex File] 分项



备注 如果在 [Output hex file] 属性上选择 [No(-no)]，当进行目标转换时只输出一个符号表文件，则可以减少目标转换的时间。

输出 hex 文件时，可以设置输出文件夹和输出文件名。

(1) 选择输出文件夹

通过设置 [Output folder for hex file] 属性或通过 [...] 按钮直接输入来设置输出文件夹。文本框内可以设定多达 259 个字符。"%BuildModeName%" 为默认设置。"%BuildModeName%" 是嵌入式宏。它替换为构建模式名称。

(2) 设置输出文件名

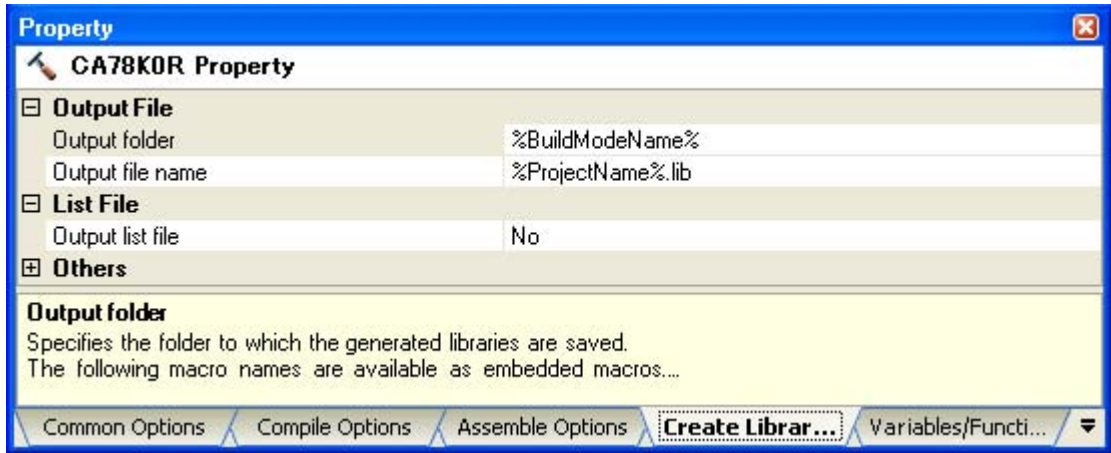
通过在 [Hex file name] 属性上直接输入来设置输出文件。文本框内可以设定多达 259 个字符。"%ProjectName%" 为默认设置。"%ProjectName%.hex" 是嵌入式宏。它替换为工程名称。

2.11 设置创建库选项

设置库选项，在工程树上选择 **Build tool** 节点并在属性面板上选择 **[Create Library Options]** 标签。可以通过此属性页给各种建库选项设置必要的值。

注意事项 仅对库工程显示此标签。

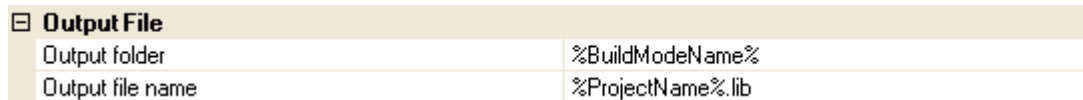
图 2-68. 属性面板: **[Create Library Options]** 标签



2.11.1 设置库文件的输出。

在工程树上选择 **build tool** 节点并在属性面板上选择 **[Create Library Options]** 标签。通过设置 **[Output File]** 分项输出库文件。

图 2-69. **[Output File]** 分项



(1) 选择输出文件夹

通过设置 **[Output folder]** 属性或通过 [...] 按钮直接输入来输出文件夹。文本框内可以设定多达 259 个字符。"%BuildModeName%" 为默认设置。"%BuildModeName%" 是嵌入式宏。它替换为构建模式名称。

(2) 设置输出文件名

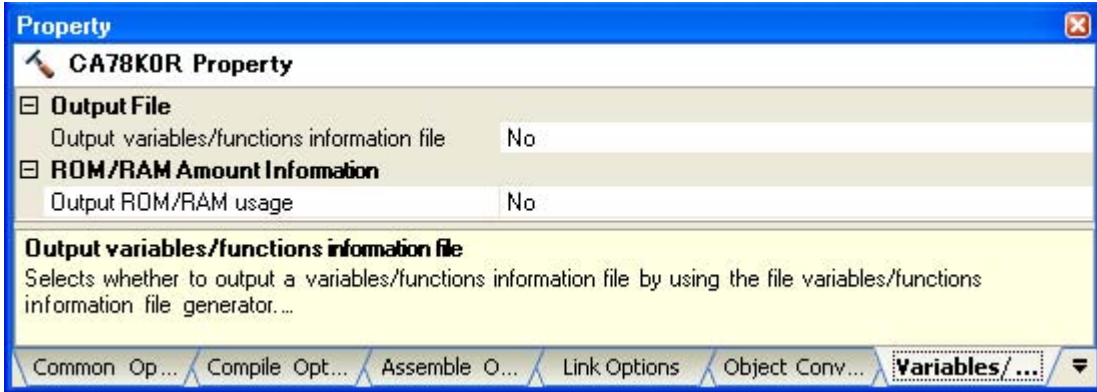
通过设置 **[Output file name]** 属性上直接输入来输出文件。文本框内可以设定多达 259 个字符。"%ProjectName%.lib" 被默认设置。"%ProjectName%.lib" 是嵌入式宏。它替换为工程名称。

2.12 变量 / 函数重定位选项

为变量 / 函数信息文件生成器进行选项设置时，在工程树上选择 Build tool 节点并在属性面板上选择 [Variables/Functions Relocation Options] 标签。

通过设置该标签中必要的属性，能设置不同的变量 / 函数重新分配选项。

图 2-70. 属性面板: [Memory Bank Relocation Options] 标签



2.12.1 有效分配的变量和函数

使用变量 / 函数信息文件生成器有效分配变量和函数。这个工具生成变量 / 函数信息文件（文件包含所有引用变量和函数的分配信息）。通过进行编译使用那个文件将分配变量到 `saddr` 区域，且分配函数到调用区域。

执行该操作的步骤如下。

- 自动生成变量 / 函数信息文件并分配变量和函数
- 编辑和使用自动生成的变量 / 函数信息文件

(1) 自动生成变量 / 函数信息文件并分配变量和函数

以下是关于自动生成变量 / 函数信息文件，以及通过构建使用文件分配变量和函数的步骤。

(a) 设置变量 / 函数信息文件的生成

在工程树上选择 build tool 节点并在属性面板上选择 [Variables/Functions Relocation Options] 标签。

通过设置 [Output variables/functions information file] 属性为 [Yes] 生成空的变量 / 函数信息文件，并加入到工程中（它也将出现在工程树的 File 节点中）。输出目标通过在 [Output folder for variables/functions information file] 属性和 [Variables/functions information file name] 属性中进行设置。

备注 如果存在具有相同名称的变量 / 函数信息文件，则使用它配置构建。

图 2-71. [Output variables/functions information file] 属性

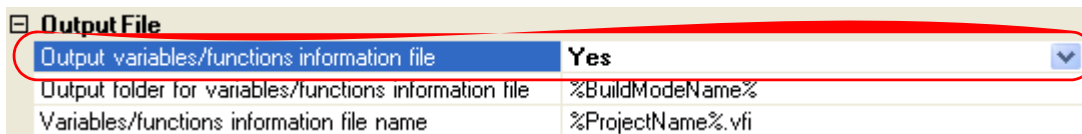
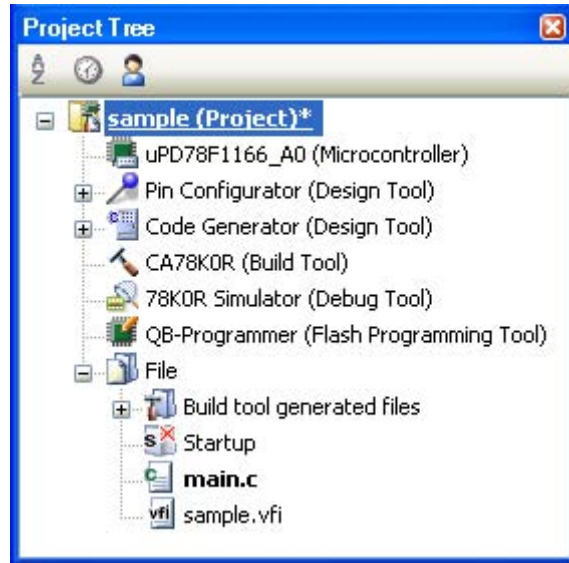


图 2-72. 工程树面板（在生成变量 / 函数信息文件之后）



变量 / 函数信息文件的输出文件夹和文件的设置可以改变。

<1> 选择输出文件夹

通过在 [Output folder for variables/functions information file] 属性上直接输入在文本框内或通过 [...] 按钮设置输出文件夹。文本框内可以设定多达 259 个字符。"%BuildModeName%" 为默认设置。

"%BuildModeName%" 是嵌入式宏。它替换为构建模式名称。

如果改变此属性，则生成空的变量 / 函数信息文件并添加到工程中（它也将出现在工程树的 Files 节点中）。

<2> 设置输出文件名

通过在 [Variables/functions information file name] 属性上直接输入设置输出文件。文本框内可以设定多达 259 个字符。"%ProjectName%.vfi" 为默认设置。"%ProjectName%.vfi" 是嵌入式宏。它替换为工程名称。

如果改变此属性，则生成空的变量 / 函数信息文件并添加到工程中（它也将出现在工程树的 Files 节点中）。

(b) 运行工程构建

运行工程构建。

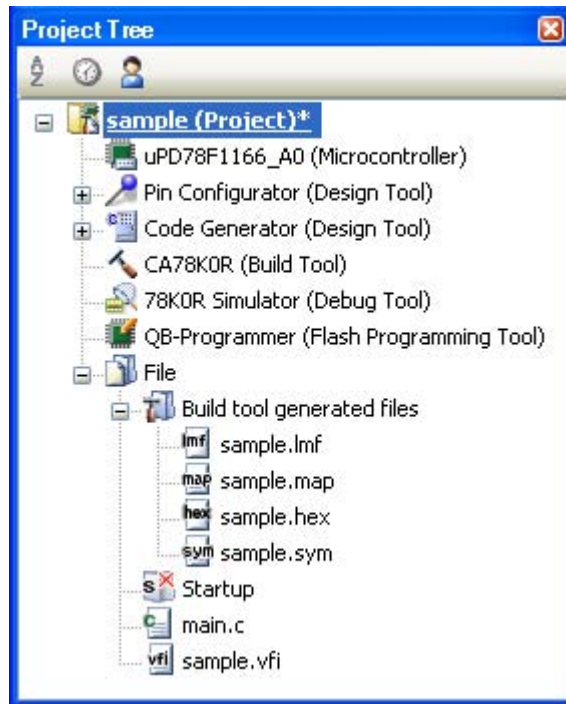
生成变量 / 函数信息文件。它将会自动输入到编译器中并重新构建。

备注 1. 通过运行重建，在（1）中会覆盖变量 / 函数信息文件。

2. 自从使用变量 / 函数信息文件重新生成对象，第二次构建将重建。

如果构建成功完成，生成带已分配变量和函数的载入模块文件。

图 2-73. 工程树面板（在生成载入模块文件之后）



(2) 编辑和使用自动生成的变量 / 函数信息文件

用户能编辑变量 / 函数信息文件。

以下是编辑在 (1) 中由用户生成的变量 / 函数信息文件的步骤和使用那个文件分配变量和函数的步骤。

(a) 编辑变量 / 函数信息文件

编辑在 (1) 中自动生成的变量 / 函数信息文件。

备注 有关自动生成的变量 / 函数信息文件格式的详细说明，参见 “3.7.1 变量 / 函数信息文件说明”。

根据以下格式描述变量 / 函数信息文件。

```

*** variable information ***
; 静态变量和常量
variable-name, number-of-references, size, reference-type, "file-name", const

; 全局变量和常量
variable-name, number-of-references, size, reference-type, , const

; 静态变量
variable-name, number-of-references, size, reference-type, "file-name"

; 全局变量
variable-name, number-of-references, size, reference-type

; 启动区域的全局变量和常量
variable-name, number-of-references, size, reference-type, , const, boot

; 启动区域的全部变量
variable-name, number-of-references, size, reference-type, , , boot

;*** Function information***
; 静态变量
function-name, number-of-references, size, "file-name"

; 全局变量
function-name, number-of-references, size

; 启动区域的全部函数
function-name, number-of-references, size, , , boot

```

备注 按优先级顺序描述变量和函数，从最高到最低。
对不能分配的变量和函数，要通过在行的开始处添加分号 (;)，并在其后做相关注释。

(b) 设置变量 / 函数信息文件的生成

在工程树上选择 build tool 节点并在属性面板上选择 [\[Variables/Functions Relocation Options\]](#) 标签。
在 [Output variables/functions information file] 属性上选择 [No]。

图 2-74. [Output File] 分项



(c) 运行工程构建

运行工程构建。
带分配变量和函数的已生成载入模块文件作为指定在变量 / 函数信息文件中。

注意事项 如果在工程中添加扩展名为 "vfi" 的文件，则作为变量 / 函数信息文件。如果添加到 Startup 节点中时，则作为变量 / 函数信息文件。
当添加变量 / 函数信息文件到工程中时，如果之前变量 / 函数信息文件已经添加，则构建仅指向最后添加的变量 / 函数信息文件，之前添加的文件不能指向。

当设置变量 / 函数信息文件时，如其没有作为构建目标被指向，而其它变量 / 函数信息文件已添加，则构建指向该文件，且不指向其它文件。

2.12.2 显示 ROM/RAM 用法

在链接到输出面板后，能使用变量 / 函数信息文件生成器来显示 ROM/RAM 用法。

在工程树上选择 build tool 节点并在属性面板上选择 [Variables/Functions Relocation Options] 标签。

显示 ROM/RAM 用法，在 [ROM/RAM Amount Information] 分项的 [Output ROM/RAM usage] 属性上选择 [Yes] (默认选择为 [No])。

图 2-75. [Output ROM/RAM usage] 属性



当运行构建时，根据下列的构建结果，ROM/RAM 用法输出到输出面板。

首先输出全部所有使用，其次为每个存储区域的用法。

图 2-76. ROM/RAM 用法显示

```

Output
===== Start build all(Wednesday, August 27, 2008 7:32:54 PM) =====
----- BuildStart(sample, DefaultBuild) -----
>vf78k0r.exe
>..\src\main.c
>lk78k0r.exe
>vf78k0r.exe
*** Memory Area Information ***
ROM : 302H byte(s) real data
RAM : 376H byte(s) real data
*** Memory Area Information in ROM ***
ROM : 302H byte(s)
*** Memory Area Information in RAM ***
RAM : 376H byte(s)
>oc78k0r.exe
----- BuildEnded(Error:0, Warning:0) -----
===== Build ended(Success:1 Projects, Failed:0 Projects)(Wednesday, August 27, 2008 7:32:57 PM) =====
[EOF]
  
```

2.13 单独设置构建选项

构建选项可设置在工程中或文件层面中。

- 工程层：参阅 "2.13.1 在工程层设置构建选项"
- 工程层：参阅 "2.13.2 在文件层设置构建选项"

2.13.1 在工程层设置构建选项

为设置工程构建选项 (主工程或子工程), 在工程树上选择 **Build tool** 节点并显示属性面板。选择组件标签, 并通过设置必要的属性设置构建选项。

- 编译器: [Compile Options] 标签
- 汇编程序: [Assemble Options] 标签
- 链接器: [Link Options] 标签
- 目标转换器: [Object Convert Options] 标签
- 库管理程序: [Create Library Options] 标签
- 变量 / 函数信息文件生成器: [Variables/Functions Relocation Options] 标签

2.13.2 在文件层设置构建选项

可以分别为添加到工程中的每个源文件设置编译和汇编选项。

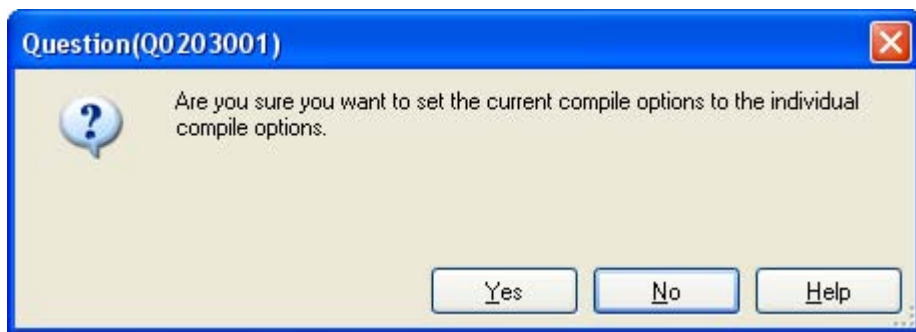
(1) 为 C 源文件设置编译选项

在工程树上选择 C 源文件并在属性面板上选择 [Build Settings] 标签。在 [Build] 分项中, 如果在 [Set individual compile option] 属性上选择 [Yes], 显示标签如下:

图 2-77. [Set individual compile option] 属性

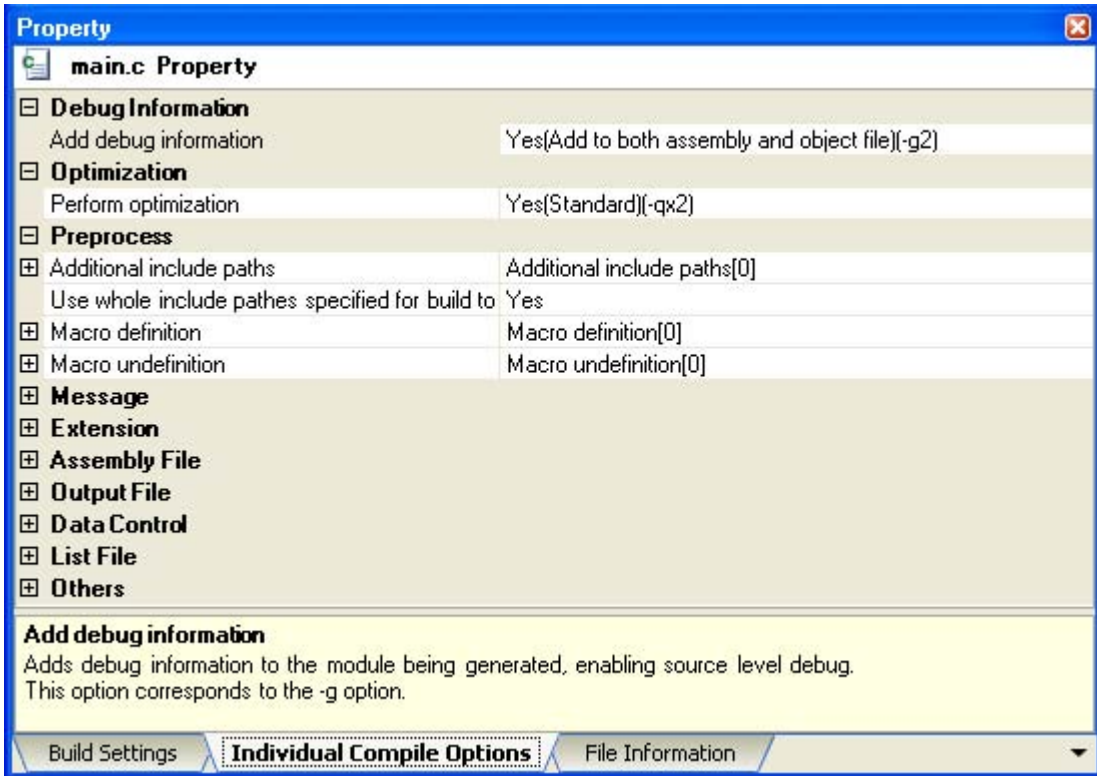


图 2-78. 消息对话框



如果在对话框中点击 [Yes] 按钮, 将显示 [Individual Compile Options] 标签。

图 2-79. 属性面板 :[Individual Compile Options] 标签



可以通过设置该标签中必须设置的属性，为 C 源文件设置编译选项。注意，此标签代替了 [Compile Options] 标签的设置（默认）。

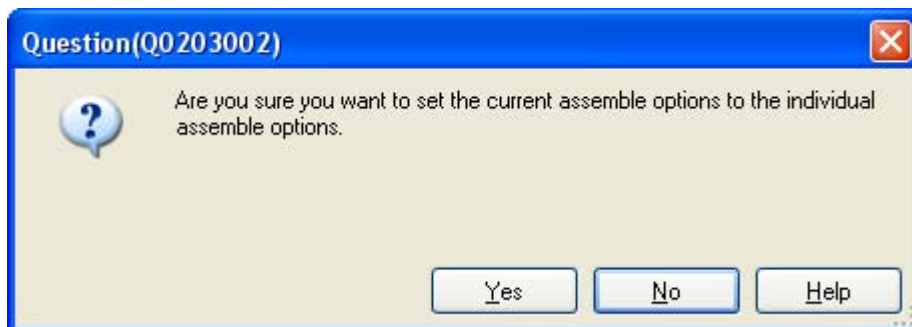
(2) 为汇编源文件设置汇编

在工程树上选择汇编源文件并在属性面板上选择 [Build Settings] 标签。在 [Build] 分项中，如果在 [Set individual assemble option] 属性上选择 [Yes], 显示标签如下：

图 2-80. [Set individual assemble option] 属性

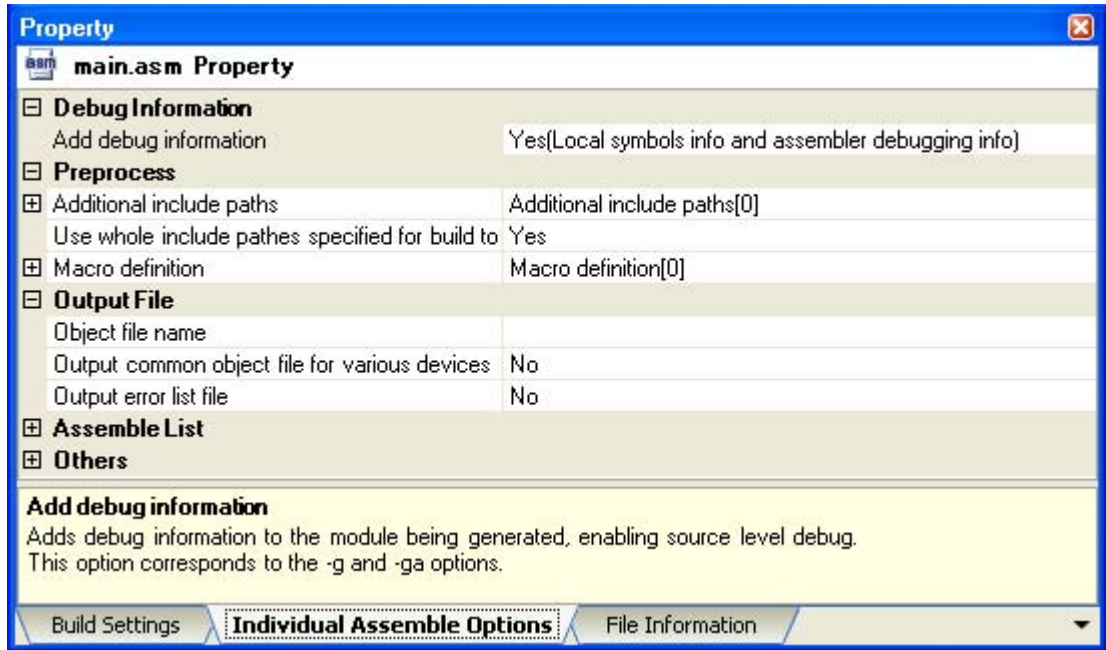


图 2-81. 消息对话框



如果在对话框中点击 [Yes] 按钮，将显示 [Individual Assemble Options] 标签。

图 2-82. 属性面板 :[Individual Assemble Options] 标签



通过设置该标签中必须设置的属性，为汇编源文件设置汇编选项。注意，此标签代替了 [Assemble Options] 标签的设置（默认）。

备注 可以由 C 源文件生成对应的汇编文件设置汇编选项。在工程树上选择 C 源文件并在属性面板上选择 [Individual Compile Options] 标签。如果在 [Assembly File] 分项的 [Output assemble file] 上选择 [Yes], 显示 [Individual Assemble Options] 标签。

2.14 设置构建操作

本章节介绍构建操作。

- 设置文件的链接顺序
- 改变子工程的文件构建顺序
- 显示构建选项列表
- 改变文件构建目标工程
- 添加构建模式
- 改变构建模式
- 删除构建模式
- 设置当前构建选项作为工程的标准

2.14.1 设置文件的链接顺序

目标模块文件和库文件的连接顺序自动确定，但是用户也可以设置该顺序。

在工程树上，选择 **Build tool** 节点，然后在快捷菜单中选择 [Set Link Order...] 打开链接顺序对话框。

图 2-83. 链接顺序对话框



下述文件名依照文件输入链接器时的顺序在 [File] 中列表。

- 添加到主工程或子工程的从源文件中生成的目标模块文件
- 直接添加到主工程和子工程工程树的目标模块文件
- 直接添加到主工程和子工程工程树的库文件

备注 默认顺序为文件添加到工程时的顺序。

通过新的源文件创建的目标模块文件和新添加的目标模块文件添加到列表中的最后的目标模块文件之后。

新添加的库文件添加到列表的末尾。

通过改变文件的显示顺序，可以设置文件输入到链接器的顺序。

要改变显示顺序，使用 [Up] 和 [Down] 按钮或拖放文件名。在改变显示顺序后，单击 [OK] 按钮。

2.14.2 改变子工程的文件构建顺序

以先子工程后主工程的顺序进行构建，但当添加多个子工程时，创建子工程的顺序为它们在工程树中的显示顺序。要在工程树中改变子工程的显示顺序，拖动要移动的子工程将其放在目标位置即可。

2.14.3 显示构建选项列表

可以显示当前工程（主工程和子工程）在属性面板上设置的创建选项列表。

如果从 [Build] 菜单中选择 [Build Option List]，当前工程的选项设置通过输出面板依照创建顺序在 [Build Tool] 中显示。

备注 可以改变创建选项列表的显示格式。

在工程树上选择 Build tool 节点，并在属性面板上选择 [Common Options] 标签。设置 [Others] 中的 [Format of build option list] 属性。

图 2-84. [Format of build option list] 属性

| Others | |
|---|--|
| Output message format | %FileName% |
| Format of build option list | %FileName% : %Program% %Options% |
| Temporary folder | |
| Commands executed before build processing | Commands executed before build processing[0] |
| Commands executed after build processing | Commands executed after build processing[0] |

"%FileName% : "%FileName% : %Program% %Options%" 按照默认设置。

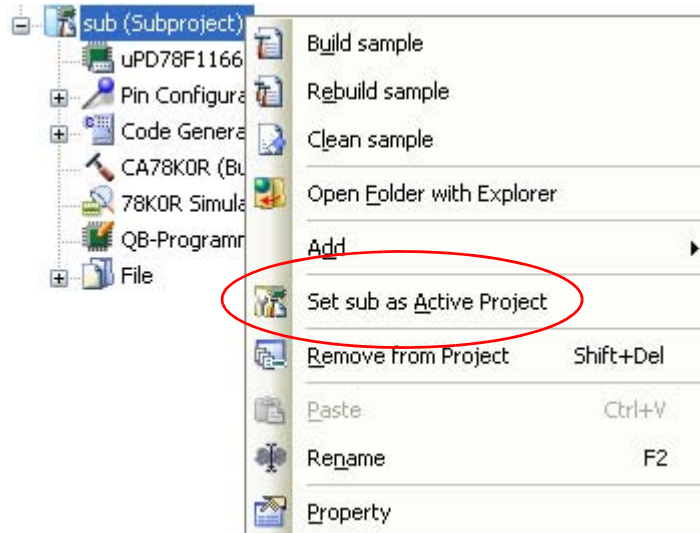
"%FileName%", "%Program%", and "%Options%" 为嵌入宏。它们被替换为正在创建的文件名，执行的程序名和创建执行中的命令行选项。

2.14.4 改变文件构建目标工程

当运行指定工程（主工程或子工程）的构建程序时，用户必须设置该工程为“活动工程”。

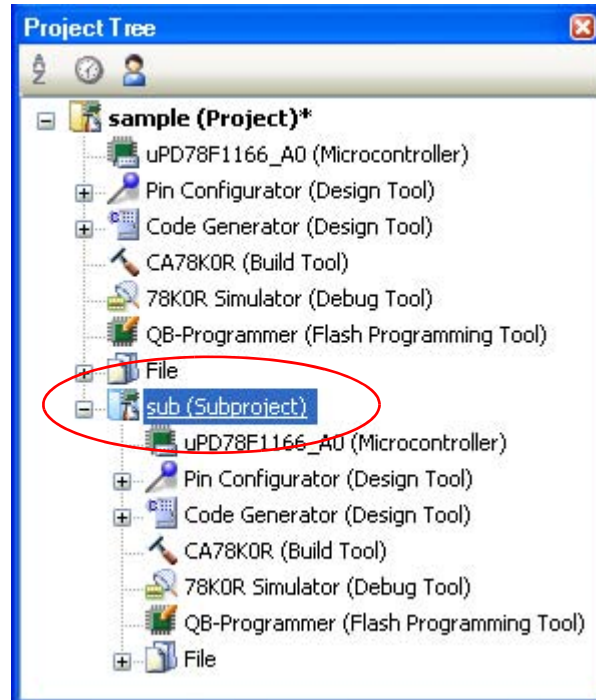
设置活动工程，在工程树中选择要作为活动工程的主工程或子工程并从快捷菜单中选择 [set selected subproject as Active Project]。

图 2-85. [Set selected project as Active Project] 条目



当工程被设置成活动工程时，该工程添加下划线。

图 2-86. 活动工程



- 备注 1.** 在创建工程后，主工程即成为活动工程。
2. 当在工程中移除已设置成为活动工程的子工程时，主工程将成为活动工程。

2.14.5 添加构建模式

当用户希望根据创建的目的改变构建选项和宏定义时，可以选择性的改变这些设置。构建选项和宏定义设置被归入“build mode”选项中，通过改变“build mode”，用户可以不必在每次都改变构建选项和宏定义设置。

构建模式默认为“DefaultBuild”。根据构建的目的添加构建模式。

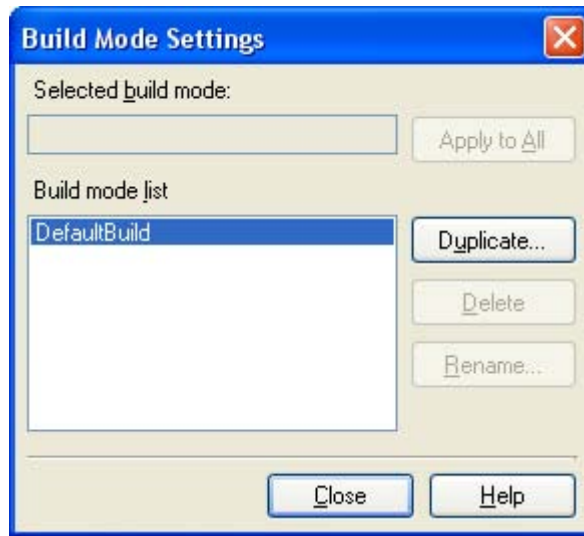
通过以下步骤添加构建模式。

(1) 创建新的构建模式

通过复制现有的构建模式创建新的构建模式。

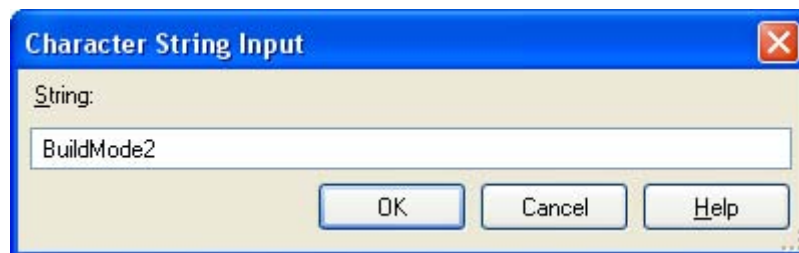
从 [Build] 菜单中选择 [Build Mode Settings...]。打开构建模式设置对话框。

图 2-87. 构建模式设置对话框



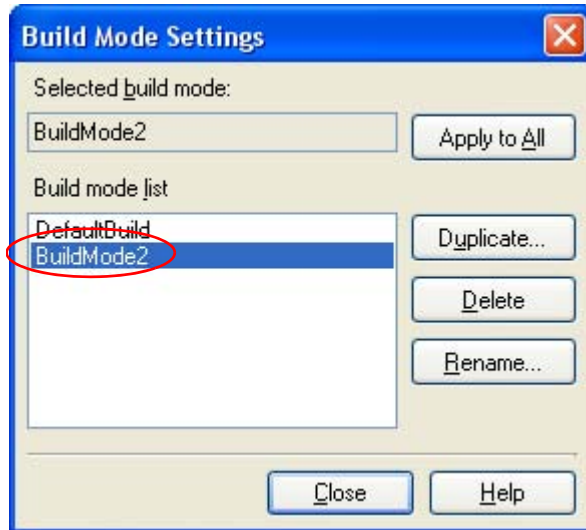
从构建模式列表中选择要复制的构建模式并单击 [Duplicate] 按钮。打开字符串输入对话框。

图 2-88. 字符串输入对话框



在对话框中，键入要创建的构建模式的名称然后单击 [OK] 按钮。具有该名的构建模式将被复制。创建的构建模式添加到属于该工程的主工程和所有子工程的构建模式中。

图 2-89. 构建模式设置对话框（在添加构建模式之后）

**(2) 改变构建模式**

改变构建模式为新创建的构建模式（参阅“2.14.6 改变构建模式”）。

(3) 改变构建模式的设置

选择工程树上的 **Build tool** 节点并改变在属性面板上的构建选项和宏定义设置。

备注 创建构建模式后，则认为工程改变。当关闭工程时，系统询问确认是否保存构建模式。

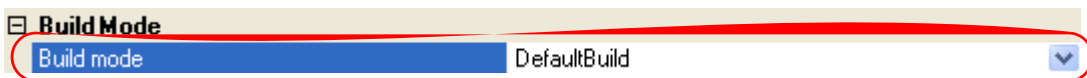
2.14.6 改变构建模式

当用户希望根据创建的目的改变构建选项和宏定义时，可以选择性的改变这些设置。构建选项和宏定义设置被归入“build mode”选项中，通过改变“build mode”，用户可以不必在每次都改变构建选项和宏定义设置。

(1) 改变主工程和子工程的构建模式

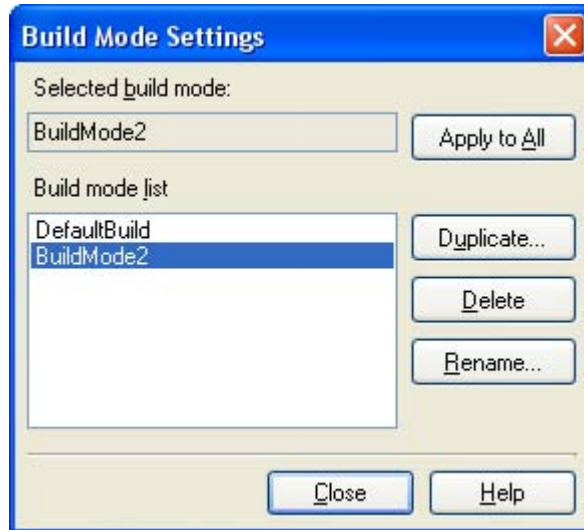
在工程树上选择目标工程的 **Build tool** 节点，并在属性面板上选择 **[Common Options]** 标签。在 **[Build Mode]** 分项中的 **[Build mode]** 上选择改变构建模式。

图 2-90. [Build Mode] 属性

**(2) 为整个工程改变构建模式**

从 **[Build]** 菜单中选择 **[Build Mode Settings...]**。打开构建模式设置对话框。

图 2-91. 构建模式设置对话框



如果选择从构建模式列表中改变构建模式，选中的构建模式显示在 [Selected build mode] 中。如果单击 [Apply to All] 按钮，主工程的构建模式和所有隶属于工程的子工程将改变为在对话框中选择的构建模式。

注意事项 如果工程选择的构建模式不存在，则根据所选构建模式名称由 "DefaultBuild" 复制，且构建模式被改变为复制的构建模式。

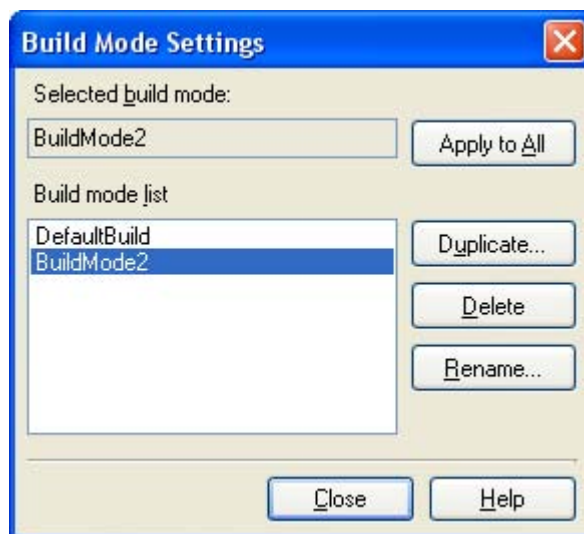
- 备注 1.** 构建模式默认为 "DefaultBuild"。参阅 "2.14.5 添加构建模式" 中关于添加构建模式方法的说明。
2. 通过从构建模式列表中选择构建模式并单击 [Rename...] 按钮，可以改变构建模式的名称。不能改变 "DefaultBuild" 的名称。

2.14.7 删除构建模式

通过构建模式设置对话框删除构建模式。

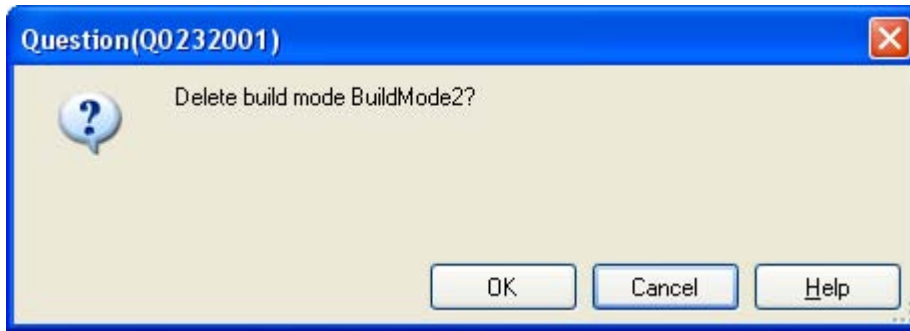
从 [Build] 菜单中选择 [Build Mode Settings...]。打开对话框。

图 2-92. 构建模式设置对话框



从构建模式列表中选择要删除的构建模式并单击 [Delete] 按钮。打开消息对话框。

图 2-93. 消息对话框



单击对话框中的 [OK] 按钮，继续操作。
选中的构建模式从工程中删除。

注意事项 不能删除 "DefaultBuild"。

2.14.8 设置当前构建选项作为工程的标准

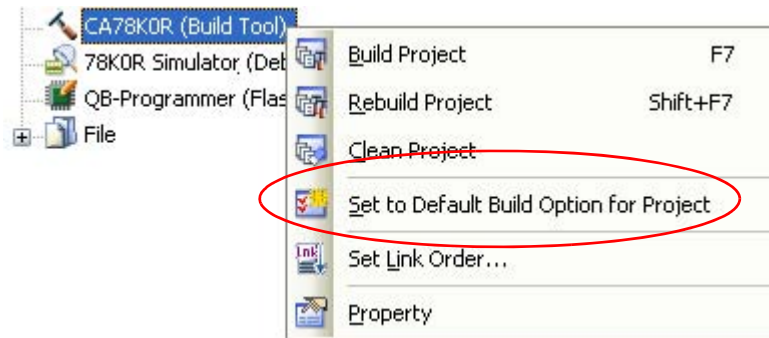
在属性面板上，如果为标准构建选项中添加设置变更，则该属性的值将加粗显示。

图 2-94. 属性面板（在改变标准构建选项之后）



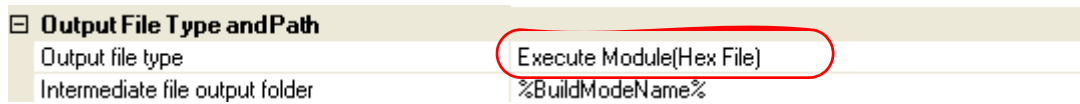
为当前选中标准构建选项（移除粗体字）的工程设置构建选项（主工程或子工程），在工程树中选择 Build tool 节点并从快捷菜单中选择 [Set to Default Build Option for Project]。

图 2-95. [Set to Default Build Option for Project] 条目



在设置它们作为标准创建选项之后的属性值显示如下。

图 2-96. 属性面板（在设置标准构建选项之后）



注意事项 选中主工程后，仅可以设定主工程。即使添加子工程，也不会设定其设置。

2.15 执行创建

本章节介绍执行构建操作。

(1) 构建类型

有下列构建类型。

表 2-1. 构建类型

| 类型 | 说明 |
|------|---|
| 构建 | 构建目标文件之外，仅执行升级文件的构建。 参见 "2.15.1 运行升级文件的构建" |
| 重新构建 | 执行对所有创建目标文件的构建。 参见 "2.15.2 执行所有文件的构建" |
| 快速构建 | 执行平行的构建与其他操作。 参见 "2.15.3 运行与其他操作平行的构建。" |
| 批量构建 | 通过工程中的构建模式，执行批量构建。 参见 "2.15.4 运行构建模式中的批量构建。" |

- 备注 1.** 按照子工程、主工程的顺序执行构建。
按照在工程树中的显示顺序创建子工程 (见 "2.14.2 改变子工程的文件构建顺序")。
- 2.** 在运行构建，重新构建，或批量构建时，如果有文件在 **编辑器面板** 中编辑，所有这些文件将被保存。

(2) 显示执行结果

构建的执行结果 (输出 build tool 的消息) 显示在**输出面板**上的每个标签里。

- 构建、重新创建、或批量创建: [All Messages] 标签和 [Build Tool] 标签
- 快速构建: [Rapid Build] 标签

图 2-97. 构建执行结果 (构建、重新构建或批量创建)

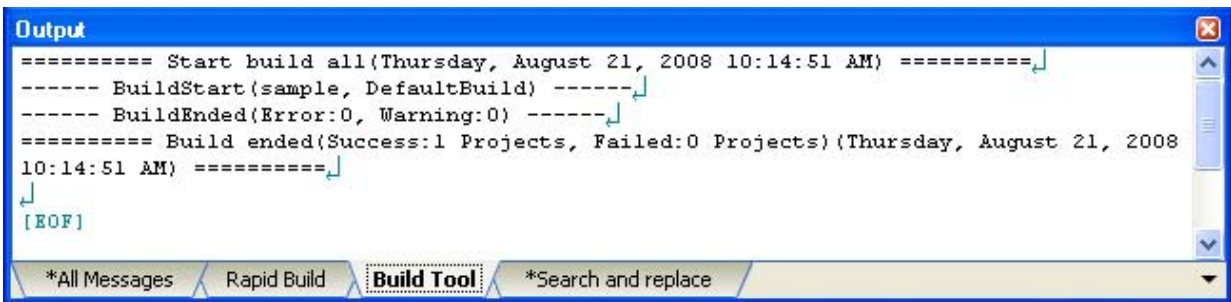
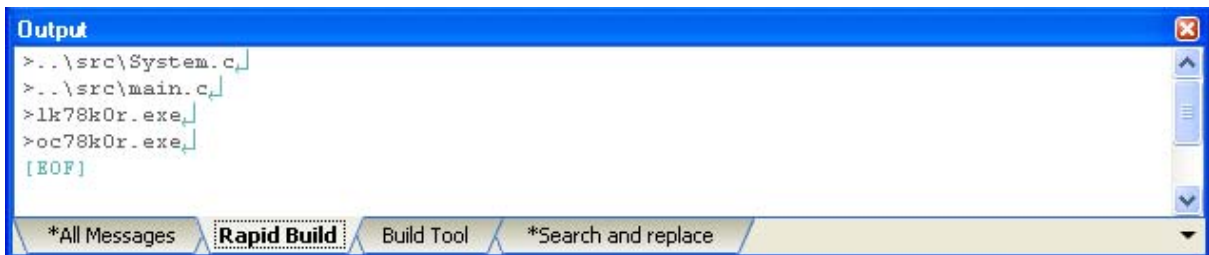


图 2-98. 构建执行结果 (快速构建)

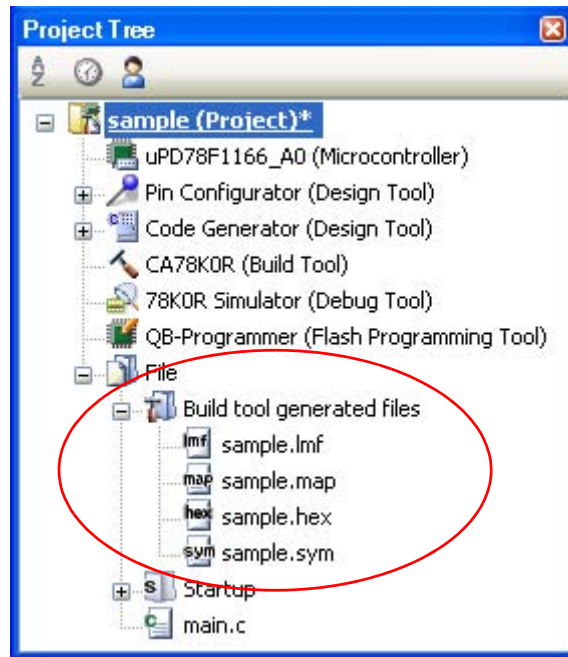


- 备注 1.** [Rapid Build] 标签中的文本变淡。
- 2.** 文件从输出消息中获得名称或行号，如果双击此消息，可以跳至此文件的相关行。

3. 当光标在显示警告或出错消息的行上按下 [F1], 可以显示与行消息相关的在线帮助。

文件由工程树面板面板下的 Build tool 创建至 Build tool generated files 节点下。

图 2-99. Build tool 生成文件



备注 文件显示在 Build tool generated files 节点之下，如下所示。

- 除库工程以外
 - 载入模块文件 (*.lmf)
 - 链接列表文件 (*.map)
 - 错误列表文件 (*.elk)
 - 十六进制文件 (*.hex, *.hxb, *.hxf)
 - 符号表格文件 (*.sym)
 - 出错列表文件 (*.eoc)
- 用于库工程
 - 库文件 (*.lib)
 - 列表文件 (*.lst)


注意事项 在构建中建立 Build tool generated files 节点。
如果构建后重新载入工程，该节点将不再出现。

2.15.1 运行升级文件的构建

构建目标文件之外，运行仅升级文件的构建。（后指“构建”）。

在整个工程（主工程和子工程）或活动工程中运行构建操作（见“2.14.4 改变文件构建目标工程”）。

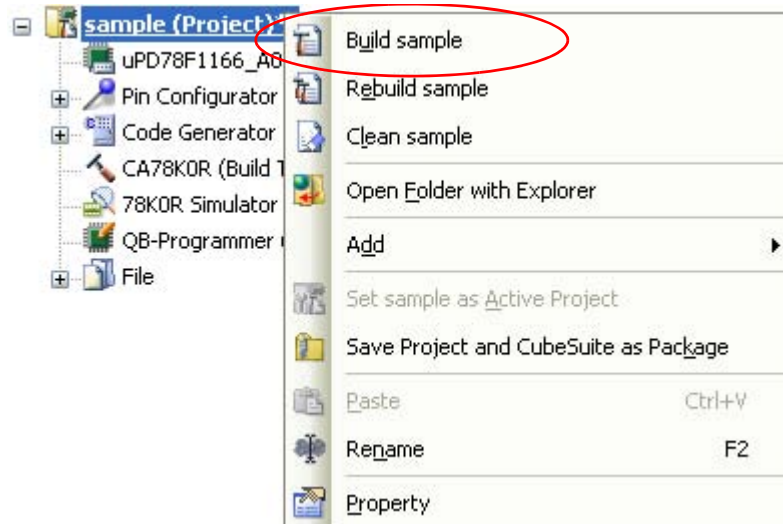
(1) 运行整个工程的构建

在工具栏上单击 。

(2) 运行活动工程的构建

选择工程，然后在快捷菜单中选择 [Build active project]。

图 2-100. [Build active project] 条目




备注 如果编辑头文件后相应的源文件未构建，则一旦执行此构建，则升级此文件依赖关系（见“2.3.7 升级文件依赖关系”）。

2.15.2 执行所有文件的构建

执行对所有创建目标文件的一次构建（后指“重建”）。

对整个工程（主工程和子工程）或活动工程执行重新构建操作（见“2.14.4 改变文件构建目标工程”）。

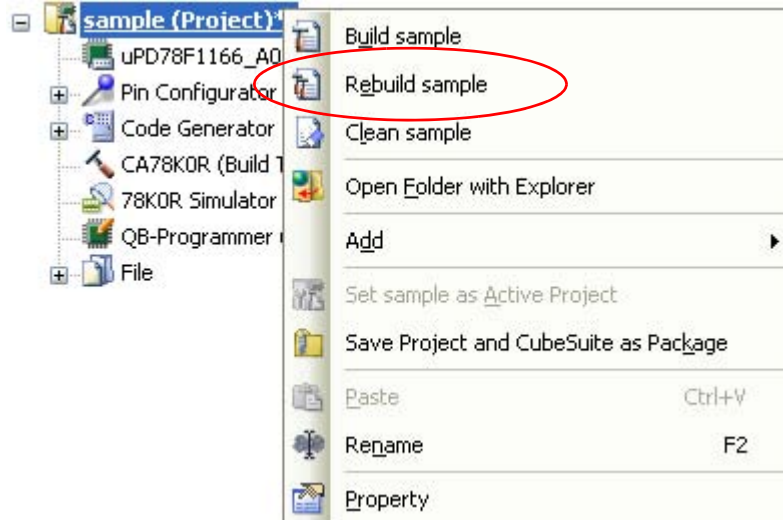
(1) 执行整个工程的重建

在工具栏上单击 。

(2) 执行活动工程的重建

选择工程，然后在快捷菜单中选择 [Rebuild active project]。

图 2-101. [Rebuild active project] 条目



2.15.3 运行与其他操作平行的构建。

CubeSuite 在下列事件之一发生时，自动开始构建（后指“快速构建”）。

- C 源文件，汇编源文件，或头文件添加到工程中并保存
- 构建目标文件添加到工程或从工程中移除
- 目标模块文件和库文件的连接顺序改变
- Build tool 或构建目标文件的属性改变时

如果设定快速构建被，可以运行与上述操作平行的构建。

要启用 / 禁用快速构建，从 [Build] 菜单中选择 [Rapid Build]。快速构建默认是启用的。

图 2-102. [Rapid Build] 条目 (当快速构建有效时)

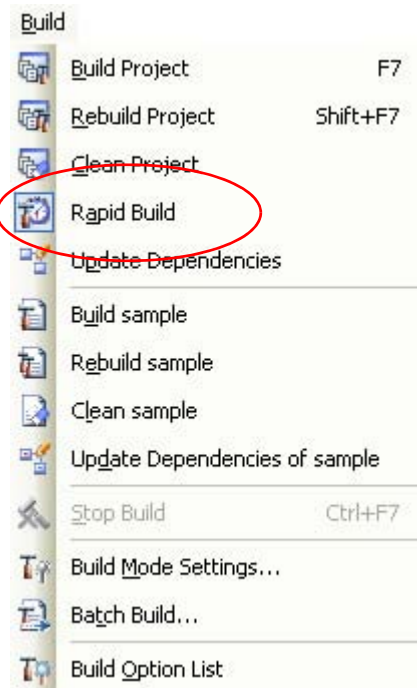
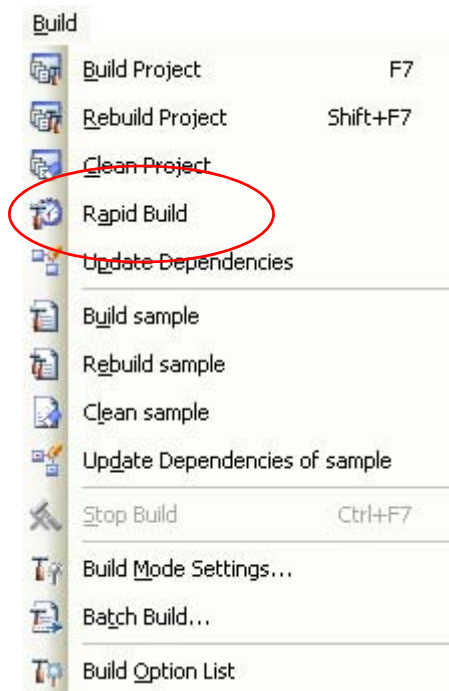


图 2-103. [Rapid Build] 条目 (当快速构建无效时)



- 备注 1.** 在编辑源文件后，建议经常按 [Ctrl] + [S] 键保存。
2. 在整个工程中设置启用 / 禁用快速构建（主工程和子工程）。
 3. 如果在执行时禁用快速构建，则快速构建在那个时刻将停止。

注意事项 该功能只有在 [编辑器面板](#) 上编辑源文件时才有效。

2.15.4 运行构建模式中的批量构建。

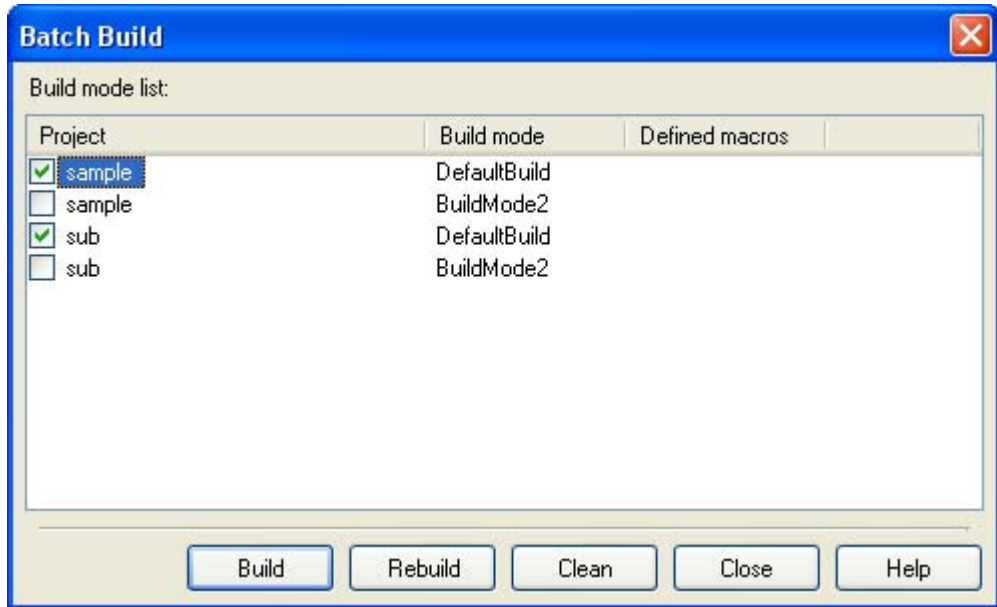
可以通过批量的方式运行构建、重建和清除工程的构建模式（主工程和子工程）（后指“批量构建”）。

备注 参阅以下有关构建，重建和清除的章节。

- Build: 参见 "2.15.1 运行升级文件的构建".
- Rebuild: 参见 "2.15.2 执行所有文件的构建".
- Clean: 参见 "2.15.8 删除中间件文件和生成的文件".

从 [Build] 菜单中选择 [Batch Build]。打开批量构建对话框。

图 2-104. 批量构建对话框



在对话框中，显示当前打开工程中主工程和子工程名称联合列表以及它们的构建模式和宏定义。

从主工程和子工程以及构建模式的联合列表复选框中选择批量构建，然后单击 [Build]、[Rebuild] 或 [Clean] 按钮。

备注 批量构建根据子工程及主工程的顺序执行。

独立的主工程或子工程选择多重构建模式时，所有选中的构建模式的子工程运行构建后，下一个子工程或主工程的构建才能运行。

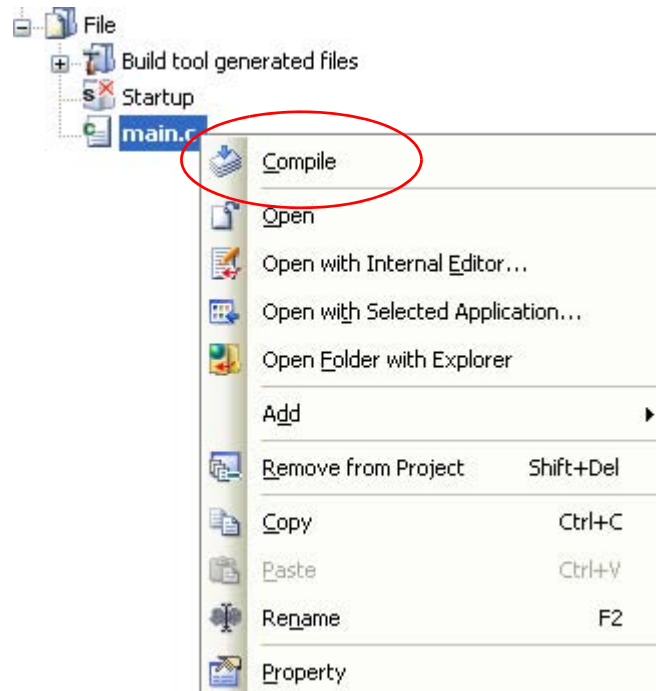
2.15.5 编译 / 汇编 独立的文件

可以编译或汇编每个添加到工程中的源文件。

(1) 编译 C 源文件

在工程树上选择 C 源文件并从快捷菜单中选择 [Compile]。

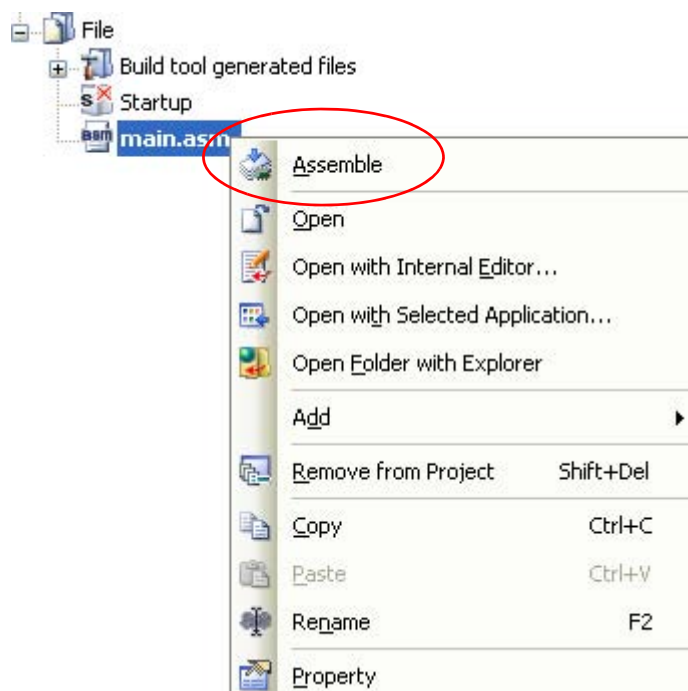
图 2-105. [Compile] 条目




(2) 汇编汇编源文件

在工程树上选择汇编源文件并从快捷菜单中选择 [Assemble]。

图 2-106. [Assemble] 条目



2.15.6 停止运行构建

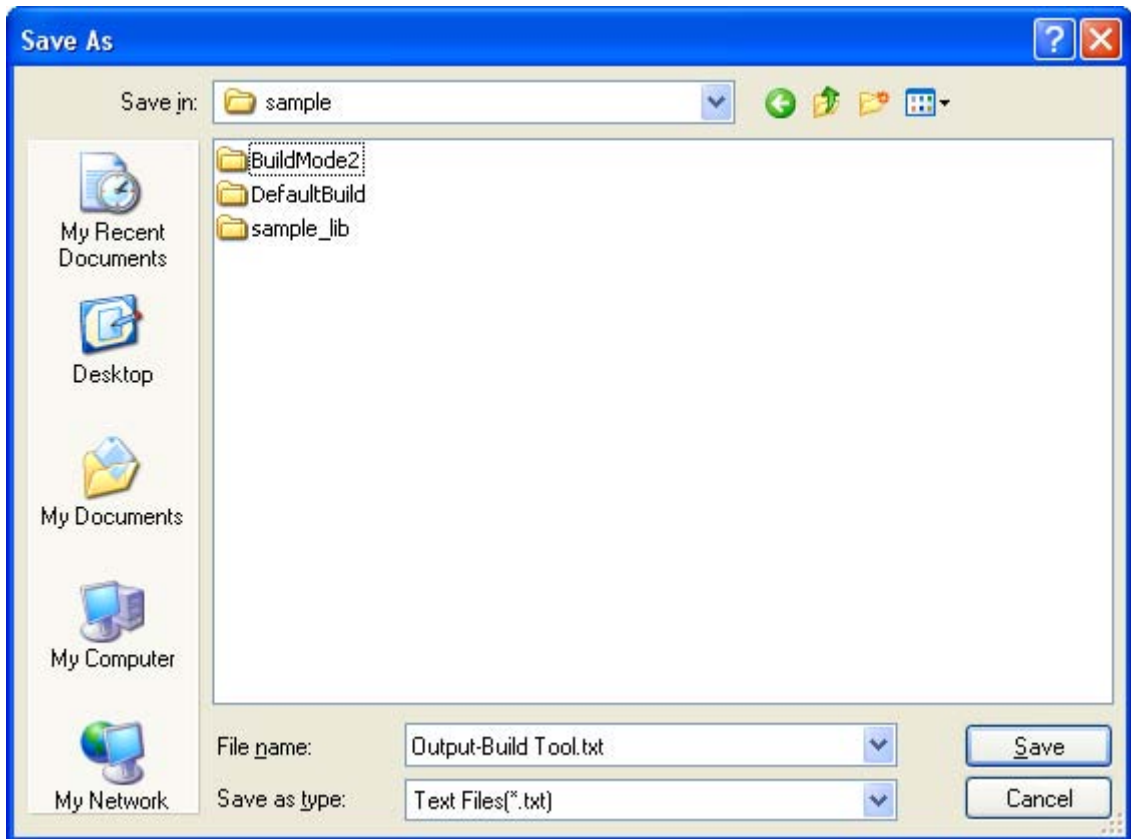
停止运行构建，重建，或批量构建，在工具栏上单击 。

2.15.7 保存构建结果到文件中

可以保存构建的执行结果（输出 Build tool 的消息）并显示在 [输出面板](#)上。

在面板上选择 [Build Tool] 标签，然后从 [File] 菜单中选择 [Save Output - Build Tool As...]。打开另存为对话框。

图 2-107. 另存为对话框



在对话框中，指定要保存的文件并点击 [Save] 按钮。

2.15.8 删除中间件文件和生成的文件

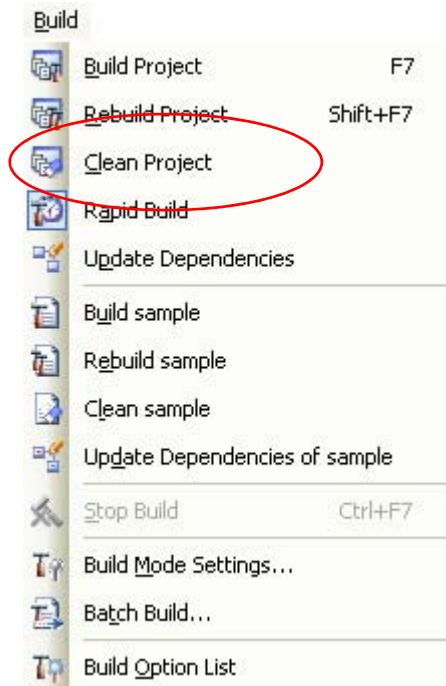
可以删除所有的中间件文件及通过运行构建输出的生成文件（后指“清除”）。

对整个工程（主工程和子工程）或活动工程运行清除操作（见“2.14.4 改变文件构建目标工程”）。

(1) 运行整个工程的清除

通过 [Build] 菜单，选择 [Clean Project]。

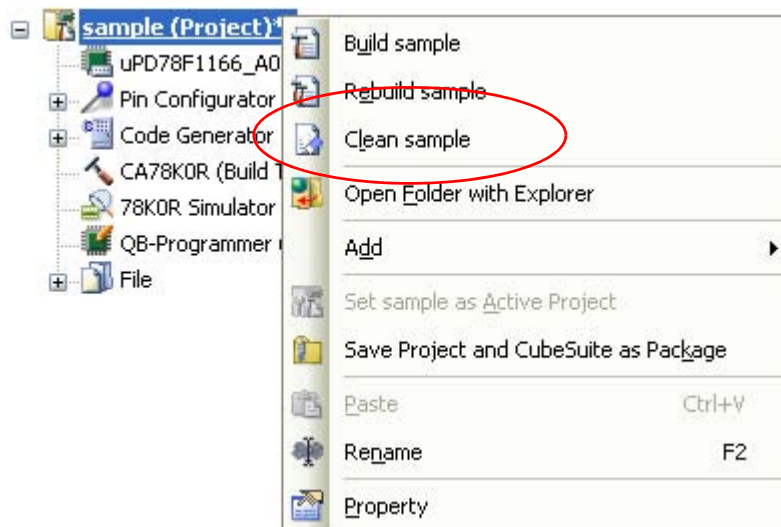
图 2-108. [Clean Project] 条目



(2) 运行活动工程的清除

选择工程，然后在快捷菜单中选择 [Clean active project]。

图 2-109. [Clean active project] 条目



2.16 堆栈使用跟踪程序

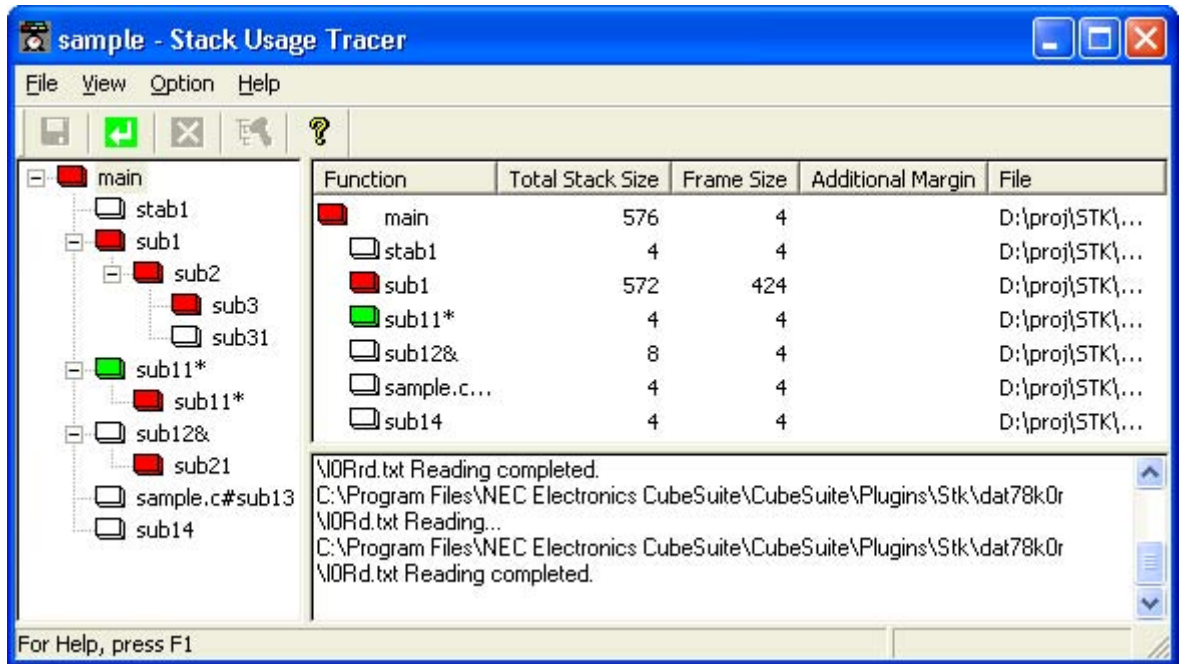
堆栈使用跟踪程序执行静态分析，并以树型格式显示由功能调用的函数，以及每个函数的堆栈信息（函数名、总堆栈大小、帧大小、附加容限和文件名）。

2.16.1 开始和退出

要从 **主窗口** 启动堆栈使用跟踪程序，选择 [Tool] 菜单 >> [Startup Stack Usage Tracer]。

在堆栈跟踪程序启动完成后，在 **堆栈用法跟踪窗口** 中树型显示区 / 列表显示区中将显示每个函数的函数调用关系和堆栈信息。

图 2-110. 启动堆栈使用跟踪程序

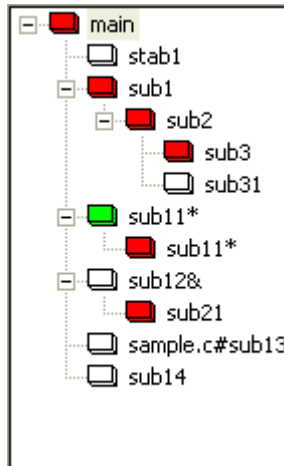




从 **堆栈用法跟踪窗口** 中退出堆栈使用跟踪程序，选择 [File] 菜单 >> [Exit sk78k0]。






2.16.2 检查调用关系

用户可以在堆栈用法跟踪窗口中的树型显示区检查函数调用关系。

图 2-111. 树型显示区



备注 在此表之下说明的是这些图标的含义，这些图标显示在字符串左边用来描述有关功能名称。
图标显示优先级是从高： 到低：.








| | |
|---|--|
|  | 此功能可以直接通过占有最大总堆栈大小来调用指定功能 |
|  | 信息 (附加空间、递归深度、以及调用功能) 已修改通过调整栈堆大小对话框或者特定文件的堆栈大小。 |
|  | 重现功能 |
|  | 堆栈用法跟踪器不需要获得此功能的任何堆栈信息 |
|  | 除上面之外 |


2.16.3 检查堆栈信息

可以从堆栈用法跟踪窗口的列表显示区检查堆栈信息（函数名，总堆栈大小，帧大小，附加容限和文件名）。

- 总堆栈大小（包括 callee 函数的堆栈大小）。
- 帧大小（不包括 callee 函数的堆栈大小）
- 附加容限（强制添加到帧大小里的值）

图 2-112. 列表显示区

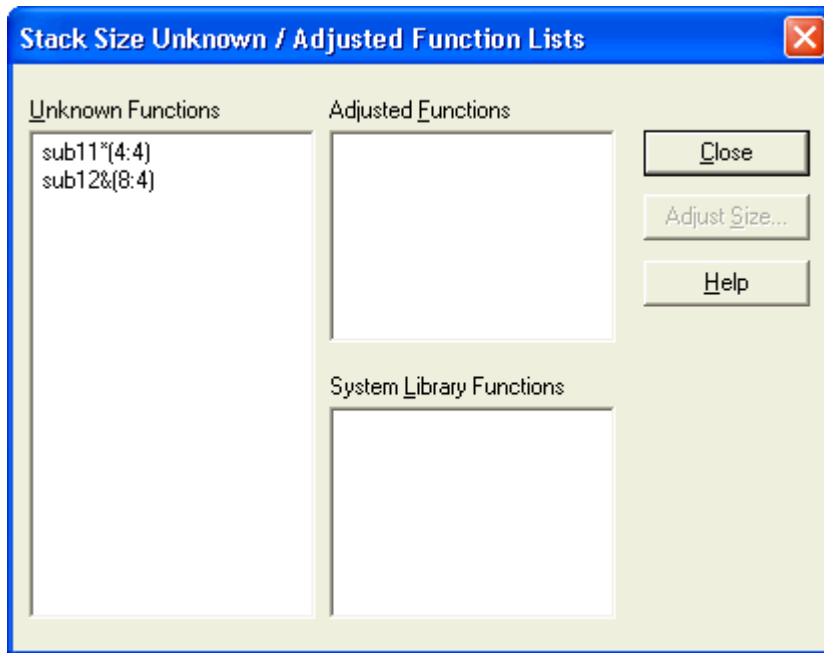
| Function | Total Stack Size | Frame Size | Additional Margin | File |
|---|------------------|------------|-------------------|-----------------|
|  main | 576 | 4 | | D:\proj\STK\... |
|  stab1 | 4 | 4 | | D:\proj\STK\... |
|  sub1 | 572 | 424 | | D:\proj\STK\... |
|  sub11* | 4 | 4 | | D:\proj\STK\... |
|  sub12& | 8 | 4 | | D:\proj\STK\... |
|  sample.c... | 4 | 4 | | D:\proj\STK\... |
|  sub14 | 4 | 4 | | D:\proj\STK\... |

备注 如果用户在堆栈使用跟踪程序运行时改变影响总堆栈大小（例如，用户在其工程中编辑文件以使总堆栈大小改变），在重新创建工程之后，可单击  按钮更新显示。

2.16.4 检查未知函数

在 [Unknown Functions] 下，用户可以在堆栈大小未知 / 校准函数列表对话框中检查堆栈使用跟踪程序无法获取堆栈信息的函数。

图 2-113. 堆栈大小未知 / 校准函数列表对话框



备注 在下述情况时函数将出现在 [Unknown Functions] 下。

- 帧大小无法测量。
- 在调整栈堆大小对话框中递归层数的未设置。
- 函数包含间接调用函数，但未在调整栈堆大小对话框中设置为 callee 函数。

2.16.5 改变帧大小

用户可以通过使用 **调整栈堆大小对话框** 或者堆栈大小设定文件，动态改变堆栈跟踪程序无法获取堆栈信息的函数或用户需要改变的函数的帧大小。

(1) 使用 **调整栈堆大小对话框**

使用 **调整栈堆大小对话框** 的步骤如下。


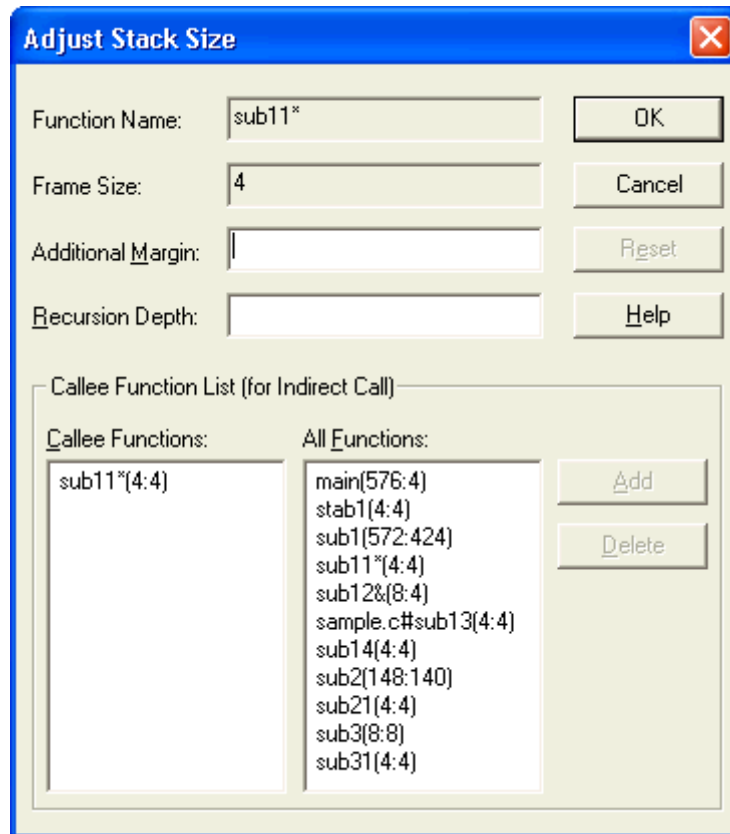
- 选择 **堆栈用法跟踪窗口** 中树型显示区的条目，然后单击工具栏 >> 。 **调整栈堆大小对话框** 打开。

图 2-114. 调整栈堆大小对话框



- 在设置 [Additional Margin]， [Recursion Depth] 和 [Callee Functions] 后，单击 [OK]。

(2) 使用堆栈大小设定文件

以下是使用堆栈大小设定文件的步骤。

- 创建堆栈大小设定文件

采用下述格式，用户需要动态设定的堆栈大小设定文件中写入函数。

文件名 [, ADD=additional margin] [, RECTIME=recursion depth] [, CALL=callee function] ...

图 2-115. 堆栈大小设定文件举例

```
# Set the frame size of function "_flib" written in assembly
# language to 50
[flib], ADD=50

# Set the frame size of function "sub2" written in C to 100
sub2, ADD=100

#Set the recursion depth of recursive function "sub3" written
# in C to 123
sub3, RECTIME=123
```

- 从堆栈用法跟踪窗口，选择 [File] 菜单 >> [Load Stack Size Specification File...]。打开对话框打开。设定堆栈大小设定文件，然后单击 [Open]。

第 3 章 构建输出列表

本章介绍通过各种命令构建后输出的列表格式和其他特征。

3.1 C 编译器

C 编译器输出下列文件。

- 汇编程序源文件
- 出错列表文件
- 预处理列表文件
- 交叉引用列表文件

备注 有关 C 编译器输入和输出的详细说明，参见“[B.1.1 I/O 文件](#)”。

3.1.1 汇编程序源文件

汇编程序源文件是 C 源代码编译结果的 ASCII 映射列表，也是对应 C 源程序的汇编语言源文件。

如果在创建汇编源文件时设定选项 (-sa)，则文件中也会包含 C 语言源代码作为注释。

要在 CubeSuite 中配置此汇编源文件，首先在[工程树面板](#)上选择 build tool 节点，然后在[属性面板](#)上的 [\[Compile Options\]](#) 标签里做设置。在 [\[Assembly File\]](#) 分项中设置 [\[Output an assemble file\]](#) 属性为 [\[Yes\]](#)。输出目标是在 [\[Common Options\]](#) 标签上设置的文件夹，具体在 [\[Output File Type And Path\]](#) 分项的 [\[Intermediate file output folder\]](#) 属性中。

```
; 78K0R C Compiler V(1)x.xx Assembler Source      Date:(2)xx xxx xxxx Time:(3)xx:xx:xx

; Command   : (4)-cf1166a0 prime.c 柚 a
; In-file   : (5)prime.c
; Asm-file  : (6)prime.asm
; Para-file : (7)

        $PROCESSOR((8)f1166a0)
(9) $DEBUG
(10)$NODEBUGA
(11)$KANJICODE SJIS
(12)$TOL_INF      03FH , 100H , 00H , 00H , 00H

(13)$DGS      FIL_NAM , .file ,      034H , 0FFFEH , 03FH , 067H , 01H , 00H
        :
(14)      EXTRN  _@RTARG0
        :
; line (15)1 : (16)#define TRUE      1
; line (15)2 : (16)#define FALSE    0
; line (15)3 : (16)#define SIZE     200
        :
(14)_main :
(17)$DGL      1 , 14
(14)      push   hl                      ; (21)[ INF ] 1 , 1
(14)      subw  sp , #08H                 ; (21)[ INF ] 2 , 1
```

```

(14)      movw    ax , sp                      ; (21)[ INF ] 2 , 1
(14)      movw    hl , ax                     ; (21)[ INF ] 1 , 1
          :
(18)??bf_main :
          :
          ; (22)*** Code Information ***
          ;
          ; (23)$FILE C:\Program Files\NEC Electronics CubeSuite\CubeSuite\CA78K0R\Vx.xx\
smp78k0r\CC78K0R\prime.c
          ;
          ; (24)$FUNC main ( 8 )
          ; (25)      bc = ( void )
          ; (26)      CODE SIZE = 116 bytes , CLOCK_SIZE = 86 clocks , STACK_SIZE = 16 bytes
          ;
          ; (27)$CALL printf ( 18 )
          ; (28)      bc = ( pointer:ax , int : [ sp + 2 ] )
          ;
          ; (27)$CALL putchar ( 20 )
          ; (28)      bc = ( int : ax ) ;
          ;
          ; (27)$CALL printf ( 25 )
          ; (28)      bc = ( pointer:ax , int : [ sp + 2 ] )
          ;
          ; (24)$FUNC printf ( 31 )
          ; (25)      bc = ( pointer s :ax , int i : [ sp + 4 ] )
          ; (26)      CODE SIZE = 23 bytes , CLOCK_SIZE = 22 clocks , STACK_SIZE = 14 bytes
          ;
          ; (24)$FUNC putchar ( 41 )
          ; (25)      bc = ( char c : x )
          ; (26)      CODE SIZE = 16 bytes , CLOCK_SIZE = 18 clocks , STACK_SIZE = 6 bytes

          ; Target chip : (19)uPD78F1166_A0
          ; Device file : (20)Vx.xx
    
```

| 条目号 | 说明 | 格式 |
|-----|---------|--|
| (1) | 版本号 | 以 "x.yz" 格式显示 |
| (2) | Date | 系统日期 (以 "DD Mmm YYYY" 格式显示) |
| (3) | 时间 | 系统时间 (以 "HH MM SS" 格式显示) |
| (4) | 命令行 | 由 "CC78K0" 输出命令行内容。 80 列后的内容从下一行的 15 列开始输出。输出分号 (;) 到第 1 列。将一个以上的空字符或制表符替换成单个空字符。 |
| (5) | C 源文件名称 | 输出指定文件名。 如果省略文件类型, 则会附加 ".c" 作为文件类型 (扩展名)。80 列后的内容从下一行的 15 列开始输出。输出分号 (;) 到第 1 列。 |

| 条目号 | 说明 | 格式 |
|------|-------------|---|
| (6) | 汇编程序源文件名 | 输出指定文件名。 如果省略文件类型,则会附加 ".asm" 作为文件类型 (扩展名)。80 列后的内容从下一行的 15 列开始输出。输出分号 (;) 到第 1 列。 |
| (7) | 参数文件内容 | 输出参数文件内容。 80 列后的内容从下一行的 15 列开始输出。输出分号 (;) 到第 1 列。将一个以上的空字符或制表符替换成单个空字符。 |
| (8) | 设备类型 | 通过 -c 选项设定字符串。 |
| (9) | 调试信息 | 输出 DEBUG 控制。 输出 \$DEBUG 或 \$NODEBUG。 |
| (10) | 汇编程序的调试信息控制 | 输出 NODEBUGA 控制。 输出 \$NODEBUGA。 |
| (11) | kanji 类型信息 | 输出 kanji 编码 (双字节编码) 类型。 输出 \$KANJI CODE SJIS, \$KANJI CODE EUC, 或 \$KANJI CODE NONE。 |
| (12) | 工具信息 | 输出工具信息、版本信息、出错信息、特定选项等 (信息以 \$TOL_INF 开始)。 |
| (13) | 符号信息 | 输出符号信息 (信息以 \$DGS 开始)。 此信息只在选定调试信息输出选项时输出。即使设定 -g1 选项,也不会输出。 |
| (14) | 汇编源程序 | 输出包含编译结果的汇编程序源文件。 |
| (15) | 行号 | 输出 C 源模块文件的行号采用右对齐的非零十进制数值表示。 |
| (16) | C 源代码 | 这是输入 C 源程序映像 80 列后的内容从下一行的 16 列开始输出。输出分号 (;) 到第 1 列。 |
| (17) | 行号信息 | 按输入的行数输出行号 (信息以 \$DGL 开始)。 此信息只在选定调试信息输出选项时输出。即使设定 -g1 选项,也不会输出。 |
| (18) | 符号信息创建标签 | 输出函数标签信息 (信息以 ?? 开始)。 此信息只在选定调试信息输出选项时输出。 |
| (19) | 编译器的目标设备 | 通过命令行选项 (-c) 或源文件显示指定的目标设备。 |
| (20) | 设备文件版本 | 显示输入设备文件的版本号。 |
| (21) | 大小, 时钟 | 为输出指令输出大小和时钟。(信息以 ;[INF] 开始)。 如果输出指令不能指定时钟号,时钟将以下列格式:"clock 1/clock 2" 输出。 在不访问数据时输出在访问内部 RAM 区域或 SFR 区域时的时钟数。 为条件分支指令,在建立条件时输出时钟数。 风险不能确定。 注意,因此,那个输出的时钟数与实际时钟数不同。 它只是参考值。 |

| 条目号 | 说明 | 格式 |
|------|---------------------|--|
| (22) | 函数信息 (开始) | 表明函数信息开始。 |
| (23) | 函数信息 (文件名) | 含完整路径的输出目标源文件名。(信息以 ;\$FILE 开始)。 |
| (24) | 函数信息 (定义函数) | 输出函数名和用十进制码定义的行号。(信息以 ;\$FUNC 开始)。 |
| (25) | 函数信息 (返回值、定义函数的参数) | 输出定义函数的返回值寄存器和参数信息 (寄存器或堆栈位置)。 |
| (26) | 函数信息 (定义函数大小、时钟、堆栈) | 输出定义函数所需的大小、时钟和最大堆栈使用量。 在这里显示由函数自身使用的堆栈大小。 如果函数调用另一函数, 被调用函数使用的堆栈大小不会添加到调用函数的堆栈大小中。 CLOCK_SIZE 是在条目 (21) 中已添加时钟数量的结果。 |
| (27) | 函数信息 (调用函数) | 输出函数名和以十进制码定义的函数调用行号。(信息以 ;\$CALL 开始)。 |
| (28) | 函数信息 (调用函数返回值, 自变量) | 在函数调用过程中, 输出返回值寄存器和自变量信息 (寄存器或堆栈位置)。 |

3.1.2 出错列表文件

出错列表文件中包含在编译过程中发生的任何出错和警告消息。

通过特定的编译选项, 可以将 C 源代码添加到出错列表中。包含 C 源代码的出错列表文件可以通过修改 C 源代码和删除注释 (如列表头), 当作 C 源文件来使用。

在 CubeSuite 中配置错误列表文件, 在 [工程树面板](#) 上, 选择 build tool 节点, 然后在 [属性面板](#) 上, 从 [\[Compile Options\]](#) 标签里做设置。在 [\[List File\]](#) 分项中, 设置 [\[Output error list file\]](#) 属性为 [\[Yes\]](#)。输出目标是在 [\[Common Options\]](#) 标签上设置的文件夹, 具体在 [\[Output File Type And Path\]](#) 分项的 [\[Intermediate file output folder\]](#) 属性中。

- 含 C 源代码的出错列表文件

```

/*
78K0R C Compiler V(1)x.xx Error List   Date:(2)xx xxx xxxx Time:(3)xx:xx:xx

Command   : (4)-cf1166a0 prime.c -se
C-file    : (5)prime.c
Err-file  : (6)prime.cer
Para-file: (7)
*/

(8) #define TRUE    1
(8) #define FALSE   0
(8) #define SIZE    200

(8) char    mark [ SIZE + 1 ] ;

(8) void main ( ) {
(8)     int i , prime , k , count ;
(8)     cont = 0 ;

```

```

*** CC78K0R error (9)E0711: (10)Undeclared 'cont' ; function 'main'
(8)   for ( i = 0 ; i <= SIZE ; i++ )
(8)       mark [ i ] = TRUE ;
(8)   for ( i = 0 ; i <= SIZE ; i++ ) {
(8)       if ( mark [ i ] ) {
(8)           prime = i + i + 3 ;
(8)           printf ( "%6d" , prime ) ;
*** CC78K0R warning (9)W0745: (10)Expected function prototype
      :
/*
(11)Target chip : uPD78F1166_A0
(12)Device file : Vx.xx
Compilation complete, (13)1 error(s) and (14)5 warning(s) found.
*/

```

| 条目号 | 说明 | 格式 |
|------|---------------|--|
| (1) | 版本号 | 以 "x.yz" 格式显示 |
| (2) | Date | 系统日期 (以 "DD Mmm YYYY" 格式显示) |
| (3) | 时间 | 系统时间 (以 "HH MM SS" 格式显示) |
| (4) | 命令行 | 由 "CC78K0" 输出命令行内容。 80 列后的内容从下一行的 15 列开始输出。一个或以上的空白制表符替换成单个空字符。 |
| (5) | C 源文件名称 | 输出指定文件名。 如果省略文件类型, 则会附加 ".c" 作为文件类型 (扩展名)。80 列后的内容从下一行的 15 列开始输出。 |
| (6) | 出错列表文件名 | 输出指定文件名。 如果省略文件类型, 则会附加 ".cer" 作为文件类型。 80 列后的内容从下一行的 15 列开始输出。 |
| (7) | 参数文件内容 | 输出参数文件内容。 80 列后的内容从下一行的 15 列开始输出。一个或以上的空白制表符替换成单个空字符。 |
| (8) | C 源代码 | 这是输入 C 源程序映像 80 列以后的内容不接在下一行。 |
| (9) | 出错消息编号 | 以 "#nnnn" 格式输出出错编号。 当 "#" 是中止性错误则输出 "F", 是致命性错误则输出 "E", 是内部错误则输出 "C", 如果是警告则输出 "W"。 "nnnn" (出错编号) 以 4 位十进制数字表示 (不允许为 0)。 |
| (10) | Error message | 输出出错消息。 80 列以后的内容不接在下一行。 |
| (11) | 编译器的目标设备 | 通过命令行选项 (-c) 或源文件显示指定的目标设备。 |
| (12) | 设备文件版本 | 显示输入设备文件的版本号。 |

| 条目号 | 说明 | 格式 |
|------|-----|---------------|
| (13) | 出错数 | 输出右对齐的非零十进制数。 |
| (14) | 警告数 | 输出右对齐的非零十进制数。 |

- 只含出错信息的出错列表文件

```
(1) prime.c ((2)18) : CC78K0R warning (3)W0745: (4)Expected function prototype
(1) prime.c ((2)20) : CC78K0R warning (3)W0745: (4)Expected function prototype
(1) prime.c ((2)26) : CC78K0R warning (3)W0622: (4)No return value
(1) prime.c ((2)37) : CC78K0R warning (3)W0622: (4)No return value
(1) prime.c ((2)44) : CC78K0R warning (3)W0622: (4)No return value

Target chip : (7)uPD78F1166_A0
Device file : (8)Vx.xx

Compilation complete, (5)0 error(s) and (6)5 warning(s) found.
```

| 条目号 | 说明 | 格式 |
|-----|---------------|--|
| (1) | C 源文件名 | 输出指定文件名。 如果省略文件类型，则会附加 ".c" 作为文件类型 (扩展名)。 |
| (2) | 行号 | 输出右对齐的非零十进制数。 |
| (3) | 出错消息编号 | 以 "#nnnn" 格式输出出错编号。 当 "#" 是中止性错误则输出 "F", 是致命性错误则输出 "E", 是内部错误则输出 "C", 如果是警告则输出 "W"。 "nnnn" (出错编号) 以 4 位十进制数字表示 (不允许为 0)。 |
| (4) | Error message | 输出出错消息。 |
| (5) | 出错数 | 输出右对齐的非零十进制数。 |
| (6) | 警告数 | 输出右对齐的非零十进制数。 |
| (7) | 编译器的目标设备 | 通过命令行选项 -c 或源文件显示指定的目标设备。 |
| (8) | 设备文件版本 | 显示输入设备文件的版本号。 |

3.1.3 预处理列表文件

预处理列表文件是只包含 C 源代码预处理结果的 ASCII 映像文件。

设定 -k 选项后，预处理列表文件可以当 C 源文件用。预处理类型已设定为 "n" 时例外。设定 -kd 选项后，则输出 #define 扩展的列表。

要在 CubeSuite 中配置预处理列表文件，在工程树面板上选择 Build tool 节点，然后在属性面板上的 [Compile Options] 标签里做设置。在 [List File] 分项中的 [Output preprocess list file] 上选择 [Yes(-p)]。输出目标是在 [Common Options] 标签上设置的文件夹，具体在 [Output File Type And Path] 分项的 [Intermediate file output folder] 属性中。

如果页宽为 80，则结果如下。

```

/*
78K0R C Compiler V(1)x.xx Preprocess List   Date:(2)xx xxx xxxx   Page: (3)xxxx

Command   : (4)-cfl166a0 prime.c -p -lw80
In-file   : (5)prime.c
PPL-file  : (6)prime.ppl
Para-file : (7)
*/

(8) 1 : (9)#define TRUE    1
(8) 2 : (9)#define FALSE   0
(8) 3 : (9)#define SIZE    200
(8) 4 : (9)
(8) 5 : (9)char    mark [ SIZE + 1 ] ;
(8) 6 : (9)

/*
(10)Target chip : uPD78F1166_A0
(11)Device file : Vx.xx
*/

```

| 条目号 | 说明 | 格式 |
|------|----------|--|
| (1) | 版本号 | 以 "x.yz" 格式显示 |
| (2) | Date | 系统日期 (以 "DD Mmm YYYY" 格式显示) |
| (3) | 页数 | 输出右对齐的非零十进制数。 |
| (4) | 命令行 | 由 "CC78K0" 输出命令行内容。 超出该行长度的内容输出到下一行 13 列的起始位置。一个或以上的空白制表符替换成单个空字符。 |
| (5) | C 源文件名称 | 输出指定文件名。 如果省略文件类型, 则会附加 ".c" 作为文件类型 (扩展名)。超出该行长度的内容输出到下一行 13 列的起始位置。 |
| (6) | 预处理列表文件名 | 输出指定文件名。 如果省略文件类型, 则附加 ".ppl" 作为文件类型 (扩展名)。超出该行长度的内容输出到下一行 13 列的起始位置。 |
| (7) | 参数文件内容 | 输出参数文件内容。 超出该行长度的内容输出到下一行 13 列的起始位置。输出分号 ";" 到第 1 列。一个或以上的空白制表符被替换为单个空字符。 |
| (8) | 行号 | 输出右对齐的非零十进制数。 |
| (9) | C 源代码 | 这输入的是 C 源代码。 超出该行长度的内容输出到下一行 9 列的起始位置。 |
| (10) | 编译器的目标设备 | 通过命令行选项 (-c) 或源文件显示指定的目标设备。 |
| (11) | 设备文件版本 | 显示输入设备文件的版本号。 |

3.1.4 交叉引用列表文件

交叉引用列表文件中包含标识符列表，例如声明、定义、引用函数和变量。其中也包含其它信息，例如属性和行数。它们以编译器编译的顺序输出。

在 CubeSuite 中配置交叉引用列表文件，在工程树面板上选择 build tool 节点，然后在属性面板上的 [Compile Options] 标签里做设置。在 [List File] 分项中的 [Output cross reference list file] 属性上选择 [Yes(-x)]。输出目标是在 [Common Options] 标签上设置的文件夹，具体在 [Output File Type And Path] 分项的 [Intermediate file output folder] 属性中。

如果页宽为 80，则结果如下。

```

78K0R C Compiler V(1)x.xx Cross reference List      Date:(2)xx xxx xxxx  Page: (3)xxxx

Command   : (4)-cf1166a0 prime.c -x -lw80
In-file   : (5)prime.c
Xref-file : (6)prime.xrf
Para-file : (7)
Inc-file  : (8)[ n ]

(9)ATTRIB (10)MODIFY (11)TYPE (12)SYMBOL (13)DEFINE (14)REFERENCE

  EXTERN   NEAR      array   mark     5          14         16         22
  EXTERN   FAR       func    main     7
  AUTO1    int       i       9          13         13         13         14
                                     15         15         15         16
                                     17         17         21
  AUTO1    int       prime   9          17         18         21         21
  AUTO1    int       k       9          21         21         21         22
  AUTO1    int       count   9          11         19         20         25

:
/*
(15)Target chip : uPD78F1166_A0
(16)Device file : Vx.xx
*/

```

| 条目号 | 说明 | 格式 |
|-----|-----------|--|
| (1) | 版本号 | 以 "x.yz" 格式显示 |
| (2) | Date | 系统日期 (以 "DD Mmm YYYY" 格式显示) |
| (3) | 页数 | 输出右对齐的非零十进制数。 |
| (4) | 命令行 | 由 "CC78K0" 输出命令行内容。 超出该行长度的内容输出到下一行 13 列的起始位置。一个或以上的空白制表符替换成单个空字符。 |
| (5) | C 源文件名称 | 输出指定文件名。 如果省略文件类型，则会附加 ".c" 作为文件类型 (扩展名)。超出该行长度的内容输出到下一行 13 列的起始位置。 |
| (6) | 交叉引用列表文件名 | 输出指定文件名。 如果省略文件类型，则附加 ".xrf" 作为文件类型。超出该行长度的内容输出到下一行 13 列的起始位置。 |

| 条目号 | 说明 | 格式 |
|------|------------|---|
| (7) | 参数文件内容 | 输出参数文件内容。 超出该行长度的内容输出到下一行 13 列的起始位置。一个或以上的空白制表符替换成单个空字符。 |
| (8) | Include 文件 | 通过命令行选项 (-c) 或源文件显示指定的目标设备。 "n" 表示起始于 "1" 的数字, 以此来标示包含文件号。超出该行长度的内容输出到下一行 13 列的起始位置。在没有包含文件时, 则该行不被输出。 |
| (9) | 符号属性 | 显示符号的属性。 用 EXTERN 来标识外部变量, EXSTC 来标识外部静态变量, INSTC 来标识内部静态变量, AUTOOnn 来标识自动变量, REGnn 来标识寄存器变量 (用 nn 来表示数值的范围, 其数值从 "1" 开始), 用 EXTYP 来标识外部 typedef 声明, INTYP 来标识内部 typedef 声明, 以 LABEL 来标识标签, 以 TAG 来标识结构体或共用体, 以 MEMBER 标识成员, 并以 PARAM 来标识函数参数。 |
| (10) | 符号可用属性 | 显示符号可用属性 (左对齐)。 用 CONST 标识常量, VLT 标识变量, CALLT 标识调用函数, SREG 标识 sreg 位变量, RWSFR 标识 sfr 变量, ROSFR 标识为只读 sfr 变量, WOSFR 标识只读 sfr 变量, VECT 标识中断函数, 用 NEAR 标识在 near 区域分配的函数和变量, 用 FAR 标识在 far 区域分配的函数和变量。 |
| (11) | 符号类型 | 显示符号类型。 类型包括字符型、整型、短整型、长整型和域型。把 "u" 加到 unsigned 型的开头。 附加类型包括空类型、浮点型、双精度型、长精度 (long double) 型、函数型、数组型、指针型、结构体型、共用体型、枚举型、位, inter, 和宏替换指令 #define。 |
| (12) | 符号名称 | 如果符号名称超过 15 个字符并处于一行中, 则名称原样输出。如果符号名称超过 15 个字符并处于一行中, 超越部分将从下一行第 23 列输出且条目 (13) 和 (14) 从下一行第 39 列输出。 |
| (13) | 符号定义行号 | 输出的行号和符号定义的文件名以行号 (5 位数字): 包含文件号 (2 位) 的格式显示。 |
| (14) | 符号引用行号 | 输出的行号和引用符号的文件名以行号 (5 位数字): 包含文件号 (2 位) 的格式显示。 如果行内容超过行的长度, 余下内容输出到下一行第 48 列的起始位置。 |
| (15) | 编译器的目标设备 | 通过命令行选项 (-c) 或源文件显示指定的目标设备。 |
| (16) | 设备文件版本 | 显示输入设备文件的版本号。 |

3.2 汇编程序

汇编程序输出如下列表：

| 输出列表文件名 | 输出列表名 |
|---------|---------|
| 汇编列表文件 | 汇编列表文件头 |
| | 汇编列表 |
| | 符号列表 |
| | 交叉引用列表 |
| 错误列表文件 | 错误列表 |

在 CubeSuite 中配置汇编列表文件，在工程树面板上，选择 build tool 节点，然后在属性面板上，从 [Assemble Options] 标签里做设置。具体在 [Assemble List] 分项中的 [Output assemble list file] 上选择 [Yes(-p)]。要输出错误列表文件，在 [Output File] 分项中设置 [Output error list file] 属性为 [Yes(-e)]。输出目标是在 [Common Options] 标签上设置的文件夹，具体在 [Output File Type And Path] 分项的 [Intermediate file output folder] 属性中。

备注 参见 “B.2.1 I/O 文件” 有关汇编文件输入和输出的详细说明。

3.2.1 汇编列表文件头

列表文件头总是输出到汇编列表文件的开始部分。

```
78K0R Assembler (1)Vx.xx (2)SAMPLE_TITLE      Date:(3)xx xxx xxxx  Page: (4)xxxx
(5)SAMPLE_SUBTITLE
Command: (6) k0rmain.asm -cf1166a0
Para-file: (7) -ks -kx
In-fine: (8) k0rmain.asm
Obj-file: (9) k0rmain.rel
Prn-file: (10)k0rmain.prn
```

| 条目号 | 说明 | 格式 |
|-----|-----------|---|
| (1) | 汇编版本号 | 以 "x.yz" 格式显示 |
| (2) | 标题字符串 | Õ@¼" -lh—°œÓªÚTITLEøÿ÷ý÷³;Óp%Š`÷³¹@µfP÷²°¥Æ°£ |
| (3) | 汇编列表日期的创建 | 汇编列表日期的创建 (用 "DD Mmm YYYY" 格式显示) |
| (4) | 页号 | 输出右对齐的非零十进制数。 |
| (5) | 副标题字符串 | 通过副标题控制指令的设定输出字符串。 |
| (6) | 命令行 | 输出命令行内容。 超出该行长度的内容输出到下一行 11 列的起始位置。一个或以上的空白制表符替换成单个空字符。 |
| (7) | 参数文件内容 | 输出参数文件内容。 超出该行长度的内容输出到下一行 11 列的起始位置。一个或以上的空白制表符替换成单个空字符。 |

| 条目号 | 说明 | 格式 |
|------|-----------|--|
| (8) | 输入源文件名 | 输出指定文件名。 如果省略文件类型,则会附加 ".asm" 作为文件类型 (扩展名)。超出该行长度的内容输出到下一行 11 列的起始位置。 |
| (9) | 输出目标模块文件名 | 输出指定文件名。 如果省略文件类型,则会附加 ".ref" 作为文件类型。超出该行长度的内容输出到下一行 11 列的起始位置。 |
| (10) | 打印文件名 | 输出指定文件名。 如果省略文件类型,则会附加 ".prn" 作为文件类型。超出该行长度的内容输出到下一行 11 列的起始位置。 |

3.2.2 汇编列表

汇编列表输出含有出错消息的汇编结果 (如果有错误产生)。

在 CubeSuite 中配置此汇编列表, 在 [工程树面板](#) 上, 选择 **build tool** 节点, 然后在 [属性面板](#) 上, 从 [\[Assemble Options\]](#) 标签里做设置。具体在 [\[Assemble List\]](#) 分项中的 [\[Output assemble list file\]](#) 上选择 [\[Yes\(-p\)\]](#)。

```

Assemble list

(1)ALNO (2)STNO (6)ADRS (8)OBJECT (3)M (4)I (5)SOURCE STATEMENT
  1      1
  2      2                      NAME    SAMPM
  :
 31     31    0000A   RFD0000                      CALL  !CONVAH
                                           ; convert ASCII <- HEX
 32     32                                           ; output BC-register <- ASCII code
 33     33    0000D   00000000                     MOV   DE , #LOWW ( STASC )
                                           ; set DE <- store ASCII code table
                                           00011   00
(7)*** ERROR E2202, STNO 33 ( 33) Illegal operand
 34     34    00012   63                          MOV   A , B
 35     35    00013   99                          MOV   [ DE ] , A
  :
Segment informations :

(9)ADRS (10)LEN (11)NAME

  FFE20   00003H   DATA
  00000   00002H   CODE
  00000   00019H   ?CSEG

Target chip : (12)uPD78xxx
Device file : (13)Vx.xx
Assembly complete, (14)1 error(s) and (15)0 warning(s) found. ( (16)33)

```

| 条目号 | 说明 | 格式 |
|------|------------|---|
| (1) | 源程序映像的行号 | 输出右对齐的非零十进制数。 |
| (2) | ミコト | 输出右对齐的非零十进制数。 包含 INCLUDE 文件和宏扩展。 |
| (3) | 显示宏 | 显示宏。 - M: 宏定义行。 - #n: 宏扩展行。"n" 是其嵌套级。 - Blank: 不是宏定义和宏扩展行。 |
| (4) | 显示 INCLUDE | 显示 INCLUDE 。J - In: 在 INCLUDE 文件里面。"n" 是其嵌套级。 - Blank: INCLUDE 文件未使用。 |
| (5) | 源语句 | 显示源语句 超出该行长度的内容输出到下一行的起始位置。 |
| (6) | 位置计数器值 | 行起始地址显示用来标识机器指令 DB, DW, DS, 和 DBIT 的标签。 以非零的十六进制格式来显示。 以非零的十六进制格式来显示。 |
| (7) | 产生错误的行 | 产生错误的行。显示所需项目。 |
| (8) | 重新配置信息 | 显示重新配置的信息。 - R: 链接器改变目标代码或符号值。 - Blank: 链接器不改变目标代码或符号值。 |
| (9) | 段地址 | 显示段的起始地址。 以非零的十六进制格式来显示。 |
| (10) | 段长度 | 显示段长度。 以非零的十六进制格式来显示。 |
| (11) | 段名称 | 显示段名称。 |
| (12) | リキヲシテ ヲシテ | 通过命令行选项 (-c) 或源文件显示指定的目标设备。 |
| (13) | 设备文件版本号 | 显示输入设备文件的版本号。 |
| (14) | 致命性错误的数量 | 输出右对齐的非零十进制数。 |
| (15) | 警告数 | 输出右对齐的非零十进制数。 |
| (16) | 最终错误行 | 输出右对齐的非零十进制数。 |

3.2.3 符号列表

符号列表输出定义在源代码中的符号 (包括局部符号)。

在 CubeSuite 中配置符号列表, 在工程树面板上, 选择 build tool 节点, 然后在属性面板上, 从 [Assemble Options] 标签里做设置。具体在 [Assemble List] 分项中的 [Output with symbol list] 属性上选择 [Yes(-ks)]。

| Symbol Table List | | | | | | | |
|-------------------|---------|---------|---------|----------|---------|---------|---------|
| (1)VALUE | (2)ATTR | (3)RTYP | (4)NAME | (1)VALUE | (2)ATTR | (3)RTYP | (4)NAME |
| | CSEG | | ?CSEG | | CSEG | | CODE |
| -----H | EXT | CONVAH | | | DSEG | | DATA |
| FFE20H | ADDR | HDTSA | | 0H | ADDR | PUB | MAIN |
| | MOD | | SAMPM | 0H | ADDR | PUB | START |
| FFE21H | ADDR | STASC | | -----H | EXT | | _@STBEG |

| 条目号 | 说明 | 格式 |
|-----|--------|--|
| (1) | 符号值 | 显示符号的值。 输出右对齐的非零十六进制数。 |
| (2) | 符号属性 | 显示符号的属性。(左对齐) <ul style="list-style-type: none"> - CSEG: 代码段名称 - DSEG: 数据段名称 - BSEG: 位段名称 - MAC: 宏名称 - MOD: 模块名称 - SET: 通过 SET 指令定义符号 - NUM:NUMBER 属性符号 - ADDR:ADDRESS 属性符号 - BIT:BIT 属性符号 (addr. 位) - SABIT:BIT 属性符号 (saddr. 位) - SFBIT:BIT 属性符号 (sfr. 位) - RBIT:BIT 属性符号 (A. 位 ,X. 位 ,PSW. 位) - SFR: 通过 EQU 指令定义 SFRs 名称 - SFRP: 通过 EQU 指令定义 SFRPs 名称 - Blank: 通过 EXTRN 或 EXTBIT 来定义外部引用符号 - *****: 未定义符号 |
| (3) | 符号引用格式 | 显示符号引用格式。(左对齐) <ul style="list-style-type: none"> - EXT: 用 EXTRN(SADDR 属性) 来定义外部引用符号 - EXTB: 用 EXTBIT(saddr. 位) 来定义外部引用符号 - PUB: 用 PUBLIC 来定义外部引用符号 - Blank: 本地符号, 段名称, 宏名称, 模块名称 - *****: 未定义符号 |
| (4) | 定义符号名称 | 显示定义的符号名称。(左对齐) |

3.2.4 交叉引用列表

交叉引用列表输出的数据, 标识符号在源代码里定义的位置 (在任意行)。

在 CubeSuite 中配置交叉引用列表文件, 在工程树面板上, 选择 build tool 节点, 然后在属性面板上, 从 [Assemble Options] 标签里做设置。具体在 [Assemble List] 分项中的 [Output with cross reference list] 属性上选择 [Yes(-kx)]。

| Cross-Reference List | | | | | | | | | |
|----------------------|----------|------|---------|---------|------------|----------|--------|--------|---------|
| (1)NAME | (2)VALUE | (3)R | (4)ATTR | (5)RTYP | (6)SEGNAME | (7)XREFS | | | |
| ?CSEG | | | CSEG | | ?CSEG | | 9.98kg | | |
| CODE | | | CSEG | | CODE | | 8.62kg | | |
| CONVAH | -----H | E | EXT | | 12@ 31 | | | | |
| DATA | | | DSEG | | DATA | | 6.80kg | | |
| HDTSA | FFE20H | ADDR | | DATA | 16# 28 | 29 | | | |
| MAIN | 0H | | ADDR | PUB | CODE | | 11@ | 9.07kg | |
| SAMPM | | | MOD | | | | 2# | | |
| START | 0H | R | ADDR | PUB | ?CSEG | | 11@ | 20 | 10.43kg |
| STASC | FFE21H | ADDR | | DATA | 17# 33 | | | | |
| __@STBEG | -----H | E | EXT | | 13@ 26 | | | | |

| 条目号 | 说明 | 格式 |
|-----|--------|--|
| (1) | 定义符号名称 | 显示定义的符号名称。(左对齐) 如果符号名称超过 16 个字符, 则名称照现样输出。项目 (2), (4), (5), (6), (7) 和 (8) 从下一行输出。 |
| (2) | 符号值 | 显示符号的值。 输出右对齐的非零十六进制数。 |
| (3) | 重新配置属性 | 显示重新配置的属性。 - R: 可浮动符号 - E: 外部符号 - Blank: 绝对符号 - *: 未定义符号 |
| (4) | 符号属性 | 显示符号的属性。(左对齐) - CSEG: 代码段名称 - DSEG: 数据段名称 - BSEG: 位段名称 - MAC: 宏名称 - MOD: 模块名称 - SET: 通过 SET 指令定义符号 - NUM:NUMBER 属性符号 - ADDR:ADDRESS 属性符号 - BIT:BIT 属性符号 (addr. 位) - SABIT:BIT 属性符号 (saddr. 位) - SFBIT:BIT 属性符号 (sfr. 位) - RBIT:BIT 属性符号 (A. 位 ,X. 位) - SFR: 通过 EQU 指令定义 SFRs 名称 - SFRP: 通过 EQU 指令定义 SFRPs 名称 - Blank: 通过 EXTRN 或 EXTBIT 来定义外部引用符号 - *****: 未定义符号 |
| (5) | 符号引用格式 | 显示符号引用格式。(左对齐) - EXT: 用 EXTRN(SADDR 属性) 来定义外部引用符号 - EXTB: 用 EXTBIT(saddr. 位) 来定义外部引用符号 - PUB: 用 PUBLIC 来定义外部引用符号 - Blank: 本地符号, 段名称, 宏名称, 模块名称 - *****: 未定义符号 |

| 条目号 | 说明 | 格式 |
|-----|---------|--|
| (6) | 定义段名称 | 显示用符号来定义的段名称。(左对齐) |
| (7) | 定义/引用行号 | 显示定义/引用行号。 - 定义行 :xxxxx# - 引用行 :xxxxx Δ (Δ = 1 空白) - EXTRN 声明、EXTBIT 声明、PUBLIC 声明 :xxxxx@ |

3.2.5 错误列表

在汇编器启动时，输出有出错消息的错误列表。

```

PASS1 Start
(1)ERROR.ASM((2)26) : RA78K0R (3)error (4)E2202: (5)Illegal operand
(1)ERROR.ASM((2)32) : RA78K0R (3)error (4)E2202: (5)Illegal operand
PASS2 Start
(1)ERROR.ASM((2)26) : RA78K0R (3)error (4)E2202: (5)Illegal operand
(1)ERROR.ASM((2)29) : RA78K0R (3)error (4)E2407: (5)Undefined symbol reference 'DTSA'
(1)ERROR.ASM((2)29) : RA78K0R (3)error (4)E2303: (5)Illegal expression
(1)ERROR.ASM((2)32) : RA78K0R (3)error (4)E2202: (5)Illegal operand
(1)ERROR.ASM((2)37) : RA78K0R (3)error (4)E2407: (5)Undefined symbol reference 'F'
(1)ERROR.ASM((2)37) : RA78K0R (3)error (4)E2303: (5)Illegal expression

```

| 条目号 | 说明 | 格式 |
|-----|---------------|---|
| (1) | 产生错误的源文件名 | 输出已产生错误的源文件名 |
| (2) | 产生错误的行 | 输出左对齐的非零数值。 |
| (3) | 出错类型 | 输出出错类型。 |
| (4) | 出错号 | 输出错误编号以"#mnnnn"格式。 输出"2"表示"??"汇编器、"3"表示链接器、"4"表示目标转换器、"5"表示库管理程序及"6"表示列表转换器。 înnn"是出错号。 |
| (5) | Error message | 输出出错消息。 |

备注 产生错误的文件和行可能不会显示。

3.3 链接器

链接器输出如下列表。

| 输出列表文件名 | 输出列表名 |
|---------|---------|
| 链接列表文件 | 链接列表文件头 |
| | Map 列表 |
| | 公用符号列表 |
| | 局部符号列表 |
| 错误列表文件 | 错误列表 |

在 CubeSuite 中配置链接列表文件，在工程树面板上选择 build tool 节点，然后在属性面板上的 [Link Options] 标签里设置。在 [Link List] 分项中的 [Output link list file] 属性上选择 [Yes]。输出错误列表文件，在 [Error List] 分项中设置 [Output error list file] 属性为 [Yes(-e)]。输出目标是在 [Common Options] 标签上设置的文件夹，具体在 [Output File Type And Path] 分项的 [Intermediate file output folder] 属性中。同时也显示 Build tool generated files 节点下的工程树面板。

备注 参见 “B.3.1 I/O 文件” 有关链接文件输入和输出的详细说明。

3.3.1 链接列表文件头

列表文件头总是输出到链接列表文件的起始部分。

```

78K0R Linker (1)Vx.xx                               Date:(2)xx xxx xxxx Page: (3)xxxx

Command:      (4)k0rmain.rel k0rsub.rel -s -ok0r.map -dk0r.dr
Para-file:    (5)
Out-file:     (6)k0rmain.lmf
Map-File:     (7)k0r.map
Direc-File:   (8)k0r.dr
Directive:    (9)MEMORY ROM : ( 0H , 0ED800H )
              (9)MEMORY RAM1 : ( 0FCF00H , 1100H )
              (9)MEMORY RAM : ( 0FE000H , 1F00H )

*** Link information ***

(10)      6 output segment(s)
(11)     9DH byte(s) real data
(12)     40 symbol(s) defined

```

| 条目号 | 说明 | 格式 |
|------|---------------|------------------------------------|
| (1) | 链接器版本号 | 以 "x.yz" 格式显示 |
| (2) | 链接列表文件日期的创建 | 链接列表文件日期的创建 (以 "DD Mmm YYYY" 格式显示) |
| (3) | 页号 | 输出右对齐的非零十进制数。 |
| (4) | 命令行映像 | 显示启动行指定选项。 |
| (5) | 参数文件内容 | 输出参数文件内容。 |
| (6) | 输出载入模块文件名 | 输出通过连接器生成的载入模块文件名。 |
| (7) | 链接列表文件名 | 输出通过连接器生成的链接列表文件名。 |
| (8) | 链接指令文件名 | 输出通过连接器输入的链接指令文件名。 |
| (9) | 链接指令文件内容 | 显示链接指令文件的内容。 |
| (10) | 输出段数到载入模块文件 | 显示输出到载入模块文件的段数。 输出右对齐的非零十进制数。 |
| (11) | 输出数据长度到载入模块文件 | 显示输出到载入模块文件的数据长度。 输出右对齐的非零十进制数。 |
| (12) | 输出符号数到载入模块文件 | 显示输出到载入模块文件的符号数。 输出右对齐的非零十进制数。 |

3.3.2 Map 列表

Map 列表输出有关段地址的数据。

在 CubeSuite 中配置 Map 列表，在工程树面板上，选择 build tool 节点，然后在属性面板上的 [Link Options] 标签里做设置。具体在 [Link List] 分项中的 [Output with map list] 属性上选择 [Yes]。

```

*** Memory map ***

(1)SPACE=REGULAR

MEMORY=(2)ROM
BASE ADDRESS=(3)00000H   SIZE=(4)ED800H
      (6)OUTPUT  (7)INPUT  (8)INPUT  (9)BASE  (10)SIZE
      SEGMENT    SEGMENT    MODULE    ADDRESS
      CODE
      CODE      SAMPM    00000H    00002H    (11)CSEG  AT
(5) * gap *
      ?CSEGOB0    00002H    000BEH
      ?CSEG        000C0H    00004H    (11)CSEG  OPT_BYTE
      ?CSEG        000C4H    00059H    (11)CSEG
      ?CSEG    SAMPM    000C4H    00017H
      ?CSEG    SAMPS    000DBH    00042H
(5) * gap *
      0011DH    ED6E3H

MEMORY=RAM1
BASE ADDRESS=(3)FCF00H   SIZE=(4)01100H
      (6)OUTPUT  (7)INPUT  (8)INPUT  (9)BASE  (10)SIZE
      SEGMENT    SEGMENT    MODULE    ADDRESS
(5) * gap *
      FCF00H    01100H

MEMORY=RAM
BASE ADDRESS=(3)FE000H   SIZE=(4)01F00H
      (6)OUTPUT  (7)INPUT  (8)INPUT  (9)BASE  (10)SIZE
      SEGMENT    SEGMENT    MODULE    ADDRESS
(5) * gap *
      DATA
      DATA      SAMPM    FFE20H    00003H    (11)DSEG  AT
(5) * gap *
      FFE23H    000DDH

Target chip : (12)uPD78xxx
Device File  : (13)Vx.xx

```

| 条目号 | 说明 | 格式 |
|-----|----------|-----------------------------------|
| (1) | 存储空间名称 | 显示存储空间名称。 |
| (2) | 存储区域名称 | 显示存储区域名称 |
| (3) | 存储区域起始地址 | 显示存储区域的起始地址。 以左对齐的非零十六进制格式来显示。 |

| 条目号 | 说明 | 格式 |
|------|----------------|--|
| (4) | 存储区域大小 | 显示存储区域的大小。 以左对齐的非零十六进制格式来显示。 |
| (5) | 输出组 | 当区域中不存在任何东西，则显示 "gap"。 |
| (6) | 输出段名到载入模块文件中 | 显示输出到载入模块文件的段名。 |
| (7) | 从目标模块文件中读取的段名。 | 显示从目标模块文件中读取的段名。 |
| (8) | 输入模块名称 | 显示输入文件的模块名称，即已存在的输入段显示在 (7) 中。 如果模块名称超过 8 个字符，则名字照样子输出。项目 (9)，(10) 和 (11) 从第 39 列下一行输出。 |
| (9) | 段起始地址 | 显示输出已分配的段起始地址。 |
| (10) | 输出段长度 | 显示输出段的长度。 |
| (11) | 段类型和重新配置属性 | 显示段类型以及重新配置的属性。 |
| (12) | 该汇编器目标设备 | 通过命令行选项 (-c) 或源文件显示指定的目标设备。 |
| (13) | 设备文件版本 | 显示输入设备文件的版本号。 |

3.3.3 公用符号列表

公用符号列表输出的是定义在输入模块中的公用符号。

在 CubeSuite 中配置公用符号列表，在工程树面板上，选择 build tool 节点，然后在属性面板上的 [Link Options] 标签里做设置。具体在 [Link List] 分项中的 [Output with public symbol list] 属性上选择 [Yes(-kp)]。

```

*** Public symbol list ***

(1)MODULE (2)ATTR (3)VALUE (4)NAME

SAMPM
      ADDR 00000H MAIN
      ADDR 000D2H START

SAMPs
      ADDR 000E9H CONVAH

      NUM FFE20H _@STBEG

      NUM FE000H _@STEND

```

| 条目号 | 说明 | 格式 |
|-----|------------------|-------------------------|
| (1) | 用已定义的公用符号命名的模块名称 | 显示用已定义的公用符号命名的输入目标模块名称。 |

| 条目号 | 说明 | 格式 |
|-----|--------|---|
| (2) | 符号属性 | 显示符号的属性。 - CSEG: 代码段名称 - DSEG: 数据段名称 - BSEG: 位段名称 - MAC: 宏名称 - MOD: 模块名称 - SET: 通过 SET 指令定义符号 - NUM:NUMBER 属性符号 - ADDR:ADDRESS 属性符号 - BIT:BIT 属性符号 (addr. 位) - SABIT:BIT 属性符号 (saddr. 位) - SFBIT:BIT 属性符号 (sfr. 位) - RBIT:BIT 属性符号 (A. 位 ,X. 位 PSW. 位) - SFR: 通过 EQU 指令定义 SFRs 名称 - SFRP: 通过 EQU 指令定义 SFRPs 名称 - Blank: 通过 EXTRN 或 EXTBIT 来定义外部引用符号 - *****: 未定义符号 |
| (3) | 符号值 | 显示公用符号值。 |
| (4) | 公用符号名称 | 显示公用符号名称。 |

3.3.4 局部符号列表

局部符号列表输出的是定义在输入模块中的局部符号。

在 CubeSuite 中配置局部符号列表，在工程树面板上，选择 build tool 节点，然后在属性面板上的 [Link Options] 标签里做设置。具体在 [Link List] 分项中的 [Output with local symbol list] 属性上选择 [Yes(-kl)]。

```

*** Local symbol list ***

(1)MODULE (2)ATTR (3)VALUE (4)NAME

SAMPM

      MOD          SAMPM
      DSEG         DATA
      ADDR FFE20H  HDTSA
      ADDR FFE21H  STASC
      CSEG         CODE
      CSEG         ?CSEG

SAMPS

      MOD          SAMPS
      CSEG         ?CSEG
      ADDR 00114H  SASC
      ADDR 0011AH  SASC1
  
```

| 条目号 | 说明 | 格式 |
|-----|------------------|-------------------------|
| (1) | 用已定义的局部符号命名的模块名称 | 显示用已定义的局部符号命名的输入目标模块名称。 |

| 条目号 | 说明 | 格式 |
|-----|--------|--|
| (2) | 符号属性 | 显示符号的属性。 - CSEG: 代码段名称 - DSEG: 数据段名称 - BSEG: 位段名称 - MAC: 宏名称 - MOD: 模块名称 - SET: 通过 SET 指令定义符号 - NUM:NUMBER 属性符号 - ADDR:ADDRESS 属性符号 - BIT:BIT 属性符号 (addr. 位) - SABIT:BIT 属性符号 (saddr. 位) - SFBIT:BIT 属性符号 (sfr. 位) - RBIT:BIT 属性符号 (A. 位 ,X. 位 PSW. 位) - SFR: 通过 EQU 指令定义 SFRs 名称 - SFRP: 通过 EQU 指令定义 SFRPs 名称 - Blank: 通过 EXTRN 或 EXTBIT 来定义外部引用符号 - *****: 未定义符号 |
| (3) | 符号值 | 显示局部符号值 |
| (4) | 局部符号名称 | 显示局部符号名称 |

3.3.5 错误列表

在链接器启动时，输出存有出错消息的错误列表。

```
LK78K0R (1)error (2)E3405: (3)Undefined symbol 'CONVAH' in file 'k0rmain.rel'
```

| 条目号 | 说明 | 格式 |
|-----|---------------|--|
| (1) | 出错类型 | 输出出错类型。 |
| (2) | 出错号 | 以 "#nnnn" 格式输出出错编号。 当 "#" 是中止性错误则输出 "F", 是致命性错误则输出 "E", 是内部错误则输出 "C", 如果是警告则输出 "W"。 "nnnn" (出错编号) 以 4 位十进制数字表示 (不允许为 0)。 |
| (3) | Error message | 输出出错消息。 |

3.4 目标转换器

目标转换器输出如下列表；

| 输出列表文件名 | 输出列表名 |
|---------|-------|
| 错误列表文件 | 错误列表 |

在 CubeSuite 中配置错误列表文件，在工程树面板上，选择 build tool 节点，然后在属性面板上，从 [Object Convert Options] 标签里做设置。在 [Error List] 分项中的 [Output error list file] 属性上选择 [Yes(-e)]。输出目标是在 [Common Options] 标签上设置的文件夹，具体在 [Output File Type And Path] 分项的 [Intermediate file output folder] 属性中。同时也显示 Build tool generated files 节点下的工程树面板。

备注 参见“B.4.1 I/O 文件”有关目标转换器的输入和输出文件的详细说明。

3.4.1 错误列表

在目标转换器启动时输出出错信息并存储在出错列表中。

输出格式与链接器输出的错误列表格式相同。

3.5 库管理程序

库管理程序输出如下列表；

| 输出列表文件名 | 输出列表名 |
|---------|---------|
| 列表文件 | 库信息输出列表 |

在 CubeSuite 中配置列表文件输出，在工程树面板上，选择 build tool 节点，然后在属性面板上的 [Create Library Options] 标签里做设置。在 [Error List] 分项中的 [Output list file] 属性上选择 [Yes]。输出目标是在 [Common Options] 标签上设置的文件夹，具体在 [Output File Type And Path] 分项的 [Intermediate file output folder] 属性中。同时也显示 Build tool generated files 节点下的工程树面板。

备注 参见“B.5.1 I/O 文件”有关库管理程序文件输入和输出的详细说明。

3.5.1 库信息输出列表

库信息输出列表输出的是库文件中与模块相关的数据。

```

78K0R librarian (1)Vx.xx                               Date:(2)xx xxx xxxx Page(3)xxxx

LIB-FILE NAME : (4)k0r.lib                            ((5)xx xxx xxxx)

(6)0001 (7)k0rmain.rel                                ((8)xx xxx xxxx)

(9)MAIN        (9)START

NUMBER OF PUBLIC SYMBOLS : (10)2

(6)0002 (7)k0rsub.rel                                ((8)xx xxx xxxx)

(9)CONVAH

NUMBER OF PUBLIC SYMBOLS : (10)1

```

| 条目号 | 说明 | 格式 |
|-----|----------|--|
| (1) | 库管理程序版本号 | 以 "x.yz" 格式显示 |
| (2) | 列表日期的创建 | 列表日期的创建 (以 "DD Mmm YYYY" 格式显示) |
| (3) | 页数 | 输出右对齐的非零十进制数。 |
| (4) | 库文件名称 | 输出指定文件名。 如果省略文件类型，则会附加 ".lib" 作为文件类型 (扩展名)。 |
| (5) | 库文件日期的创建 | 库文件日期的创建 (以 "DD Mmm YYYY" 格式显示) |
| (6) | 模块序列号 | 数字指定从 0001 开始。 |

| 条目号 | 说明 | 格式 |
|------|--------------|--|
| (7) | 模块名称 | 显示模块名称。 如果省略文件类型, 则会附加 ".ref" 作为文件类型 (扩展名)。 |
| (8) | 模块日期的创建 | 模块日期的创建 (以 "DD Mmm YYYY" 格式显示) |
| (9) | 公用符号名称 | 显示公用符号名称。 |
| (10) | 定义在模块中的公用模块数 | 显示定义在模块中的公用模块数。 输出右对齐的非零十进制数。 |

3.6 列表转换器

列表转换器输出如下列表。

| 输出列表文件名 | 输出列表名 |
|-----------|--------|
| 完全的汇编列表文件 | 完全汇编列表 |
| 错误列表文件 | 错误列表 |

在 CubeSuite 中配置完全汇编列表文件, 在 **工程树面板** 上, 选择 build tool 节点, 然后在 **属性面板** 上的 **[Assemble Options]** 标签里做设置。具体在 **[Assemble List]** 分项中的 **[Execute list converter]** 属性上选择 **[Yes]**。输出错误列表文件, 在 **[Assemble List]** 分项中设置 **[Output list converter error list file]** 属性为 **[Yes(-e)]**。输出目标是在 **[Common Options]** 标签上设置的文件夹, 具体在 **[Output File Type And Path]** 分项的 **[Intermediate file output folder]** 属性中。

备注 参见 “**B.6.1 I/O 文件**” 有关列表转换器输入和输出文件的详细说明。

3.6.1 完全汇编列表

完全汇编列表是嵌入绝对值的汇编列表和输出列表。
输出格式与汇编器输出的汇编列表格式相同。

3.6.2 错误列表

在列表转换器启动时输出出错信息并存储在出错列表中。
输出格式与汇编器输出的错误列表格式相同。

3.7 变量 / 函数信息文件生成器

变量 / 函数信息文件生成器输出以下文件。
- [变量 / 函数信息文件说明](#)

备注 有关变量 / 函数信息文件生成器的输入和输出文件的详细说明, 参见 “**B.7.1 I/O 文件**”。

3.7.1 变量 / 函数信息文件说明

变量 / 函数信息文件包含有效分配的变量和函数的信息。

在 CubeSuite 中配置变量 / 函数信息文件, 在 **工程树面板** 上, 选择 build tool 节点, 然后在 **属性面板** 上的 **[Variables/Functions Relocation Options]** 标签里做设置。在 **[Output File]** 分项中, 设置 **[Output variables/functions information file]** 属性为 **[Yes]**。在 **[Output folder for variables/functions information file]** 属性和 **[Variables/functions information file name]** 属性中设定输出目标。同时也显示在 **Build tool generated files** 节点下的工程树上。

```

;VF78K0R (1)Vx.xx
; 注意：在行开头的分号含义是该行为注释。
;           请在该项的每一节中参考 " 格式信息 "。
;(2)*** format information ***
;[sreg]
;variable,count,size,type,"file",const ;static-const
;variable,count,size,type,,const ;global-const
;variable,count,size,type,"file" ;static
;variable,count,size,type ;global
;variable,count,size,type,,const,boot ;global-const in boot
;variable,count,size,type,,boot ;global in boot
;;type : near=1 , far=2 , sreg=0
;
;[callt]
;variable,count,type,"file" ;static
;variable,count,type ;global
;variable,count,type,,boot ;global in boot
;;type : near=1 , far=2 , callt=0
;
;(3)*** gap information ***
;[callt-gap]
;(4)START (5)SIZE
; 00080H 00040H
;[base-gap]
;(4)START (5)SIZE
; 00190H 00E70H
; 01004H 00FFCH
; 02004H 00008H
; 02018H 000ECH
; 0210CH 03EF4H
; 060F7H 00001H
; 06100H 09EFCH
;[saddr-gap]
;(4)START (5)SIZE
; FFE26H 000BAH
;
;(6)*** variable information ***
[sreg]
(7)f,(8)3,(9)1,(10)1
(7)flash_a,(8)2,(9)2,(10)1
(7)flash_b,(8)2,(9)2,(10)1
;(7)var1,(8)1,(9)2,(10)1,(11)"flash.c",(12)const
;(7)var2,(8)1,(9)2,(10)1,,const
(7)var3,(8)1,(9)4,(10)1,(11)"flash.c"
;(7)boot_a,(8)1,(9)2,(10)0,,(13)boot
;(7)boot_b,(8)1,(9)2,(10)0,,(13)boot
;

```

```
;(14)*** function information ***
[callt]
;(15)f1,(16)1,(17)1,(18)"flash.c"
;(15)f2,(16)1,(17)1
;(15)func,(16)1,(17)1,,(19)boot
```

| 条目号 | 说明 | 格式 |
|------|---------------|---|
| (1) | 版本号 | 以 "x.yz" 格式显示 |
| (2) | 格式信息 (开始) | 显示变量和函数信息的初始格式信息。 |
| (3) | 空闲区域信息 (开始) | 显示 saddr 区域, BASE 区域, 和调用区域的初始空闲信息。 通过在行开头添加分号 (;) 用来注释。 |
| (4) | 空闲区域信息 (起始地址) | 显示空闲区域的起始地址。 |
| (5) | 空闲区域信息 (大小) | 显示空闲区域的大小。 |
| (6) | 变量信息 (开始) | 表明变量信息开始。 变量信息按优先级顺序输出, 从最高到最低。 由于在启动区域定义的 const,sreg , 及静态变量和变量, 在 flash 区域引用它们时不能分配到 saddr 区域, 通过在行的开头添加分号 (;) 来作注释。 |
| (7) | 变量信息 (变量名) | 输出变量名。 |
| (8) | 变量信息 (引用数) | 输出变量的引用数。 |
| (9) | 变量信息 (大小) | 显示变量的大小。 |
| (10) | 变量信息 (引用类型) | 显示变量的引用类型。 near: 1 (改变 near 区域为 saddr 区域) far: 2 (改变 far 区域为 saddr 区域) sreg: 0 (按照 sreg 说明已分配到 saddr 区域)。 |
| (11) | 变量信息 (文件名) | 输出目标源文件名称并用引号 (" ") 环绕。 尽管输出了静态变量, 但不能输出全局变量。 |
| (12) | 变量信息 (常量) | "const" 用于输出常量。 |
| (13) | 变量信息 (启动区域变量) | 如果在启动区域定义变量并在 flash 区域引用, 则输出 "boot"。 |
| (14) | 函数信息 (开始) | 表明函数信息开始。 函数信息按优先级顺序输出, 从最高到最低。 由于在 flash 区域的函数, 调用函数和静态函数不能分配到 saddr 区域, 通过在行的开头添加分号并注释这些变量。 |
| (15) | 函数信息 (函数名) | 输出函数名。 |
| (16) | 函数信息 (引用数) | 输出函数的引用数。 |

| 条目号 | 说明 | 格式 |
|------|--------------|--|
| (17) | 函数信息（引用类型） | 显示函数的引用类型。 near: 1 (改变 near 区域为 callt 区域) far: 2 (改变 far 区域为 callt 区域) sreg: 0 (已经分配给 callt 区域) |
| (18) | 函数信息（文件名） | 输出目标源文件名称并用引号 (" ") 环绕。 尽管输出了静态函数，但不能输出全局函数。 |
| (19) | 函数信息（启动区域函数） | 如果在启动区域定义函数并在 flash 区域引用，则输出 "boot"。 |

第 4 章 示例程序

本章介绍 CubeSuite 中于 CA78K0 (build tool) 有关联的示例程序列表。

4.1 C 编译器

本章节介绍了与 C 编译器相关联的示例程序列表。

4.1.1 C 源文件

```
#define TRUE    1
#define FALSE   0
#define SIZE    200

char    mark [ SIZE + 1 ] ;

void main ( void ) {
    int    i , prime , k , count ;

    count = 0 ;

    for ( i = 0 ; i <= SIZE ; i++ )
        mark [ i ] = TRUE ;
    for ( i = 0 ; i <= SIZE ; i++ ) {
        if ( mark [ i ] ) {
            prime = i + i + 3 ;
            printf ( "%6d" , prime ) ;
            count++ ;
            if ( ( count%8 ) == 0 ) putchar ( '\n' ) ;
            for ( k = i + prime ; k <= SIZE ; k += prime )
                mark [ k ] = FALSE ;
        }
    }
    printf ( "\n%d primes found." , count ) ;
}

void printf ( char *s , int i ) {
    int    j ;
    char    *ss ;

    j = i ;
    ss = s ;
}

void putchar ( char c ) {
    char    d ;
    d = c ;
}
```

备注 当文件编译完成后输出下列警告。

```
prime.c(18) : CC78K0R warning W0745 : Expected function prototype
prime.c(20) : CC78K0R warning W0745 : Expected function prototype
prime.c(26) : CC78K0R warning W0622 : No return value
prime.c(35) : CC78K0R warning W0622 : No return value
prime.c(41) : CC78K0R warning W0622 : No return value
```

4.2 汇编程序

本章节介绍了与汇编器相关联的示例程序列表。

4.2.1 k0rmain.asm

```

        NAME      SAMPM
; *****
;      HEX -> ASCII Conversion Program
;              main-routine
; *****

PUBLIC  MAIN , START
EXTRN  CONVAH
EXTRN  @_STBEG

DATA   DSEG      AT      0FFE20H
HDTSA : DS       1
STASC : DS       2

CODE   CSEG      AT      0H
MAIN  : DW       START

        CSEG
START :

        ; chip initialize
        MOVW     SP , @_STBEG

        MOV      HDTSA , #1AH
        MOVW     HL , #LOWW ( HDTSA )      ; set hex 2-code data in HL register

        CALL     !CONVAH                   ; convert ASCII <- HEX
                                           ; output BC-register <- ASCII code

        MOVW     DE , #LOWW ( STASC )      ; set DE <- store ASCII code table
        MOV      A , B
        MOV      [ DE ] , A
        INCW     DE
        MOV      A , C
        MOV      [ DE ] , A
        BR      $$

        END

```

4.2.2 k0rsub.asm

```

NAME      SAMPS
; *****
;      HEX -> ASCII Conversion Program
;
;          sub-routine
;  input condition      : ( HL )          <- hex 2 code
;  output condition    : BC-register     <- ASCII 2 code
; *****

PUBLIC  CONVAH

        CSEG
CONVAH :
        XOR      A , A
        ROL4     [ HL ]          ; hex lower code load
        CALL     !SASC
        MOV      B , A          ; store result

        XOR      A , A
        ROL4     [ HL ]          ; hex lower code load
        CALL     !SASC
        MOV      C , A          ; store result
        RET

; *****
;      subroutine      convert ASCII code
;
;  input      Acc ( lower 4bits )      <- hex code
;  output     Acc                      <- ASCII code
; *****

SASC :
        CMP      A , #0AH          ; check hex code > 9
        BC      $SASC1
        ADD      A , #07H          ; bias ( +7H )
SASC1 :
        ADD      A , #30H          ; bias ( +30H )
        RET

        END

```

第 5 章 注意事项

本章介绍了使用 CubeSuite 和 CA78K0 命令的注意事项。

(1) Kanji 代码 (双字节代码) 分类

要使用包含有 EUC 的源代码, 需要将环境变量 LANG78K 设置为 euc, 或指定 -ze 选项。

当使用 CubeSuite 时, 在属性面板上, 对 C 源文件可通过 [Compile Options] 标签设置 [Extension] 选项的 [Kanji character code of source] 属性, 或对汇编源文件可通过 [Assemble Options] 标签设置 [Others] 选项的 [Kanji character code of source files] 属性。

(2) 编译选项说明

当使用 CA78K0 时, 需注意以下几点:

- 当不可多重设定的选项进行了多次设定时, 以最后的设定为准。
- 不能省略 -c 选项后的类型设定。在编译过程中, 如果设定的选项与在 C 源文件中不同, 则设定的选项优先。此时会输出警告消息。
- 如果设定了帮助选项, 则忽略所有其它选项。

(3) 源文件名称

源文件名中除去扩展名外的部分 (主名) 默认为模块名。因此, 对源文件名有些限制。

- 关于文件名的长度, 在主操作系统允许的范围内配置主文件名中的主名和扩展名, 并使用点 (.) 作为主名和扩展名的分割符。
- 由主文件名和扩展名能使用的字符组成主操作系统允许的字符, 圆括号 (()), 分号 (;) 以及逗号 (,) 除外。注意连字符 (-) 不能用作文件名或文件名的首字符使用。不能设定包含空格和双字节字符的文件名。
- 在参数文件中, 文件名和路径名不能使用 (#) 字符。

(4) 使用汇编源文件作为输出。

当一个 C 源文件包含使用汇编语言 (例如 #asm 块或 __asm 声明) 的描述时, 加载模块文件的创建顺序是编译、汇编然后链接。

如果想使用 C 编译器先输出汇编源文件后进行汇编, 而不是直接输出目标文件, 就如使用汇编语言编写的文件, 需要注意以下事项:

- 如果包含 #asm 块和 __asm 声明的 C 源文件, 设置 -a 或 -sa 选项来启用汇编说明, 并汇编输出汇编源文件。
- 当使用 CubeSuite 时, 通过属性面板, 在 [Compile Options] 标签上, 设置 [Asseembly File] 分项的 [Output an assemble file] 属性, 设定为输出汇编源文件, 如果只输出汇编源文件, 则在 [Individual Compile Options] 标签上, 设置 [Asseembly File] 分项的 [Output an assemble file] 属性输出汇编源文件。
- 使用 CubeSuite, 在设定汇编源文件的输出时汇编器启动, 忽略编译选项 -o/-no 的设定。

(5) 堆栈符号生成设定选项 (-s)

为了锁定一个栈区，要在链接过程中设定链接选项 (-s)。

使用 CubeSuite 时，通过在 [Link Options] 标签中 [Stack] 分项上进行设置，该标签在属性面板上。使用 CubeSuite 时，在设定的源文件中包含 C 源代码时，则自动附加 -s 选项。

(6) 使用目标转换器

通过设定 -r (目标地址排序) 和 -u (指定存储值) 选项启用目标转换器。

使用 CubeSuite 时，在属性面板上，通过 [Object Convert Options] 标签，配置在 [Hex File] 中的 [Hex File Filling] 属性。

默认设定的选项。

如果当排序后的 ROM 代码未存储到目标地址时 (执行时用 “across processing” 或 “tape out” 标识)，则会发生异常中断错误。因此，确保设定 -r (不要取消设定)。

(7) 目标存储值设定选项 (-u)

如果通过目标转换器选项 (-u) 设定了起始地址，从起始地址或代码所在地址开始存储。不能在内部 RAM 区 (ED800H to FFFFFH) 进行存储。

格式说明描述如下：

```
-ufilling-value[, [start-address], size]
```

备注 [] 可以省略。

(8) 包含文件相关度

在 CubeSuite 中检查包含文件相关度时，忽略条件声明例如 #if 和注释。因此，无需构建的包含文件被错认为需要构建的文件 (在下述的例子中，判断 header1.h 和 header5.h 是需要构建的文件)。

```
#if      0
#include  "header1.h"      /* Dependence relationship judged to exist */
#else
#include  "header2.h"      /* Dependence relationship to exist */
#endif

#define   AAA
#ifdef   AAA
#include  "header3.h"      /* Dependence relationship to exist */
#else
#include  "header4.h"      /* Dependence relationship to exist */
#endif

/*
#include  "header5.h"      /* Dependence relationship judged to exist */
*/
```

在 CubeSuite 中检查包含文件相关度时，忽略条件声明例如 #if 和注释。因此，需构建的包含文件被错误当作无需构建的文件 (在下述的例子中，判断 header1.h 和 header5.h 为无需构建的文件)。

```

/* comment */ #include "header6.h" /* Dependence relationship judged not to exist */

/*
comment
*/ #include "header7.h" /* Dependence relationship judged not to exist */

```

(9) 使用网络

如果在共享文件系统的网络上安放的文件夹中创建临时文件，当使用特定类型的网络软件时会发生文件争用的现象，导致异常操作。通过合理地配置选项及环境变量避免此类争用的出现。

使用 CubeSuite 时，避免在网络环境中使用临时文件。

(10) 当使用自动编程时

以下介绍有关自动编程的命令选项的注意事项。

(a) 链接器

使用 `-zb` 选项创建启动区域的载入模块文件。

以下的 `-zb` 选项，在 flash ROM 区域设定起始地址。

当使用 CubeSuite 时，通过 [\[Link Options\]](#) 标签的**属性面板**上，在 **[Device]** 分项中配置 **[Set flash start address]** 属性和 **[Flash start address]** 属性。

创建 flash 区域的载入模块文件，输入 flash 区域的载入模块文件和 flash 区域的目标模块文件并重新链接它们。

分配 flash 区域的目标模块文件在要比使用 `-zb` 选项设定的地址更后面的地址上。

在启动区的载入模块文件以及 flash 区域的目标模块文件中混合小模式和中模式是允许的。注意，然而，flash 区的目标模块文件必须即不是小模式也不是中模式。

(b) 目标转换器

输入启动区的载入模块文件和 flash 区的目标模块文件，重新链接它们，并输入输出的载入模块文件到目标转换器中。

通过此时设定的 `-zf` 选项，能分别输出启动区的 hex 文件 (*.hxb) 和 flash 区的 hex 文件 (*.hxf)。

使用 CubeSuite 时，通过 [\[Object Convert Options\]](#) 标签，在**属性面板**上，在 **[Hex File]** 分项中配置 **[Split hex file]** 属性。

(11) 使用变量 / 函数信息文件生成器**(a) 函数调用的记号**

如果在调用时函数的参数丢失类型声明，且在变量 / 函数信息文件中设定的参数有调用指令分配的函数地址，则会由于不一致的函数接口造成程序的运行。

- C 源代码

```
int func_c() /* callt in .vfi */
{
    return 0 ;
}

void func()
{
    func2(func_c); /* W0553 */
}

int func2(int (*p)(void))
{
}
```

- 变量 / 函数信息文件

```
*** variable information ***
[sreg]
;
*** function information ***
[callt]
func_c,2,2
func2,1,2
```

如果遇到上面情况，则 C 编译器输出警告 W0553。

在函数调用原型声明中包含类型的声明。

在变量 / 函数信息文件中，也注释有关函数参数代码中的函数名称的调用说明。

(b) 如果用 AT 起始地址来设定 #pragma 节指令

通过用 AT 起始地址，分配变量和函数给调用表区域或 `saddr` 区域来设定的 `#pragma` 节指令在节中定义，可能会引起不正确的运转情况。

- C 源代码

```
#pragma section @@DATA @FCDATA AT 0FCF00H
#define dn1l    (*(int *)0xfcfc00)
int __near nil; /* sreg in .vfi */
__sreg int x1, x2;

void func()
{
    x1 = nil;
    x2 = dn1l;
}

void main()
{
    nil = 0x10;
    func();
}
```

- 变量 / 函数信息文件

```
*** variable information ***
[sreg]
;x2,1,2,0
;x1,1,2,0
;
*** function information ***
[callt]
func,1,2
```

在以上的 C 源代码中，假设变量 `x1` 和 `x2` 的值都为 `0x10`。但如果在变量 / 函数信息文件中变量 `ni1` 分配在 `saddr` 区 (从 `0xfce20`)，则程序将不会有意运行：变量 `x1` 将有 `ni1` 的值，为 `0x10`，且变量 `x2` 将有地址 `0xfcfc00` 的值。

变量 / 函数信息文件生成器不能为在节中由带指定 AT 起始地址的 `#pragma` 节指令定义的变量和函数指定 `sreg/callt`。

如果编辑变量 / 函数信息文件，不为以上函数和变量指定分配调用表区域或 `saddr` 区域。

(c) 处理警告

当通过处理变量 / 函数信息文件生成器分配函数地址到调用表区域，则可能输出警告。

- C 源代码 (当存储模式为中或大模式时)

```
void f1(void (*fp)()){
void f2(void){
void ( *fp )( void );
void main( void )
{
    f1(f2);                /* W0510: Pointer mismatch in function */
    f1((void (*)())f2);    /* Fine if cast is made */
    fp = f2; /* W0416: Illegal type and size (far/near) pointer combination */
    fp = (void (*)())f2;   /* Fine if cast is made */
}
```

- 变量 / 函数信息文件

```
;*** variable information ***
[sreg]
fp,4,4,2
;
;*** function information ***
[callt]
f2,7,2
f1,2,2
```

虽然这不会引起程序运行的任何问题，如果希望消除这个警告，在代码中进行安排即处理函数指针。

(d) 引用计数

对函数调用的引用函数地址进行计数。因此，可能没有正确地对引用的数量计数。

(e) 由编译器生成局部符号的输出

由编译器生成的局部符号也输出到变量 / 函数信息文件中，当对它们进行注释。

- 变量 / 函数信息文件

```
[sreg]
;L0003,2,1,2,"t08.c",const
;
;*** function information ***
[callt]
```

(f) 改变库文件或载入模块文件的扩展名

如果使用变量 / 函数信息文件生成器，不能改变库文件 (.lib) 或载入模块文件 (.lmf) 的扩展名。

如果改变这些，则变量 / 函数不符合进行输出处理。

附录 A 窗口参考

本章节介绍创建过程中使用的窗口 / 面板 / 对话框。

A.1 说明

下面列出了构建过程中使用的窗口 / 面板 / 对话框。

表 A-1. 窗口 / 面板 / 对话框的列表

| 窗口 / 面板 / 对话框的名称 | 功能描述 |
|---------------------|---|
| 主窗口 | 当 CubeSuiteis 启动时打开的第一个窗口。 |
| 工程树面板 | 该面板用于在树状视图中显示工程组件。 |
| 属性面板 | 该面板用于显示工程树面板中选定的 build tool、file 或 category 的详细信息以及改变信息的设置。 |
| 编辑器面板 | 该面板用于显示 / 编辑文本文件 / 源文件。 |
| 输出面板 | 该面板用于显示从 build tool 输出的消息或使用查找和替换对话框的批量查找的结果。 |
| 添加文件对话框 | 该对话框用于创建新的文件并添加到工程中。 |
| 添加文件夹和文件对话框 | 该对话框用于添加现有文件和各级文件夹到工程中。 |
| 字符串输入对话框 | 该对话框用于输入和编辑一行中的字符。 |
| 文本编辑对话框 | 该对话框用于输入和编辑多中的文本。 |
| 路径编辑对话框 | 该对话框用于编辑和添加路径。 |
| 链接顺序对话框 | 该对话框用于显示目标模块文件和库文件并输入到链接器以及配置这些链接的顺序。 |
| 构建模式设置对话框 | 该对话框用于添加和删除构建模式和以批处理的方式设置当前构建模式。 |
| 批量构建对话框 | 该对话框用于批量的构建、重新构建和清理每个已有工程的构建模式。 |
| 查找和替换对话框 | 此对话框用于查找和替换指定字符。 |
| 跳转定位对话框 | 该对话框用于移动光标到指定位置。 |
| 进度状态对话框 | 该对话框用于显示进程是如何处理的。 |
| 选项对话框 | 该对话框被用于配置 CubeSuite 环境。 |
| 用户设置对话框 | 此对话框允许自定义主窗口中的工具栏和菜单。 |
| 新建工具栏对话框 | 此对话框用于在主窗口中创建新工具栏。 |
| 重命名工具栏对话框 | 该对话框用于编辑由用户创建的工具栏的名称。 |
| 自定义键盘对话框 | 该对话框用于分配不同命令的快捷键。 |
| 重排命令对话框 | 该对话框用以改变主窗口中菜单条目和按钮的排列（包含加法和除法）。 |
| 添加现有的文件对话框 | 该对话框用于选择现有文件并添加到工程中。 |
| 文件夹浏览对话框 | 该对话框用于选择文件夹并可在调用请求下检索文件夹。 |
| 设定启动区变量 / 函数信息文件对话框 | 该对话框用于选择引导区变量 / 函数信息文件并设置在对话框的调用请求中。 |
| 设定引导区载入模块文件对话框 | 该对话框用于选择引导区加载模块文件并设置在对话框的调用请求中。 |
| 另存为对话框 | 该对话框用于将编辑中的文件或每个面板的内容保存到有名称的文件中。 |
| 打开程序对话框 | 该对话框用于选择应用程序打开文件。 |
| 选择程序对话框 | 此对话框用于选择外部工具的可执行文件。 |
| 选择外部文本编辑器对话框 | 此对话框用于选择外部文本编辑器的可执行文件。 |

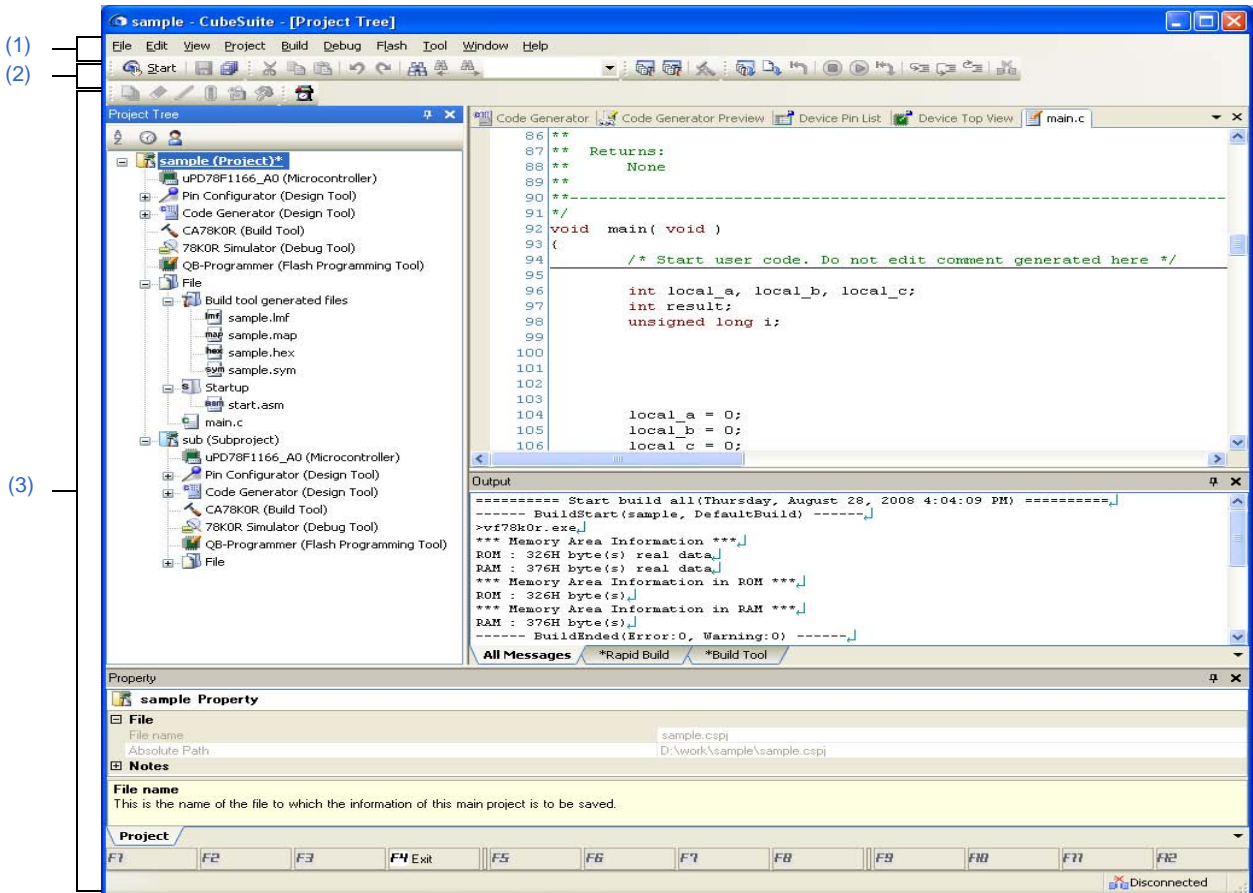
| 窗口 / 面板 / 对话框的名称 | 功能描述 |
|--------------------|---|
| 堆栈用法跟踪窗口 | 当堆栈使用跟踪程序启动时打开的第一个窗口。 |
| 堆栈大小未知 / 校准函数列表对话框 | 该对话框用于显示包含下列函数的函数列表：堆栈使用跟踪程序不能获取堆栈信息的函数；信息主动发生变化的函数以及堆栈使用跟踪程序强制设定附加容限的函数。 |
| 调整堆大小对话框 | 该对话框用于改变选定函数的信息。 |
| 打开对话框 | 该对话框用于打开存在堆栈大小的指定文件。 |

主窗口

当 CubeSuite 启动时打开的第一个窗口。

窗口用于控制用户程序的执行以及打开编译进程面板。

图 A-1. 主窗口



下列条目说明如下。

- [如何打开]
- [各部分说明]

[如何打开]

- 选择窗口 \mathbb{P} [start] >> [All programs] >> [NEC Electronics CubeSuite] >> [CubeSuite]

[各部分说明]

(1) 菜单条

显示与构建相关的菜单。

每个菜单项内容可在[用户设置对话框](#)中自定义。

(a) [Project]

[Project] 菜单显示工程操作及其它菜单项。

| | |
|-----------------------|--|
| Add New Subproject... | 关闭当前的工程并打开创建工程对话框来创建新工程。 如果当前打开的工程或文件已修改且还未保存，会显示确认消息并询问是否要保存。 |
| Open Project... | 关闭当前的工程并打开工程对话框来然后打开已存在的工程。 如果当前打开的工程或文件已修改且还未保存，会显示确认消息并询问是否要保存。 |
| 常用工程 | 显示平铺菜单用于打开或保存常用的工程。 |
| 1 路径 | [Opens your favorite project registered with [Favorite Projects] >> [1 Register to Favorite Project]. 如果没有工程收藏过，显示 "工程收藏夹"。 |
| 2 路径 | [Opens your favorite project registered with [Favorite Projects] >> [2 Register to Favorite Project]. 如果没有工程收藏过，显示 "工程收藏夹"。 |
| 3 路径 | [Opens your favorite project registered with [Favorite Projects] >> [3 Register to Favorite Project]. 如果没有工程收藏过，显示 "工程收藏夹"。 |
| 4 路径 | [Opens your favorite project registered with [Favorite Projects] >> [4 Register to Favorite Project]. 如果没有工程收藏过，显示 "工程收藏夹"。 |
| 1 收藏到工程收藏夹 | 添加当前工程路径到 [1 path]，该项在 [Favorite Projects] 中。 |
| 2 收藏到工程收藏夹 | 添加当前工程路径到 [2 path]，该项在 [Favorite Projects] 中。 |
| 3 收藏到工程收藏夹 | 添加当前工程路径到 [3 path]，该项在 [Favorite Projects] 中。 |
| 4 收藏到工程收藏夹 | 添加当前工程路径到 [4 path]，该项在 [Favorite Projects] 中。 |
| Add | 显示平铺菜单并添加子工程到工程中。 |
| Add Subproject... | 打开添加现存的子工程对话框并添加现存的子工程到工程中。 |
| Add New Subproject... | 打开建立的子工程对话框并添加新的子工程到工程中。 |
| Add File... | 打开添加现有的文件对话框添加选定的文件到工程。 |
| Add New File... | 打开添加文件对话框并创建指定文件类型的文件然后添加文件到工程中。 添加的文件可用与文件扩展名关联的应用程序打开。 |
| Add New Category | 添加新的 Category 节点到 File 节点的根部。该操作允许改变类别名。 默认类别名称是 "新建类别"。新建分项名称可与已存在的 Category 节点的名称相同。 注意：当 build tool 在运行时此菜单不可用。 |
| 设置已选工程或子工程为活动工程。 | 设置已选工程或子工程为活动工程。 |
| Close Project | 关闭当前的工程。 如果当前打开的工程或文件已修改且还未保存，会显示确认消息并询问是否要保存。 |
| Save Project | 保存当前工程到工程文件的配置信息。 |

| | |
|----------------------------|---|
| Save Project As... | 打开保存工程对话框并使用另外的名称保存当前工程到工程文件的配置信息。 |
| Remove from Project | 从工程中移除所选工程或子工程。 此子工程文件或文件本身没有从文件系统中删除。 |
| 以包的形式保存工程和 CubeSuite... | 通过保存到文件夹中保存 CubeSuite 和工程的设置。 |

(b) [Build]

[Build] 菜单显示构建处理及其他菜单项。

| | |
|--|---|
| 构建工程 | 构建工程。当子工程添加到工程中时，子工程也同时创建。 注意：当 build tool 在运行时此菜单不可用。 |
| 重新构建工程 | 重新构建工程。在子工程添加到工程中后也可以重新构建。 注意：当 build tool 在运行时此菜单不可用。 |
| Clean Project | 清除工程。在子工程添加到工程中后也可以清除。 注意：当 build tool 在运行时此菜单不可用。 |
| 快速构建 | 设定快速构建工程为可用（默认）和不可用。 |
| 更新依从 | 在工程构建后相关文件更新。当子工程被添加到工程后，子工程中的文件通过构建也被更新。 |
| 构建活动工程 | 构建活动工程。 如果活动工程是主工程，它的子工程不能构建。 注意：当 build tool 在运行时此菜单不可用。 |
| 重新构建活动工程 | 重新构建活动工程。 如果活动工程是主工程，它的子工程不能重新构建。 注意：当 build tool 在运行时此菜单不可用。 |
| Clean active project | 清除活动工程。 如果活动工程是主工程，它的子工程不能清除。 注意：当 build tool 在运行时此菜单不可用。 |
| Update Dependencies of active project | 在工程构建后相关文件更新。 |
| 停止构建 | 取消构建，重建，批量构建和清除操作。 |
| Build Mode Settings... | 打开 构建模式设置对话框 修改和添加构建模式。 |
| Batch Build... | 打开 批量构建对话框 进行批量构建。 |
| Build Option List | 在 输出面板 中列出当前构建选项的设置。 |




(2) 工具栏

显示在构建过程中使用过的按钮。

工具栏上的按钮可以在[用户设置对话框](#)中自定义。可以通过同一对话框建立新的工具栏。

(a) 构建工具栏

构建工具栏显示在构建过程中使用的按钮。

| | |
|---|---|
|  | 构建工程。当子工程添加到工程中时，子工程也同时创建。 注意：当 build tool 在运行时此按钮不可用。 |
|  | 重新构建工程。在子工程添加到工程中后也可以重新构建。 注意：当 build tool 在运行时此按钮不可用。 |
|  | 取消构建、重建、批量构建和清除操作。 |

(3) 面板显示区

在此区域显示以下面板。

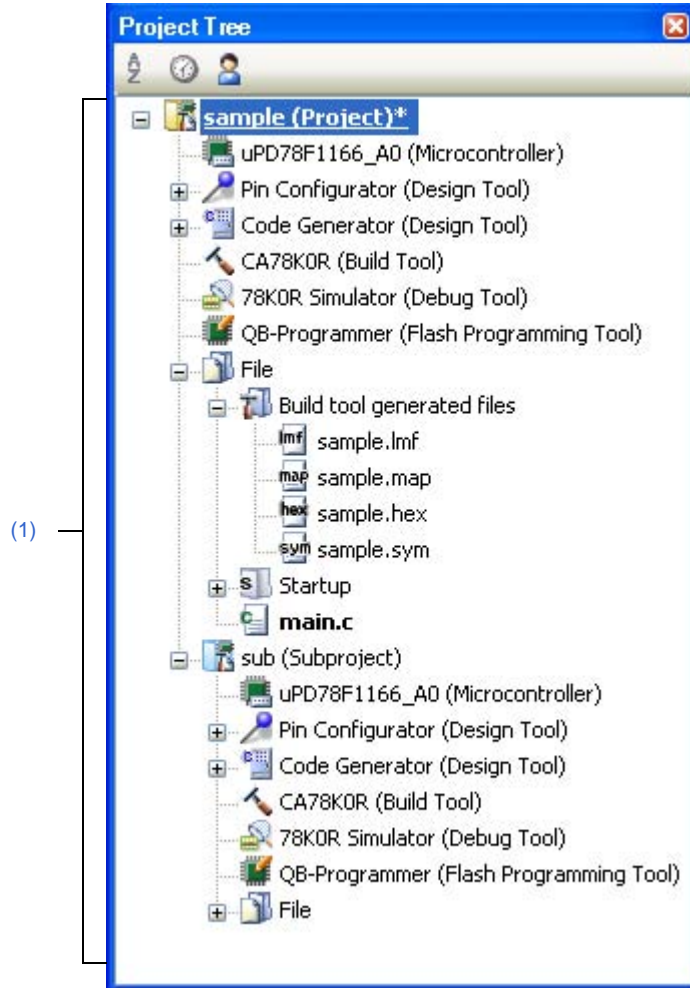
- 工程树面板
- 属性面板
- 编辑器面板
- 输出面板

参阅各面板章中有关显示内容的详细说明。

工程树面板

该面板用于在树结构视图中显示工程组件例如 build tool，源文件等。

图 A-2. 工程树面板



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [[Edit] 菜单 (仅用于工程树面板)]
- [快捷菜单]

【如何打开】

- 通过 [View] 菜单，选择 [Project Tree]。

【各部分说明】

(1) 工程树区域

工程组件显示在树结构视图中下列特定的节点中。

| 节点 | 说明 |
|--|-----|
| <i>Project name (Project)</i> (后指 “Project 节点”) | 工程名 |

| 节点 | 说明 |
|---|---|
| <i>Build tool name</i> (Build tool) (后指“Build tool 节点”) | 在工程中使用的 build tool（编译器、汇编器等）。 |
| 文件 (后指“File 节点”) | 下列添加到工程中的文件显示在节点根部。 - C 源文件 (*.c) - 汇编源文件 (*.asm) - 头文件 (*.h, *.inc) - 目标文件 (*.rel) - 库文件 (*.lib) - 链接指令文件 (*.dr, *.dir) - 变量和函数信息文件 (*.vfi) - 其他文件 (doc, xml, 等。) |
| Build tool generated files (后指“Build tool generated files 节点”) | 下列由 build tool 生成的文件直接出现在构建时创建的节点下。 - 除库工程以外 载入模块文件 (*.lmf) 链接列表文件 (*.map) 错误列表文件 (*.elk) 十六进制文件 (*.hex, *.hxb, *.hxf) 符号表格文件 (*.sym) 出错列表文件 (*.eoc) - 用于库工程 库文件 (*.lib) 列表文件 (*.lst) 不能重命名，删除和移动在节点下显示的文件。 此节点总是置于 File 节点的较低位置。 如果在构建后重新载入工程，此节点将不再出现。 |
| Startup (后指“Startup 节点”) | 此节点用于将除标准启动文件外的文件添加到工程。 此节点总是置于 File 节点的较低位置。 |
| 分项名称 (后指“Category 节点”) | 用户创建用于文件归类的分项（参阅“2.3.5 将文件归类到分项中”）。 此节点总是置于 File 节点的较低位置。 |
| <i>Subproject name</i> (Subproject) (后指“Subproject 节点”) | 添加到工程的子工程。 |

当选择各组件（节点或文件）时，在[属性面板](#)中显示详细的信息（属性）。用户可以改变设置。

备注 当选定一个以上的组件时，仅显示所有组件共有的标签。
当选定多个文件且它们共有的属性值不同时，则对应的值字段显示为空白。

此区域有以下功能。

(a) 添加文件

用户可以通过下述的步骤之一完成文件添加。
在 File 节点下添加文件。

<1> 添加现有的文件

- 选择任一 Project 节点，Subproject 节点，File 节点或文件。然后通过 [File] 菜单选择 [Add] >> [Add File...]。出现[添加现有的文件对话框](#)。选择文件进行添加。

- 选择任一 Project 节点， Subproject 节点， File 节点或文件。然后通过快捷菜单中选择 [Add] >> [Add File...]。出现**添加现有的文件对话框**。选择文件进行添加。
- 使用 Windows 资源管理器拷贝文件，然后将鼠标指针指向该区。通过 [Edit] 菜单，选择 [Paste]。
- 使用 Windows 资源管理器拖动文件然后将其放在用户想要添加文件的区域。

备注 如果从 Windows 资源管理器拖动文件而后放在工程树低位的空白区域，视为放在主工程中。

<2> 当新文件添加时

- 选择任一 Project 节点， Subproject 节点， File 节点或文件。然后通过 [File] 菜单选择 [Add] >> [Add New File...]。出现**添加文件对话框**。创建指定文件。
- 选择任一 Project 节点， Subproject 节点， File 节点或文件。然后通过快捷菜单选择 [Add] >> [Add New File...]。出现**添加文件对话框**。创建指定文件。

备注 在**添加文件对话框**中的指定位置创建空白文件。

(b) 从工程中移除文件

用户可以通过下述的步骤之一来完成将文件从工程中移除。
在该项操作中移除的文件不会在文件系统中删除。

- 从工程中选择用户想移除的文件。然后通过 [Project] 菜单选择 [Remove from Project]。
- 从工程中选择用户想移除的文件。然后通过快捷菜单选择 [Remove from Project]。

(c) 移动文件

可以通过下述步骤移动文件。
文件在 File 节点下移动。

- 拖动用户想要移动的文件然后将其放在目的地。

- 备注 1.** 当文件放在主工程或子工程中时保留特定选项。
- 2.** 当文件在不同工程之间或同一工程的主工程和子工程之间拖放时，复制文件，而没有移动。注意：该操作没有在每个文件中保留特定选项设置。

(d) 添加分项

可以通过下述程序来添加 Category 节点。
在 File 节点下添加 Category 节点。

- 通过 [Project] 菜单选择 [Add New Category]。
- 通过 Project 节点， Subproject 节点或 File 节点中的任意快捷菜单中选择 [Add New Category]。

- 备注 1.** 默认分项名称为 "New category"。
- 2.** 新的分项名可以修改为与现有 Category 节点名相同的名称。

(e) 移动分项

可以通过下列程序移动 Category 节点。
Category 节点在 File 节点下移动。

- 拖动用户想要移动的 Category 节点然后将其放在目的地。

- 备注 1.** 当拖放 Category 节点在主工程或子工程中时，保留 Category 节点中文件的特定选项设置。
- 2.** 当 Category 节点在不同工程之间或同一工程的主工程和子工程之间拖放时，会复制 Category 节点，而不是移动。注意：未保留包含在 Category 节点中的每个文件的特定选项设置。

(f) 添加文件夹

用户可以通过下列步骤通过资源管理器添加文件夹。

添加文件夹在 **File** 节点下。

文件夹作为分项添加。

- 从资源管理器中拖动文件夹，然后放在目的地。打开**添加文件夹和文件对话框**。设定要添加文件的类型和文件夹中的子目录级别。

注意事项 用户可以同时拖放文件夹和文件到该区。

(g) 修改子工程的显示顺序按照构建顺序放置

子工程从顶部开始按照构建顺序显示。因此构建顺序可以通过改变子工程的显示顺序来改变。

工程必须按照先子工程后主工程的构建顺序。

(h) 配置标准构建选项

当标准构建选项被改变时，属性以粗体格式在**属性面板**中显示。

用户可以通过下述步骤改变标准构建选项为当前设置（取消粗体）。

- 选择 **Build tool** 节点然后在快捷菜单中选择 **[Set to Default Build Option for Project]**。

备注 标准构建选项的配置对于所有工程（主工程和子工程）均有效。

(i) 文件和分项的排序

可以根据下列步骤按照文件名、时间标记或用户定义对 **File** 和 **Category** 节点进行排序。

- 选择工具栏中的按钮之一。

下列表格对按钮做了说明。



默认选定为默认。

| 按钮 | 说明 |
|----|---|
| | 按照名称对 File 和 Category 节点进行排序。 : 升序 : 降序 : 升序 |
| | 按照时间标记对 File 和 Category 节点进行排序。 : 降序 : 升序 : 降序 |
| | 按照用户定义（默认）对 File 和 Category 节点进行排序。 用户可以通过拖放文件和 Category 节点改变显示的顺序。 |

(j) 显示编辑中的文件

当**编辑器面板**中编辑的文件添加到工程时，且文件没有保存过，文件名后存在“*”。当文件保存过时，删除“*”。

| | |
|-------------|---------|
| 保存过的文件 | main.c |
| 编辑后没有保存过的文件 | main.c* |



(k) 以粗体格式显示源文件，由于设置了特定的构建选项。

选项不同于工程常用选项（个性编译选项，个性汇编选项）的源文件图标变的不同于一般图标。

| | |
|-------------|--------|
| 带工程常用选项的文件 | main.c |
| 带个性化创建选项的文件 | main.c |



(l) 高亮显示有只读属性的文件

添加到工程中的只读文件以斜体格式显示。

| | |
|-----------|---|
| 不带只读属性的文件 |  main.c |
| 有只读属性的文件 |  <i>main.c</i> |




(m) 高亮显示文件不存在。

添加到工程但不存在的文件变成灰色且图标颜色淡化。

| | |
|--------|---|
| 存在的文件 |  main.c |
| 不存在的文件 |  <i>main.c</i> |

(n) 高亮显示构建目标文件

<1> 在构建（快速构建），重新构建，编译或汇编过程中出现错误的文件被高亮显示，如下例所示。

| | |
|------------|---|
| 没有错误或警告的文件 |  main.c |
| 有错误的文件 |  main.c |
| 有警告的文件 |  main.c |

备注 1. 有错误和警告的文件高亮显示成红色。

2. 当构建选项（常用选项或个性选项）或构建模式改变时，高亮显示取消。



<2> 下述文件的名称显示为**粗体**。

- 在编辑后没有编译的源文件
- 执行清除后的源文件
- 构建工具选项改变后的源文件
- 任何构建模式改变后的源文件

备注 工程打开后立刻以粗体格式显示文件名。构建执行后取消以粗体格式显示。

(o) 非高亮显示构建目标文件



设置无构建目标为高亮显示的文件如下例所示。

| | |
|---------|---|
| 构建目标文件 |  main.c |
| 无构建目标文件 |  main.c |

(p) 高亮显示没有改变的工程



添加到工程的文件组件和工程组件的属性改变，工程名后存在“*”，并且以粗体格式显示。

当工程保存时，取消粗体显示。

| | |
|---------|--|
| 还未改变的工程 |  sample (Project) |
| 还未改变的工程 |  sample (Project)* |

(q) 高亮显示活动的工程

在活动工程下添加下划线。

| | |
|--------|---|
| 未活动的工程 |  sample (Project)* |
| 活动工程 |  <u>sample (Project)*</u> |

(r) 运行编辑器

在**编辑器面板**中打开有特定扩展名的文件。当在**选项对话框**中设定使用外部编辑器时，打开外部编辑器的文件。打开其他文件使用与 OS 相关联的应用程序。

注意事项 如果带扩展名的文件与 OS 无关联，则文件无法显示。

用户可以通过下述的步骤之一来打开编辑器。

- 双击文件。
- 选择文件然后在快捷菜单中选择 [Open]。
- 选择文件然后按 [Enter] 键。

文件可以在**编辑器面板**中打开，如下。

- C 源文件 (*.c)
- 汇编源文件 (.asm)
- 头文件 (.h, .inc)
- 链接指令文件 (.dr, .dir)
- 变量和函数信息文件 (*.vfi)
- Map 文件 (.map)
- 符号表格文件 (.sym)
- Hex 文件 (.hex, .hxb, .hxf)
- 文本文件 (.txt)

备注 用户可以使用在打开文件下方而不是那些被列在**编辑器面板**上方的方法。

- 拖动文件并将其放入**编辑器面板**。
- 选择文件然后在快捷菜单中选择 [Open with Internal Editor...]

[[Edit] 菜单 (仅用于工程树面板)]

| | |
|--------|---|
| Copy | 复制选择的文件或 Category 节点到剪贴板。 在编辑文件名或分项名时，复制选定的字符到剪贴板。 注意：当文件或 Category 节点没有选定时，仅该菜单可用。 |
| Paste | 将剪贴板的内容插入到工程树上所选择节点的相同层中（如果选定 category 节点，插入到它的下面一层）。 当编辑文件名或分项名时，插入剪贴板的内容。 注意：当文件或 Category 节点存在时，仅该菜单可用。然而，当选定多个 File 和 Category 节点且 build tool 正在执行操作时，菜单失效。 |
| Rename | 可以重命名选定的 project、subproject、file 和 category 节点 按 [Enter] 键确认重命名。按 [ESC] 键取消重命名。 当文件选定时，实际文件名同样改变。 当选择的文件添加到其它工程时，那些文件名同样改变。 注意：只有选定 Project, Subproject, File 和 Category 节点时，该菜单可用。注意：当 build tool 正在执行操作时，不允许重命名。 |

[快捷菜单]**(1) 当选择 Project 节点时**

| | |
|--------|---|
| 构建活动工程 | 构建活动工程。 如果活动工程是主工程，它的子工程不能构建。 注意：当 build tool 在运行时此菜单不可用。 |
|--------|---|

| | |
|---|--|
| 重新构建 <i>活动工程</i> | 重新构建活动工程。 如果活动工程是主工程，它的子工程不能重新构建。 注意：当 build tool 在运行时此菜单不可用。 |
| Clean <i>active project</i> | 清除活动工程。 如果活动工程是主工程，它的子工程不能清除。 注意：当 build tool 在运行时此菜单不可用。 |
| Open Folder with Explorer | 使用资源管理器打开包含选定工程的工程文件的文件夹。 |
| Add | 显示层叠菜单用以添加子工程和文件到工程。 |
| Add Subproject... | 打开添加现有子工程对话框用以添加被选择的子工程到工程。 |
| Add New Subproject... | 打开创建工程对话框用以添加创建子工程到工程。 |
| Add File... | 打开添加现有的文件对话框添加选定的文件到工程。 |
| Add New File... | 打开添加文件对话框用以创建具有选择文件类型的文件并添加到工程。 添加的文件可用与文件扩展名关联的应用程序打开。 |
| Add New Category | 添加新建 Category 节点到 File 节点的根部。该操作允许改变类别名。 多达 200 个字符可设定。 默认分项名称为 "New category"。新的分项名可以修改为与现有 Category 节点名相同的名称。 当 build tool 正在运行，并且如果分项有 20 级嵌套时，该菜单不可用。 |
| Set <i>selected project</i> as Active Project | 设置选定的工程作为活动工程。 |
| 以包的形式保存工程和 CubeSuite... | 通过保存到文件夹中保存 CubeSuite 和工程的设置。 |
| Paste | 该菜单总是失效。 |
| Rename | 可以重命名选定的工程。 |
| Property | 在属性面板上显示选定工程的属性。 |

(2) 当 **Subproject** 节点选定时

| | |
|-----------------------------|--|
| 构建 <i>活动工程</i> | 构建活动工程。 注意：当 build tool 在运行时此菜单不可用。 |
| 重新构建 <i>活动工程</i> | 重新构建活动工程。 注意：当 build tool 在运行时此菜单不可用。 |
| Clean <i>active project</i> | 清除活动工程。 注意：当 build tool 在运行时此菜单不可用。 |
| Open Folder with Explorer | 使用资源管理器打开包含选定子工程的子工程文件的文件夹。 |
| Add | 显示层叠菜单用以添加 subproject ， files 和 Category 节点到工程。 |

| | |
|---|--|
| Add Subproject... | 打开添加现有子工程对话框用以添加被选择的子工程到工程。 不能添加子工程到另一个子工程。 |
| Add New Subproject... | 打开创建工程对话框用以添加创建子工程到工程。 不能添加子工程到另一个子工程。 |
| Add File... | 打开添加现有的文件对话框添加选定的文件到工程。 |
| Add New File... | 打开添加文件对话框用以创建具有选择文件类型的文件并添加到工程。 添加的文件可用与文件扩展名关联的应用程序打开。 |
| Add New Category | 添加新建 Category 节点到 File 节点的根部。该操作允许改变类别名。 多达 200 个字符可设定。 默认分项名称为 "New category"。新的分项名可以修改为与现有 Category 节点名相同的名称。 当 build tool 正在运行，并且如果分项有 20 级嵌套时，该菜单不可用。 |
| Set selected subproject as Active Project | 设置选择的子工程作为活动工程。 |
| Remove from Project | 从工程中移除选择的子工程。 在该项操作中子工程文件本身没有从文件系统中删除。 当选择的子工程是活动工程时，它不能从工程中移除。 注意：当 build tool 在运行时此菜单不可用。 |
| Paste | 该菜单总是失效。 |
| Rename | 用户可以重命名选择的子工程。 |
| Property | 在属性面板上显示选定子工程的属性。 |

(3) 当 Build tool 节点选择时

| | |
|-------------------|--|
| 构建工程 | 构建选择工程（主工程或子工程）。当子工程添加到工程中时，子工程也同时创建。 注意：当 build tool 在运行时此菜单不可用。 |
| 重新构建工程 | 重新构建选择工程（主工程或子工程）。在子工程添加到工程中后也可以重新构建。 注意：当 build tool 在运行时此菜单不可用。 |
| Clean Project | 清除选择工程（主工程或子工程）。在子工程添加到工程中后也可以清除。 注意：当 build tool 在运行时此菜单不可用。 |
| 设置工程的默认构建工程 | 设置当前构建选项为选择工程的标准选项。当子工程添加时，它不设置。 当不同于标准选项的构建选项设定时，它的属性以粗体格式显示。 |
| Set Link Order... | 打开 链接顺序对话框 显示目标模块文件和库文件用以设置它们的连接顺序。 注意：当 build tool 在运行时此菜单不可用。 |
| Property | 在 属性面板 上显示选择 build tool 的属性。 |

(4) 当 File 节点选择时

| | |
|---------------------|--|
| Add | 显示层叠菜单用以添加文件和 Category 节点到工程。 |
| Add File... | 打开 添加现有的文件对话框 添加选定的文件到工程。文件直接添加到该节点下。 添加文件可以使用与文件扩展名对应的应用程序打开。文件直接添加到该节点。 |
| Add New File... | 打开 添加文件对话框 用以创建具有选择文件类型的文件并添加到工程。文件直接添加到该节点下。 添加的文件可用与文件扩展名关联的应用程序打开。 |
| Add New Category | 添加新的 Category 节点到该节点的根部。用户可以重命名分项。 多达 200 个字符可设定。 默认分项名称为 "New category"。新的分项名可以修改为与现有 Category 节点名相同的名称。 当 build tool 正在运行，并且如果分项有 20 级嵌套时，该菜单不可用。 |
| Remove from Project | 该菜单总是失效。 |
| Copy | 该菜单总是失效。 |
| Paste | 将剪贴板的内容插入到与 File 节点相同的级别中。 |
| Rename | 该菜单总是失效。 |
| Property | 在 属性面板 上显示选择 Category 节点的属性。 |

(5) 当文件选择时

| | |
|------------------------------|---|
| Compile | 编译选择的 C 源文件。 注意该菜单仅当 C 源文件（除非创建目标文件以外的文件）选择时显示。 注意：当 build tool 在运行时此菜单不可用。 |
| Assemble | 汇编选择的汇编源文件。 注意该菜单仅当 C 源文件（除非创建目标文件以外的文件）选择时显示。 注意：当 build tool 在运行时此菜单不可用。 |
| Open | 使用与文件扩展名对应的应用程序打开选择的文件（参阅“(r) 运行编辑器”）。 注意：该菜单当多个文件选择时无效。 |
| Open with Internal Editor... | 使用 编辑器面板 打开选择的文件。 注意：该菜单当多个文件选择时无效。 |

| | |
|-----------------------------------|--|
| Open with Selected Application... | 打开 打开程序对话框 用以添加选择的文件到项目。 注意：该菜单当多个文件选择时无效。 |
| Open Folder with Explorer | 使用资源管理器打开包含选择文件的文件夹。 |
| Add | 显示层叠菜单用以添加文件和 Category 节点到工程。 |
| Add File... | 打开 添加现有的文件对话框 添加选定的文件到工程。该文件添加到与选择文件相同的级别。 |
| Add New File... | 打开 添加文件对话框 用以创建具有选择文件类型的文件并添加到工程。该文件添加到与选择文件相同的级别。 添加的文件可用与文件扩展名关联的应用程序打开。 |
| Add New Category | 在与被选择文件的相同级别添加新的 Category 节点。用户可以重命名分项。 多达 200 个字符可设定。 默认分项名称为 "New category"。新的分项名可以修改为与现有 Category 节点名相同的名称。 当 build tool 正在运行，并且如果分项有 20 级嵌套时，该菜单不可用。 |
| Remove from Project | 从工程中移除选择的文件。 在该项操作中移除的文件本身没有从文件系统中删除。 注意：当 build tool 在运行时此菜单不可用。 |
| Copy | 拷贝所选择的文件到剪贴板。 在编辑文件名时，选择的字符被拷贝到剪贴板。 |
| Paste | 该菜单总是失效。 |
| Rename | 用户可以重命名选择的文件。 现有的文件也重命名。 当添加选择的文件到另一个工程时，它也会重命名。 |
| Property | 在 属性面板 上显示选择文件的属性。 |

(6) 当选择 **Build tool generated files** 节点

| | |
|----------|--------------------------|
| Property | 在 属性面板 上显示该节点的属性。 |
|----------|--------------------------|

(7) 选择 **Startup** 节点时

| | |
|---------------------|---|
| Add | 显示层叠菜单用以添加文件和 Category 节点到工程。 |
| Add File... | 打开 添加现有的文件对话框 添加选定的文件到工程。文件直接添加到该节点下。 添加的文件可用与文件扩展名关联的应用程序打开。 |
| Add New File... | 打开 添加文件对话框 用以创建具有选择文件类型的文件并添加到工程。文件直接添加到该节点下。 添加的文件可用与文件扩展名关联的应用程序打开。 |
| Add New Category | 添加新的 Category 节点到该节点的根部。用户可以重命名分项。 多达 200 个字符可设定。 默认分项名称为 "New category"。新的分项名可以修改为与现有 Category 节点名相同的名称。 当 build tool 正在运行，并且如果分项有 20 级嵌套时，该菜单不可用。 |
| Remove from Project | 该菜单总是失效。 |
| Copy | 该菜单总是失效。 |
| Paste | 将剪贴板的内容插入到与 File 节点相同的级别中。 |
| Rename | 该菜单总是失效。 |
| Property | 在 属性面板 上显示该节点的属性。 |

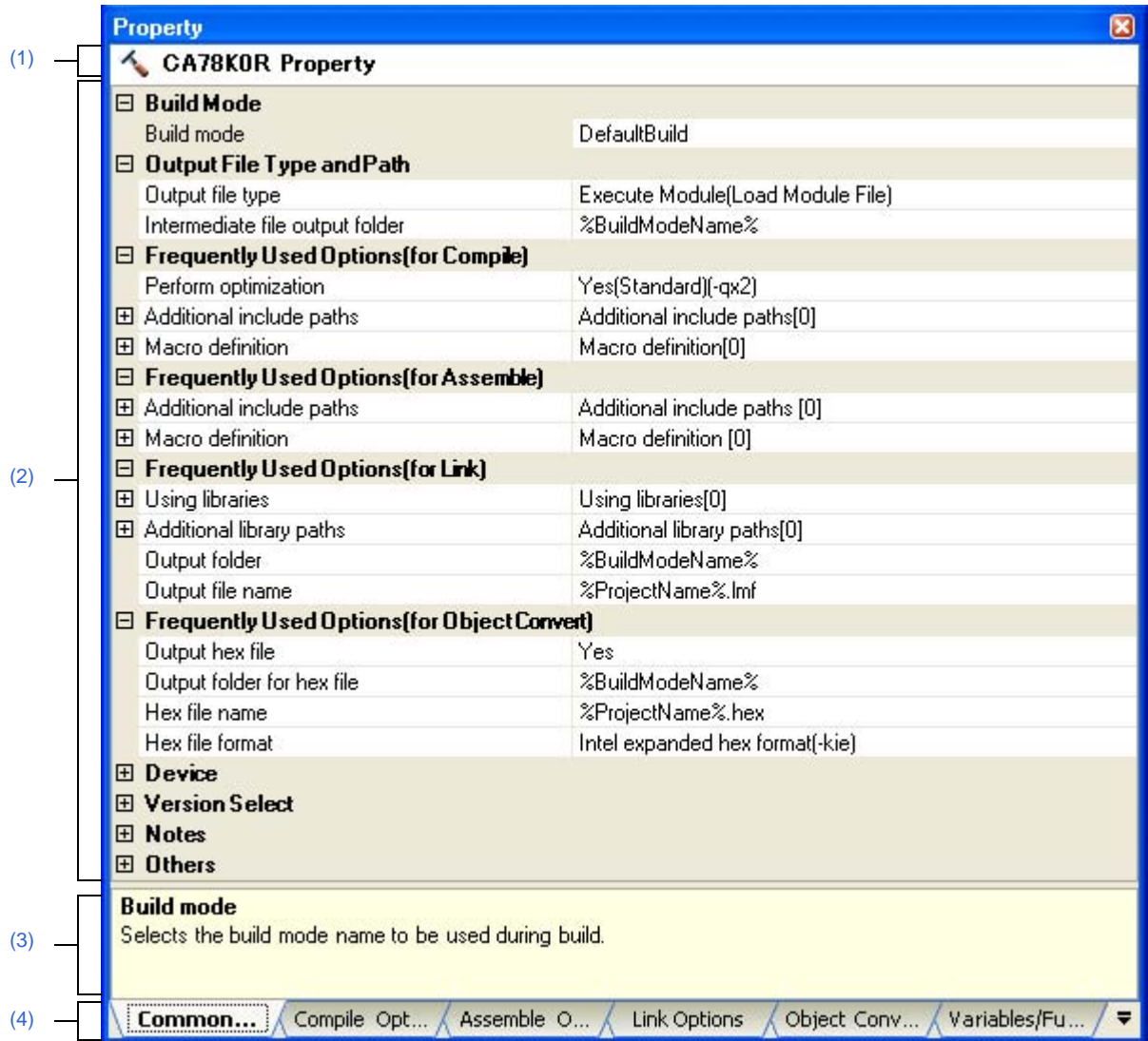
(8) 当 Category 节点选择时

| | |
|---------------------|--|
| Add | 显示层叠菜单用以添加文件和 Category 节点到工程。 |
| Add File... | 打开添加现有的文件对话框添加选定的文件到工程。文件直接添加到该节点下。 添加的文件可用与文件扩展名关联的应用程序打开。 |
| Add New File... | 打开添加文件对话框用以创建具有选择文件类型的文件并添加到工程。文件直接添加到该节点下。 添加的文件可用与文件扩展名关联的应用程序打开。 |
| Add New Category | 添加新的 Category 节点到该节点的根部。用户可以重命名分项。 多达 200 个字符可设定。 默认分项名称为 "New category"。新的分项名可以修改为与现有 Category 节点名相同的名称。 当 build tool 正在运行，并且如果分项有 20 级嵌套时，该菜单不可用。 |
| Remove from Project | 从工程中移除选择的 Category 节点。 注意：当 build tool 在运行时此菜单不可用。 |
| Copy | 拷贝所选择的 Category 节点到剪贴板。 在编辑分项名时，选择的字符拷贝到剪贴板。 |
| Paste | 插入剪贴板的内容到选择 Category 节点的更低级别中。 当分项名被编辑时，插入剪贴板的内容。 |
| Rename | 用户可以重命名选择的 Category 节点。 |
| Property | 在属性面板上显示选择 Category 节点的属性。 |

属性面板

该面板用以显示通过各分项在工程树面板上选择的 Build tool 节点、File 或 Category 节点的详细信息以及改变信息的设置。

图 A-3. 属性面板



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [[Edit] 菜单 (仅用于工程树面板)]
- [快捷菜单]

[如何打开]

- 在**工程树面板**上，选择 **Build tool** 节点，**file** 或者 **category** 节点，然后选择 **[View]** 菜单或快捷菜单的 **[Property]** 属性。

备注 当选择**工程树面板**上的 **Build tool** 节点、**file** 或 **category** 节点时，属性面板会打开，显示选择节点的详细信息。

[各部分说明]

(1) 选择的节点区

显示**工程树面板**上选择的节点名称。
当选择多个节点时，该区为空白。

(2) 详细信息显示 / 改变区

该区内，显示通过列表中各分项在**工程树面板**上选择的 **Build tool** 节点、**File** 或 **Category** 节点的详细信息。并且信息的设置可以直接改变。

标记 说明分项中的所有条目都打开。标记 说明所有条目都折叠。用户可以通过单击这些标记或双击分项名来打开 / 折叠这些条目。

标记 **HEX** 说明仅十六进制数允许输入到文字框。

参阅每个列表的章节，其中列表是关于包含在分项中的显示 / 设置细节以及它们的内容的列表。

(3) 属性说明区

显示在详细信息显示 / 改变区选择的分项和它们内容的主要说明。

(4) 标签选定区域

显示详细信息的分项通过选择一个标签改变。

此面板包含下述列表（参阅包含有每个显示 / 设置细节的标签的章节）。

(a) 当在工程子目录结构面板中选择 **Build tool** 节点时

- [\[Common Options\]](#) 标签
- [\[Compile Options\]](#) 标签
- [\[Assemble Options\]](#) 标签
- [\[Link Options\]](#) 标签
- [\[Object Convert Options\]](#) 标签
- [\[Create Library Options\]](#) 标签
- [\[Variables/Functions Relocation Options\]](#) 标签

(b) 当在工程子目录结构面板中选择文件时

- [\[Build Settings\]](#) 标签（对于 C 源文件、汇编源文件、链接指令文件，变量 / 函数信息文件，目标文件和库文件）
- [\[Individual Compile Options\]](#) 标签（对于 C 源文件）
- [\[Individual Assemble Options\]](#) 标签（对于汇编源文件^{注意}）
- [\[File Information\]](#) 标签

注 当在 [\[Assembly File\]](#) 分项中通过 [\[Individual Compile Options\]](#) 标签在 [\[Output assemble file\]](#) 中选择 [\[Yes\]](#) 时，该列表也同时显示。

(c) 当在工程子目录结构面板中选择 **Category** 节点、**File** 节点、**Build tool generated files** 节点或 **Startup** 节点时

- [\[Category Information\]](#) 标签

备注 当在**工程树面板**选择多个组件时，仅显示所有组件共用的列表。如果属性值修改，则对于所有选择组件都有效。

[[Edit] 菜单 (仅用于工程树面板)]

| | |
|------------|---------------------------|
| Undo | 取消属性值之前的编辑操作。 |
| Cut | 当编辑属性值时，剪切选择的字符并拷贝它们到剪贴板。 |
| Copy | 拷贝选择的属性字符到剪贴板。 |
| Paste | 当编辑属性值时，插入剪贴板上的内容。 |
| Delete | 当编辑属性值时，删除选择的字符串。 |
| Select All | 当编辑属性值时，选择选定属性的所有字符。 |

[快捷菜单]

| | |
|----------------------|--|
| Undo | 取消属性值之前的编辑操作。 |
| Cut | 当编辑属性值时，剪切选择的字符并拷贝它们到剪贴板。 |
| Copy | 拷贝选择的属性字符到剪贴板。 |
| Paste | 当编辑属性值时，插入剪贴板上的内容。 |
| Delete | 当编辑属性值时，删除选择的字符串。 |
| Select All | 当编辑属性值时，选择选定属性的所有字符。 |
| Reset to Default | 恢复选择条目的配置到工程的默认配置。 对于 [Individual Compile Options] 标签和 [Individual Assemble Options] 标签，恢复到常用选项的配置。 |
| Reset All to Default | 恢复所有当前列表的配置到工程的默认配置。 对于 [Individual Compile Options] 标签和 [Individual Assemble Options] 标签，恢复到常用选项的配置。 |

[Common Options] 标签

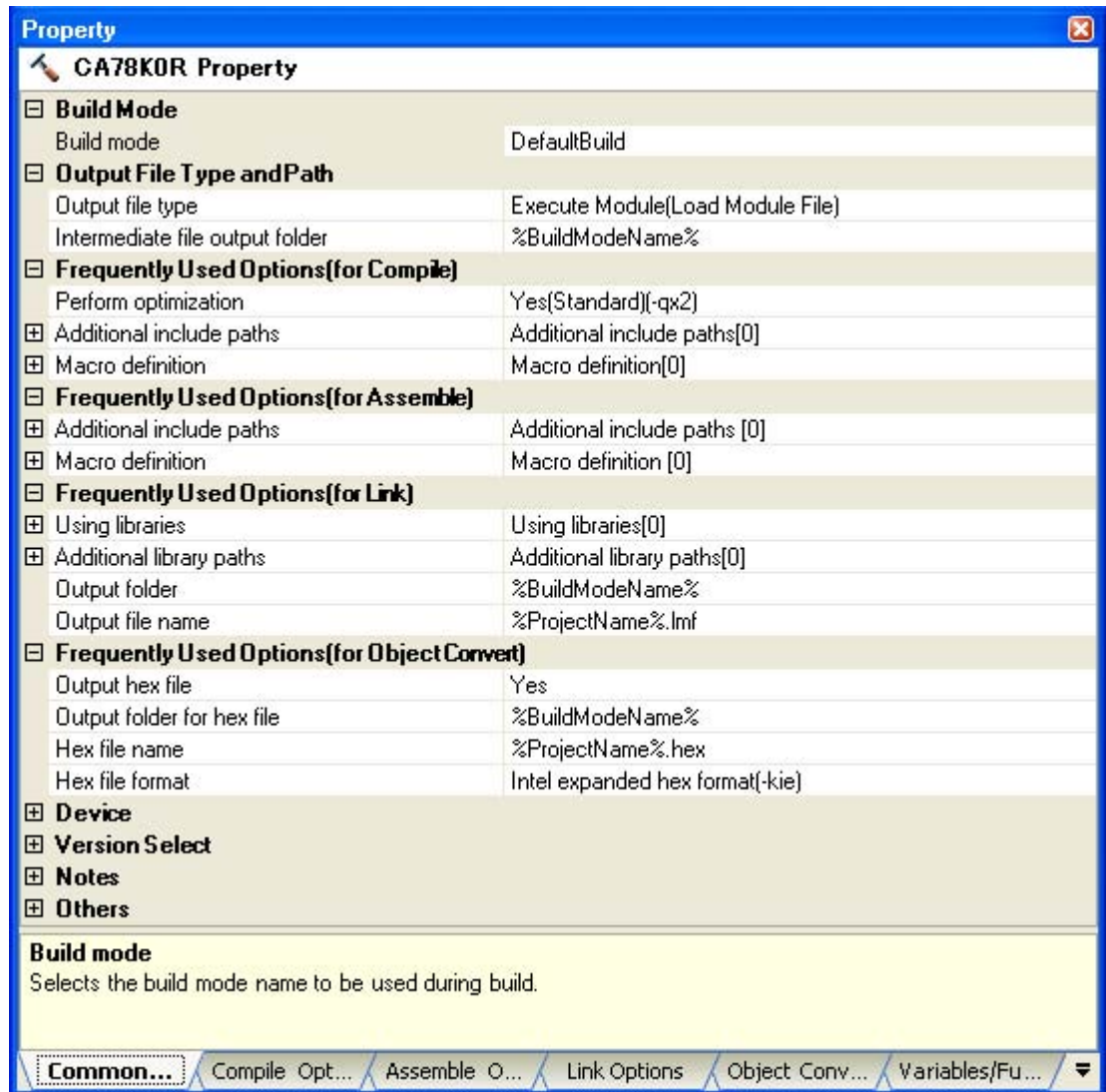
此标签说明根据下列及可修改配置的不同对关于 **build tool** 的详细情况做了分类。

- (1) [Build Mode]
- (2) [Output File Type and Path]
- (3) [Frequently Used Options(for Compile)]
- (4) [Frequently Used Options(for Compile)]
- (5) [Frequently Used Options(for Link)]
- (6) [Frequently Used Options(for Object Convert)]
- (7) [Device]
- (8) [Version Select]
- (9) [Notes]
- (10) [Others]

备注 如果 [Frequently Used Options] 分项属性被改变，属性值与之有着相同名称的标签将会发生相应的改变。

| [Common Options] 标签中的分项 | Corresponding 标签 |
|--|-----------------------------|
| [Frequently Used Options(for Compile)] 分项 | [Compile Options] 标签 |
| [Frequently Used Options(for Compile)] 分项 | [Assemble Options] 标签 |
| [Frequently Used Options(for Link)] 分项 | [Link Options] 标签 |
| [Frequently Used Options(for Object Convert)] 分项 | [Object Convert Options] 标签 |

图 A-4. 属性面板 :[Common Options] 标签



[各分项说明]

(1) [Build Mode]

有关构建模式及修改配置信息的详细说明。

| | | | |
|-----------------------|-------------|-----------------------|-----------------------|
| 构建模式 | 构建时选择使用构建模式 | | |
| | 默认值 | 默认构建 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | 默认构建 | 在建立新的工程时采用默认的构建模式来构建。 |
| 构建模式加入到工程中 (非默认构建) | | 构建时构建模式加入到工程中 (非默认构建) | |

(2) [Output File Type and Path]

有关输出文件类型及路径方面的详细说明及可修改的相关配置信息。

| | | | | |
|-----------|--|---|----------------------------|--|
| 输出文件类型 | <p>在构建时选择生成文件的类型。</p> <p>此文件类型的设置必须在调试中进行。</p> <p>除了工程库，只显示 [Execute Module(Load Module File)] 和 [Execute Module(Hex File)]。在 [Hex File] 的 [Output hex file] 属性设置成 [YES] 时，只有 [Execute Module(Load Module File)] 显示，[Hex File] 在 [Object Convert Options] 标签中。</p> <p>只在 [LIBRARY] 下进行库工程操作。</p> | | | |
| | 默认值 | 可执行模块 (载入模块文件) | | |
| | 如何改变 | 通过下拉列表选择。 | | |
| | 限制 | 可执行模块 (载入模块文件) | 通过可执行格式 (载入模块文件) 构建生成文件。 | |
| | | 执行模块 (Hex 文件) | 通过可执行格式 (HEX 文件) 构建生成文件。 | |
| Library | | 通过库格式 (库文件) 构建生成文件。 | | |
| 关联文件输出文件夹 | <p>给中间文件指定路径及文件夹 (object module files (*.rel), cross-reference list files (*.xrf), 等等) 以备用于输出。</p> <p>如果指定的是相对路径，则参考路径为主工程或子工程的文件夹。</p> <p>如果指定的是绝对路径，则参考路径为主工程或子工程的文件夹。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%BuildModeName%: 用生成模式名字替换</p> <p>如果为空，则作为已设置工程文件夹。</p> | | | |
| | 默认值 | %BuildModeName% | | |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示文件夹浏览对话框的来编辑。 | | |
| | 限制 | 多达 247 个字符 | | |

(3) [Frequently Used Options(for Compile)]

有关用于编译的常用选项方面的详细说明及可修改的相关配置信息。

| | | | |
|--------------------------|---|---|---------------------|
| 进行优化 | 选择编译优化的类型。 相当于编译器的 <code>-qx</code> 选项。 | | |
| | 默认值 | Yes(Standard)(-qx2) | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(Speed precedence)(-qx1) | 进行优化时运行速度优先。 |
| | | Yes(Standard)(-qx2) | 进行优化时同时考虑运行速度和模块大小。 |
| | | Yes(Code size precedence)(-qx3) | 进行优化时模块大小优先。 |
| Yes(Detail setting) | | 在 [Compile Options] 标签中出现出现 [Optimization(Details)] 分项。进行优化时，在分项中选定的选项有优先级。 如果将 [Optimization(Details)] 分项中所有属性选定为 [No(-nq)] 时，优化将不会起作用。 | |
| No(-nq) | 优化将不起作用。 | | |
| Additional include paths | 指定附加的编译中的 include 路径。 下面的宏变量是内嵌的宏。 <code>%BuildModeName%</code> : 用生成模式名字替换 <code>%ProjectName%</code> : 用工程名称替换。 <code>%CubeSuitePath%</code> : 用 CubeSuite 的绝对安装路径替换。 当省略此选项时，仅查询编译器的标准文件夹。路径的参考点是工程文件夹。 相当于编译器的 <code>-i</code> 选项。 指定的 include 路径显示为子属性。 | | |
| | 默认值 | Additional include paths[<i>number of defined items</i>] | |
| | 如何改变 | 点击 [...] 按钮，通过 路径编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 | |
| | 限制 | 多达 259 个字符 多达 64 项可以定义，但这也包括多个被用于链接工具的路径。 | |
| 宏定义 | 指定需要定义的宏名称。 通过 "宏名称 = 定义值" 这种格式来指定，每行一个宏。这 " <code>=def</code> " 部分被省略，在这种情况下，" <code>1</code> " 为默认定义值。 相当于编译器的 <code>-d</code> 选项。 指定的宏显示为子属性。 | | |
| | 默认值 | 宏定义 [<i>定义条目的数目</i>] | |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 | |
| | 限制 | 多达 256 个字符 设定最多可达 30 个条目。 | |

(4) [Frequently Used Options(for Compile)]

有关用于汇编的常用选项方面的详细说明及可修改的相关配置信息。

| | | |
|--------------------------|--|---|
| Additional include paths | <p>在汇编时指定附加的 include 路径。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%BuildModeName%: 用生成模式名字替换</p> <p>%ProjectName%: 用工程名称替换。</p> <p>%CubeSuitePath%: 用 CubeSuite 的绝对安装路径替换。</p> <p>当忽略该选项时, 仅查询汇编程序标准文件夹。路径的参考点是工程文件夹。</p> <p>相当于汇编程序的 -i 选项。</p> <p>指定的 include 路径显示为子属性。</p> | |
| | 默认值 | Additional include paths[<i>number of defined items</i>] |
| | 如何改变 | <p>点击 [...] 按钮, 通过路径编辑对话框来编辑。</p> <p>对于子属性, 可以用文本框直接输入。</p> |
| | 限制 | <p>多达 259 个字符</p> <p>设定最多可达 64 个条目。尽管这也包含链接工具使用的路径。</p> |
| 宏定义 | <p>指定需要定义的宏名称。</p> <p>通过 "宏名称 = 定义值" 这种格式来指定, 每行一个宏。这 "=def" 部分被省略, 在这种情况下, "1" 为默认定义值。</p> <p>相当于汇编程序的 -d 选项。</p> <p>指定的宏显示为子属性。</p> | |
| | 默认值 | 宏定义 [定义条目的数目] |
| | 如何改变 | <p>点击 [...] 按钮, 通过文本编辑对话框来编辑。</p> <p>对于子属性, 可以用文本框直接输入。</p> |
| | 限制 | <p>多达 256 个字符</p> <p>设定最多可达 30 个条目。</p> |

(5) [Frequently Used Options(for Link)]

有关用于链接的常用选项方面的详细说明及可修改的相关配置信息。

此分项在库工程中显示为不可见。

| | | |
|-----|--|---|
| 使用库 | <p>指定使用的除标准库以外的库文件名 (*.lib)。</p> <p>每行添加一个文件。</p> <p>从库文件路径查找库文件。</p> <p>相当于链接器的 -b 选项。</p> <p>指定的库文件名作为子属性来显示。</p> | |
| | 默认值 | 使用库 [定义条目的数目] |
| | 如何改变 | <p>点击 [...] 按钮, 通过文本编辑对话框来编辑。</p> <p>对于子属性, 可以用文本框直接输入。</p> |
| | 限制 | <p>多达 259 个字符</p> <p>设定最多可达 64 个条目。</p> |

| | | |
|---------|---|--|
| 附加库文件路径 | 指定除标准库以外的查找路径。 下面的宏变量是内嵌的宏。 %BuildModeName% : 用生成模式名字替换 %ProjectName% : 用工程名称替换。 %CubeSuitePath% : 用 CubeSuite 的绝对安装路径替换。 库文件从库路径中查找。如果指定的是相对路径, 参考路径是工程文件夹。 相当于链接程序的 -i 选项。 指定的库路径作为子属性来显示。 | |
| | 默认值 | 附加的库路径 [定义条目的数目] |
| | 如何改变 | 点击 [...] 按钮, 通过 路径编辑对话框 来编辑。 对于子属性, 可以用文本框直接输入。 |
| | 限制 | 多达 259 个字符 多达 64 个项目可以设定。 |
| 输出文件夹 | 指定保存生成模块的文件夹。 如果指定的是相对路径, 则参考路径为主工程或子工程的文件夹。 如果指定的是绝对路径, 则参考路径为主工程或子工程的文件夹。 下面的宏变量是内嵌的宏。 %BuildModeName% : 用生成模式名字替换 如果为空, 则作为已设置工程文件夹。 | |
| | 默认值 | %BuildModeName% |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示 文件夹浏览对话框 的来编辑。 |
| | 限制 | 多达 247 个字符 |
| 输出文件名 | 指定输出的加载模块文件名。 使用 ".lmf" 扩展名。如果扩展名省略, 会自动加上 ".lmf"。 相当于链接程序的 -o 选项。 下面的宏变量是内嵌的宏。 %ProjectName% : 用工程名称替换。 如果为空, 默认为 " %ProjectName%.lmf "。 | |
| | 默认值 | %ProjectName%.lmf |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 多达 259 个字符 |

(6) [Frequently Used Options(for Object Convert)]

有关用于对象转换的常用选项方面的详细说明及可修改的相关配置信息。
此分项在库工程中显示为不可见。

| | | |
|-----------|--|-------------|
| 输出 hex 文件 | 选择是否输出 hex 文件。 相当于目标转换程序的 -o 选项。 | |
| | 默认值 | Yes |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| No(-no) | | 不输出 hex 文件。 |

| | | | |
|--------------------------------|---|--|--|
| hex 文件的输出文件夹 | 指定文件夹用于保存 hex 文件。 相当于目标转换程序的 -o 选项。 如果指定的是相对路径，则参考路径为主工程或子工程的文件夹。 如果指定的是绝对路径，则参考路径为主工程或子工程的文件夹。 下面的宏变量是内嵌的宏。 %BuildModeName%: 用生成模式名字替换 如果为空，则作为已设置工程文件夹。 只有当 [Output hex file] 属性设置成 [Yes]，该属性才会显示。 | | |
| | 默认值 | %BuildModeName% | |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示文件夹浏览对话框的来编辑。 | |
| | 限制 | 多达 247 个字符 | |
| Hex 文件名 | 指定 hex 文件名。 相当于目标转换程序的 -o 选项。 可以任意指定扩展名。 下面的宏变量是内嵌的宏。 %ProjectName%: 用工程名称替换。 只有当 [Output hex file] 属性设置成 [Yes]，该属性才会显示。 | | |
| | 默认值 | %ProjectName%.hex | |
| | 如何改变 | 直接在文本框中输入。 | |
| | 限制 | 多达 259 个字符 | |
| Hex 文件格式 | 选择生成 hex 文件的格式。 相当于目标转换程序的 -k 选项。 当 [Output hex file] 属性为 [No(-no)] 时，不显示该属性。 | | |
| | 默认值 | Intel 扩展 hex 格式 (-kie) | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Intel 标准 hex 格式 (-ki) | 指定 Intel 标准的 hex 格式作为 hex 文件的生成格式。 |
| | | Intel 扩展 hex 格式 (-kie) | 指定 Intel 扩展的 hex 格式作为 hex 文件的生成格式。 |
| | | Motorola S 型格式 (标准地址) (-km) | 采用 Motorola S 型格式 (标准格式) 作为 hex 文件的生成格式。 |
| Motorola S 型格式 (32-位地址) (-kme) | | 采用 Motorola S 型格式 (32-位地址) 作为 hex 文件的生成格式。 | |
| 扩展的 Tektronix hex 格式 (-kt) | 采用扩展的 Tektronix 十六进制格式作为 hex 文件的生成格式。 | | |

(7) [Device]

显示有关设备方面的详细信息以及可修改的配置信息。

| | | |
|------|---|--|
| 安全编号 | 给闪存芯片设备指定安全编号。 这对应于链接器的 <code>-gi</code> 选项。 | |
| | 默认值 | 0xffffffffffffffff |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0x00000000000000000000 ~ 0xffffffffffffffff (20- 数字 (10- 字节) 十六进制数) |

(8) [Version Select]

有关 build tool 的版本及修改配置信息的详细说明。

| | | |
|---------------|--|---|
| 使用编译包安装文件夹 | 显示已安装使用的编译包的文件夹。 | |
| | 默认值 | 安装文件夹名 |
| | 如何改变 | 不允许改变。 |
| 使用编译包版本 | 选择已用编译包的版本。 此设置适用于所有的构建模式。 | |
| | 默认值 | 通常最新版本已安装。 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | 通常最新版本已安装。 使用最新版本的已安装编译包。 已安装编译包的版本 在编译包中使用已选版本。 |
| 最近的编译包版本要预先安装 | 选中 [Using compiler package version] 属性中 [Always latest version which was installed] 的选项，使用中的编译包版本会显示。 此设置适用于所有的构建模式。 只在选中 [Using compiler package version] 属性的 [Always latest version which was installed] 选项，该属性才会显示。 | |
| | 默认值 | 最近安装的编译包版本 |
| | 如何改变 | 不允许改变。 |

(9) [Notes]

有关注意事项的详细信息及可修改的配置信息的说明。

| | | |
|----|---|--|
| 备注 | 添加备注到 build tool 中。 每项一行。 此设置适用于所有的构建模式。 加入的备注作为子属性显示。 | |
| | 默认值 | 备注 [number-of-characters] |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 256 个字符 设定最多可达 256 个条目。 |

(10)[Others]

有关 build tool 的其他详细说明及可修改配置信息。

| | | | |
|------------------------------|---|--|--------------------|
| 输出消息格式 | <p>特定的构建消息格式</p> <p>下面的宏变量是内嵌的宏。</p> <p>%Program%: 在执行状态下替换程序名称。</p> <p>%Options%: 在构建执行状态下替换命令行选项。</p> <p>%FileName%: 替换正构建的文件名。</p> <p>如果是空白的, 就指定设定成 "%ProjectName%%Options%"。</p> | | |
| | 默认值 | %FileName% | |
| | 如何改变 | 直接输入在文本框内 (多达 256 个字符) 或从下拉列表中选择。 | |
| | 限制 | %FileName% | 在输出消息中显示文件名。 |
| | | %FileName%: %Options% | 在输出消息中显示文件名和命令行选项。 |
| %FileName%: %Options% | 在输出消息中显示程序名和命令行选项。 | | |
| 构建选项列表格式 | <p>指定构建选项列表的显示格式 (参阅 "2.14.3 显示构建选项列表")。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%Program%: 在执行状态下替换程序名称。</p> <p>%Options%: 在构建执行状态下替换命令行选项。</p> <p>%FileName%: 替换正构建的文件名。</p> | | |
| | 默认值 | %FileName% : %Program% %Options% | |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示字符串输入对话框的来编辑。 | |
| | 限制 | 多达 256 个字符 | |
| | | | |
| 临时文件夹 | <p>指定文件夹作为存放通过在执行过程中对每条包含在 build tool 中的命令保存而产生的临时文件的文件夹。</p> <p>对应各命令中的 -t 选项。</p> <p>如果设定的是采用相对路径, 那么主工程或子工程文件夹的路径就参照该路径。</p> <p>如果指定的是绝对路径, 则参考路径为主工程或子工程的文件夹。</p> <p>如果为空, 则作为已设置工程文件夹。</p> | | |
| | 默认值 | 空白 | |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示文件夹浏览对话框的来编辑。 | |
| | 限制 | 多达 200 个字符 | |
| | | | |
| 在构建过程之前执行的命令 | <p>在构建过程之前指定已执行的命令。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%ProjectFolder%: 用工程文件夹的绝对路径替换。</p> <p>%OutputFolder%: 用输出文件夹的绝对路径替换。</p> <p>%OutputFile%: 用输出文件的绝对路径替换。</p> <p>指定的命令显示为子属性。</p> | | |
| | 默认值 | 在构建过程之前执行的命令 [多条定义项] | |
| | 如何改变 | 点击 [...] 按钮, 通过文本编辑对话框来编辑。 对于子属性, 可以用文本框直接输入。 | |
| | 限制 | 多达 1023 个字符 设定最多可达 64 个条目。 | |
| | | | |

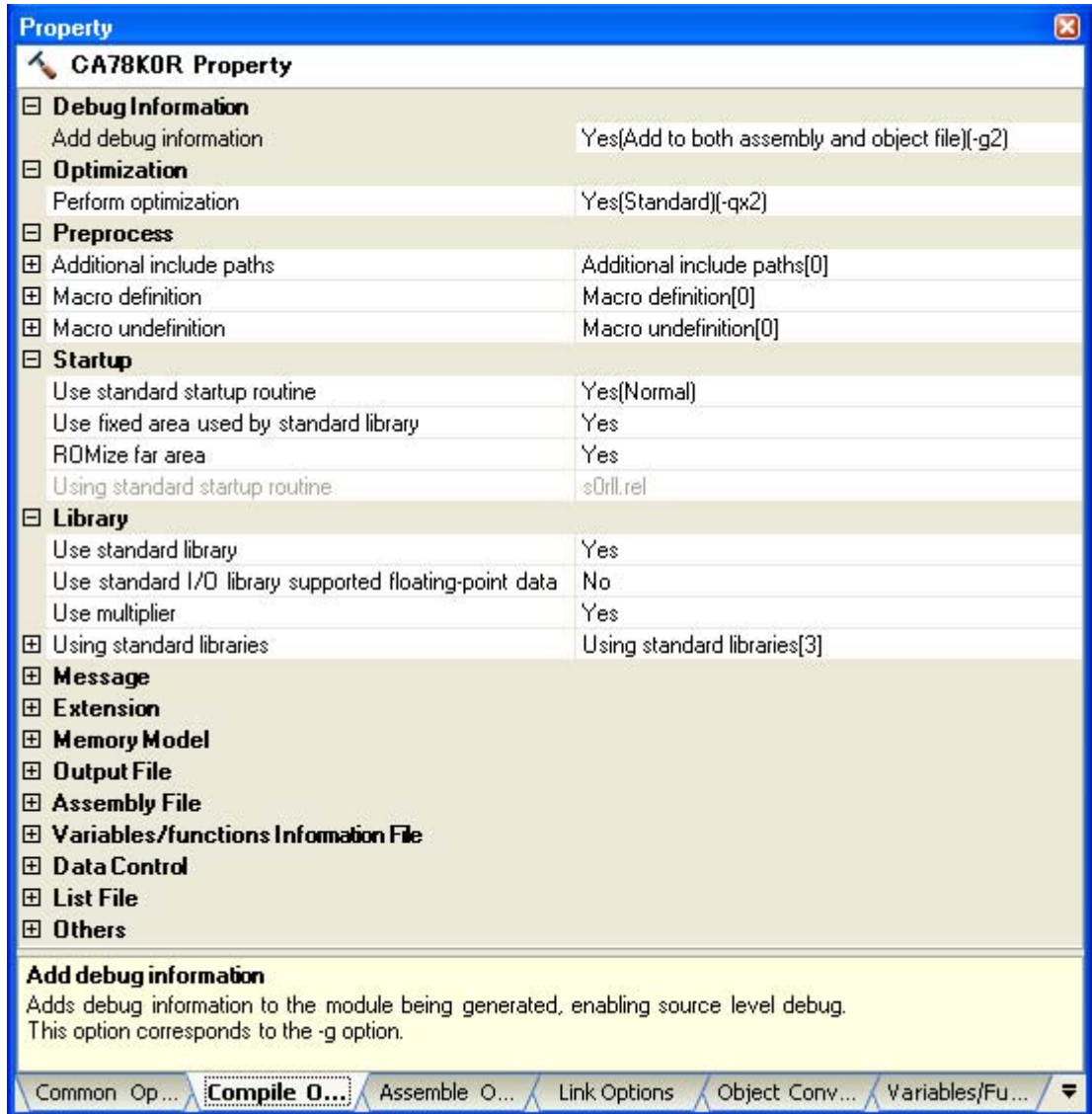
| | | |
|---------------|--|--|
| 在构建过程之后执行过的命令 | <p>在构建过程之后指定已执行的命令。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%ProjectFolder%: 用工程文件夹的绝对路径替换。</p> <p>%OutputFolder%: 用输出文件夹的绝对路径替换。</p> <p>%OutputFile%: 用输出文件的绝对路径替换。</p> <p>指定的命令显示为子属性。</p> | |
| 默认值 | 在构建过程之后执行的命令 [<i>多条定义项</i>] | |
| 如何改变 | <p>点击 [...] 按钮，通过 文本编辑对话框 来编辑。</p> <p>对于子属性，可以用文本框直接输入。</p> | |
| 限制 | <p>多达 1023 个字符</p> <p>设定最多可达 64 个条目。</p> | |

[Compile Options] 标签

此标签显示按以下分类的编译器及可更改设置的详细信息。

- (1) [Debug Information]
- (2) [Optimization]
- (3) [Optimization(Details)]
- (4) [Preprocess]
- (5) [Startup]
- (6) [Library]
- (7) [Message]
- (8) [Extension]
- (9) [Memory Model]
- (10) [Output File]
- (11) [Assembly File]
- (12) [Variables/Functions Information File]
- (13) [Data Control]
- (14) [List File]
- (15) [Others]

图 A-5. 属性面板 : [Compile Options] 标签



[各分项说明]

(1) [Debug Information]

显示详细的调试信息，设置可以更改。

| | | |
|--------|---|---|
| 加入调试信息 | 选择是否在将生成的模块中加入调试信息来允许源文件级调试。 相当于编译器的 <code>-g</code> 选项。 | |
| | 默认值 | 是 (添加到汇编和目标文件中)(-g2) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | 是 (只添加到目标文件中)(-g1) 在将要产生的目标模块文件中加入调试信息。 |
| | 是 (添加到汇编和目标文件中)(-g2) 在将要产生的目标模块文件中加入调试信息。 | |
| | 不 (-ng) 在将要产生的目标模块文件中加入调试信息。 | |

(2) [Optimization]

显示详细的优化信息，此设置可以改变。

| | | | |
|---------------------|--|---|---------------------|
| 进行优化 | 选择编译优化的类型。 相当于编译器的 <code>-qx</code> 选项。 | | |
| | 默认值 | Yes(Standard)(-qx2) | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(Speed precedence)(-qx1) | 进行优化时运行速度优先。 |
| | | Yes(Standard)(-qx2) | 进行优化时同时考虑运行速度和模块大小。 |
| | | Yes(Code size precedence)(-qx3) | 进行优化时模块大小优先。 |
| Yes(Detail setting) | | [Optimization(Details)] 分项说明如下：进行优化时，在分项中选定的选项有优先级。 在 [Optimization(Details)] 所有的分项属性中 [No] 属性选择时，优化将不会起作用。 | |
| No(-nq) | 不指定优化。 | | |

(3) [Optimization(Details)]

显示详细的优化信息，此设置可以改变。

仅当 [\[Optimization\]](#) 的 `[Perform optimization]` 属性为 `[Yes(Detail setting)]` 时，分项才会显示。

| | | | | | |
|--|---|---|--|---|----|
| 交换计算公式的次序 | 选择是否输出通过交换公式的执行顺序来提高寄存器的利用率代码。 相当于编译器的 <code>-qw</code> 选项。 | | | | |
| | 默认值 | Yes(Swap order of formula operations)(-qw) | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(Swap order of formula operations)(-qw)</td> <td>交换计算公式的次序。</td> </tr> <tr> <td>No</td> <td>不交换计算公式的次序。</td> </tr> </tbody> </table> | Yes(Swap order of formula operations)(-qw) | 交换计算公式的次序。 | No |
| Yes(Swap order of formula operations)(-qw) | 交换计算公式的次序。 | | | | |
| No | 不交换计算公式的次序。 | | | | |
| 分配自动变量给寄存器或者 SADDR 空间 | 选择是否自动分配自动变量给寄存器和 <code>saddr</code> 空间。 相当于编译器的 <code>-qx</code> 选项。 | | | | |
| | 默认值 | 是 (-qx) | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>是 (-qx)</td> <td>自动分配自动变量给寄存器和 <code>saddr</code> 空间。</td> </tr> <tr> <td>No</td> <td>自动变量不自动分配到寄存器或 <code>saddr</code> 区域。</td> </tr> </tbody> </table> | 是 (-qx) | 自动分配自动变量给寄存器和 <code>saddr</code> 空间。 | No |
| 是 (-qx) | 自动分配自动变量给寄存器和 <code>saddr</code> 空间。 | | | | |
| No | 自动变量不自动分配到寄存器或 <code>saddr</code> 区域。 | | | | |
| 将寄存器变量分配到寄存器或 <code>saddr</code> 区域 | 选择是否将寄存器变量分配到寄存器或 <code>saddr</code> 区域。 相当于编译器的 <code>-qr</code> 选项。 | | | | |
| | 默认值 | No | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-qr)</td> <td>Assigns register variables to registers and assigns them also to the <code>saddr</code> area.</td> </tr> <tr> <td>No</td> <td>不分配寄存器变量到 <code>saddr</code> 区域。</td> </tr> </tbody> </table> | Yes(-qr) | Assigns register variables to registers and assigns them also to the <code>saddr</code> area. | No |
| Yes(-qr) | Assigns register variables to registers and assigns them also to the <code>saddr</code> area. | | | | |
| No | 不分配寄存器变量到 <code>saddr</code> 区域。 | | | | |
| 对字符型 (<code>char</code>) 不使用符号扩展 | 选择是否执行无泛积分扩展的字符相关计算。 相当于编译器的 <code>-qc</code> 选项。 | | | | |
| | 默认值 | 是 (-qc) | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>是 (-qc)</td> <td>不采用泛积分扩展法进行字符 - 相关性的计算^{注意}</td> </tr> <tr> <td>No</td> <td>采用泛积分扩展法进行字符 - 相关性的计算。</td> </tr> </tbody> </table> | 是 (-qc) | 不采用泛积分扩展法进行字符 - 相关性的计算 ^{注意} | No |
| 是 (-qc) | 不采用泛积分扩展法进行字符 - 相关性的计算 ^{注意} | | | | |
| No | 采用泛积分扩展法进行字符 - 相关性的计算。 | | | | |
| 定义字符为 <code>unsigned</code> 字符 | 选择是否不采用 <code>unsigned</code> 字符限定词来定义字符。 相当于编译器的 <code>-qu</code> 选项。 | | | | |
| | 默认值 | No | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>是 (-qu)</td> <td>将无修饰符的字符型解释为 <code>unsigned</code> 字符型。</td> </tr> <tr> <td>No</td> <td>不将无修饰符的字符型解释为 <code>unsigned</code> 字符型。</td> </tr> </tbody> </table> | 是 (-qu) | 将无修饰符的字符型解释为 <code>unsigned</code> 字符型。 | No |
| 是 (-qu) | 将无修饰符的字符型解释为 <code>unsigned</code> 字符型。 | | | | |
| No | 不将无修饰符的字符型解释为 <code>unsigned</code> 字符型。 | | | | |

| | | |
|---|--|---|
| 优化分支指令 | 选择是否优化分支指令 相当于编译器的 <code>-qj</code> 选项。 | |
| | 默认值 | 是 (<code>-qj</code>) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | 是 (<code>-qj</code>) 优化分支指令。 No 不指定优化分支指令。 |
| 用库代码替换固定代码 (大小优先的优化) | 选择是否用库代码替换固定代码。 相当于编译器的 <code>-ql</code> 选项。 | |
| | 默认值 | 是 (不可替换) (<code>-ql1</code>) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | 是 (不可替换) (<code>-ql1</code>) 不用库代码替换固定代码。 进行优化时模块大小优先。 |
| | 是 (只在处理之前替换 / 之后功能) (<code>-ql2</code>) 仅将函数前 / 后的程序化处理替换为库代码。 | |
| Yes(Replace process before/after function, use low level libraries and subroutinize same codes) (<code>-ql3</code>) 仅将函数前 / 后的程序化处理替换为库代码。 同样, 使用低级别库, 并进行相同代码的子程序化。 | | |
| No 不用库代码替换固定代码。 进行优化时运行速度优先。 | | |
| 产生相关分支的 <code>switch</code> 分支表。 | 选择是否产生相关分支的 <code>switch</code> 分支表。 相当于编译器的 <code>-qx</code> 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes (<code>-qt</code>) 产生相关分支的 <code>switch</code> 分支表。 No 不指定产生相关分支的 <code>switch</code> 分支表。 |
| 调试优化 | 选择是否进行调试优化。 相当于编译器的 <code>-d</code> 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes (<code>-qg</code>) 实现调试优化。 No 不用为调试指定执行优化。 |

注 当指定选项 `-qc` 时的计算结果如下:

| 计算目标 | 计算结果 |
|----------------------------------|---------------|
| unsigned 字符类型变量和 unsigned 字符类型变量 | unsigned 字符类型 |
| unsigned 字符类型变量和 signed 字符类型变量 | unsigned 字符类型 |

| 计算目标 | 计算结果 |
|-----------------------------------|---------------|
| 有符号字符类型变量和有符号字符类型变量 | 有符号字符类型 |
| 从 -128 到 255 的常量和 unsigned 字符类型变量 | unsigned 字符类型 |
| 从 -128 到 127 的常量和有符号字符类型变量 | 有符号字符类型 |
| 带前缀 U 的从 0 到 255 的常量和有符号字符类型变量 | unsigned 字符类型 |

(4) [Preprocess]

显示详细的预处理信息，设置可以改变。

| | | |
|--------------------------|--|--|
| Additional include paths | 指定附加的编译中的 include 路径。 下面的宏变量是内嵌的宏。 %BuildModeName%: 用生成模式名字替换 %ProjectName%: 用工程名称替换。 %CubeSuitePath%: 用 CubeSuite 的绝对安装路径替换。 当省略此选项时，仅查询编译器的标准文件夹。路径的参考点是工程文件夹。 相当于编译器的 -i 选项。 指定的 include 路径显示为子属性。 | |
| | 默认值 | Additional include paths[<i>number of defined items</i>] |
| | 如何改变 | 点击 [...] 按钮，通过 路径编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 259 个字符 设定最多可达 64 个条目。尽管这也包含链接工具使用的路径。 |
| 宏定义 | 指定需要定义的宏名称。 通过 "宏名称= 定义值" 这种格式来指定，每行一个宏。"= 定义值" 部分可以省略，此时默认值为 1。 相当于编译器的 -d 选项。 指定的宏显示为子属性。 | |
| | 默认值 | 宏定义 [<i>定义条目的数目</i>] |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 256 个字符 设定最多可达 30 个条目。 |
| 取消宏定义 | 指定需要取消定义的宏名称。 通过 "宏名称" 这种格式来指定，每行一个宏。 相当于编译器的 -u 选项。 指定的宏显示为子属性。 | |
| | 默认值 | 取消宏定义 [<i>定义条目的数目</i>] |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 256 个字符 设定最多可达 30 个条目。 |

(5) [Startup]

显示详细的启动信息，此设置可改变。

| | | | |
|---------------------|---|---|--|
| 使用标准启动程序 | 选择是否在链接时链接，由编译器提供的写有启动程序的目标模块文件。 | | |
| | 默认值 | - 当选择 [Memory Model] 的 [Output objects for flash] 属性为 [Yes(-zf)] 时 [Yes(For flash area)] - 当选择 [Output objects for flash] 为 [No] [Yes(Normal)] | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(Normal) | 链接编译器提供的目标模块文件。 当 [Output objects for flash] 属性选为 [Yes(-zf)] 时，不显示此项。 |
| | | Yes(For boot area) | 链接编译器提供的引导区域目标模块文件。 当 [Output objects for flash] 属性选为 [Yes(-zf)] 时，不显示此项。 |
| Yes(For flash area) | | 链接编译器提供的闪存区域目标模块文件。 当 [Output objects for flash] 属性选为 [No] 时，不显示此项。 | |
| No | | 不链接编译器提供的目标模块文件。 | |
| 使用标准库使用的固定区域。 | 选择是否使用标准库 brk, sbrk, malloc, calloc, realloc, free, exit, rand, srand, div, ldiv, strtok, atof, strtod, 数学函数和浮点计算的运行库使用的固定区域 (RAM)。 如果这些函数不会被用到，选择 [No] 可以保留 RAM。 当 [Use standard startup routine] 属性为 [No] 时，不显示该属性。 | | |
| | 默认值 | Yes | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes | 使用标准库使用的固定区域。 |
| | | No | 不使用标准库使用的固定区域。 |
| ROM 化 far 区域 | 选择如何进行 ROM 化 far 区域。 当 [Use standard startup routine] 属性为 [No] 时，不显示该属性。 | | |
| | 默认值 | Yes | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes | ROM 化 far 区域 |
| | | No | 不要 ROM 化 far 区域 |

| | | |
|----------|---|------------|
| 使用标准启动程序 | 显示当前的设置下，在链接时使用的标准启动程序的文件名。 ^注 当 [Use standard startup routine] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 使用启动程序文件名。 |
| | 如何改变 | 不允许改变。 |

注 启动程序文件的命名规则如下：

```
s0r<model><lib><flash>.rel
```

< 模式 >

| | |
|---|---|
| m | 当存储模式既不是小模式也不是中模式，且也不能进行 ROM 化 far 区域时。 |
| l | 当存储模式为大模式时，且也进行了 far 区域 ROM 化。 |

<lib>

| | |
|------|-------------------|
| None | 当标准库使用的固定区域没有使用时。 |
| l | 当标准库使用的固定区域使用时。 |

<flash>

| | |
|------|-----------------|
| None | 当标准目标生成后 |
| b | 当引导区域目标生成后 |
| e | 当 flash 区域目标生成后 |

(6) [Library]

显示库的详细信息以及可修改的配置信息。

| | | |
|-----------------|--|---|
| 使用标准库 | 选择链接时是否链接标准库。 | |
| | 默认值 | Yes |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes 链接时链接标准库。 No 链接时不链接标准库。 |
| 使用标准 I/O 支持的浮点数 | 选择是否使用支持浮点数输入输出的 sprintf, sscanf, printf, vprintf, 和 vsprintf。 仅当 [Use standard library] 属性为 [Yes]，[Memory Model] 的 [Use static model] 属性为 [No] 时，该属性才显示。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes 使用支持输入输出的浮点数标准库。 No 不使用支持输入输出的浮点数标准库。 |

| | | |
|---------|--|--------------------|
| 使用乘法和除法 | 选择是否使用支持乘法和除法的标准库。 是否存在乘法器和除法器取决于使用的微控制器。 当微控制器不包含乘法器、除法器或 [Use standard library] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | Yes |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| No | | 不使用支持乘法和除法的标准库。 |
| 使用乘法 | 选择是否使用支持乘法的标准库。 是否存在乘法器取决于使用的微控制器。 当微控制器不包含乘法器，[Use standard library] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | Yes |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| No | | 不使用支持乘法的标准库。 |
| 使用标准库 | 显示当前设置中链接使用的标准库的文件名和文件数目。 链接库文件名作为子属性来显示。 ^注 当 [Use standard library] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 使用标准库 [使用的标准库数目] |
| | 如何改变 | 不允许改变。 |

注 库文件的命名规则如下：

```
cl0r<mul/div><model><float><flash>.lib
```

<mul/div>

| | |
|------|-------------|
| None | 当没有使用乘法器时 |
| x | 当使用乘数时 |
| d | 当使用乘法器和除法器时 |

<模式>

| | |
|---|----------------|
| m | 当存储模式是小模式或中模式时 |
| l | 当存储模式是大模式时 |

<float>

| | |
|------|----------------------|
| None | 当没有使用支持浮点数的标准 I/O 库时 |
| f | 当使用标准库和运行库时 |

<flash>

| | |
|------|-----------------|
| None | 当标准或引导区域目标生成后 |
| e | 当 flash 区域目标生成后 |

(7) [Message]

显示消息的详细信息以及可修改的配置信息。

| | | | |
|---------------------------|--|---------------------------|-----------------|
| 冗长模式 | 在构建中选择是否在输出面板中显示编译器执行的状态。 相当于编译器的 <code>-v</code> 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-v) | 在构建中显示编译器的执行状态。 |
| No | | 在构建中不显示编译器的执行状态。 | |
| 警告级别 | 选择在编译期间的警告显示级别。 相当于编译器的 <code>-w</code> 选项。 | | |
| | 默认值 | 常规输出 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | 没有输出 (<code>-w0</code>) | 不输出警告信息。 |
| | | 常规输出 | 输出常规警告信息。 |
| 详细输出 (<code>-w2</code>) | | 输出详细警告信息。 | |

(8) [Extension]

显示扩展的详细信息以及可修改的配置信息。

| | | |
|--------------|--|----------------|
| 允许 C++ 格式的注释 | 选择是否允许使用 C++ 格式的注释 (<code>"/**/</code>). 相当于编译器的 <code>-zp</code> 选项。 | |
| | 默认值 | Yes(-zp) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-zp) |
| No | | 不允许 C++ 格式的注释。 |
| 允许注释嵌套 | 选择是否允许使用嵌套注释 (<code>"/**/</code>). 相当于编译器的 <code>-zc</code> 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-zc) |
| No | | 不允许使用嵌套注释。 |

| | | | |
|------------------|---|----------------------|------------------------------------|
| 源文件的 Kanji 字符编码 | 选择源文件的 Kanji 字符编码 相当于编译器的 <code>-zs</code> , <code>-ze</code> , 和 <code>-zn</code> 选项。 | | |
| | 默认值 | Shift_JIS(-zs) | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Shift_JIS(-zs) | 将源文件的 kanji 代码用 Shift_JIS 来解释。 |
| | | EUC-JP(-ze) | 将源文件的 kanji 代码用 EUC-JP 来解释。 |
| Unspecified(-zn) | | 将源文件不包含 Kanji 代码在解释。 | |
| 符合 ANSI 标准 | 选择是否禁用非 ANSI 标准功能以及使用 ANSI 标准的功能。 相当于编译器的 <code>-za</code> 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-za) | 禁用非 ANSI 标准功能以及使用 ANSI 标准的功能。 |
| | | No | 允许非 ANSI 标准的功能。 |
| 禁用 int 扩展功能 | 选择是否禁止字符型和 unsigned 字符型参数和返回值的整型扩展。 相当于编译器的 <code>-zb</code> 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-zb) | 允许字符型和无符号字符型参数和返回值的 int 扩展。 |
| | | No | 禁止字符型和 unsigned 字符型参数和返回值的 int 扩展。 |

(9) [Memory Model]

显示详细的存储模式信息，此设置可以改变。

| | | | |
|--------------------------------|---|----------------------------------|----------------|
| 存储模式类型 | 指定存储模式类型。 相当于编译器的 <code>-m</code> 选项。 | | |
| | 默认值 | 中模式 (代码 1M 字节 / 数据 64K 字节)(-mm) | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | 小模式 (代码 64K 字节 / 数据 64K 字节)(-ms) | 指定存储模式为小模式。 |
| | | 中模式 (代码 1M 字节 / 数据 64K 字节)(-mm) | 指定存储模式为中模式。 |
| 大模式 (代码 1M 字节 / 数据 1M 字节)(-ml) | | 指定存储模式为大模式。 | |
| 输出 flash 的目标 | 选择是否输出 flash 的目标。 相当于编译器的 <code>-zf</code> 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-zf) | 输出 flash 的目标。 |
| | | No | 不输出 flash 的目标。 |

| | | |
|------------------|--|--|
| flash 区域的起始地址 | 使用不含 0x 的十六进制数来设定 flash 区域的起始地址。 相当于编译器的 -zz 选项。 | |
| | 默认值 | 空白 |
| | 如何改变 | 直接输入 |
| | 限制 | 0C0 到 0EDFFF (十六进制数) |
| flash 区域分支表的起始地址 | 使用不含 0x 的十六进制数来设定 flash 区域分支表的起始地址。 相当于编译器的 -zt 选项。 | |
| | 默认值 | No |
| | 如何改变 | 空白 |
| | 限制 | 直接输入 0C0 到 0EDFFF (十六进制数) |
| 设定镜像区域 | 设定源代码区域的镜像。 MAA 为处理器模式控制寄存器 (PMC) 的 0 位。 相当于编译器的 -mi 选项。 如果通过 [Link Options] 标签, 在 [Device] 分项中改变 [Specify mirror area] 属性的值, 则在这个属性上设置为相同数值。 | |
| | 默认值 | MAA=0(-mi0) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | MAA=0(-mi0) 假设设置 MAA=0。 MAA=1(-mi1) 假设设置 MAA=1。 |

(10)[Output File]

显示详细的输出文件信息, 此设置可以改变。

| | | |
|---------------|--|---|
| 输出各种设备的共同目标文件 | 选择是否输出各种设备的共同目标文件。 相当于编译程序的 -common 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-common) 输出共同目标文件到各种设备。 No 不设定输出共同目标文件到各种设备。 |

(11)[Assembly File]

显示详细的汇编文件信息, 此设置可以改变。

| | | |
|--------|---|---|
| 输出汇编文件 | 选择是否输出汇编文件。 相当于编译器的 -a、-sa 和 -li 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(With no C source info)(-a) 输出汇编文件 (包含 C 源代码信息)。 Yes(With C source info(unexpanded include file contents))(-sa) 输出汇编文件 (包含 C 源代码信息 (包含的文件内容没有被扩展))。 Yes(With C source info(expanded include file contents))(-sa,-li) 输出汇编文件 (包含 C 源代码信息 (扩展包含的文件内容))。 No 不输出汇编文件。 |

(12)[Variables/Functions Information File]

显示详细的变量 / 函数信息文件信息，此设置可改变。

| | | |
|------------------|---|--|
| 使用变量 / 函数信息文件 | 这是用于给变量和函数调用表区域分配 <code>saddr</code> 区域的变量 / 函数信息文件。 查询并显示在此工程中注册的有效变量 / 函数信息文件。 相当于编译器的 <code>-ma</code> 选项。 | |
| | 默认值 | <i>添加变量 / 函数信息文件名到工程中</i> |
| | 如何改变 | 不允许改变。 |
| 引导区域的变量 / 函数信息文件 | 指定用于引导区域工程的变量 / 函数信息文件。 相当于编译器的 <code>-ma</code> 选项。 如果指定的是相对路径，则参考路径为主工程或子工程的文件夹。 如果指定的是绝对路径，则参考路径为主工程或子工程的文件夹。 当 [Memory Model] 的 <code>[Output objects for flash]</code> 属性为 <code>[No]</code> 时，不显示该属性。 | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示 设定启动区变量 / 函数信息文件对话框 的来编辑。 |
| | 限制 | 多达 259 个字符 |

(13)[Data Control]

显示详细的数据控制信息，此设置可以改变。

| | | | |
|------------------|--|-----------|------------------------------------|
| 从 MSB 中分配区位给指令 | 选择是否从 MSB 中分配多个区位给指令。 相当于编译器的 -rb 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-rb) | 从 MSB 中分配多个区位给指令。 |
| | | No | 从 LSB 中分配多个区位给指令。 |
| 整合架构成员 | 选择是否禁止在结构中为成员（包含 2 个或更多字节）插入对齐数据来对其地址。 相当于编译器的 -rc 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-rc) | 禁止在结构中为成员（包含 2 个或更多字节）插入对齐数据来对其地址。 |
| | | No | 在结构中为成员（包含 2 个或更多字节）插入对齐数据来对其地址。 |
| 在 1 字节单元里进行间接引用。 | 选择是否在 1 字节单元里进行间接引用。 相当于编译器的 -ra 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-ra) | 在 1 字节单元里进行间接引用。 |
| | | No | 在 1 字节单元里不进行间接引用。 |

| | | | |
|--|---|---|---|
| 将静态变量分配到 <code>saddr</code> 区域 | 选择分配到 <code>saddr</code> 区域静态变量的类型。 相当于编译器的 <code>-rs</code> 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(Size of char)(-rs1) | 将字符型和 <code>unsigned</code> 字符型自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char, short, int)(-rs2) | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、 <code>near</code> 指针类型自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char, short, int, long)(-rs4) | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、长整型、 <code>unsigned</code> 长整型、指针类型自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Structure, union, array)(-rsm) | 将结构、联合和数组自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char and structure, union, array)(-rs1m) | 将字符型、 <code>unsigned</code> 字符型、联合和数组自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char, short, int and structure, union, array)(-rs2m) | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、 <code>near</code> 指针类型、结构体、联合体以及数组类型自动变量分配到 <code>saddr</code> 区域。 |
| Yes(Size of char, short, int, long and structure, union, array)(-rs) | | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、长整型、 <code>unsigned</code> 长整型、指针类型、及结构、联合和数组自动变量分配到 <code>saddr</code> 区域。 | |
| No | 不分配静态变量到 <code>saddr</code> 区域。 | | |

| | | | |
|--|---|---|---|
| 将外部变量分配到 saddr 区域 | 选择分配到 saddr 区域外部变量的类型。 相当于编译器的 -rd 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(Size of char)(-rd1) | 将字符型和 unsigned 字符型外部变量分配到 saddr 区域。 |
| | | Yes(Size of char, short, int)(-rd2) | 将字符型、 unsigned 字符型、短整型、 unsigned 短整型、整型、 unsigned 整型、枚举型、 near 指针类型外部变量分配到 saddr 区域。 |
| | | Yes(Size of char, short, int, long)(-rd4) | 将字符型、 unsigned 字符型、短整型、 unsigned 短整型、整型、 unsigned 整型、枚举型、长整型、 unsigned 长整型、指针类型外部变量分配到 saddr 区域。 |
| | | Yes(Structure, union, array)(-rdm) | 将结构、联合和数组外部变量分配到 saddr 区域。 |
| | | Yes(Size of char and structure, union, array)(-rd1m) | 将字符型、 unsigned 字符型、联合和数组外部变量分配到 saddr 区域。 |
| | | Yes(Size of char, short, int and structure, union, array)(-rd2m) | 将字符型、 unsigned 字符型、短整型、 unsigned 短整型、整型、 unsigned 整型、枚举型、 near 指针类型、结构体、联合体以及数组类型外部变量分配到 saddr 区域。 |
| Yes(Size of char, short, int, long and structure, union, array)(-rd) | | 将字符型、 unsigned 字符型、短整型、 unsigned 短整型、整型、 unsigned 整型、枚举型、长整型、 unsigned 长整型、指针类型、及结构、联合和数组外部变量分配到 saddr 区域。 | |
| No | 不分配外部变量到 saddr 区域。 | | |
| 指定 ROM 数据的分配目标 | 指定地方分配 ROM 数据。 相当于编译器的 -rf 和 -rm 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(far area)(-rf) | 分配 ROM 数据到 far 区域。 |
| | | Yes(near area)(-rm) | 分配 ROM 数据到 near 区域。 |
| No | 不指定 ROM 数据的分配目标。 | | |

(14)[List File]

显示详细的列表文件信息，此设置可改变。

| | | | |
|----------|---------------------------------------|-----------|------------|
| 输出预处理表文件 | 选择是否输出预处理文件。 相当于编译器的 -p 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-p) | 输出预处理表文件。 |
| | | No | 不输出预处理表文件。 |

| | | | | | |
|--|--|---|----------|--|----|
| 不输出注释 | 选择是否禁止将注释输出到预处理表文件中。 相当于编译器的 <code>-kc</code> 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | No | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kc)</td> <td>不将注释输出到预处理表文件。</td> </tr> <tr> <td>No</td> <td>将注释输出到预处理表文件。</td> </tr> </tbody> </table> | Yes(-kc) | 不将注释输出到预处理表文件。 | No |
| Yes(-kc) | 不将注释输出到预处理表文件。 | | | | |
| No | 将注释输出到预处理表文件。 | | | | |
| 扩展 <code>#define</code> 预处理指令 | 选择是否扩展 <code>#define</code> 指令到预处理表文件中。 相当于编译器的 <code>-kd</code> 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | No | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kd)</td> <td>扩展 <code>#define</code> 指令到预处理表文件中。</td> </tr> <tr> <td>No</td> <td>不扩展 <code>#define</code> 指令到预处理表文件中。</td> </tr> </tbody> </table> | Yes(-kd) | 扩展 <code>#define</code> 指令到预处理表文件中。 | No |
| Yes(-kd) | 扩展 <code>#define</code> 指令到预处理表文件中。 | | | | |
| No | 不扩展 <code>#define</code> 指令到预处理表文件中。 | | | | |
| 扩展 <code>#if</code> 、 <code>#ifdef</code> 、 <code>#ifndef</code> 预处理指令 | 选择是否扩展 <code>#if</code> 、 <code>#ifdef</code> 、 <code>#ifndef</code> 指令到预处理表文件中。 相当于编译器的 <code>-kf</code> 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | Yes(-kf) | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kf)</td> <td>执行扩展输出 <code>#if</code>、<code>#ifdef</code>、<code>#ifndef</code> 指令到预处理表文件中。</td> </tr> <tr> <td>No</td> <td>不执行扩展输出 <code>#if</code>、<code>#ifdef</code>、<code>#ifndef</code> 指令到预处理表文件中。</td> </tr> </tbody> </table> | Yes(-kf) | 执行扩展输出 <code>#if</code> 、 <code>#ifdef</code> 、 <code>#ifndef</code> 指令到预处理表文件中。 | No |
| Yes(-kf) | 执行扩展输出 <code>#if</code> 、 <code>#ifdef</code> 、 <code>#ifndef</code> 指令到预处理表文件中。 | | | | |
| No | 不执行扩展输出 <code>#if</code> 、 <code>#ifdef</code> 、 <code>#ifndef</code> 指令到预处理表文件中。 | | | | |
| 扩展 <code>#include</code> 预处理指令 | 选择是否扩展 <code>#include</code> 指令到预处理表文件中。 相当于编译器的 <code>-ki</code> 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | No | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-ki)</td> <td>扩展 <code>#include</code> 指令到预处理表文件中。</td> </tr> <tr> <td>No</td> <td>不扩展 <code>#include</code> 指令到预处理表文件中。</td> </tr> </tbody> </table> | Yes(-ki) | 扩展 <code>#include</code> 指令到预处理表文件中。 | No |
| Yes(-ki) | 扩展 <code>#include</code> 指令到预处理表文件中。 | | | | |
| No | 不扩展 <code>#include</code> 指令到预处理表文件中。 | | | | |

| | | | | | | | |
|---------------------------|---|--|---------------------------|----------------------|-------------------------|-----------------------|----|
| 扩展 #line 预处理指令 | 选择是否扩展 #line 指令到预处理表文件中。 相当于编译器的 -kl 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | | | |
| | 默认值 | Yes(-kl) | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kl)</td> <td>扩展 #line 指令到预处理表文件中。</td> </tr> <tr> <td>No</td> <td>不扩展 #line 指令到预处理表文件中。</td> </tr> </tbody> </table> | Yes(-kl) | 扩展 #line 指令到预处理表文件中。 | No | 不扩展 #line 指令到预处理表文件中。 | |
| Yes(-kl) | 扩展 #line 指令到预处理表文件中。 | | | | | | |
| No | 不扩展 #line 指令到预处理表文件中。 | | | | | | |
| 输出行号 | 选择是否输出行号到预处理表文件中。 相当于编译器的 -kn 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | | | |
| | 默认值 | Yes(-kn) | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kn)</td> <td>将行号输出到预处理表文件。</td> </tr> <tr> <td>No</td> <td>不将行号输出到预处理表文件。</td> </tr> </tbody> </table> | Yes(-kn) | 将行号输出到预处理表文件。 | No | 不将行号输出到预处理表文件。 | |
| Yes(-kn) | 将行号输出到预处理表文件。 | | | | | | |
| No | 不将行号输出到预处理表文件。 | | | | | | |
| 输出错误列表文件 | 选择是否输出错误列表文件。 相当于编译器的 -e 和 -se 选项。 | | | | | | |
| | 默认值 | No | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(Without C source)(-e)</td> <td>输出错误列表文件（不包含 C 源代码）。</td> </tr> <tr> <td>Yes(With C source)(-se)</td> <td>输出错误列表文件（包含 C 源代码）。</td> </tr> <tr> <td>No</td> <td>不输出错误列表文件。</td> </tr> </tbody> </table> | Yes(Without C source)(-e) | 输出错误列表文件（不包含 C 源代码）。 | Yes(With C source)(-se) | 输出错误列表文件（包含 C 源代码）。 | No |
| Yes(Without C source)(-e) | 输出错误列表文件（不包含 C 源代码）。 | | | | | | |
| Yes(With C source)(-se) | 输出错误列表文件（包含 C 源代码）。 | | | | | | |
| No | 不输出错误列表文件。 | | | | | | |
| 输出交叉引用列表文件 | 选择是否输出交叉引用列表文件。 相当于编译器的 -x 选项。 | | | | | | |
| | 默认值 | No | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-x)</td> <td>输出交叉引用列表文件</td> </tr> <tr> <td>No</td> <td>不输出交叉引用列表文件。</td> </tr> </tbody> </table> | Yes(-x) | 输出交叉引用列表文件 | No | 不输出交叉引用列表文件。 | |
| Yes(-x) | 输出交叉引用列表文件 | | | | | | |
| No | 不输出交叉引用列表文件。 | | | | | | |
| 带格式馈送控制代码输出 | 选择是否将格式馈送控制代码输出到表文件（预处理表文件，错误表文件和交叉引用列表文件）。 相当于编译器的 -lf 选项。 仅当 [Output error list file] 属性选择为 [Yes] 或 [Output preprocess list file] 属性选择为 [Yes(-p)] 或 [Output cross reference list file] 属性选择为 [Yes(-x)] 时，显示该属性。 | | | | | | |
| | 默认值 | No | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-lf)</td> <td>输出格式馈送代码到表文件。</td> </tr> <tr> <td>No</td> <td>不输出格式馈送代码到表文件。</td> </tr> </tbody> </table> | Yes(-lf) | 输出格式馈送代码到表文件。 | No | 不输出格式馈送代码到表文件。 | |
| Yes(-lf) | 输出格式馈送代码到表文件。 | | | | | | |
| No | 不输出格式馈送代码到表文件。 | | | | | | |

| | | |
|---------|---|-----------------------|
| 1 行字符数目 | 指定表文件每行字符数（预处理表文件，错误表文件和交叉引用列表文件）。 相当于编译器的 <code>-lw</code> 选项。 仅当 [Output error list file] 属性选择为 [Yes] 或 [Output preprocess list file] 属性选择为 [Yes(-p)] 或 [Output cross reference list file] 属性选择为 [Yes(-x)] 时，显示该属性。 | |
| | 默认值 | 132 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 72 ~ 132（十进制数） |
| 1 页的行数 | 指定表文件每页行数（预处理表文件，错误表文件和交叉引用列表文件）。 如果是 0，将不会生成分页符。 相当于编译器的 <code>-ll</code> 选项。 仅当 [Output error list file] 属性选择为 [Yes] 或 [Output preprocess list file] 属性选择为 [Yes(-p)] 或 [Output cross reference list file] 属性选择为 [Yes(-x)] 时，显示该属性。 | |
| | 默认值 | 0 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0, 和 20 ~ 65535（十进制数） |
| Tab 宽度 | 指定表文件的 Tab 宽度（预处理表文件，错误表文件和交叉引用列表文件）。 相当于编译器的 <code>-lt</code> 选项。 仅当 [Output error list file] 属性选择为 [Yes] 或 [Output preprocess list file] 属性选择为 [Yes(-p)] 或 [Output cross reference list file] 属性选择为 [Yes(-x)] 时，显示该属性。 | |
| | 默认值 | 8 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0 ~ 8（十进制数） |

(15)[Others]

显示编译的其它详细信息以及可修改的配置信息。

| | | |
|-----------|--|---|
| 编译之前执行的命令 | 指定编译之前要执行的命令。 下面的宏变量是内嵌的宏。 %ProjectFolder% : 用工程文件夹的绝对路径替换。 %OutputFolder% : 用输出文件夹的绝对路径替换。 %OutputFile% : 用输出文件的绝对路径替换。 %InputFile% : 用将要编译文件的绝对路径替换。 %CompiledFile% : 用正在编译输出文件的绝对路径替换。 指定的命令显示为子属性。 | |
| | 默认值 | 编译之前执行的命令 [定义条目的数目] |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 1023 个字符 设定最多可达 64 个条目。 |

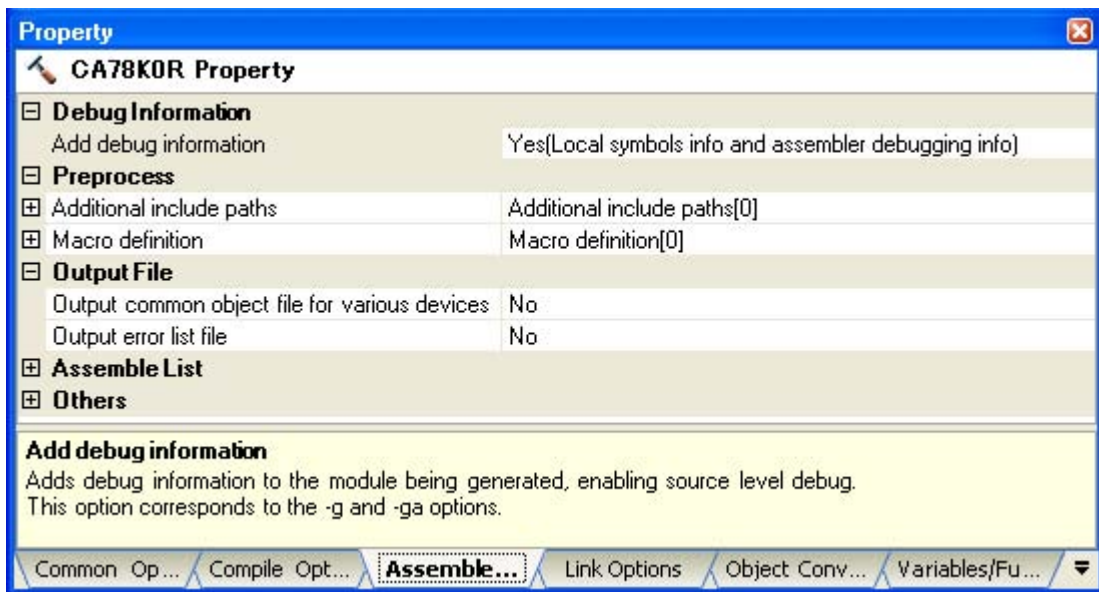
| | | |
|-----------|--|---|
| 编译之后执行的命令 | <p>指定编译之后要执行的命令。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%ProjectFolder%: 用工程文件夹的绝对路径替换。</p> <p>%OutputFolder%: 用输出文件夹的绝对路径替换。</p> <p>%OutputFile%: 用输出文件的绝对路径替换。</p> <p>%InputFile%: 用将要编译文件的绝对路径替换。</p> <p>%CompiledFile%: 用正在编译输出文件的绝对路径替换。</p> <p>指定的命令显示为子属性。</p> | |
| | 默认值 | 编译之后执行的命令 [<i>定义条目的数目</i>] |
| | 如何改变 | <p>点击 [...] 按钮，通过 文本编辑对话框 来编辑。</p> <p>对于子属性，可以用文本框直接输入。</p> |
| | 限制 | <p>多达 1023 个字符</p> <p>设定最多可达 64 个条目。</p> |
| 其他附加选项 | <p>输出需要加入的附加编译选项。</p> <p>设置的选项加在编译选项组的最后。</p> | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示 字符串输入对话框 的来编辑。 |
| | 限制 | 多达 259 个字符 |

[Assemble Options] 标签

此标签显示了按如下分类的汇编程序和可更改设置的详细信息。

- (1) [Debug Information]
- (2) [Preprocess]
- (3) [Output File]
- (4) [Assemble List]
- (5) [Others]

图 A-6. 属性面板 : [Assemble Options] 标签



[各分项说明]

(1) [Debug Information]

显示详细的调试信息，设置可以更改。

| | | |
|--------|--|-----------------------------------|
| 加入调试信息 | 选择是否在将生成的模块中加入调试信息来允许源文件级调试。 相当于编译器的 <code>-g</code> 和 <code>-ga</code> 选项。 | |
| 默认值 | Yes(Local symbols info and assembler debugging info) | |
| 如何改变 | 通过下拉列表选择。 | |
| 限制 | Yes(Assembler debugging info)(<code>-ng,-ga</code>) | 在将要产生的目标模块文件中加入调试信息（汇编调试符号信息）。 |
| | Yes(Local symbols info and assembler debugging info) | 在将要产生的目标模块文件中加入调试信息（局部符号和汇编调试信息）。 |
| | No(<code>-ng,-nga</code>) | 在将要产生的目标模块文件中加入调试信息。 |

(2) [Preprocess]

显示详细的预处理信息，设置可以改变。

| | | |
|--------------------------|--|--|
| Additional include paths | 在汇编时指定附加的 include 路径。 下面的宏变量是内嵌的宏。 <code>%BuildModeName%</code> : 用生成模式名字替换 <code>%ProjectName%</code> : 用工程名称替换。 <code>%CubeSuitePath%</code> : 用 CubeSuite 的绝对安装路径替换。 当忽略该选项时，仅查询汇编程序标准文件夹。路径的参考点是工程文件夹。 相当于汇编程序的 <code>-i</code> 选项。 指定的 include 路径显示为子属性。 | |
| 默认值 | Additional include paths[<i>number of defined items</i>] | |
| 如何改变 | 点击 [...] 按钮，通过 路径编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 | |
| 限制 | 多达 259 个字符 设定最多可达 64 个条目。尽管这也包含链接工具使用的路径。 | |
| 宏定义 | 指定需要定义的宏名称。 通过 "宏名称 = 定义值" 这种格式来指定，每行一个宏。" = 定义值" 部分可以省略，此时默认值为 1。 相当于汇编程序的 <code>-d</code> 选项。 指定的宏显示为子属性。 | |
| 默认值 | 宏定义 [<i>定义条目的数目</i>] | |
| 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 | |
| 限制 | 多达 256 个字符 设定最多可达 30 个条目。 | |

(3) [Output File]

显示详细的输出文件信息，设置可以改变。

| | | |
|---------------|---|----------------|
| 输出各种设备的共同目标文件 | 选择是否输出各种设备的共同目标文件。 相当于汇编程序的 <code>-common</code> 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-common) |
| No | | 输出 78K0 的目标文件。 |
| 输出错误列表文件 | 选择是否输出错误列表文件。 相当于汇编程序的 <code>-e</code> 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-e) |
| No | | 不输出错误列表文件。 |

(4) [Assemble List]

显示详细的汇编列表信息，此设置可改变。

| | | |
|--------------|---|---------------------|
| 输出汇编列表文件 | 选择是否输出汇编列表文件。 相当于汇编程序的 <code>-p</code> 选项。 | |
| | 默认值 | Yes(-p) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-p) |
| No(-np) | | 不输出汇编列表文件。 |
| 执行列表转换器 | 选择是否在可执行模块生成后执行列表转换。 列表转换在库生成期间不会被执行。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | Yes |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| No | | 不在可执行模块生成后执行列表转换。 |
| 输出列表转换错误列表文件 | 选择是否在列表转换执行期间输出错误列表文件。 相当于列表转换程序的 <code>-e</code> 选项。 当 [Output assemble list file] 属性为 [No]，且 [Execute list converter] 的属性为 [No] 时，不显示该属性。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-e) |
| No | | 在列表转换执行期间不输出错误列表文件。 |

| | | | | | |
|-------------|---|---|----------|--------------------|----------|
| 汇编列表信息输出 | 选择是否输出汇编列表信息到汇编表文件。 相当于汇编程序的 <code>-ka</code> 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | Yes | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes</td> <td>输出汇编列表信息到汇编表文件。</td> </tr> <tr> <td>No(-nka)</td> <td>不输出汇编列表信息到汇编表文件。</td> </tr> </tbody> </table> | Yes | 输出汇编列表信息到汇编表文件。 | No(-nka) |
| Yes | 输出汇编列表信息到汇编表文件。 | | | | |
| No(-nka) | 不输出汇编列表信息到汇编表文件。 | | | | |
| 符号列表输出 | 选择是否输出符号列表信息到汇编列表文件。 相当于汇编程序的 <code>-ks</code> 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | No | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-ks)</td> <td>输出符号列表信息到汇编列表文件。</td> </tr> <tr> <td>No</td> <td>不输出符号列表信息到汇编列表文件。</td> </tr> </tbody> </table> | Yes(-ks) | 输出符号列表信息到汇编列表文件。 | No |
| Yes(-ks) | 输出符号列表信息到汇编列表文件。 | | | | |
| No | 不输出符号列表信息到汇编列表文件。 | | | | |
| 交叉引用列表输出 | 选择是否输出交叉引用列表信息到汇编列表文件。 相当于汇编程序的 <code>-kx</code> 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | No | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kx)</td> <td>输出交叉引用列表信息到汇编列表文件。</td> </tr> <tr> <td>No</td> <td>不输出交叉引用列表信息到汇编列表文件。</td> </tr> </tbody> </table> | Yes(-kx) | 输出交叉引用列表信息到汇编列表文件。 | No |
| Yes(-kx) | 输出交叉引用列表信息到汇编列表文件。 | | | | |
| No | 不输出交叉引用列表信息到汇编列表文件。 | | | | |
| 带格式馈送控制代码输出 | 选择是否输出格式馈送代码到表文件。 相当于汇编程序的 <code>-lf</code> 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | No | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-lf)</td> <td>输出格式馈送代码到表文件。</td> </tr> <tr> <td>No</td> <td>不输出格式馈送代码到表文件。</td> </tr> </tbody> </table> | Yes(-lf) | 输出格式馈送代码到表文件。 | No |
| Yes(-lf) | 输出格式馈送代码到表文件。 | | | | |
| No | 不输出格式馈送代码到表文件。 | | | | |
| 1 行字符数目 | 指定表文件每行字符数。 相当于汇编程序的 <code>-lw</code> 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | 132 | | | |
| | 如何改变 | 直接在文本框中输入。 | | | |
| | 限制 | 72 ~ 2046 (十进制数) | | | |

| | | |
|--------|--|-------------------------|
| 1 页的行数 | 指定表文件每页行数。 如果是 0，将不会生成分页符。 相当于汇编程序的 -ll 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 0 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0，和 20 ~ 32767（十进制数） |
| Tab 宽度 | 指定列表文件的 tab 宽度 相当于汇编程序的 -lt 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 8 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0 ~ 8（十进制数） |
| 头标题 | 指定汇编表文件的页头标题。 可以指定包含有双字节和单字节空格字符的字符串。 相当于汇编程序的 -lh 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 多达 60 个单字节字符（30 个双字节字符） |

(5) [Others]

显示汇编的其它详细信息以及可修改的配置信息。

| | | |
|------------------|---|----------------------------------|
| 源文件的 Kanji 字符编码 | 选择源文件的 Kanji 字符编码 相当于汇编程序的 -zs、-ze 和 -zn 选项。 | |
| | 默认值 | Shift_JIS(-zs) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Shift_JIS(-zs) |
| EUC-JP(-ze) | | 将源文件的 kanji 代码用 EUC-JP 来解释。 |
| Unspecified(-zn) | | 将源文件不包含 Kanji 代码在解释。 |
| 允许汇编 78K0 的宏 | 选择是否使由 78K0 单片机汇编程序生成的汇编程序源文件汇编可用。 相当于汇编程序的 -compati 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-compati) |
| No | | 使对 78K0 微控制器汇编程序生成的汇编程序源文件汇编不可用。 |

| | | |
|-----------|---|---|
| 汇编之前执行的命令 | <p>指定汇编之前要执行的命令。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%ProjectFolder%: 用工程文件夹的绝对路径替换。</p> <p>%OutputFolder%: 用输出文件夹的绝对路径替换。</p> <p>%OutputFile%: 用输出文件的绝对路径替换。</p> <p>%InputFile%: 用将要汇编文件的绝对路径替换。</p> <p>%AssembledFile%: 用正在汇编输出文件的绝对路径替换。</p> <p>指定的命令显示为子属性。</p> | |
| | 默认值 | 汇编之前执行的命令 [定义条目的数目] |
| | 如何改变 | <p>点击 [...] 按钮，通过 文本编辑对话框 来编辑。</p> <p>对于子属性，可以用文本框直接输入。</p> |
| | 限制 | <p>多达 1023 个字符</p> <p>设定最多可达 64 个条目。</p> |
| 汇编之后执行的命令 | <p>指定汇编之后要执行的命令。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%ProjectFolder%: 用工程文件夹的绝对路径替换。</p> <p>%OutputFolder%: 用输出文件夹的绝对路径替换。</p> <p>%OutputFile%: 用输出文件的绝对路径替换。</p> <p>%InputFile%: 用将要汇编文件的绝对路径替换。</p> <p>%AssembledFile%: 用正在汇编输出文件的绝对路径替换。</p> <p>指定的命令显示为子属性。</p> | |
| | 默认值 | 汇编之后执行的命令 [定义条目的数目] |
| | 如何改变 | <p>点击 [...] 按钮，通过 文本编辑对话框 来编辑。</p> <p>对于子属性，可以用文本框直接输入。</p> |
| | 限制 | <p>多达 1023 个字符</p> <p>设定最多可达 64 个条目。</p> |
| 其他附加选项 | <p>输出需要加入的附加汇编选项。</p> <p>这里设置的选项加在汇编选项组的最后。</p> | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示 字符串输入对话框 的来编辑。 |
| | 限制 | 多达 259 个字符 |

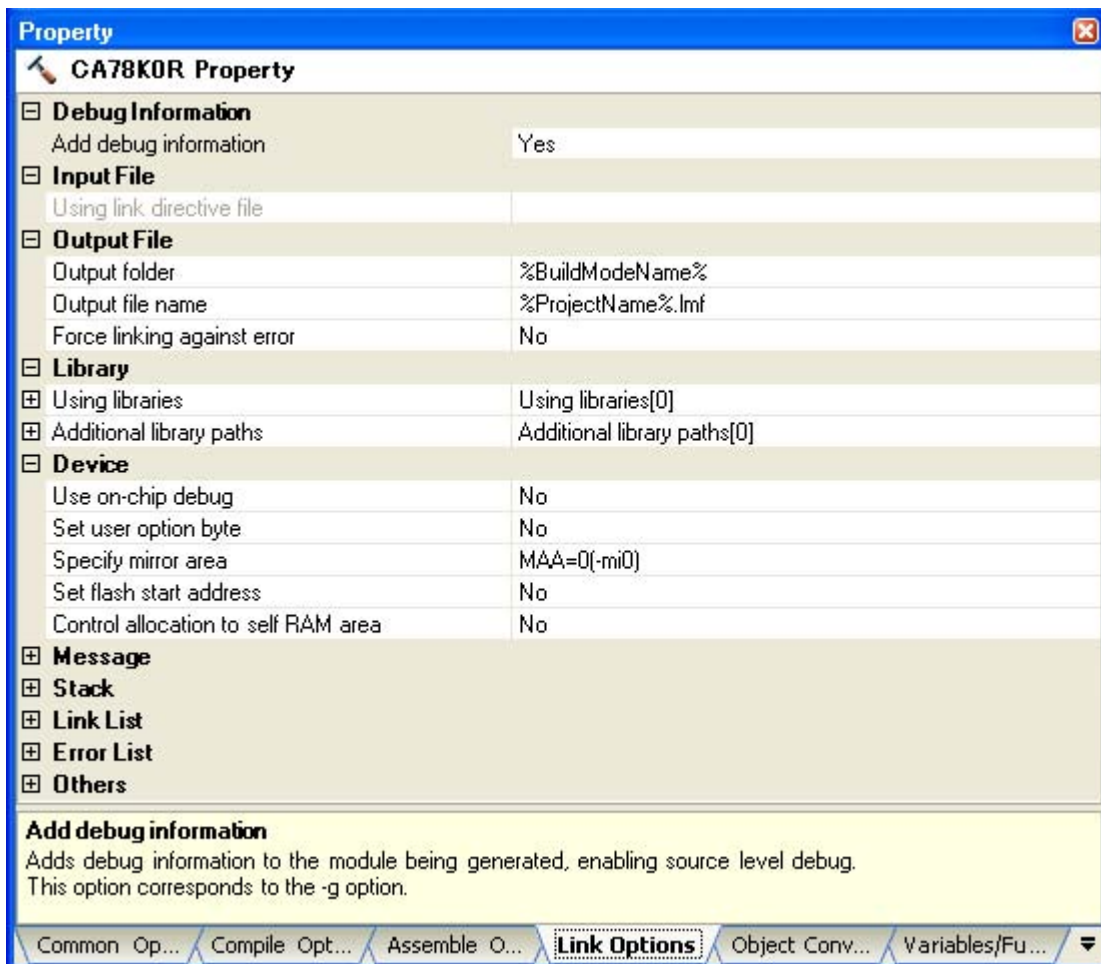
[Link Options] 标签

此标签显示了按如下分类的链接程序和可更改设置的详细信息。

- (1) [Debug Information]
- (2) [Input File]
- (3) [Output File]
- (4) [Library]
- (5) [Device]
- (6) [Message]
- (7) [Stack]
- (8) [Link List]
- (9) [Error List]
- (10) [Others]

注意事项 对库工程此标签不会显示。

图 A-7. 属性面板 : [Link Options] 标签



[各分项说明]

(1) [Debug Information]

显示详细的调试信息，设置可以更改。

| | | |
|---------|---|----------------------|
| 加入调试信息 | 选择是否在将生成的模块中加入调试信息来允许源文件级调试。 相当于连接器的 -g 选项。 | |
| | 默认值 | Yes |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| 不 (-ng) | | 在将要产生的目标模块文件中加入调试信息。 |

(2) [Input File]

显示详细的输入文件信息，此设置可改变。

| | | |
|-----------|--|---------------|
| 使用链接指令文件。 | 显示用作链接的链接指令文件。 对应于链接程序的 -d 选项。 | |
| | 默认值 | 添加链接指令文件名到工程中 |
| | 如何改变 | 不允许改变。 |

(3) [Output File]

显示详细的输出文件信息，设置可以改变。

| | | |
|-------|--|---|
| 输出文件夹 | 指定保存生成模块的文件夹。 如果指定的是相对路径，则参考路径为主工程或子工程的文件夹。 如果指定的是绝对路径，则参考路径为主工程或子工程的文件夹。 下面的宏变量是内嵌的宏。 %BuildModeName% : 用生成模式名字替换 如果为空，则作为已设置工程文件夹。 | |
| | 默认值 | %BuildModeName% |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示文件夹浏览对话框的来编辑。 |
| | 限制 | 多达 247 个字符 |
| 输出文件名 | 指定输出的加载模块文件名。 使用 ".lbf" 扩展名。如果扩展名省略，会自动加上 ".lbf"。 相当于链接程序的 -o 选项。 下面的宏变量是内嵌的宏。 %ProjectName% : 用工程名称替换。 如果为空，默认为 "%ProjectName%.lbf"。 | |
| | 默认值 | %ProjectName%.lbf |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 多达 259 个字符 |

| | | | | | |
|----------|--|---|---------|-------------------|----|
| 出错时强制链接。 | 选择是否在链接出错时强制生成加载模块文件。 相当于链接程序的 -j 选项。 | | | | |
| | 默认值 | No | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tr> <td>Yes(-j)</td> <td>在链接出错时强制生成加载模块文件。</td> </tr> <tr> <td>No</td> <td>在链接出错时不生成加载模块文件。</td> </tr> </table> | Yes(-j) | 在链接出错时强制生成加载模块文件。 | No |
| Yes(-j) | 在链接出错时强制生成加载模块文件。 | | | | |
| No | 在链接出错时不生成加载模块文件。 | | | | |

(4) [Library]

显示库的详细信息以及可修改的配置信息。

| | | |
|---------|---|--|
| 使用库 | 指定使用的除标准库以外的库文件名 (*.lib)。 每行添加一个文件。 从库文件路径查找库文件。 相当于链接器的 -b 选项。 指定的库文件名作为子属性来显示。 | |
| | 默认值 | 使用库 [定义条目的数目] |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 259 个字符 设定最多可达 64 个条目。 |
| 附加库文件路径 | 指定除标准库以外的查找路径。 下面的宏变量是内嵌的宏。 %BuildModeName%: 用生成模式名字替换 %ProjectName%: 用工程名称替换。 %CubeSuitePath%: 用 CubeSuite 的绝对安装路径替换。 库文件从库路径中查找。如果指定的是相对路径，参考路径是工程文件夹。 相当于链接程序的 -i 选项。 指定的库路径作为子属性来显示。 | |
| | 默认值 | 附加的库路径 [定义条目的数目] |
| | 如何改变 | 点击 [...] 按钮，通过 路径编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 259 个字符 设定最多可达 64 个条目。 |

(5) [Device]

显示设备的详细信息以及可修改的配置信息。

| | | |
|----------------|--|-----------------------------------|
| 使用片上调试 | 选择是否使用片上调试。 更改片上调试的控制数值，调试镜像区域的起始地址和调试镜像区域的大小。 相当于链接器的 <code>-go</code> 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-go) 设置片上调试。 No 不设置片上调试。 |
| OCD 选项设置值 | 用不含 0x 的十六进制数设定片上调试选项控制值。 相当于链接器的 <code>-go</code> 选项。 当 [Use on-chip debug] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0 ~ FF (十六进制数) |
| 调试镜像区域的起始地址 | 使用不含 0x 的十六进制数来设定调试镜像区域的起始地址。 相当于链接器的 <code>-go</code> 选项。 如果这是空的，假设已经设定 "内部 ROM 结束地址 - 1024 + 1"。 当 [Use on-chip debug] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 内部 ROM 结束地址 - 1024 + 1 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0 ~ FFFFF (十六进制数) |
| 调试监控区大小 [byte] | 用十进制数指定调试监控区的大小。 相当于链接器的 <code>-go</code> 选项。 如果为空，默认值为 "1024"。 当 [Use on-chip debug] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 1024 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 88 ~ 1024 (十进制数) |
| 设置用户选项字节 | 选择是否设置用户选项字节。 相当于链接器的 <code>-gb</code> 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-gb) 设置用户选项字节 No 不设置用户选项字节 |
| 用户选项字节值 | 用不带 0x 的十六进制数来设定用户选项字节值。 相当于链接器的 <code>-gb</code> 选项。 当 [Set user option byte] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0 ~ FFFFFFF (十六进制数) |

| | | |
|-----------------------|---|--|
| 设定镜像区域 | 在 RAM 空间中选择区域分配给做过镜像的段。 相当于链接器的 <code>-mi</code> 选项。 如果通过 [Compile Options] 标签, 在 [Output File] 分项中改变 [Specify mirror area] 属性的值, 则在这个属性上设置为相同数值。 | |
| | 默认值 | MAA=0(-mi0) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | MAA=0(-mi0) |
| MAA=1(-mi1) | | 当 MAA=1 时, 分配段给区域。 |
| 分配段给 64KB 边界区域的最后一个字节 | 选择是否分配段给 64KB 边界区域的最后一个字节。 相当于链接器的 <code>-ccza</code> 选项。 通过 [Compile Options] 标签, 在 [Extension] 分项中, 当 [Follow ANSI Standard] 属性为 [No] , 则此属性不会显示。 | |
| | 默认值 | - 当添加 C 源文件到工程中时 [No] - 当只有汇编源文件作为源代码文件添加到工程中时。 [Yes(-ccza)] - 立即在创建工程之后 [Yes(-ccza)] |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-ccza) |
| No | | 不分配段给每个 64KB 边界区域的最后一个字节。 |
| 设置 flash 开始地址 | 选择是否为内置 flash ROM 产品设置 flash 起始地址。 相当于链接程序的 <code>-zb</code> 选项。 对不包含 flash ROM 自编程功能的不设置该属性。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-zb) |
| No | | 不为内置的 flash ROM 设置 flash 开始地址。 |
| Flash 开始地址 | 显示与 [Compile Options] 标签的 [Memory Model] 分项上 [Start address of flash area] 属性的数值相同的值。 相当于链接程序的 <code>-zb</code> 选项。 当 [Set flash start address] 属性为 [No] 时, 不显示该属性。 | |
| | 默认值 | 空白 |
| | 如何改变 | 不允许改变。 |

| | | | |
|----------------|---|---|----------------------|
| 引导区加载模块文件名 | 指定引导区加载模块文件名。 相当于链接程序的 <code>-zf</code> 选项。 如果这项为空，则会产生连接错误。确保指定引导区加载模块文件名。 如果指定的是相对路径，则参考路径为主工程或子工程的文件夹。 如果指定的是绝对路径，则参考路径为主工程或子工程的文件夹。 This property is displayed 仅当 [Set flash start address] 的属性选择为 [Yes(-zb)]，或当 C 源文件加到工程中且将 [Memory Model]，来自 [Compile Options] 标签，的 [Output objects for flash] 的属性选择为 [Yes(-zf)]。 | | |
| | 默认值 | 空白 | |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示 设定引导区载入模块文件对话框 的来编辑。 | |
| | 限制 | 多达 259 个字符 | |
| 控制自身 RAM 区域的分配 | 选择是否控制自有 RAM 空间的分配。 相当于链接器的 <code>-self</code> 和 <code>-selfw</code> 选项。 对不包含 flash ROM 自编程功能的不设置该属性。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(Error message)(-self) | 禁止自有 RAM 区域的分配并输出错误。 |
| | | Yes(Warning message)(-selfw) | 当分配自身 RAM 区域时，输出警告。 |
| No | | 把自身 RAM 区域作为内部 RAM 区域使用。 | |

(6) [Message]

显示消息的详细信息以及可修改的配置信息。

| | | |
|---------------------------|--|---------------------------|
| 警告级别 | 选择在连接期间的警告显示级别 相当于链接程序的 <code>-w</code> 选项。 | |
| | 默认值 | 常规输出 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | 没有输出 (<code>-w0</code>) |
| 常规输出 | | 输出常规警告信息。 |
| 详细输出 (<code>-w2</code>) | | 输出详细警告信息。 |

(7) [Stack]

显示堆栈的详细信息以及可修改的配置信息。

| | | |
|----------|--|---|
| 生成堆栈分解符号 | 选择是否生成堆栈分解符号。 相当于链接器的 -s 选项。 如果 [Startup]，来自 [Compile Options] 标签，的属性 [Use standard startup routine] 为 [Yes]，则该属性会被设置为 [Yes(-s)] 且不能改变。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-s) 生成堆栈分解符号。 No 不生成堆栈分解符号。 |
| 区域名称 | 指定生成堆栈分解符号的内存区名。 如果省略了该区域名，则假定值为“RAM”。 相当于链接器的 -s 选项。 当 [Generate stack solution symbol] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示字符串输入对话框的来编辑。 |
| | 限制 | 多达 256 个字符 |

(8) [Link List]

显示详细的链接列表信息，此设置可改变。

| | | |
|----------|--|--|
| 输出链接列表文件 | 选择是否输出链接列表文件。 相当于链接器的 -p 选项。 | |
| | 默认值 | Yes |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes 输出链接列表文件。 No(-np) 不输出链接列表文件。 |
| 链接指令信息输出 | 选择是否将链接指令信息输出至链接列表文件。 相当于链接程序的 -kd 选项。 当 [Output link list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | Yes |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes 将链接指令信息输出至链接列表文件。 No(-nkd) 不将链接指令信息输出至链接列表文件。 |
| 带局部符号表输出 | 选择是否输出符号表信息到链接列表文件。 相当于链接程序的 -kl 选项。 当 [Output link list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-kl) 输出局部符号表信息到链接列表文件。 No 不输出局部符号表信息到链接列表文件。 |

| | | |
|-------------|--|--|
| 公共符号表输出 | 选择是否输出公共符号表信息到链接列表文件。 相当于链接器的 <code>-kp</code> 选项。 当 [Output link list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-kp) 输出公共符号表信息到链接列表文件。 No 不输公共部符号表信息到链接列表文件。 |
| map 列表输出 | 选择是否输出 map 列表信息到链接列表文件。 相当于链接器的 <code>-km</code> 选项。 当 [Output link list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | Yes |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes 输出 map 列表信息到链接列表文件。 No(-nkm) 不输出 map 列表信息到链接列表文件。 |
| 带格式馈送控制代码输出 | 选择是否输出格式馈送代码到表文件。 相当于链接程序的 <code>-lf</code> 选项。 当 [Output link list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-lf) 输出格式馈送代码到表文件。 No 不输出格式馈送代码到表文件。 |
| 1 页的行数 | 指定表文件每页行数。 如果是 0，将不会生成分页符。 相当于链接程序的 <code>-ll</code> 选项。 当 [Output link list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 0 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0, 和 20 ~ 32767 (十进制数) |

(9) [Error List]

显示详细的错误表信息，此设置可改变。

| | | |
|----------|--|--|
| 输出错误列表文件 | 选择是否输出错误列表文件。 相当于链接器的 <code>-e</code> 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-e) 输出错误列表文件 No 不输出错误列表文件。 |

(10)[Others]

显示链接的其它详细信息以及可修改的配置信息。

| | | |
|-------------|---|---|
| 进行链接之前执行的命令 | <p>设定链接之前要执行的命令。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%ProjectFolder%: 用工程文件夹的绝对路径替换。</p> <p>%OutputFolder%: 用输出文件夹的绝对路径替换。</p> <p>%OutputFile%: 用输出文件的绝对路径替换。</p> <p>%LinkedFile%: 用正在连接输出文件的绝对路径替换。</p> <p>指定的命令显示为子属性。</p> | |
| | 默认值 | 链接之前执行的命令 [定义条目的数目] |
| | 如何改变 | <p>点击 [...] 按钮，通过文本编辑对话框来编辑。</p> <p>对于子属性，可以用文本框直接输入。</p> |
| | 限制 | <p>多达 1023 个字符</p> <p>设定最多可达 64 个条目。</p> |
| 链接之后执行的命令 | <p>指定链接之后要执行的命令。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%ProjectFolder%: 用工程文件夹的绝对路径替换。</p> <p>%OutputFolder%: 用输出文件夹的绝对路径替换。</p> <p>%OutputFile%: 用输出文件的绝对路径替换。</p> <p>%LinkedFile%: 用正在连接输出文件的绝对路径替换。</p> <p>指定的命令显示为子属性。</p> | |
| | 默认值 | 链接之后执行的命令 [<i>number of defined items</i>] |
| | 如何改变 | <p>点击 [...] 按钮，通过文本编辑对话框来编辑。</p> <p>对于子属性，可以用文本框直接输入。</p> |
| | 限制 | <p>多达 1023 个字符</p> <p>设定最多可达 64 个条目。</p> |
| 其他附加选项 | <p>输入需要加入的附加链接选项。</p> <p>这里设置的选项加在链接选项组的最后。</p> | |
| | 默认值 | 空白 |
| | 如何改变 | <p>直接在文本框中输入或通过点击 [...] 按钮而显示字符串输入对话框的来编辑。</p> |
| | 限制 | 多达 259 个字符 |

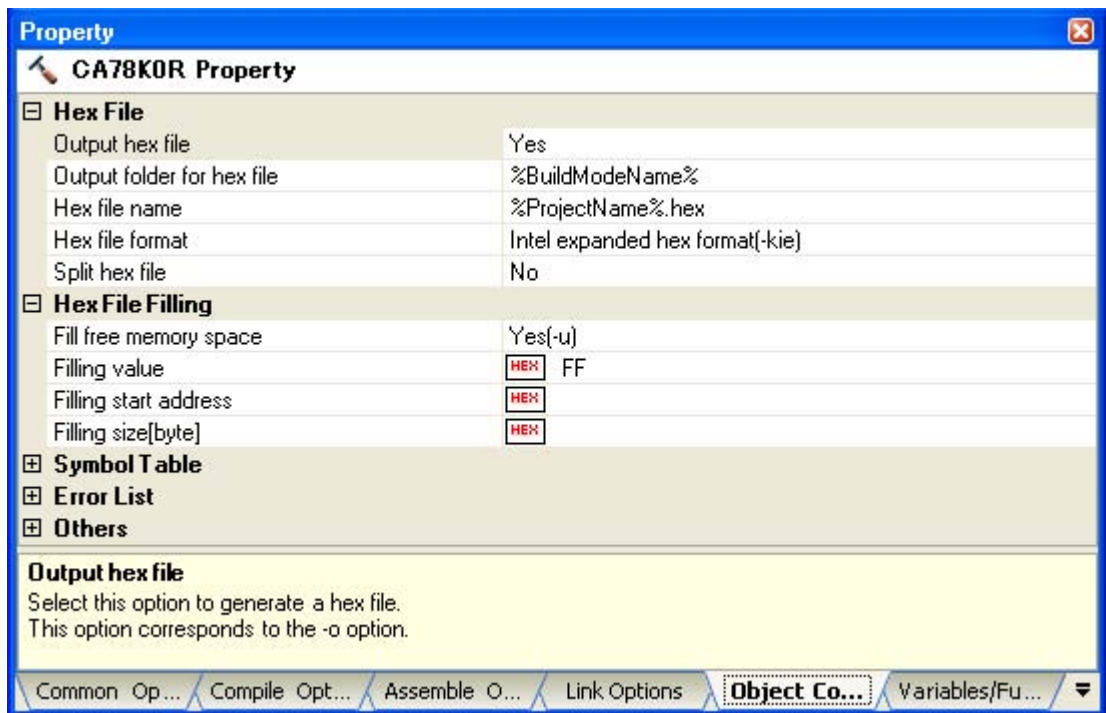
[Object Convert Options] 标签

此标签显示了按如下分类的目标转换器和可更改设置的详细信息。

- (1) [Hex File]
- (2) [Hex File Filling]
- (3) [Symbol Table]
- (4) [Error List]
- (5) [Others]

注意事项 对库工程此标签不会显示。

图 A-8. 属性面板 :[Object Convert Options] 标签



[各分项说明]

(1) [Hex File]

显示详细的 HEX 文件信息，此设置可改变。

| | | | |
|--------------|---|---|-------------|
| 输出 hex 文件 | 选择是否输出 hex 文件。 相当于目标转换程序的 <code>-o</code> 选项。 | | |
| | 默认值 | Yes | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes | 输出 Hex 文件。 |
| | | No(-no) | 不输出 hex 文件。 |
| hex 文件的输出文件夹 | 指定文件夹用于保存 hex 文件。 相当于目标转换程序的 <code>-o</code> 选项。 如果指定的是相对路径，则参考路径为主工程或子工程的文件夹。 如果指定的是绝对路径，则参考路径为主工程或子工程的文件夹。 下面的宏变量是内嵌的宏。 %BuildModeName% : 用生成模式名字替换 如果为空，则作为已设置工程文件夹。 当 [Output hex file] 属性为 [No(-no)] 时，不显示该属性。 | | |
| | 默认值 | %BuildModeName% | |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示文件夹浏览对话框的来编辑。 | |
| | 限制 | 多达 247 个字符 | |
| Hex 文件名 | 指定 hex 文件名。 相当于目标转换程序的 <code>-o</code> 选项。 可以任意指定扩展名。 下面的宏变量是内嵌的宏。 %ProjectName% : 用工程名称替换。 当 [Output hex file] 属性为 [No(-no)] 时，不显示该属性。 | | |
| | 默认值 | %ProjectName%.hex | |
| | 如何改变 | 直接在文本框中输入。 | |
| | 限制 | 多达 259 个字符 | |

| | | | |
|--------------------------------|--|--|--|
| Hex 文件格式 | 选择生成 hex 文件的格式。 相当于目标转换程序的 -k 选项。 当 [Output hex file] 属性为 [No(-no)] 时，不显示该属性。 | | |
| | 默认值 | Intel 扩展 hex 格式 (-kie) | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Intel 标准 hex 格式 (-ki) | 指定 Intel 标准的 hex 格式作为 hex 文件的生成格式。 |
| | | Intel 扩展 hex 格式 (-kie) | 指定 Intel 扩展的 hex 格式作为 hex 文件的生成格式。 |
| | | Motorola S 型格式 (标准地址) (-km) | 采用 Motorola S 型格式 (标准格式) 作为 hex 文件的生成格式。 |
| Motorola S 型格式 (32-位地址) (-kme) | | 采用 Motorola S 型格式 (32-位地址) 作为 hex 文件的生成格式。 | |
| 扩展的 Tektronix hex 格式 (-kt) | 采用扩展的 Tektronix 十六进制格式作为 hex 文件的生成格式。 | | |
| 分割 hex 文件 | 当为有内置闪存的产品连接指定引导区 ROM 程序时，选择是否将文件分割为单独的 hex 格式文件，引导区和其它区各一个。 相当于目标转换程序的 -zf 选项。 对不包含 flash ROM 自编功能的不设置该属性。 当 [Output hex file] 属性为 [No(-no)] 时，不显示该属性。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-zf) | 将文件分割为单独的 HEX 文件：引导区和其它区各一个。 |
| | | No | 不将文件分割为单独的 HEX 文件：引导区和其它区各一个。 |

(2) [Hex File Filling]

显示详细的 HEX 文件填充信息，此设置可改变。

| | | |
|----------|---|---------------------------|
| 填充空存储器区域 | 没有写入 HEX 格式文件的地址区域可能被写入不需要的代码。指定是否提前写入代码防止程序读取这些地址而导致程序出错。 相当于目标转换程序的 -u 选项。 当 [Hex File] 的 [Output hex file] 属性为 [No(-no)] 时，不显示该属性。 | |
| | 默认值 | Yes(-u) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-u) |
| No(-nu) | | 不提前将代码写到没有写入 HEX 格式文件的地址。 |

| | | |
|--------------|--|---|
| 填充值 | 指定填充值，无 0x 的十六进制值（例如：FF），将写到没有写入 HEX 格式文件的地址。 如果没有指定，默认值为“FF”。 相当于目标转换程序的 -u 选项。 当 [Fill free memory space] 属性为 [No(-nu)] 时，不显示该属性。 | |
| | 默认值 | FF |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0 ~ FF（十六进制数） |
| 填充起始地址 | 以无 0x 的十六进制数指定填充起始地址（例如：100A0）。 如果为空，默认值为“0”。 如果设置该属性，也需要设置 [Filling size[byte]] 属性。如果 [Filling size[byte]] 属性为空，则指定的该属性无效。 相当于目标转换程序的 -u 选项。 当 [Fill free memory space] 属性为 [No(-nu)] 时，不显示该属性。 | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0 到 <i>程序区的最大地址</i> （十六进制数） |
| 填充字节数 [byte] | 以无 0x 的十六进制数指定从填充起始地址开始的字节数（例如：F00）。 如果改变 [Filling start address] 属性的结果不在该属性可指定范围内，那么该属性为空。 相当于目标转换程序的 -u 选项。 当 [Fill free memory space] 属性为 [No(-nu)] 时，不显示该属性。 | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 1 ~ 0xFFFF - <i>填充起始地址</i> + 0x1（十六进制数） |

(3) [Symbol Table]

显示详细的符号表信息，可以更改此设置。

| | | |
|---------|-----------------------------------|-----------|
| 输出符号表文件 | 选择是否输出符号表文件。 相当于目标转换程序的 -s 选项。 | |
| | 默认值 | Yes(-s) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-s) |
| No(-ns) | | 不输出符号表文件。 |

(4) [Error List]

显示详细的错误表信息，此设置可改变。

| | | | |
|----------|--|-----------|------------|
| 输出错误列表文件 | 选择是否输出错误列表文件。 相当于链接器的 -e 选项。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-e) | 输出错误列表文件 |
| | | No | 不输出错误列表文件。 |

(5) [Others]

显示目标转换的其他详细信息，此设置可改变。

| | | | |
|-------------|---|--|--|
| 目标转换之前执行的命令 | 指定目标转换之前要执行的命令。 下面的宏变量是内嵌的宏。 %ProjectFolder%: 用工程文件夹的绝对路径替换。 %OutputFolder%: 用输出文件夹的绝对路径替换。 %OutputFile%: 用输出文件的绝对路径替换。 %InputFile%: 用正在进行目标转换的输入文件的绝对路径替换。 %ObjectConvertedFile%: 用正在进行目标转换的输出文件的绝对路径替换。 指定的命令显示为子属性。 | | |
| | 默认值 | 目标转换之前执行的命令 [定义条目的数目] | |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 | |
| | 限制 | 多达 1023 个字符 | |
| | | 设定最多可达 64 个条目。 | |
| 目标转换之后执行的命令 | 指定目标转换之后要执行的命令。 下面的宏变量是内嵌的宏。 %ProjectFolder%: 用工程文件夹的绝对路径替换。 %OutputFolder%: 用输出文件夹的绝对路径替换。 %OutputFile%: 用输出文件的绝对路径替换。 %InputFile%: 用正在进行目标转换的输入文件的绝对路径替换。 %ObjectConvertedFile%: 用正在进行目标转换的输出文件的绝对路径替换。 指定的命令显示为子属性。 | | |
| | 默认值 | 目标转换之后执行的命令 [定义条目的数目] | |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 | |
| | 限制 | 多达 1023 个字符 | |
| | | 设定最多可达 64 个条目。 | |
| 其他附加选项 | 输入需要加入的附加目标转换选项。 这里设置的选项加在目标转换选项组的最后。 | | |
| | 默认值 | 空白 | |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示 字符串输入对话框 的来编辑。 | |
| | 限制 | 多达 259 个字符 | |

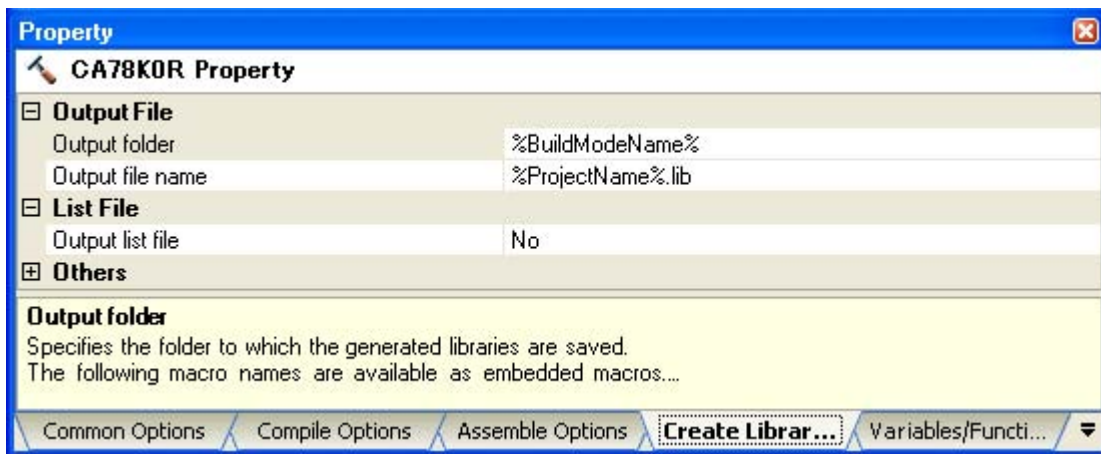
[Create Library Options] 标签

此标签显示了按如下分类的库生成程序和可更改设置的详细信息。

- (1) [Output File]
- (2) [List File]
- (3) [Others]

注意事项 仅对库工程显示此标签。

图 A-9. 属性面板: [Create Library Options] 标签

**[各分项说明]****(1) [Output File]**

显示详细的输出文件信息，设置可以改变。

| | | |
|-------|---|---|
| 输出文件夹 | 指定保存生成库的文件夹。 如果指定的是相对路径，则参考路径为主工程或子工程的文件夹。 如果指定的是绝对路径，则参考路径为主工程或子工程的文件夹。 下面的宏变量是内嵌的宏。 %BuildModeName%: 用生成模式名字替换 如果为空，则作为已设置工程文件夹。 | |
| | 默认值 | %BuildModeName% |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示文件夹浏览对话框的来编辑。 |
| | 限制 | 多达 259 个字符 |

| | | |
|-------|--|-------------------|
| 输出文件名 | 指定输出的库文件名。 使用 ".lib" 扩展名。如果扩展名省略，会自动加上 ".lib"。 下面的宏变量是内嵌的宏。 %ProjectName%: 用工程名称替换。 | |
| | 默认值 | %ProjectName%.lib |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 多达 259 个字符 |

(2) [List File]

显示详细的列表文件信息，此设置可改变。

| | | |
|-------------|--|-------------------|
| 输出表文件 | 选择是否输出带库生成程序的表文件。 相当于列表子命令的 -o 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| No | | 不输出列表文件（库文件模块信息）。 |
| 公共符号信息 | 选择是否输出公共符号信息到带库生成程序的表文件。 相当于列表子命令的 -public 选项。 当 [Output list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| No | | 不输出公共符号表信息到列表文件。 |
| 带格式馈送控制代码输出 | 选择是否输出格式馈送代码到表文件。 相当于库生成程序的 -lf 选项。 当 [Output list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-lf) |
| No | | 不输出格式馈送代码到表文件。 |
| 1 行字符数目 | 指定表文件每行字符数。 相当于库生成程序的 -lw 选项。 当 [Output list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 132 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 72 ~ 260（十进制数） |

| | | |
|--------|---|------------------------|
| 1 页的行数 | 指定表文件每页行数。 如果是 0，将不会生成分页符。 相当于库生成程序的 -ll 选项。 当 [Output list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 0 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0, 和 20 ~ 32767 (十进制数) |

(3) [Others]

显示库的其它详细信息以及可修改的配置信息。

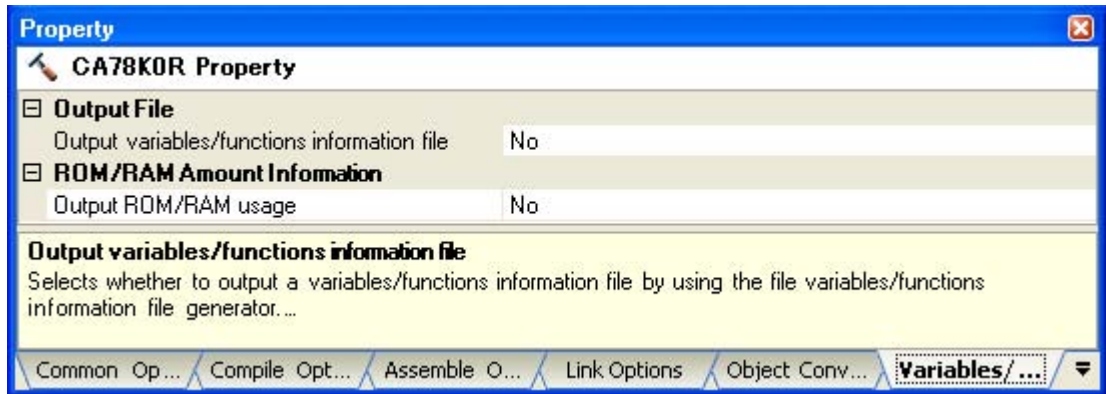
| | | |
|------------|--|--|
| 库生成之前执行的命令 | 指定库生成之前要执行的命令。 下面的宏变量是内嵌的宏。 %ProjectFolder%: 用工程文件夹的绝对路径替换。 %OutputFolder%: 用输出文件夹的绝对路径替换。 %OutputFile%: 用输出文件的绝对路径替换。 %LibraryFile%: 用正在生成的库文件的绝对路径替换。 指定的命令显示为子属性。 | |
| | 默认值 | 库生成之前执行的命令 [定义条目的数目] |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 1023 个字符 设定最多可达 64 个条目。 |
| 库生成之后执行的命令 | 指定库生成之后要执行的命令。 下面的宏变量是内嵌的宏。 %ProjectFolder%: 用工程文件夹的绝对路径替换。 %OutputFolder%: 用输出文件夹的绝对路径替换。 %OutputFile%: 用输出文件的绝对路径替换。 %LibraryFile%: 用正在生成的库文件的绝对路径替换。 指定的命令显示为子属性。 | |
| | 默认值 | 库生成之后执行的命令 [定义条目的数目] |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 1023 个字符 设定最多可达 64 个条目。 |
| 其他附加选项 | 输入需要加入的附加库生成程序选项。 这里设置的选项加在库生成程序选项组的最后。 | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示 字符串输入对话框 的来编辑。 |
| | 限制 | 多达 259 个字符 |

[Variables/Functions Relocation Options] 标签

此标签显示按如下分类的变量 / 函数信息文件生成器和可更改配置的详细信息。

- (1) [Output File]
- (2) [Margin]
- (3) [ROM/RAM Amount Information]

图 A-10. 属性面板: [Memory Bank Relocation Options] 标签

**[各分项说明]****(1) [Output File]**

显示详细的输出文件信息，设置可以改变。

| | | | | | |
|-------------------|---|---|-----|---------------|----|
| 输出变量 / 函数信息文件 | 选择是否输出变量 / 函数信息文件。 | | | | |
| | 默认值 | No | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes</td> <td>输出变量 / 函数信息文件</td> </tr> <tr> <td>No</td> <td>不输出变量 / 函数信息文件。</td> </tr> </tbody> </table> | Yes | 输出变量 / 函数信息文件 | No |
| Yes | 输出变量 / 函数信息文件 | | | | |
| No | 不输出变量 / 函数信息文件。 | | | | |
| 变量 / 函数信息文件的输出文件夹 | 指定保存变量 / 函数信息文件的路径。 这相当于变量 / 函数信息文件生成器中的 -vo 选项。 如果指定的是相对路径，则参考路径为主工程或子工程的文件夹。 如果指定的是绝对路径，则参考路径为主工程或子工程的文件夹。 下面的宏变量是内嵌的宏。 %BuildModeName%: 用生成模式名字替换 如果为空，则作为已设置工程文件夹。 当 [Output variables/function information file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | %BuildModeName% | | | |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示文件夹浏览对话框的来编辑。 | | | |
| | 限制 | 多达 247 个字符 | | | |

| | | |
|--------------|---|-------------------|
| 变量 / 函数信息文件名 | 指定变量 / 函数信息文件名。 这相当于变量 / 函数信息文件生成器中的 -vo 选项。 使用 ".vfi" 扩展名。如果扩展名省略, 会自动加上 ".vfi"。 下面的宏变量是内嵌的宏。 %ProjectName% : 用工程名称替换。 当 [Output variables/function information file] 属性为 [No] 时, 不显示该属性。 | |
| | 默认值 | %ProjectName%.vfi |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 多达 259 个字符 |

(2) [Margin]

显示余量的详细信息以及可修改的配置信息。

当 [Output File] 分项的 [Output variables/function information file] 属性选中为 [No] 时, 不显示该属性。

| | | |
|------------|--|----------------|
| saddr 区域界限 | 为 saddr 区域指定界限。 在通过变量 / 函数信息文件生成器分配变量给 saddr 区域之后, 由于处理顺序和排列的关系, 在编译或链接期间可能会产生排列错误。在这种状况下, 在 saddr 区域设置边界能够避免这种错误。 这相当于变量 / 函数信息文件生成器中的 -vs 选项。 | |
| | 默认值 | 0 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0 ~ 192 (十进制数) |

(3) [ROM/RAM Amount Information]

显示有关 ROM/RAM 用法的详细信息以及可更改的配置。

| | | |
|---------------|--|-----------------|
| 输出 ROM/RAM 用法 | 选择是否在 输出面板 上显示 ROM/RAM 用法。 这相当于变量 / 函数信息文件生成器中的 -vx 选项。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| No | | 不输出 ROM/RAM 用法。 |

[Build Settings] 标签

此标签显示关于每个 C 源文件，汇编源文件，链接指令文件，变量 / 函数信息文件，目标文件，以及库文件的详细信息，分成以下各类型且配置可以修改。

(1) [Build]

图 A-11. 属性面板 : [Build Settings] 标签 (在选定 C 源文件时)

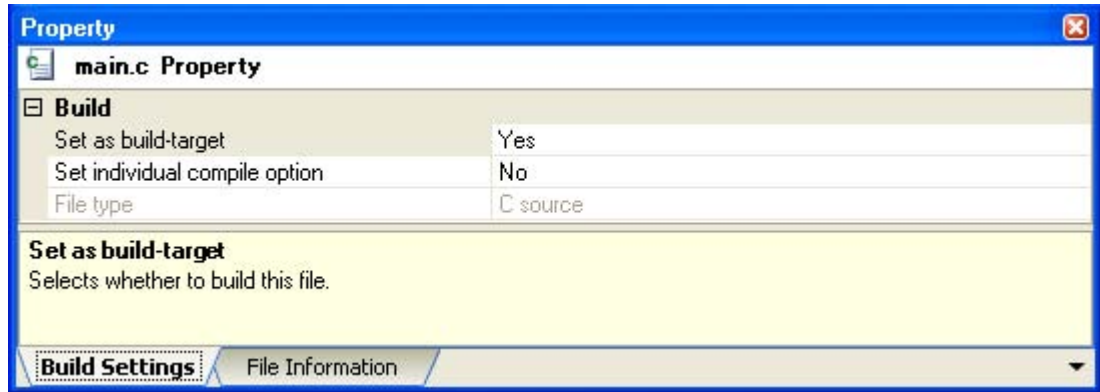


图 A-12. 属性面板 : [Build Settings] 标签 (在选定汇编源文件时)

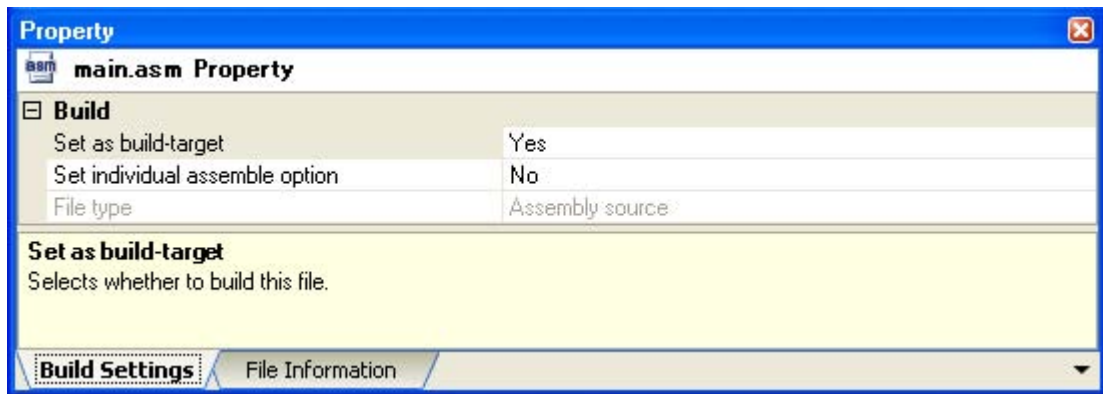


图 A-13. 属性面板 : [Build Settings] 标签 (在选定链接指令文件时)

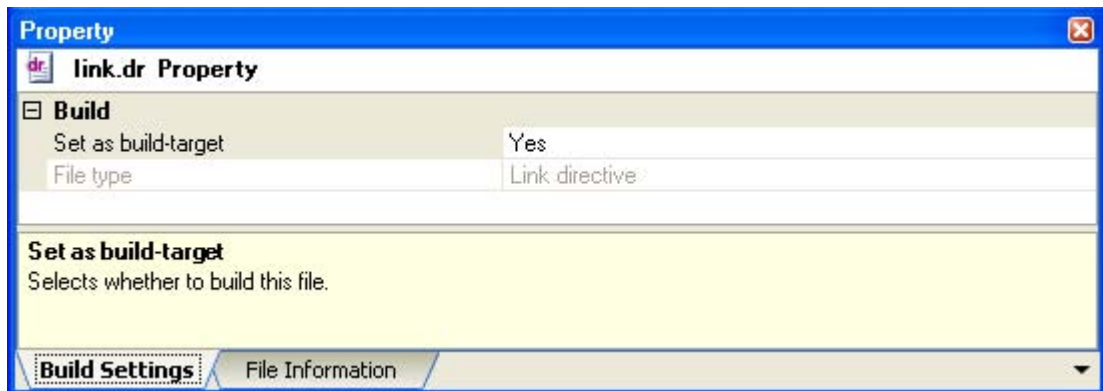


图 A-14. 属性面板 : [Build Settings] 标签 (在选定变量 / 函数信息文件时)

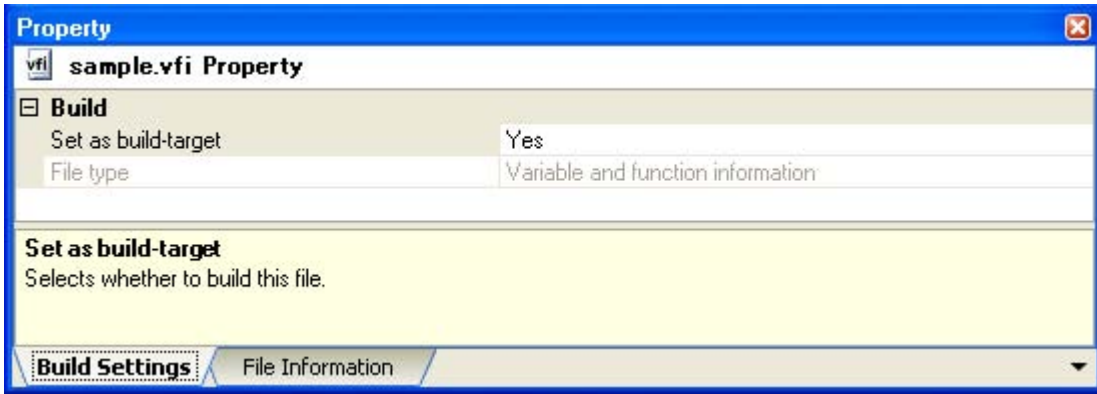


图 A-15. 属性面板 : [Build Settings] 标签 (在选定目标文件时)

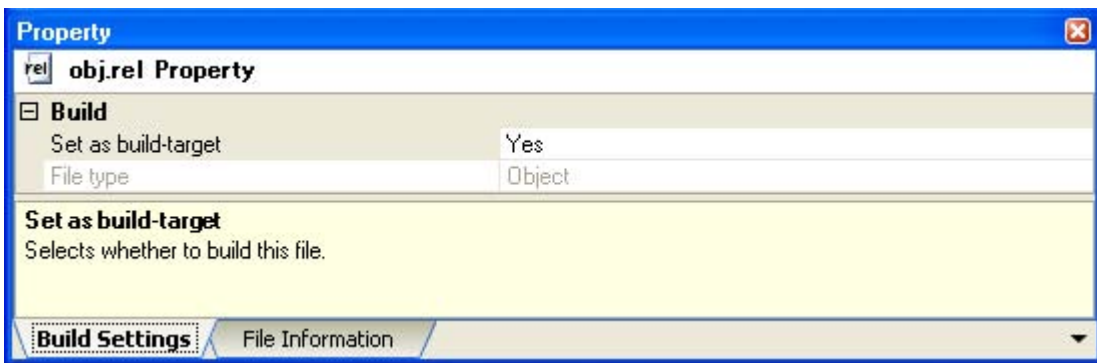
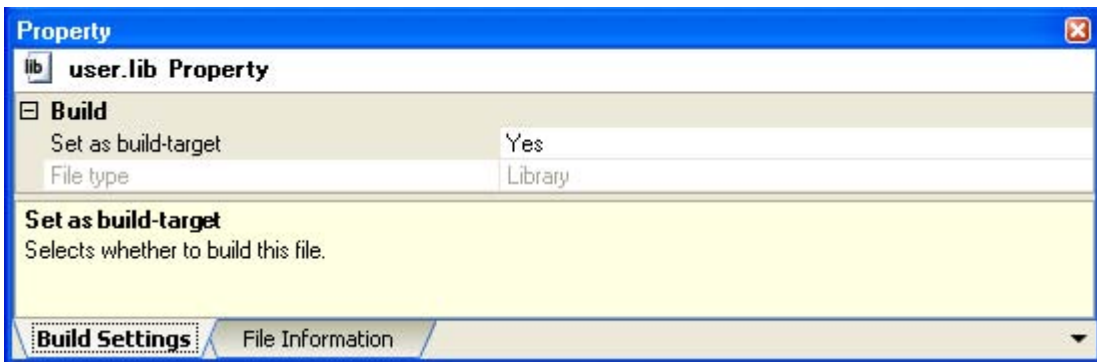


图 A-16. 属性面板 : [Build Settings] 标签 (在选定库文件时)



[各分项说明]

(1) [Build]

显示构建的详细信息以及可修改的配置信息。

| | | | |
|-----------|---|--|---------------------------|
| 设置为构建目标 | 选择是否构建已选文件。 | | |
| | 默认值 | Yes | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes | 构建选定文件 |
| | | No | 不构建选定文件。 |
| 设置个性化编译选项 | 选择是否设置编译选项并区别于选定 C 源文件的工程设置。 该属性只有当 C 源文件在 工程树面板 中选中并且在 [Build] 分项中的 [Set as build-target] 选为 [Yes] 才会有显示。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes | 选择编译选项并区别于选定 C 源文件的工程设置。 |
| | | No | 不设置区别于选定 C 源文件的工程设置的编译选项。 |
| 设置个性化汇编选项 | 选择是否设置区别于选定汇编源文件的工程设置汇编选项。 该属性只有当汇编源文件在 工程树面板 中选中并且在 [Build] 分项中的 [Set as build-target] 被选为 [Yes] 才会有显示。 | | |
| | 默认值 | No | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes | 选择区别于选定汇编源文件的工程设置编译选项。 |
| | | No | 不设置区别于选定汇编源文件的工程设置的编译选项。 |
| 文件类型 | 显示选定的文件类型 | | |
| | 默认值 | C 源 (在 C 源文件选定时) 汇编源代码 (在汇编源文件选定时) 链接指令 (在链接指令文件选定时) 变量和函数信息 (在变量 / 函数信息文件选中时) 目标 (在目标文件中时) 库 (在库文件选中时) | |
| | 如何改变 | 不允许改变。 | |

[Individual Compile Options] 标签

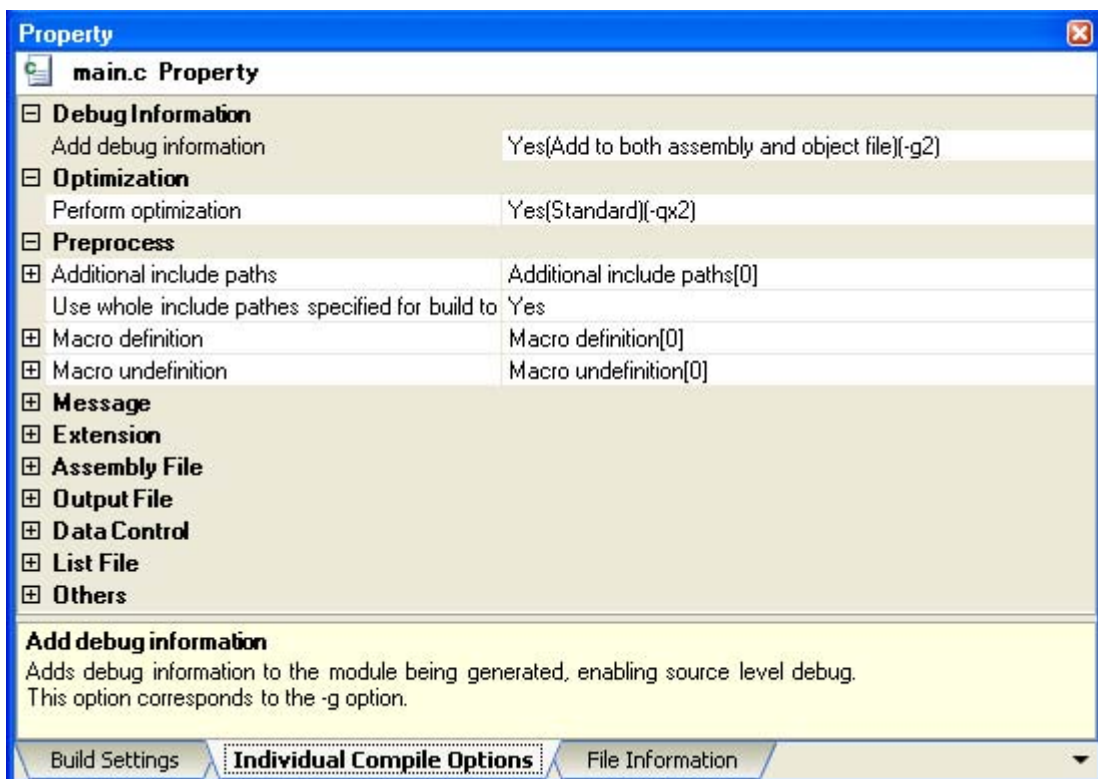
此标签说明了根据下列可修改配置类别把 C 源文件的详细信息做了分类。

注意：此标签代替了 [Compile Options] 标签的设置。如果设置通过 [Compile Options] 标签改变，属性就会显示成黑体字。

备注 此标签的显示只有通过选中 [Build]，此选项属于 [Build Settings] 标签，并将 [Set individual compile option] 属性设置为 [YES]。

- (1) [Debug Information]
- (2) [Optimization]
- (3) [Optimization(Details)]
- (4) [Preprocess]
- (5) [Message]
- (6) [Extension]
- (7) [Assembly File]
- (8) [Output File]
- (9) [Data Control]
- (10) [List File]
- (11) [Others]

图 A-17. 属性面板 :[Individual Compile Options] 标签



[各分项说明]

(1) [Debug Information]

显示详细的调试信息，设置可以更改。

| | | | |
|--------|---|------------------------|----------------------|
| 加入调试信息 | 选择是否在将生成的模块中加入调试信息来允许源文件级调试。 相当于编译器的 <code>-g</code> 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | 是 (只添加到目标文件中)(-g1) | 在将要产生的目标模块文件中加入调试信息。 |
| | | 是 (添加到汇编和目标文件中)(-g2) | 在将要产生的目标模块文件中加入调试信息。 |
| No | | 在将要产生的目标模块文件中加入调试信息。 | |

(2) [Optimization]

显示详细的优化信息，此设置可以改变。

| | | | |
|---------------------|--|---|---------------------|
| 进行优化 | 选择编译优化的类型。 相当于编译器的 <code>-qx</code> 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(Speed precedence)(-qx1) | 进行优化时运行速度优先。 |
| | | Yes(Standard)(-qx2) | 进行优化时同时考虑运行速度和模块大小。 |
| | | Yes(Code size precedence)(-qx3) | 进行优化时模块大小优先。 |
| Yes(Detail setting) | | [Optimization(Details)] 分项说明如下：进行优化时，在分项中选定的选项有优先级。 如果将 [Optimization(Details)] 分项中所有属性选定为 [No(-nq)] 时，优化将不会起作用。 | |
| No(-nq) | 不指定优化。 | | |

(3) [Optimization(Details)]

显示详细的优化信息，此设置可以改变。

仅当 [Optimization] 的 [Perform optimization] 属性为 [Yes(Detail setting)] 时，分项才会显示。

| | | | |
|-------------------------------|---|--|---|
| 交换计算公式的次序 | 选择是否输出通过交换公式的执行顺序来提高寄存器的利用率代码。 相当于编译器的 -qw 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(Swap order of formula operations)(-qw) | 交换计算公式的次序。 |
| | | No | 不交换计算公式的次序。 |
| 分配自动变量给寄存器或者 SADDR 空间 | 选择是否自动分配自动变量给寄存器和 saddr 空间。 相当于编译器的 -qx 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | 是 (-qv) | 自动分配自动变量给寄存器和 saddr 空间。 |
| | | No | 自动变量不自动分配到寄存器或 saddr 区域。 |
| 将寄存器变量分配到寄存器或 saddr 区域 | 选择是否将寄存器变量分配到寄存器或 saddr 区域。 相当于编译器的 -qr 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-qr) | Assigns register variables to registers and assigns them also to the saddr area. |
| | | No | 不分配寄存器变量到 saddr 区域。 |
| 对字符型 (char) 不使用符号扩展 | 选择是否执行无泛积分扩展的字符相关计算。 相当于编译器的 -qc 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | 是 (-qc) | 不采用泛积分扩展法进行字符 - 相关性的计算 ^{注意} |
| | | No | 采用泛积分扩展法进行字符 - 相关性的计算。 |
| 定义字符为 unsigned 字符 | 选择是否不采用 unsigned 字符限定词来定义字符。 相当于编译器的 -qu 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | 是 (-qu) | 将无修饰符的字符型解释为 unsigned 字符型。 |
| | | No | 不将无修饰符的字符型解释为 unsigned 字符型。 |

| | | | | | | | | | | |
|---|---|--|----------------|------------------------------|---------------------------|-------------------------------|---|--|----|------------------------------|
| 优化分支指令 | 选择是否优化分支指令 相当于编译器的 -qj 选项。 | | | | | | | | | |
| | 默认值 | 常用选项的配置 | | | | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>是 (-qj)</td> <td>优化分支指令。</td> </tr> <tr> <td>No</td> <td>不指定优化分支指令。</td> </tr> </tbody> </table> | 是 (-qj) | 优化分支指令。 | No | 不指定优化分支指令。 | | | | |
| 是 (-qj) | 优化分支指令。 | | | | | | | | | |
| No | 不指定优化分支指令。 | | | | | | | | | |
| 用库代码替换固定代码 (大小优先的优化) | 选择是否用库代码替换固定代码。 相当于编译器的 -ql 选项。 | | | | | | | | | |
| | 默认值 | 常用选项的配置 | | | | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>是 (不可替换)(-ql1)</td> <td>不用库代码替换固定代码。 进行优化时模块大小优先。</td> </tr> <tr> <td>是 (只在处理之前替换 / 之后功能)(-ql2)</td> <td>仅将函数前 / 后的程序化处理替换为库代码。</td> </tr> <tr> <td>Yes(Replace process before/after function, use low level libraries and subroutinize same codes)(-ql3)</td> <td>之前只替换处理程序, 并在库函数之后, 也使用低级库并附加常规化的重复代码。</td> </tr> <tr> <td>No</td> <td>不用库代码替换固定代码。 进行优化时运行速度优先。</td> </tr> </tbody> </table> | 是 (不可替换)(-ql1) | 不用库代码替换固定代码。 进行优化时模块大小优先。 | 是 (只在处理之前替换 / 之后功能)(-ql2) | 仅将函数前 / 后的程序化处理替换为库代码。 | Yes(Replace process before/after function, use low level libraries and subroutinize same codes)(-ql3) | 之前只替换处理程序, 并在库函数之后, 也使用低级库并附加常规化的重复代码。 | No | 不用库代码替换固定代码。 进行优化时运行速度优先。 |
| | 是 (不可替换)(-ql1) | 不用库代码替换固定代码。 进行优化时模块大小优先。 | | | | | | | | |
| 是 (只在处理之前替换 / 之后功能)(-ql2) | 仅将函数前 / 后的程序化处理替换为库代码。 | | | | | | | | | |
| Yes(Replace process before/after function, use low level libraries and subroutinize same codes)(-ql3) | 之前只替换处理程序, 并在库函数之后, 也使用低级库并附加常规化的重复代码。 | | | | | | | | | |
| No | 不用库代码替换固定代码。 进行优化时运行速度优先。 | | | | | | | | | |
| 产生相关分支的 switch 分支表。 | 选择是否产生相关分支的 switch 分支表。 相当于编译器的 -qx 选项。 | | | | | | | | | |
| 默认值 | 常用选项的配置 | | | | | | | | | |
| 如何改变 | 通过下拉列表选择。 | | | | | | | | | |
| 限制 | <table border="1"> <tbody> <tr> <td>Yes(-qt)</td> <td>产生相关分支的 switch 分支表。</td> </tr> <tr> <td>No</td> <td>不指定产生相关分支的 switch 分支表。</td> </tr> </tbody> </table> | | Yes(-qt) | 产生相关分支的 switch 分支表。 | No | 不指定产生相关分支的 switch 分支表。 | | | | |
| Yes(-qt) | 产生相关分支的 switch 分支表。 | | | | | | | | | |
| No | 不指定产生相关分支的 switch 分支表。 | | | | | | | | | |
| 调试优化 | 选择是否进行调试优化。 相当于编译器的 -d 选项。 | | | | | | | | | |
| | 默认值 | 常用选项的配置 | | | | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-qg)</td> <td>实现调试优化。</td> </tr> <tr> <td>No</td> <td>不用为调试指定执行优化。</td> </tr> </tbody> </table> | Yes(-qg) | 实现调试优化。 | No | 不用为调试指定执行优化。 | | | | |
| Yes(-qg) | 实现调试优化。 | | | | | | | | | |
| No | 不用为调试指定执行优化。 | | | | | | | | | |

注 当指定选项 **-qc** 时的计算结果如下:

| 计算目标 | 计算结果 |
|----------------------------------|---------------|
| unsigned 字符类型变量和 unsigned 字符类型变量 | unsigned 字符类型 |
| unsigned 字符类型变量和 signed 字符类型变量 | unsigned 字符类型 |

| 计算目标 | 计算结果 |
|-----------------------------------|---------------|
| 有符号字符类型变量和有符号字符类型变量 | 有符号字符类型 |
| 从 -128 到 255 的常量和 unsigned 字符类型变量 | unsigned 字符类型 |
| 从 -128 到 127 的常量和有符号字符类型变量 | 有符号字符类型 |
| 带前缀 U 的从 0 到 255 的常量和有符号字符类型变量 | unsigned 字符类型 |

(4) [Preprocess]

显示详细的预处理信息，设置可以改变。

| | | |
|----------------------------------|--|--|
| Additional include paths | 指定附加的编译中的 include 路径。 下面的宏变量是内嵌的宏。 %BuildModeName%: 用生成模式名字替换 %ProjectName%: 用工程名称替换。 %CubeSuitePath%: 用 CubeSuite 的绝对安装路径替换。 当省略此选项时，仅查询编译器的标准文件夹。路径的参考点是工程文件夹。 相当于编译器的 -i 选项。 指定的 include 路径显示为子属性。 | |
| | 默认值 | Additional include paths[<i>number of defined items</i>] |
| | 如何改变 | 点击 [...] 按钮，通过 路径编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 259 个字符 多达 64 项可以定义，但这也包括多个被用于链接工具的路径。 |
| 在 build tool 中使用整个特定的 include 路径 | 选择是否在 [Additional include paths] 属性中编译使用特定 include 路径，该属性属于 [Preprocess] 分项，该分项出自被使用的 build tool 中的 [Compile Options] 标签。 相当于编译器的 -i 选项。 | |
| | 默认值 | Yes |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes 编译使用过 build tool 属性中的特定 include 路径。 No 在使用过 build tool 属性中的不使用特定的 include 路径。 |
| 宏定义 | 指定需要定义的宏名称。 通过 "宏名称 = 定义值" 这种格式来指定，每行一个宏。这 "=def" 部分被省略，在这种情况下，"1" 为默认定义值。 相当于编译器的 -d 选项。 指定的宏显示为子属性。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 256 个字符 设定最多可达 30 个条目。 |

| | | |
|-------|---|--|
| 取消宏定义 | 指定需要取消定义的宏名称。 通过 "宏名称" 这种格式来指定, 每行一个宏。 相当于编译器的 <code>-u</code> 选项。 指定的宏显示为子属性。 | |
| | 默认值 | 取消宏定义 [定义条目的数目] |
| | 如何改变 | 点击 [...] 按钮, 通过 文本编辑对话框 来编辑。 对于子属性, 可以用文本框直接输入。 |
| | 限制 | 多达 256 个字符 设定最多可达 30 个条目。 |

(5) [Message]

显示消息的详细信息以及可修改的配置信息。

| | | | |
|------------|--|--|-----------|
| 冗长模式 | 在构建中选择是否在输出面板中显示编译器执行的状态。 相当于编译器的 <code>-v</code> 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-v) 在构建中显示编译器的执行状态。 No 在构建中不显示编译器的执行状态。 | |
| 警告级别 | 选择在编译期间的警告显示级别。 相当于编译器的 <code>-w</code> 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | 没有输出 (-w0) | 不输出警告信息。 |
| | | 常规输出 | 输出常规警告信息。 |
| 详细输出 (-w2) | 输出详细警告信息。 | | |

(6) [Extension]

显示扩展的详细信息以及可修改的配置信息。

| | | |
|--------------|--|---|
| 允许 C++ 格式的注释 | 选择是否允许使用 C++ 格式的注释 ("//"). 相当于编译器的 <code>-zp</code> 选项。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-zp) 允许 C++ 格式的注释。 No 不允许 C++ 格式的注释。 |
| 允许注释嵌套 | 选择是否允许使用嵌套式注释 ("/*"). 相当于编译器的 <code>-zc</code> 选项。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-zc) 允许使用嵌套注释。 No 不允许使用嵌套注释。 |

| | | |
|-------------------------------|---|---|
| 源文件的 Kanji 字符编码 | 选择源文件的 Kanji 字符编码 相当于编译器的 <code>-zs</code> , <code>-ze</code> , 和 <code>-zn</code> 选项。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | <code>Shift_JIS(-zs)</code> |
| <code>EUC-JP(-ze)</code> | | 将源文件的 kanji 代码用 <code>EUC-JP</code> 来解释。 |
| <code>Unspecified(-zn)</code> | | 将源文件不包含 Kanji 代码在解释。 |
| 符合 ANSI 标准 | 选择是否禁用非 ANSI 标准功能以及使用 ANSI 标准的功能。 相当于编译器的 <code>-za</code> 选项。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | <code>Yes(-za)</code> |
| <code>No</code> | | 允许非 ANSI 标准的功能。 |
| 禁用 int 扩展功能 | 选择是否禁止字符型和 <code>unsigned</code> 字符型参数和返回值的整型扩展。 相当于编译器的 <code>-zb</code> 选项。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | <code>Yes(-zb)</code> |
| <code>No</code> | | 禁止字符型和 <code>unsigned</code> 字符型参数和返回值的 int 扩展。 |

(7) [Assembly File]

显示详细的汇编文件信息，此设置可以改变。

| | | |
|---|---|---|
| 输出汇编文件 | 选择是否输出汇编文件。 相当于编译器的 <code>a</code> , <code>-sa</code> , 和 <code>-li</code> 选项。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | <code>Yes(With no C source info)(-a)</code> |
| <code>Yes(With C source info(unexpanded include file contents))(-sa)</code> | | 输出汇编文件（包含 C 源代码信息（包含的文件内容没有被扩展））。 |
| <code>Yes(With C source info(expanded include file contents))(-sa,-li)</code> | | 输出汇编文件（包含 C 源代码信息（扩展包含的文件内容））。 |
| <code>No</code> | | 不输出汇编文件。 |

(8) [Output File]

显示详细的输出文件信息，设置可以改变。

| | | |
|--------|---|------------|
| 目标文件名称 | 在编译后指定名称的目标文件生成。 如果此区域是空的，保存文件扩展名以 <code>.rel</code> 代替 <code>.c</code> 。 相当于编译器的 <code>-o</code> 选项。 | |
| | 默认值 | 空白 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 多达 259 个字符 |

| | | |
|---------------|---|-------------------|
| 输出各种设备的共同目标文件 | 选择是否输出各种设备的共同目标文件。 相当于编译程序的 <code>-common</code> 选项。 | |
| | 默认值 | <i>常用选项的配置</i> |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-common) |
| No | | 不设定输出共同目标文件到各种设备。 |

(9) [Data Control]

显示详细的数据控制信息，此设置可以改变。

| | | |
|------------------|--|----------------------------------|
| 从 MSB 中分配区位给指令 | 选择是否从 MSB 中分配多个区位给指令。 相当于编译器的 <code>-rb</code> 选项。 | |
| | 默认值 | <i>常用选项的配置</i> |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-rb) |
| No | | 从 LSB 中分配多个区位给指令。 |
| 整合架构成员 | 选择是否禁止在结构中为成员（包含 2 个或更多字节）插入对齐数据来对其地址。 相当于编译器的 <code>-rc</code> 选项。 | |
| | 默认值 | <i>常用选项的配置</i> |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-rc) |
| No | | 在结构中为成员（包含 2 个或更多字节）插入对齐数据来对其地址。 |
| 在 1 字节单元里进行间接引用。 | 选择是否在 1 字节单元里进行间接引用。 相当于编译器的 <code>-ra</code> 选项。 | |
| | 默认值 | <i>常用选项的配置</i> |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-ra) |
| No | | 在 1 字节单元里不进行间接引用。 |

| | | | |
|--|---|---|---|
| 将静态变量分配到 <code>saddr</code> 区域 | 选择分配到 <code>saddr</code> 区域静态变量的类型。 相当于编译器的 <code>-rs</code> 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(Size of char)(-rs1) | 将字符型和 <code>unsigned</code> 字符型自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char, short, int)(-rs2) | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、 <code>near</code> 指针类型自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char, short, int, long)(-rs4) | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、长整型、 <code>unsigned</code> 长整型、指针类型自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Structure, union, array)(-rsm) | 将结构、联合和数组自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char and structure, union, array)(-rs1m) | 将字符型、 <code>unsigned</code> 字符型、联合和数组自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char, short, int and structure, union, array)(-rs2m) | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、 <code>near</code> 指针类型、结构体、联合体以及数组类型自动变量分配到 <code>saddr</code> 区域。 |
| Yes(Size of char, short, int, long and structure, union, array)(-rs) | | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、长整型、 <code>unsigned</code> 长整型、指针类型、及结构、联合和数组自动变量分配到 <code>saddr</code> 区域。 | |
| No | 不分配静态变量到 <code>saddr</code> 区域。 | | |

| | | | |
|--|---|---|---|
| 将外部变量分配到 <code>saddr</code> 区域 | 选择分配到 <code>saddr</code> 区域外部变量的类型。 相当于编译器的 <code>-rd</code> 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(Size of char)(-rd1) | 将字符型和 <code>unsigned</code> 字符型自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char, short, int)(-rd2) | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、 <code>near</code> 指针类型自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char, short, int, long)(-rd4) | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、长整型、 <code>unsigned</code> 长整型、指针类型自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Structure, union, array)(-rdm) | 将结构、联合和数组自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char and structure, union, array)(-rd1m) | 将字符型、 <code>unsigned</code> 字符型、联合和数组自动变量分配到 <code>saddr</code> 区域。 |
| | | Yes(Size of char, short, int and structure, union, array)(-rd2m) | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、 <code>near</code> 指针类型、结构体、联合体以及数组类型自动变量分配到 <code>saddr</code> 区域。 |
| Yes(Size of char, short, int, long and structure, union, array)(-rd) | | 将字符型、 <code>unsigned</code> 字符型、短整型、 <code>unsigned</code> 短整型、整型、 <code>unsigned</code> 整型、枚举型、长整型、 <code>unsigned</code> 长整型、指针类型、及结构、联合和数组自动变量分配到 <code>saddr</code> 区域。 | |
| No | 不分配外部变量到 <code>saddr</code> 区域。 | | |
| 指定 ROM 数据的分配目标 | 指定地方分配 ROM 数据。 相当于编译器的 <code>-rf</code> 和 <code>-rm</code> 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(far area)(-rf) | 分配 ROM 数据到 <code>far</code> 区域。 |
| | | Yes(near area)(-rm) | 分配 ROM 数据到 <code>near</code> 区域。 |
| No | 不指定 ROM 数据的分配目标。 | | |

(10)[List File]

显示详细的列表文件信息，此设置可改变。

| | | | |
|----------|---|-----------|------------|
| 输出预处理表文件 | 选择是否输出预处理文件。 相当于编译器的 <code>-p</code> 选项。 | | |
| | 默认值 | 常用选项的配置 | |
| | 如何改变 | 通过下拉列表选择。 | |
| | 限制 | Yes(-p) | 输出预处理表文件。 |
| | | No | 不输出预处理表文件。 |

| | | | | | |
|--|--|---|----------|--|----|
| 不输出注释 | 选择是否禁止将注释输出到预处理表文件中。 相当于编译器的 <code>-kc</code> 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | 常用选项的配置 | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kc)</td> <td>不将注释输出到预处理表文件。</td> </tr> <tr> <td>No</td> <td>将注释输出到预处理表文件。</td> </tr> </tbody> </table> | Yes(-kc) | 不将注释输出到预处理表文件。 | No |
| Yes(-kc) | 不将注释输出到预处理表文件。 | | | | |
| No | 将注释输出到预处理表文件。 | | | | |
| 扩展 <code>#define</code> 预处理指令 | 选择是否扩展 <code>#define</code> 指令到预处理表文件中。 相当于编译器的 <code>-kd</code> 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | 常用选项的配置 | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kd)</td> <td>扩展 <code>#define</code> 指令到预处理表文件中。</td> </tr> <tr> <td>No</td> <td>不扩展 <code>#define</code> 指令到预处理表文件中。</td> </tr> </tbody> </table> | Yes(-kd) | 扩展 <code>#define</code> 指令到预处理表文件中。 | No |
| Yes(-kd) | 扩展 <code>#define</code> 指令到预处理表文件中。 | | | | |
| No | 不扩展 <code>#define</code> 指令到预处理表文件中。 | | | | |
| 扩展 <code>#if</code> 、 <code>#ifdef</code> 、 <code>#ifndef</code> 预处理指令 | 选择是否扩展 <code>#if</code> 、 <code>#ifdef</code> 、 <code>#ifndef</code> 指令到预处理表文件中。 相当于编译器的 <code>-kf</code> 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | 常用选项的配置 | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kf)</td> <td>执行扩展输出 <code>#if</code>、<code>#ifdef</code>、<code>#ifndef</code> 指令到预处理表文件中。</td> </tr> <tr> <td>No</td> <td>不执行扩展输出 <code>#if</code>、<code>#ifdef</code>、<code>#ifndef</code> 指令到预处理表文件中。</td> </tr> </tbody> </table> | Yes(-kf) | 执行扩展输出 <code>#if</code> 、 <code>#ifdef</code> 、 <code>#ifndef</code> 指令到预处理表文件中。 | No |
| Yes(-kf) | 执行扩展输出 <code>#if</code> 、 <code>#ifdef</code> 、 <code>#ifndef</code> 指令到预处理表文件中。 | | | | |
| No | 不执行扩展输出 <code>#if</code> 、 <code>#ifdef</code> 、 <code>#ifndef</code> 指令到预处理表文件中。 | | | | |
| 扩展 <code>#include</code> 预处理指令 | 选择是否扩展 <code>#include</code> 指令到预处理表文件中。 相当于编译器的 <code>-ki</code> 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | 常用选项的配置 | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-ki)</td> <td>扩展 <code>#include</code> 指令到预处理表文件中。</td> </tr> <tr> <td>No</td> <td>不扩展 <code>#include</code> 指令到预处理表文件中。</td> </tr> </tbody> </table> | Yes(-ki) | 扩展 <code>#include</code> 指令到预处理表文件中。 | No |
| Yes(-ki) | 扩展 <code>#include</code> 指令到预处理表文件中。 | | | | |
| No | 不扩展 <code>#include</code> 指令到预处理表文件中。 | | | | |

| | | | | | | | |
|---------------------------|---|--|---------------------------|----------------------|-------------------------|-----------------------|----|
| 扩展 #line 预处理指令 | 选择是否扩展 #line 指令到预处理表文件中。 相当于编译器的 -kl 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | | | |
| | 默认值 | 常用选项的配置 | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kl)</td> <td>扩展 #line 指令到预处理表文件中。</td> </tr> <tr> <td>No</td> <td>不扩展 #line 指令到预处理表文件中。</td> </tr> </tbody> </table> | Yes(-kl) | 扩展 #line 指令到预处理表文件中。 | No | 不扩展 #line 指令到预处理表文件中。 | |
| Yes(-kl) | 扩展 #line 指令到预处理表文件中。 | | | | | | |
| No | 不扩展 #line 指令到预处理表文件中。 | | | | | | |
| 输出行号 | 选择是否输出行号到预处理表文件中。 相当于编译器的 -kn 选项。 当 [Output preprocess list file] 属性为 [No] 时，不显示该属性。 | | | | | | |
| | 默认值 | 常用选项的配置 | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kn)</td> <td>将行号输出到预处理表文件。</td> </tr> <tr> <td>No</td> <td>不将行号输出到预处理表文件。</td> </tr> </tbody> </table> | Yes(-kn) | 将行号输出到预处理表文件。 | No | 不将行号输出到预处理表文件。 | |
| Yes(-kn) | 将行号输出到预处理表文件。 | | | | | | |
| No | 不将行号输出到预处理表文件。 | | | | | | |
| 输出错误列表文件 | 选择是否输出错误列表文件。 相当于编译器的 -e 和 -se 选项。 | | | | | | |
| | 默认值 | 常用选项的配置 | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(Without C source)(-e)</td> <td>输出错误列表文件（不包含 C 源代码）。</td> </tr> <tr> <td>Yes(With C source)(-se)</td> <td>输出错误列表文件（包含 C 源代码）。</td> </tr> <tr> <td>No</td> <td>不输出错误列表文件。</td> </tr> </tbody> </table> | Yes(Without C source)(-e) | 输出错误列表文件（不包含 C 源代码）。 | Yes(With C source)(-se) | 输出错误列表文件（包含 C 源代码）。 | No |
| Yes(Without C source)(-e) | 输出错误列表文件（不包含 C 源代码）。 | | | | | | |
| Yes(With C source)(-se) | 输出错误列表文件（包含 C 源代码）。 | | | | | | |
| No | 不输出错误列表文件。 | | | | | | |
| 输出交叉引用列表文件 | 选择是否输出交叉引用列表文件。 相当于编译器的 -x 选项。 | | | | | | |
| | 默认值 | 常用选项的配置 | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-x)</td> <td>输出交叉引用列表文件</td> </tr> <tr> <td>No</td> <td>不输出交叉引用列表文件。</td> </tr> </tbody> </table> | Yes(-x) | 输出交叉引用列表文件 | No | 不输出交叉引用列表文件。 | |
| Yes(-x) | 输出交叉引用列表文件 | | | | | | |
| No | 不输出交叉引用列表文件。 | | | | | | |
| 带格式馈送控制代码输出 | 选择是否将格式馈送控制代码输出到表文件（预处理表文件，错误表文件和交叉引用列表文件）。 相当于编译器的 -lf 选项。 仅当 [Output error list file] 属性选择为 [Yes] 或 [Output preprocess list file] 属性选择为 [Yes(-p)] 或 [Output cross reference list file] 属性选择为 [Yes(-x)] 时，显示该属性。 | | | | | | |
| | 默认值 | 常用选项的配置 | | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-lf)</td> <td>输出格式馈送代码到表文件。</td> </tr> <tr> <td>No</td> <td>不输出格式馈送代码到表文件。</td> </tr> </tbody> </table> | Yes(-lf) | 输出格式馈送代码到表文件。 | No | 不输出格式馈送代码到表文件。 | |
| Yes(-lf) | 输出格式馈送代码到表文件。 | | | | | | |
| No | 不输出格式馈送代码到表文件。 | | | | | | |

| | | |
|---------|---|-----------------------|
| 1 行字符数目 | 指定表文件每行字符数（预处理表文件，错误表文件和交叉引用列表文件）。 相当于编译器的 <code>-lw</code> 选项。 仅当 [Output error list file] 属性选择为 [Yes] 或 [Output preprocess list file] 属性选择为 [Yes(-p)] 或 [Output cross reference list file] 属性选择为 [Yes(-x)] 时，显示该属性。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 72 ~ 132（十进制数） |
| 1 页的行数 | 指定表文件每页行数（预处理表文件，错误表文件和交叉引用列表文件）。 如果是 0，将不会生成分页符。 相当于编译器的 <code>-ll</code> 选项。 仅当 [Output error list file] 属性选择为 [Yes] 或 [Output preprocess list file] 属性选择为 [Yes(-p)] 或 [Output cross reference list file] 属性选择为 [Yes(-x)] 时，显示该属性。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0, 和 20 ~ 65535（十进制数） |
| Tab 宽度 | 指定表文件的 Tab 宽度（预处理表文件，错误表文件和交叉引用列表文件）。 相当于编译器的 <code>-lt</code> 选项。 仅当 [Output error list file] 属性选择为 [Yes] 或 [Output preprocess list file] 属性选择为 [Yes(-p)] 或 [Output cross reference list file] 属性选择为 [Yes(-x)] 时，显示该属性。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 0 ~ 8（十进制数） |

(11) [Others]

显示编译的其它详细信息以及可修改的配置信息。

| | | |
|-----------|--|---|
| 编译之前执行的命令 | 指定编译之前要执行的命令。 下面的宏变量是内嵌的宏。 %ProjectFolder%: 用工程文件夹的绝对路径替换。 %OutputFolder%: 用输出文件夹的绝对路径替换。 %OutputFile%: 用输出文件的绝对路径替换。 %InputFile%: 用将要编译文件的绝对路径替换。 %CompiledFile%: 用正在编译输出文件的绝对路径替换。 指定的命令显示为子属性。 | |
| | 默认值 | 编译之前执行的命令 [定义条目的数目] |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 1023 个字符 设定最多可达 64 个条目。 |

| | | |
|-----------|--|--|
| 编译之后执行的命令 | 指定编译之后要执行的命令。 下面的宏变量是内嵌的宏。 %ProjectFolder%: 用工程文件夹的绝对路径替换。 %OutputFolder%: 用输出文件夹的绝对路径替换。 %OutputFile%: 用输出文件的绝对路径替换。 %InputFile%: 用将要编译文件的绝对路径替换。 %CompiledFile%: 用正在编译输出文件的绝对路径替换。 指定的命令显示为子属性。 | |
| | 默认值 | 编译之后执行的命令 [定义条目的数目] |
| | 如何改变 | 点击 [...] 按钮, 通过 文本编辑对话框 来编辑。 对于子属性, 可以用文本框直接输入。 |
| | 限制 | 多达 1023 个字符 设定最多可达 64 个条目。 |
| 其他附加选项 | 输出需要加入的附加编译选项。 设置的选项加在编译选项组的最后。 | |
| | 默认值 | <i>常用选项的配置</i> |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示 字符串输入对话框 的来编辑。 |
| | 限制 | 多达 259 个字符 |

[Individual Assemble Options] 标签

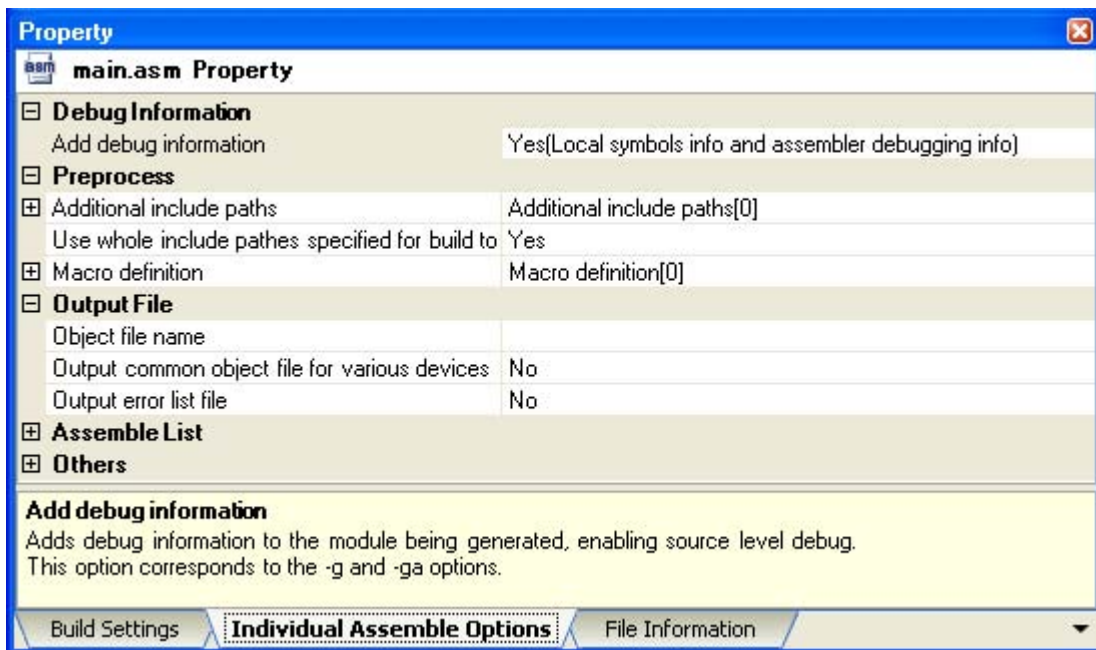
此标签说明了根据下列可修改配置类别把汇编源文件的详细信息做了分类。

注意：此标签代替了 [Assemble Options] 标签的设置。如果设置通过 [Assemble Options] 标签改变，属性就会显示成黑体字。

- 备注 1.** 此标签的显示只有通过选中 [Build]，此选项属于 [Build Settings] 标签，并将 [Set individual assemble option] 属性设置为 [YES]。
2. 此标签当选定 C 源文件并且在 [Output assemble file] 的属性设置为 [Yes] 也会显示，该属性属于 [Assembly File] 分项，此分项在 [Individual Compile Options] 标签中选择。

- (1) [Debug Information]
- (2) [Preprocess]
- (3) [Output File]
- (4) [Assemble List]
- (5) [Others]

图 A-18. 属性面板 :[Individual Assemble Options] 标签



[各分项说明]

(1) [Debug Information]

显示详细的调试信息，设置可以更改。

| | | |
|--------|--|-----------------------------------|
| 加入调试信息 | 选择是否在将生成的模块中加入调试信息来允许源文件级调试。 相当于编译器的 <code>-g</code> 和 <code>-ga</code> 选项。 | |
| 默认值 | 常用选项的配置 | |
| 如何改变 | 通过下拉列表选择。 | |
| 限制 | Yes(Assembler debugging info)(-ng,-ga) | 在将要产生的目标模块文件中加入调试信息（汇编调试符号信息）。 |
| | Yes(Local symbols info and assembler debugging info) | 在将要产生的目标模块文件中加入调试信息（局部符号和汇编调试信息）。 |
| | No(-ng,-nga) | 在将要产生的目标模块文件中加入调试信息。 |

(2) [Preprocess]

显示详细的预处理信息，设置可以改变。

| | | |
|----------------------------------|--|--|
| Additional include paths | 在汇编时指定附加的 include 路径。 下面的宏变量是内嵌的宏。 <code>%BuildModeName%</code> : 用生成模式名字替换 <code>%ProjectName%</code> : 用工程名称替换。 <code>%CubeSuitePath%</code> : 用 CubeSuite 的绝对安装路径替换。 当忽略该选项时，仅查询汇编程序标准文件夹。路径的参考点是工程文件夹。 相当于汇编程序的 <code>-i</code> 选项。 指定的 include 路径显示为子属性。 | |
| 默认值 | Additional include paths[<i>number of defined items</i>] | |
| 如何改变 | 点击 [...] 按钮，通过 路径编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 | |
| 限制 | 多达 259 个字符 多达 64 项可以定义，但这也包括多个被用于链接工具的路径。 | |
| 在 build tool 中使用整个特定的 include 路径 | 选择是否在 [Additional include paths] 属性中汇编使用特定 include 路径，该属性属于 [Preprocess] 分项，该分项出自被使用的 build tool 中的 [Assemble Options] 标签。 相当于汇编程序的 <code>-i</code> 选项。 | |
| 默认值 | Yes | |
| 如何改变 | 通过下拉列表选择。 | |

| | | |
|-----|---|---|
| 宏定义 | <p>指定需要定义的宏名称。</p> <p>通过 " 宏名称= 定义值" 这种格式来指定, 每行一个宏。这 "<i>=def</i>" 部分被省略, 在这种情况下, "1" 为默认定义值。</p> <p>相当于汇编程序的 <i>-d</i> 选项。</p> <p>指定的宏显示为子属性。</p> | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | <p>点击 [...] 按钮, 通过文本编辑对话框来编辑。</p> <p>对于子属性, 可以用文本框直接输入。</p> |
| | 限制 | <p>多达 256 个字符</p> <p>设定最多可达 30 个条目。</p> |

(3) [Output File]

显示详细的输出文件信息, 设置可以改变。

| | | | | | | |
|---------------|---|---|--------------|----------------|----|----------------|
| 目标文件名称 | <p>在汇编后指定名称的目标文件生成。</p> <p>如果此区域是空的, 保存文件扩展名以 <i>.rel</i> 代替 <i>.asm</i>。</p> <p>这相当于汇编器的 <i>-o</i> 选项。</p> | | | | | |
| | 默认值 | 空白 | | | | |
| | 如何改变 | 直接在文本框中输入。 | | | | |
| | 限制 | 多达 259 个字符 | | | | |
| 输出各种设备的共同目标文件 | <p>选择是否输出各种设备的共同目标文件。</p> <p>相当于汇编程序的 <i>-common</i> 选项。</p> | | | | | |
| | 默认值 | 常用选项的配置 | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | |
| | 限制 | <table border="1"> <tr> <td>Yes(-common)</td> <td>输出共同目标文件到各种设备。</td> </tr> <tr> <td>No</td> <td>输出 78K0 的目标文件。</td> </tr> </table> | Yes(-common) | 输出共同目标文件到各种设备。 | No | 输出 78K0 的目标文件。 |
| Yes(-common) | 输出共同目标文件到各种设备。 | | | | | |
| No | 输出 78K0 的目标文件。 | | | | | |
| 输出错误列表文件 | <p>选择是否输出错误列表文件。</p> <p>相当于汇编程序的 <i>-e</i> 选项。</p> | | | | | |
| | 默认值 | 常用选项的配置 | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | |
| | 限制 | <table border="1"> <tr> <td>Yes(-e)</td> <td>输出错误列表文件</td> </tr> <tr> <td>No</td> <td>不输出错误列表文件。</td> </tr> </table> | Yes(-e) | 输出错误列表文件 | No | 不输出错误列表文件。 |
| Yes(-e) | 输出错误列表文件 | | | | | |
| No | 不输出错误列表文件。 | | | | | |

(4) [Assemble List]

显示详细的汇编列表信息, 此设置可改变。

| | | | | | | |
|----------|--|---|---------|----------|---------|------------|
| 输出汇编列表文件 | <p>选择是否输出汇编列表文件。</p> <p>这相当于汇编器的 <i>-p</i> 选项。</p> | | | | | |
| | 默认值 | 常用选项的配置 | | | | |
| | 如何改变 | 通过下拉列表选择。 | | | | |
| | 限制 | <table border="1"> <tr> <td>Yes(-p)</td> <td>输出汇编列表文件</td> </tr> <tr> <td>No(-np)</td> <td>不输出汇编列表文件。</td> </tr> </table> | Yes(-p) | 输出汇编列表文件 | No(-np) | 不输出汇编列表文件。 |
| Yes(-p) | 输出汇编列表文件 | | | | | |
| No(-np) | 不输出汇编列表文件。 | | | | | |

| | | |
|--------------|--|---------------------|
| 执行列表转换器 | 选择是否在可执行模块生成后执行列表转换。 列表转换在库生成期间不会被执行。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| No | | 不在可执行模块生成后执行列表转换。 |
| 输出列表转换错误列表文件 | 选择是否在列表转换执行期间输出错误列表文件。 相当于列表转换程序的 -e 选项。 当 [Output assemble list file] 属性为 [No]，且 [Execute list converter] 的属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-e) |
| No | | 在列表转换执行期间不输出错误列表文件。 |
| 汇编列表信息输出 | 选择是否输出汇编列表信息到汇编表文件。 相当于汇编程序的 -ka 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| No(-nka) | | 不输出汇编列表信息到汇编列表文件。 |
| 符号列表输出 | 选择是否输出符号列表信息到汇编列表文件。 相当于汇编程序的 -ks 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | |
| | 默认值 | 常用选项的配置 |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes(-ks) |
| No | | 不输出符号列表信息到汇编列表文件。 |

| | | | | | |
|-------------|--|---|----------|--------------------|----|
| 交叉引用列表输出 | 选择是否输出交叉引用列表信息到汇编列表文件。 这对应于汇编器的 -kx 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | 常用选项的配置 | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-kx)</td> <td>输出交叉引用列表信息到汇编列表文件。</td> </tr> <tr> <td>No</td> <td>不输出交叉引用列表信息到汇编列表文件。</td> </tr> </tbody> </table> | Yes(-kx) | 输出交叉引用列表信息到汇编列表文件。 | No |
| Yes(-kx) | 输出交叉引用列表信息到汇编列表文件。 | | | | |
| No | 不输出交叉引用列表信息到汇编列表文件。 | | | | |
| 带格式馈送控制代码输出 | 选择是否输出格式馈送代码到表文件。 相当于汇编程序的 -lf 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | 常用选项的配置 | | | |
| | 如何改变 | 通过下拉列表选择。 | | | |
| | 限制 | <table border="1"> <tbody> <tr> <td>Yes(-lf)</td> <td>输出格式馈送代码到表文件。</td> </tr> <tr> <td>No</td> <td>不输出格式馈送代码到表文件。</td> </tr> </tbody> </table> | Yes(-lf) | 输出格式馈送代码到表文件。 | No |
| Yes(-lf) | 输出格式馈送代码到表文件。 | | | | |
| No | 不输出格式馈送代码到表文件。 | | | | |
| 1 行字符数目 | 指定表文件每行字符数。 相当于汇编程序的 -lw 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | 常用选项的配置 | | | |
| | 如何改变 | 直接在文本框中输入。 | | | |
| | 限制 | 72 ~ 2046 (十进制数) | | | |
| 1 页的行数 | 指定表文件每页行数。 如果是 0，将不会生成分页符。 相当于汇编程序的 -ll 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | 常用选项的配置 | | | |
| | 如何改变 | 直接在文本框中输入。 | | | |
| | 限制 | 0, 和 20 ~ 32767 (十进制数) | | | |
| Tab 宽度 | 指定列表文件的 tab 宽度 相当于汇编程序的 -lt 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | | | | |
| | 默认值 | 常用选项的配置 | | | |
| | 如何改变 | 直接在文本框中输入。 | | | |
| | 限制 | 0 ~ 8 (十进制数) | | | |

| | | |
|------|--|--|
| 头标题 | 指定汇编表文件的页头标题。 可以指定包含有双字节和单字节空格字符的字符串。 相当于汇编程序的 -lh 选项。 当 [Output assemble list file] 属性为 [No] 时，不显示该属性。 | |
| 默认值 | 常用选项的配置 | |
| 如何改变 | 直接在文本框中输入。 | |
| 限制 | 多达 60 个单字节字符（30 个双字节字符） | |

(5) [Others]

显示汇编的其它详细信息以及可修改的配置信息。

| | | |
|-----------------|---|----------------------------------|
| 源文件的 Kanji 字符编码 | 选择源文件的 Kanji 字符编码 相当于汇编程序的 -zs、-ze 和 -zn 选项。 | |
| 默认值 | 常用选项的配置 | |
| 如何改变 | 通过下拉列表选择。 | |
| 限制 | Shift_JIS(-zs) | 将源文件的 kanji 代码用 Shift_JIS 来解释。 |
| | EUC-JP(-ze) | 将源文件的 kanji 代码用 EUC-JP 来解释。 |
| | Unspecified(-zn) | 将源文件不包含 Kanji 代码在解释。 |
| 允许汇编 78K0 的宏 | 选择是否使由 78K0 单片机汇编程序生成的汇编程序源文件汇编可用。 相当于汇编程序的 -compati 选项。 | |
| 默认值 | 常用选项的配置 | |
| 如何改变 | 通过下拉列表选择。 | |
| 限制 | Yes(-compati) | 使对 78K0 微控制器汇编程序生成的汇编程序源文件汇编可用。 |
| | No | 使对 78K0 微控制器汇编程序生成的汇编程序源文件汇编不可用。 |
| 汇编之前执行的命令 | 指定汇编之前要执行的命令。 下面的宏变量是内嵌的宏。 %ProjectFolder%: 用工程文件夹的绝对路径替换。 %OutputFolder%: 用输出文件夹的绝对路径替换。 %OutputFile%: 用输出文件的绝对路径替换。 %InputFile%: 用将要汇编文件的绝对路径替换。 %AssembledFile%: 用正在汇编输出文件的绝对路径替换。 指定的命令显示为子属性。 | |
| 默认值 | 汇编之前执行的命令 [定义条目的数目] | |
| 如何改变 | 点击 [...] 按钮，通过文本编辑对话框来编辑。 对于子属性，可以用文本框直接输入。 | |
| 限制 | 多达 1023 个字符 设定最多可达 64 个条目。 | |

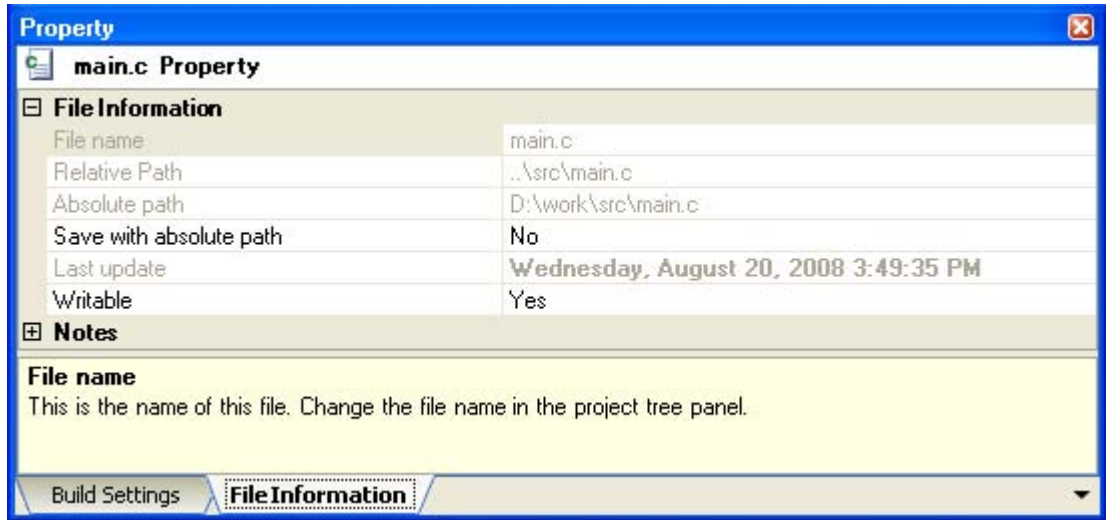
| | | |
|-----------|---|---|
| 汇编之后执行的命令 | <p>指定汇编之后要执行的命令。</p> <p>下面的宏变量是内嵌的宏。</p> <p>%ProjectFolder%: 用工程文件夹的绝对路径替换。</p> <p>%OutputFolder%: 用输出文件夹的绝对路径替换。</p> <p>%OutputFile%: 用输出文件的绝对路径替换。</p> <p>%InputFile%: 用将要汇编文件的绝对路径替换。</p> <p>%AssembledFile%: 用正在汇编输出文件的绝对路径替换。</p> <p>指定的命令显示为子属性。</p> | |
| | 默认值 | 汇编之后执行的命令 [定义条目的数目] |
| | 如何改变 | <p>点击 [...] 按钮，通过 文本编辑对话框 来编辑。</p> <p>对于子属性，可以用文本框直接输入。</p> |
| | 限制 | <p>多达 1023 个字符</p> <p>设定最多可达 64 个条目。</p> |
| 其他附加选项 | <p>输出需要加入的附加汇编选项。</p> <p>这里设置的选项加在汇编选项组的最后。</p> | |
| | 默认值 | <i>常用选项的配置</i> |
| | 如何改变 | 直接在文本框中输入或通过点击 [...] 按钮而显示 字符串输入对话框 的来编辑。 |
| | 限制 | 多达 259 个字符 |

[File Information] 标签

此标签显示下列各类可修改配置文件的详细信息。

- (1) [File Information]
- (2) [Notes]

图 A-19. 属性面板 : [File Information] 标签

**[各分项说明]****(1) [File Information]**

有关文件显示方面的详细信息说明以及可修改的配置信息。

| | | |
|-----------|--------------------------|----------------|
| 文件名称 | 文件名显示 在工程树面板上重命名文件名称。 | |
| | 默认值 | 文件名称 |
| | 如何改变 | 不允许改变。 |
| 相对路径 | 在工程文件夹中显示文件的相对路径。 | |
| | 默认值 | 在工程文件夹中文件的相对路径 |
| | 如何改变 | 不允许改变。 |
| 绝对路径 | 显示文件的绝对路径。 | |
| | 默认值 | 文件的绝对路径 |
| | 如何改变 | 不允许改变。 |
| 采用绝对路径来保存 | 选择是否要采用绝对路径保存文件位置。 | |
| | 默认值 | No |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes |
| | No | 采用相对路径保存文件位置 |

| | | |
|------|--|--|
| 最新升级 | 显示文件最新修改过的日期和时间。 | |
| | 默认值 | 文件升级日期和时间 |
| | 如何改变 | 不允许改变。 |
| 可写入 | 选择是否可写入文件 如果没有权限去改变文件属性，该属性将显示成灰色，表示不能修改属性。 | |
| | 默认值 | Yes (when the file is write enabled) 非 (当文件不可以写入) |
| | 如何改变 | 通过下拉列表选择。 |
| | 限制 | Yes 使文件可写入 No 使文件不可写入 |

(2) [Notes]

有关注意事项的详细信息及可修改的配置信息的说明。

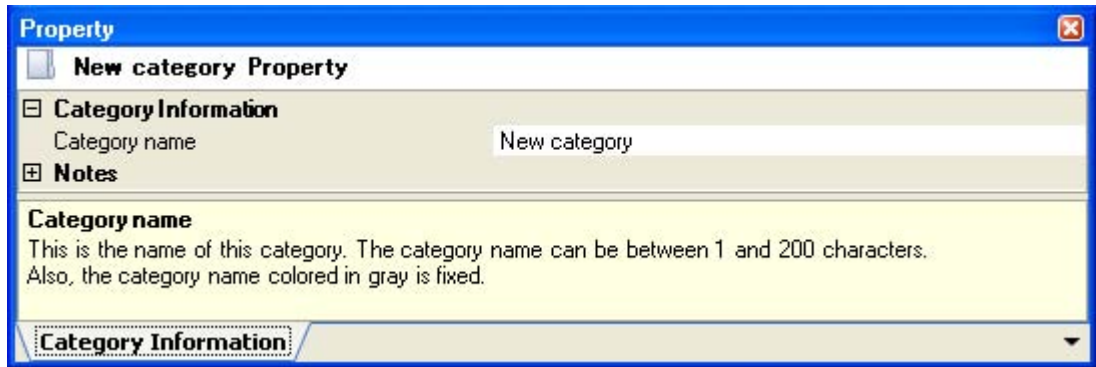
| | | |
|----|-----------------------------------|--|
| 备注 | 添加备注到文件 每项一行。 加入的备注作为子属性显示。 | |
| | 默认值 | 备注 [<i>number-of-characters</i>] |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 256 个字符 多达 256 条备注可以设定。 |

[Category Information] 标签

此标签显示用户添加分项的详细信息，根据下列可更改配置分为 File 节点 ,Build tool generated files 节点以及 Startup 节点。

- (1) [Category Information]
- (2) [Notes]

图 A-20. 属性面板 : [Category Information] 标签

**[各分项说明]****(1) [Category Information]**

显示有关分项方面的详细信息及更改过的配置。

| | | |
|------|--|-------------|
| 分项名称 | 设定分项文件的分项名称。 File 节点 ,Build tool generated files 节点 , 以及 Startup 节点的属性灰色显示说明不能修改属性。 | |
| | 默认值 | 文件分项名 |
| | 如何改变 | 直接在文本框中输入。 |
| | 限制 | 1 到 200 个字符 |

(2) [Notes]

有关注意事项的详细信息及可修改的配置信息的说明。

不显示 File 节点 ,Build tool generated files 节点 , 以及 Startup 节点的分项。

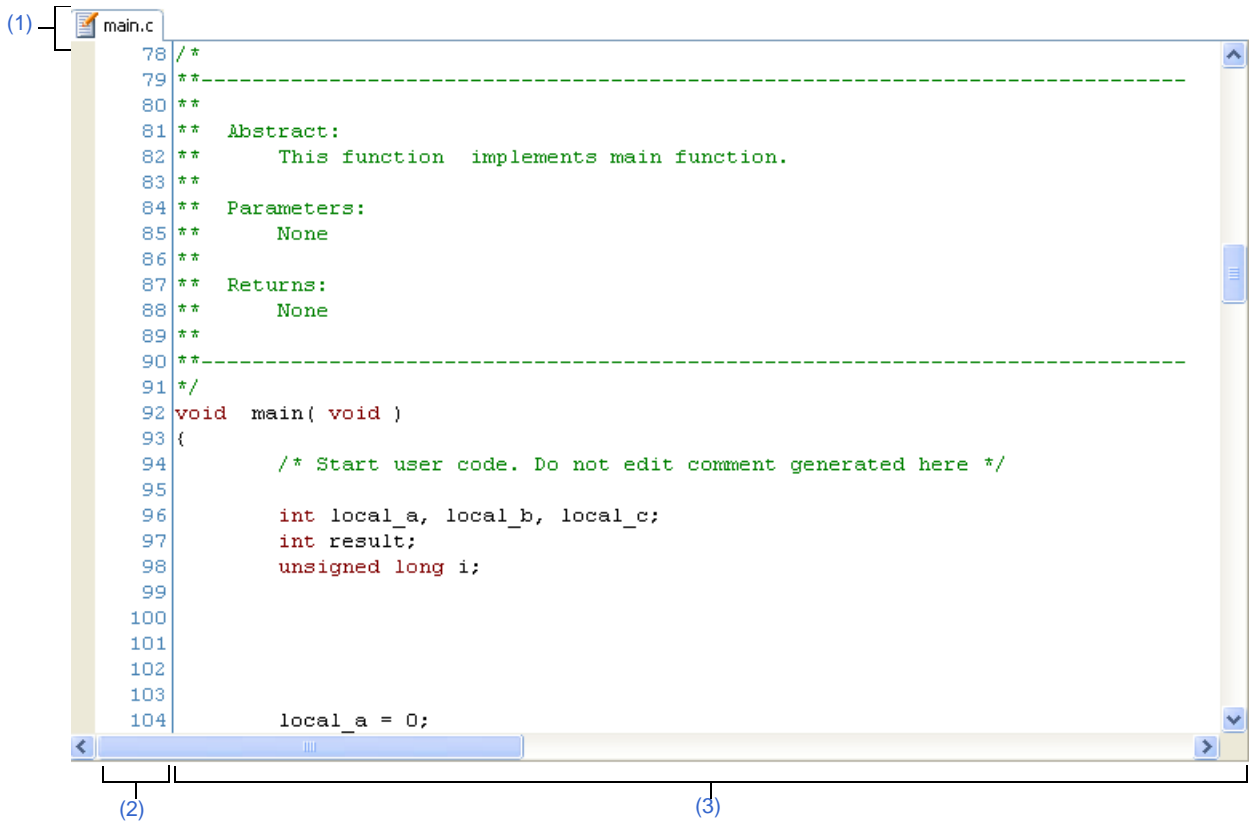
| | | |
|----|--|---|
| 备注 | 在文件的分项中加入备注。 每项一行。 加入的备注作为子属性显示。 | |
| | 默认值 | 备注 [number-of-characters] |
| | 如何改变 | 点击 [...] 按钮，通过 文本编辑对话框 来编辑。 对于子属性，可以用文本框直接输入。 |
| | 限制 | 多达 256 个字符 多达 256 条备注可以设定。 |

编辑器面板

该面板用于显示 / 编辑文本文件 / 源文件。
该面板可以重复打开（最多：100 个面板）。

备注 当打开源文件之前下载载入模块文件时显示消息。

图 A-21. 编辑器面板



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [[File] menu (only available for the Editor panel)]
- [[Edit] menu] (only available for the Editor panel)]
- [快捷菜单]

[如何打开]

- 在工程树面板上，双击文件。
- 在工程树面板上，选择源文件，然后通过快捷菜单选择 [Open]。
- 在工程树面板上，选择源文件，然后通过快捷菜单选择 [Open with Internal Editor...]
- 在工程树面板上，通过快捷菜单选择 [Add] >> [Add New File...]，然后创建文本文件 / 源文件。

[各部分说明]

(1) 标题栏

显示打开的文本文件 / 源文件名。
对显示在每个文件末尾的标记说明如下。

| 标记 | 说明 |
|--------|---------------|
| * | 编辑文件的内容改变。 |
| (不可编辑) | 打开的文本文件不能写入。 |
| ID 号 | 相同的文本文件被重复打开。 |

(2) 行号区

显示打开的文本文件 / 源文件的行号。

(3) 字符区

显示文本文件 / 源文件的字符和编辑文本文件 / 源文件的字符。
此区域有以下功能。

(a) 字符编辑

可以通过键盘输入字符。
多个快捷键可用于增强编辑功能。

(b) 文件监视器

下列监视功能应用于管理源文件。

- 如果不是 **CubeSuite** 改变了当前所显示文件的内容，则显示消息询问是否存储文件。可以选择存或不存。

备注 通过设置 [选项对话框](#) 下列条目可自定义。

- 显示字体
- 制表位间隔
- 控制字符（控制字符包含空格符号）的显示 / 隐藏 / 颜色
- 保留字 / 注释的颜色

[[File] menu (only available for the Editor panel)]

下列的条目是在编辑面板中的 **[File]** 菜单中独有的（其他条目通用于所有的面板）。

| | |
|---------------|---|
| 关闭 文件名 | 关闭当前正在编辑的编辑面板。 在面板的内容之前未保存过时，显示确认消息。 |
| 保存 文件名 | 覆盖当前编辑中的编辑器面板的内容。 注意：当文件没有被保存或文件不允许写入时，通过选择 [Save file name As...] 进行相同的操作。 |
| 保存 文件名为 ... | 从 另存为对话框 中打开最新保存的当前编辑器面板的编辑中的内容。 |
| Page Setup... | 打开窗口页面设置对话框。 |
| 打印 ... | 打开窗口的打印对话框打印当前正在编辑的编辑面板的内容。 |

[[Edit] menu] (only available for the Editor panel)]

下列的条目是在编辑器面板中的 **[Edit]** 菜单中独有的（其他条目全部无效）。

| | |
|------|--|
| Undo | 取消编辑面板之前的操作并恢复字符和光标位置（最大撤销次数为 100 次）。 |
| Redo | 取消之前在编辑面板的 [Undo] 操作并恢复字符和光标位置。 |
| Cut | 剪切选择的字符并拷贝到剪贴板。 |
| Copy | 拷贝选择的字符到剪贴板。 |

| | |
|------------|---|
| Paste | 插入（插入模式）或者改写（改写模式）拷贝到剪贴板的字符到光标位置。 当剪贴板的内容不能识别为字符时，操作无效。 |
| Delete | 在光标位置删除字符。 当存在选择区域时，所有选择区域内的字符都删除。 |
| Select All | 在当前正在编辑的文本文件中选择从开始到结尾所有字符。 |
| Find... | 打开 查找和替换对话框 通过 [Quick Search] 标签。 当存在选择区时，查找仅在选择区范围内进行。 |
| Replace... | 打开 查找和替换对话框 中的 [Quick Replace] 标签。 当存在选择区时，替换仅在选择区范围内进行。 |
| Move To... | 打开 跳转定位对话框 并移动光标到指定行。 |

[快捷菜单]

[Characters area/Line number area]

| | |
|----------------|---|
| 跳转到函数 | 该菜单总是失效。 |
| 返回到最后的鼠标指针所在位置 | 返回到鼠标指针转移之前的所在位置。 |
| 前进到下一个鼠标指针位置。 | 跳转到 [Back To Last Cursor Position] 操作之前的位置。 |
| Cut | 剪切选定的字符并粘贴到剪贴板。 |
| Copy | 复制选定的字符到剪贴板。 |
| Paste | 插入剪贴板的内容到光标所在位置。 |
| 在新建面板中打开 | 打开有相同内容的新建编辑面板作为当前编辑面板（最新打开编辑面板的标题栏显示文件名和 ID 号码）。 可以重复打开多达 100 个的编辑面板。 |

输出面板

该面板是用于显示从 build tool 中输出的消息或从[查找和替换对话框](#)中批量查找的结果。消息分别显示在按输出工具分类的标签上。

图 A-22. 输出面板



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [[File] menu (only available for the Output panel)]
- [[Edit] menu (only available for the Output panel)]
- [快捷菜单]

[如何打开]

- 通过 [View] 菜单，选择 [Output]。

[各部分说明]

(1) 消息区

显示消息并输出每个工具的查找结果。

在构建结果中 / 查找结果 (批量查找) 显示，每次构建时删除以前消息显示的新消息 / 查找已完成 (而不是 [All Messages] 标签)。

备注 可显示多达 50 万行消息。如果超过 1 行或者更多行消息被输出，那么超过的行会被删除，首先删除最前的行。

这条消息的颜色与下列提及的输出消息的类型有区分 (在 [General - Font and Color] 分项中设置的字符显色 / 背景颜色，此分项在[选项对话框](#)中)。

| 消息类型 | 示例 (默认) | | 说明 |
|----------------|---------|------|------|
| Normal message | AaBbCc | 字符颜色 | 黑色 |
| | | 背景颜色 | 白色 |
| Warning | AaBbCc | 字符颜色 | 蓝色 |
| | | 背景颜色 | 正常颜色 |
| Error message | AaBbCc | 字符颜色 | 红色 |
| | | 背景颜色 | 浅灰色 |

此区域有以下功能。

(a) 标签跳转

当输出消息被双击，或在光标越过消息按下 [Enter]，**编辑器面板**会出现且显示文件的目标行数。可以跳转到构建时从出错信息输出生成的源文件中的某行。

(b) 显示在线帮助

选中快捷菜单中 [Help for Message]，或在光标停在显示警告消息或出错消息的这行上按 [F1] 键，则关于该行消息的在线帮助会显示。

(c) 保存日志


在显示选定标签上的内容是通过选择 [Save Output - tab name As...] 项，从 [File] 菜单中打开**另存为对话框** (标签上的没有选中的消息将不保存) 可以保存到文本文件中 (*.TXT)。

(2) 标签选定区域

选择输出消息的标签。

标签显示如下：

| 标签名称 | 说明 |
|--------------|---------------------------------------|
| All Messages | 通过输出命令显示所有消息。(当执行快速构建除外) |
| 快速构建 | 通过运行快速构建从 build tool 中显示消息输出。 |
| Build Tool | 通过运行构建 / 重建 / 清除从 build tool 中显示消息输出。 |
| 查找和替换 | 在 查找和替换对话框 中显示批量结果。 |

注意事项 当新消息输出到未选定的标签上，标签不会自动改变。
如果这种情况的话， 添加到标签会产生新消息输出。

[[File] menu (only available for the Output panel)]

下列的条目是在输出面板中的 [File] 菜单中独有的 (其他条目通用于所有的面板)。

| | |
|------------------|--|
| 保存输出 - 标签名称 | 保存当前选中标签的内容到已保存过的文本文件中 (*.TXT)(见 "(c) 保存日志")。运行程序后当条目第一次选中，此操作与选择 [Save Output - tab name As...] 相同。 |
| 保存输出 - 标签名称为 ... | 打开 另存为对话框 保存当前选中的标签内容到指定的文本文件中 (*.TXT)(见 "(c) 保存日志")。 |

[[Edit] menu (only available for the Output panel)]

下列的条目是在输出面板中的 [Edit] 菜单中独有的 (其他条目全部无效)。

| | |
|------------|---|
| Copy | 复制选定的字符到剪切板。 |
| Select All | 选择此面板上所有的显示消息。 |
| Find... | 打开 查找和替换对话框 中的 [Quick Search] 标签。 |
| Replace... | 打开 查找和替换对话框 中的 [Whole Replace] 标签。 |

[快捷菜单]

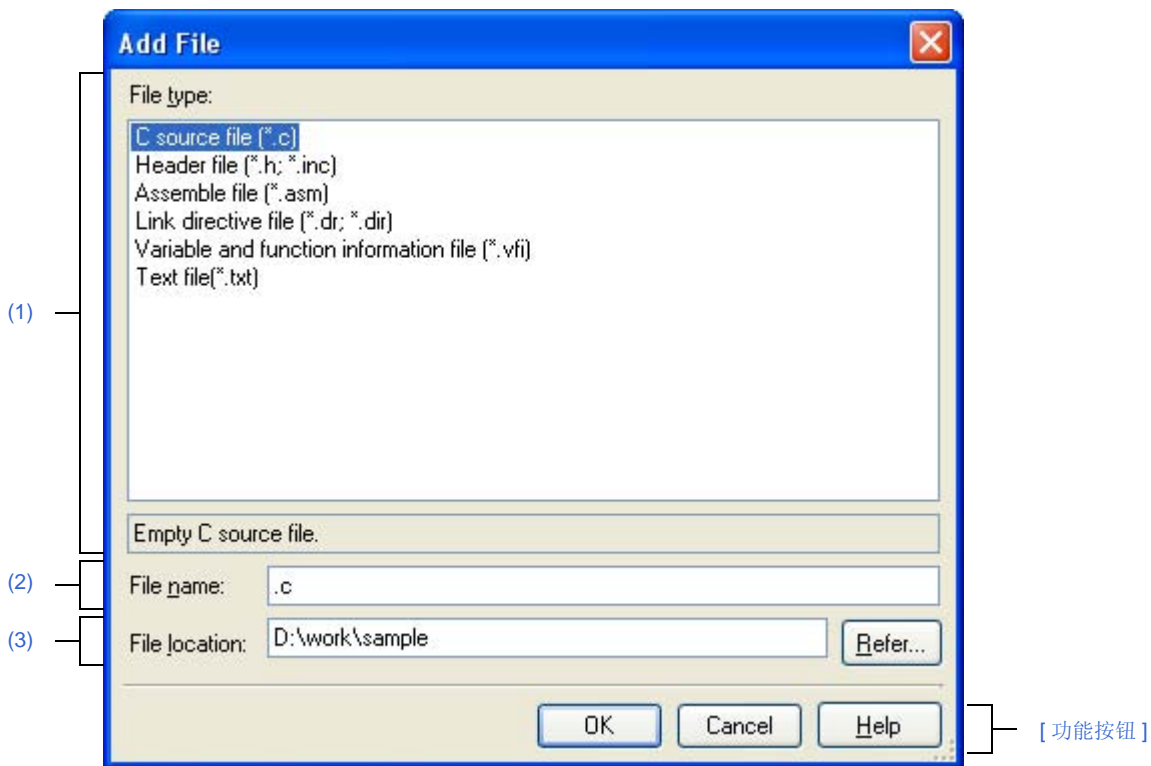
| | |
|------------|----------------|
| Copy | 复制选定的字符到剪切板。 |
| Select All | 选择此面板上所有的显示消息。 |

| | |
|----------------|--|
| Clear | 删除此面板上所有的显示消息。 |
| Stop Searching | 取消当前的查找操作。 当查找操作不发生，此命令无效。 |
| 消息帮助 | 显示关于当前光标消息的在线帮助。 注意：警告 / 出错消息的在线帮助。 |

添加文件对话框

该对话框用于创建新的文件并添加到工程中。

图 A-23. 添加文件对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [File] 菜单，选择 [Add] >> [Add File...]
- 在工程树面板上，选择 Project 节点、Subproject 节点、File 节点或 category 节点，并从快捷菜单中选择 [Add] >> [Add New File...]

[各部分说明]

(1) [File type] 区

选择创建的文件类型。

当文件类型选择时，说明在下一级对话框中显示。

文件类型如下所示。

- C 源文件 (*.c)
- 头文件 (*.h; *.inc)
- 汇编文件 (*.asm)
- 链接指令文件 (*.dr; *.dir)
- 变量和函数信息文件 (*.vfi)
- 文本文件 (*.txt)

(2) [File name] 区域

直接键入创建文件的文件名。

默认文件扩展名为“txt”。

备注 如果扩展名没有指定，则添加在 [File type] 区选定的那个。同时，如果不同于在 [File type] 区中选择的扩展名设定，则在 [File type] 区选择的扩展名将添加（例如，如果指定“aaa.txt”作为文件名并选择“C 源文件 (*.c)”作为文件类型，则文件名为“aaa.txt.c”）。

(3) [File location] 区

通过直接键入路径或通过 [Refer...] 按钮选择路径的方法来指定创建文件的位置。

文件的默认位置是工程文件夹路径。

(a) 按钮

| | |
|--------|----------------------------------|
| 引用 ... | 打开文件夹浏览对话框。 当选择文件夹时，添加路径到文字框。 |
|--------|----------------------------------|

备注 1. 当文字框左侧空白，则看作为指定的工程文件夹。

2. 当相对路径使用，则路径认作为是工程文件夹。

备注 可以在 [File name] 区域和 [File location] 区中键入路径名和文件名的字符总数多达 259 个。当输入受到限制时，则在 [File name] 区域的工具提示中显示下列消息。

| 消息 | 说明 |
|--|---|
| 包含路径的文件名称太长。文件名称长度控制在 259 个字符内。 | 包含路径的文件名超过 259 字符。 |
| 指定路径中包含的文件夹不存在。 | 路径中包含不存在的文件夹。 |
| 文件名称和路径名称无效。The following characters cannot be used: ?, /, :, *, ?, “, <, >, | The file name with the invalid path is designated. 字符 \、/、:、*、"、<、> 和 不能用作文件名和文件夹名。 |

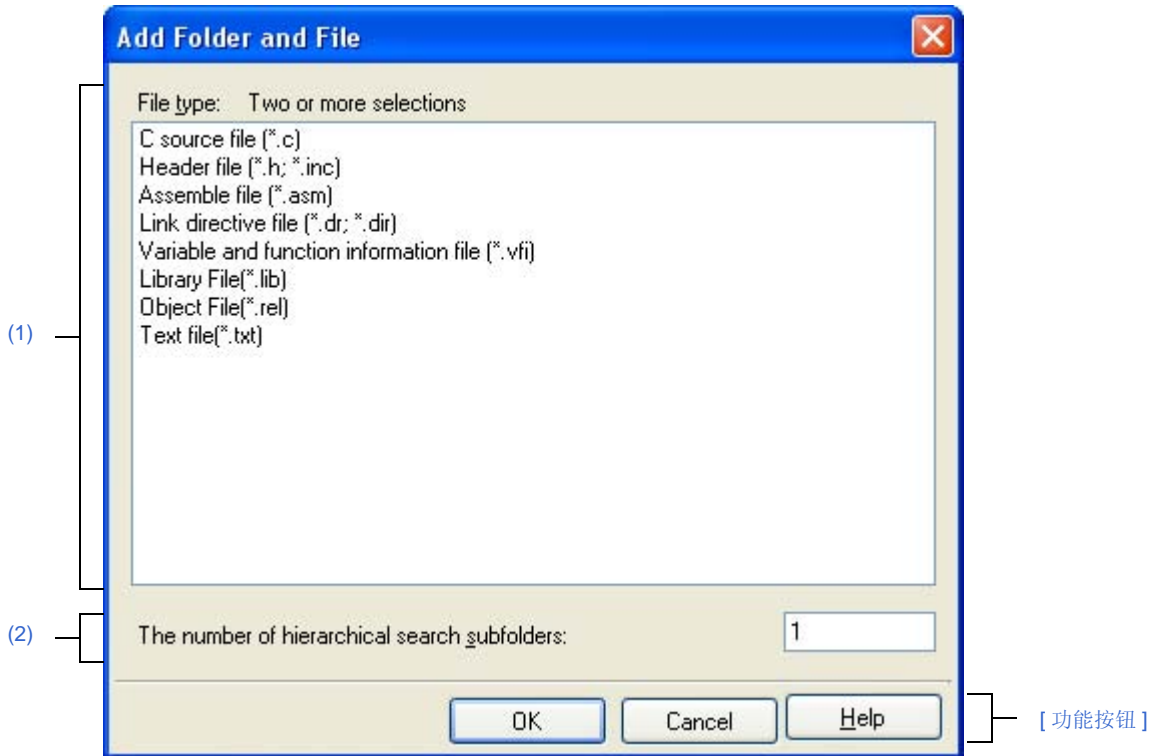
[功能按钮]

| 按钮 | 功能 |
|--------|---------------------------------------|
| OK | 创建带文件名的文件，并添加到工程中，使用编辑器面板打开。然后关闭此对话框。 |
| Cancel | 不创建文件并关闭此对话框。 |
| Help | 显示此对话框的在线帮助。 |

添加文件夹和文件对话框

该对话框用于添加现有文件和各级文件夹到工程中。
该文件夹添加作为分项。

图 A-24. 添加文件夹和文件对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 从资源管理器中拖动文件夹，并放在工程树面板上。

[各部分说明]

(1) [File type] 区

选择文件类型将其添加到工程。

使用左键在保持 [Ctrl 或 Shift] 键按下的状态下选取多个文件类型，可以实现多重类型的选择。

如果没有选择内容，则假定所有类型都已选择。

显示的文件类型如下所示。

- C 源文件 (*.c)
- 头文件 (*.h; *.inc)
- 汇编文件 (*.asm)
- 链接指令文件 (*.dr; *.dir)
- 变量和函数信息文件 (*.vfi)
- 库文件 (*.lib)
- 目标文件 (*.rel)
- 文本文件 (*.txt)

(2) [Subfolder level to search] 区

直接键入添加到工程的子文件夹的级别数。

默认为“1”。

备注 允许上限为 10 的十进制数。当输入受到限制时，则在工具提示中显示下列消息。

| 消息 | 说明 |
|---------------------|--------------------|
| 不能设定小于 0 或大于 10 的值。 | 大于 10 的子文件夹级别数的设定。 |
| 用十进制设定。 | 非十进制的数或字符串的设定。 |

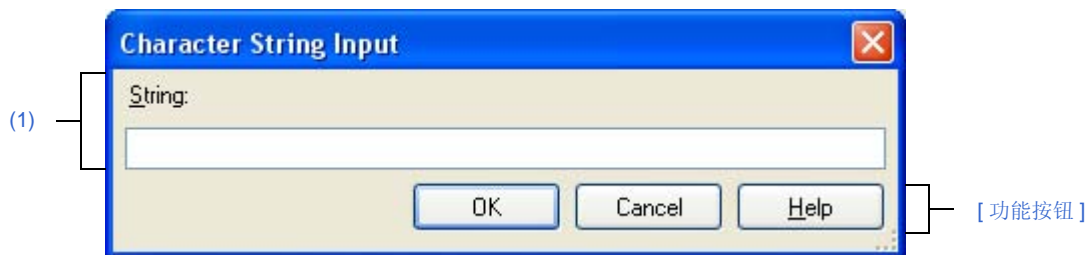
[功能按钮]

| 按钮 | 功能 |
|--------|-------------------------------|
| OK | 拖放文件夹和文件夹内的文件，并添加到工程，然后关闭对话框。 |
| Cancel | 不添加文件夹和文件，然后关闭此对话框。 |
| Help | 显示此对话框的在线帮助。 |

字符串输入对话框

该对话框用于输入和编辑一行中的字符。

图 A-25. 字符串输入对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过属性面板，选择下列属性，然后点击 [...] 按钮。
 - 通过 [Common Options] 标签的 [Others] 分项中的 [Output folder for hex file]。
 - 通过 [Compile Options] 标签的 [Others] 分项中的 [Output folder]。
 - 通过 [Assemble Options] 标签的 [Others] 分项中的 [Output folder]。
 - 通过 [Link Options] 标签的 [Stack] 分项中的 [Area name]，和 [Others] 分项中的 [Other additional options]。
 - 通过 [Object Convert Options] 标签的 [Others] 分项中的 [Output folder]。
 - 通过 [Individual Compile Options] 标签的 [Others] 分项中的 [Output folder]。
 - 通过 [Individual Assemble Options] 标签的 [Others] 分项中的 [Output folder]。
- 在 [General - External Tools] 分项中，其属于选项对话框，在新注册区中检查 [Require options at start-up]。当外部工具通过 [Tool] 菜单启动时，对话框自动打开。

[各部分说明]

(1) [String] 区

在一行中输入字符。

默认情况下，打开带编辑框的对话框会影响已选定调用对话框的当前属性值。

不允许链接中断。

备注 最多输入字符可多达 32767 个字符。当输入受到限制时，则在工具提示中显示下列消息。

| 消息 | 说明 |
|------------------------------|------------------------|
| 大于属性中限制的最大数，则不能设定已调用此对话框的字符。 | 字符数超过了已调用该对话框属性限制的最大数。 |

[功能按钮]

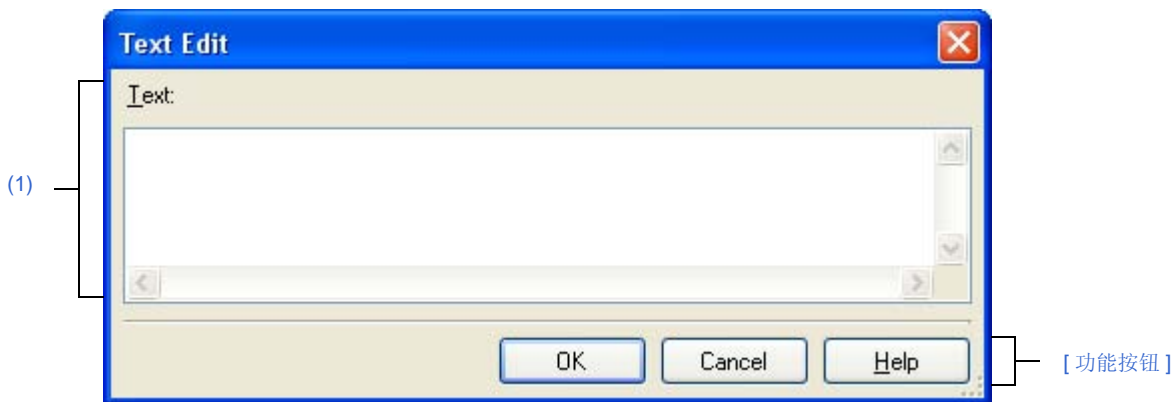
| 按钮 | 功能 |
|----|-----------------------------|
| OK | 影响已输入字符到调用该对话框的属性中，然后关闭对话框。 |

| 按钮 | 功能 |
|--------|-------------------------------|
| Cancel | 不会影响已输入字符在调用该对话框的属性中，然后关闭对话框。 |
| Help | 显示此对话框的在线帮助。 |

文本编辑对话框

该对话框用于输入和编辑多中的文本。

图 A-26. 文本编辑对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过属性面板，选择下列属性，然后点击 [...] 按钮。
 - 通过 [Common Options] 标签， [Frequently Used Options(for Compile)] 分项的 [Macro definition]， [Frequently Used Options(for Compile)] 分项的 [Macro definition]， [Frequently Used Options(for Link)] 分项的 [Using libraries]， [Notes] 分项的 [Memo]， [Others] 分项的 [Commands executed before build processing] 和 [Commands executed after build processing]。
 - 通过 [Compile Options] 标签， [Preprocess] 分项的 [Macro definition] 和 [Macro undefinition]， [Others] 分项的 [Commands executed before compile processing] 和 [Commands executed after compile processing]。
 - 通过 [Assemble Options] 标签， [Preprocess] 分项的 [Macro definition]， [Others] 分项的 [Commands executed before assemble processing] 和 [Commands executed after assemble processing]。
 - 通过 [Link Options] 标签， [Library] 分项的 [Using libraries]， [Others] 分项的 [Commands executed before link processing] 和 [Commands executed after link processing]。
 - 通过 [Object Convert Options] 标签， [Others] 分项的 [Commands executed before object convert processing] 和 [Commands executed after object convert processing]。
 - 通过 [Create Library Options] 标签， [Others] 分项的 [Commands executed before making library] 和 [Commands executed after making library]。
 - 通过 [Individual Compile Options] 标签， [Preprocess] 分项的 [Macro definition] 和 [Macro undefinition]， [Others] 分项的 [Commands executed before compile processing] 和 [Commands executed after compile processing]。
 - 通过 [Individual Assemble Options] 标签， [Preprocess] 分项的 [Macro definition]， [Others] 分项的 [Commands executed before assemble] 和 [Commands executed after assemble]。

[各部分说明]

(1) [Text] 区

输入和编辑多行文本。

默认情况下，打开带编辑框的对话框会影响已选定调用对话框的当前属性值。

备注 允许多达 65535 行和 65535 个字符。当输入受到限制时，则在工具提示中显示下列消息。

| 消息 | 说明 |
|---|------------------------|
| 大于属性中限制的最大数，则不能设定已调用此对话框的字符。显示当前的字符数在超限行开头的括号内。 | 字符数超过了已调用该对话框属性限制的最大数。 |

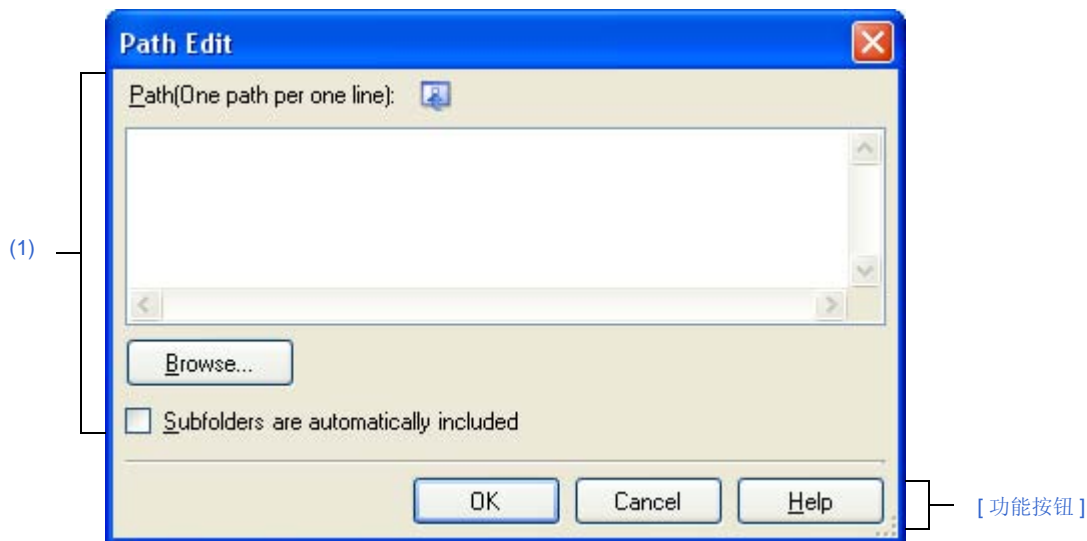
[功能按钮]

| 按钮 | 功能 |
|--------|--------------------------|
| OK | 打开和关闭对话框会影响已输入在文本框内的文本。 |
| Cancel | 打开和关闭该对话框不影响已输入在文本框内的文本。 |
| Help | 显示此对话框的在线帮助。 |

路径编辑对话框

该对话框用于编辑和添加路径。

图 A-27. 路径编辑对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过属性面板，选择下列属性，然后点击 [...] 按钮。
 - 通过 [Common Options] 标签， [Frequently Used Options(for Compile)] 分项的 [Additional include paths]， [Frequently Used Options(for Compile)] 分项的 [Additional include paths]，和 [Frequently Used Options(for Link)] 分项的 [Additional library paths]。
 - 通过 [Compile Options] 标签， [Preprocess] 分项的 [Additional include paths]。
 - 通过 [Assemble Options] 标签， [Preprocess] 分项的 [Additional include paths]。
 - 通过 [Link Options] 标签， [Library] 分项的 [Additional include paths]。
 - 通过 [Individual Compile Options] 标签， [Preprocess] 分项的 [Additional include paths]。
 - 通过 [Individual Assemble Options] 标签， [Preprocess] 分项的 [Additional include paths]。

[各部分说明]

(1) 路径编辑区

编辑或添加路径。

(a) [Path(One path per one line)]

编辑或添加路径通过直接输入路径。

路径可以多行指定。在行中指定路径。

默认时，在该区域文本框内容受到已打开该对话框的影响。

添加路径可以使用下列方法之一。

- 点击 [Browse...] 按钮，并在文件夹浏览对话框中选择文件夹。
- 使用例如资源管理器之类的工具来拖放文件夹。

备注 允许最多达 10000 行。可用的最大字符数受到 Windows 操作系统规定的限制。当输入受到限制时，则在工具提示中显示下列消息。

| 消息 | 说明 |
|---|--|
| 指定路径。 | 字段为空。 |
| 路径过长，指定路径的字符数小于或等于已调用对话框中属性的最大限定数。 | 包含路径的文件名超过了在原来路径定义中的字符数限制。 |
| 指定路径中包含的文件夹不存在。 | 路径中包含不存在的文件夹。 |
| 文件名称和路径名称无效。The following characters cannot be used: ?, /, :, *, ?, " , <, >, | The file name with the invalid path is designated. 字符 \, /, :, *, " , <, >, , 不能作为文件名和文件夹名使用。 |
| 如果指定的路径数或文件数超过了最大数，则不能设定调用方的行数。 | 输入的路径或文件的数目超过了由调用者设定的路径或文件的最大数。 |

(b) 按钮

| | |
|-----------|---|
| Browse... | 打开文件夹浏览对话框。 当文件夹选定后，路径添加到 [Path(One path per one line)]。 |
|-----------|---|

(c) [Subfolders are automatically included]

在核对该复选框后，通过 [Browse...] 按钮指定路径，并添加路径到 [Path(One path per one line)] 中包含子文件夹（多达 5 层）。

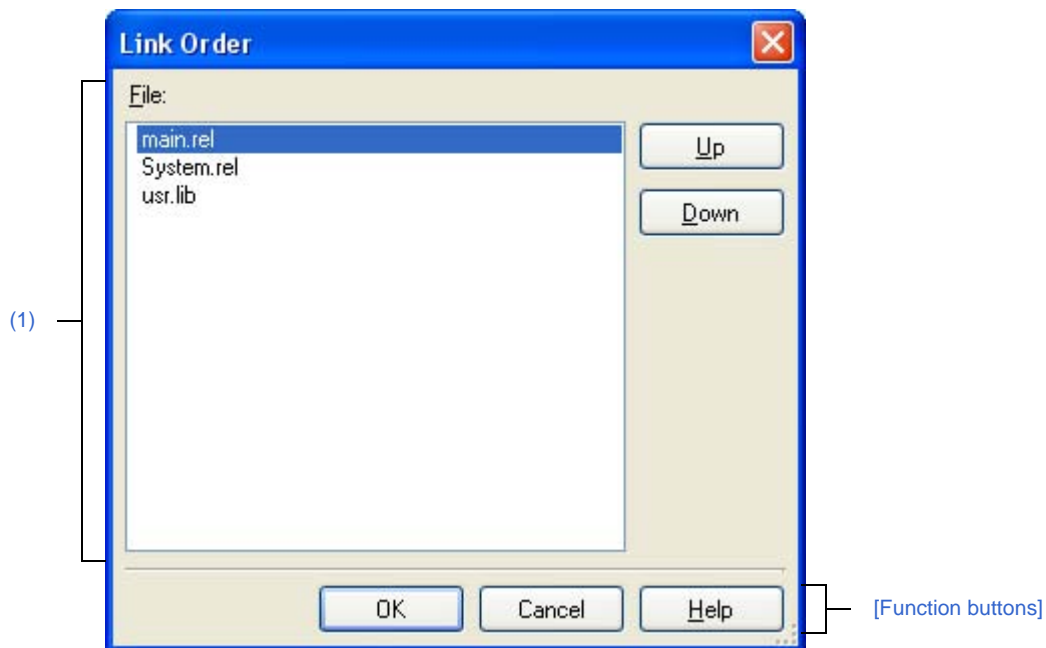
[功能按钮]

| 按钮 | 功能 |
|--------|--------------------------------|
| OK | 调用该对话框然后关闭对话框会对已输入在属性中的路径有影响。 |
| Cancel | 调用该对话框然后关闭对话框不会对已输入在属性中的路径有影响。 |
| Help | 显示此对话框的在线帮助。 |

链接顺序对话框

该对话框用于显示 输入到链接器的目标模块文件和库文件 并 配置它们的链接顺序。

图 A-28. 链接顺序对话框



下列条目说明如下。

- [How to open]
- [各部分说明]
- [Function buttons]

[How to open]

- 在工程树面板上, 选择 Build tool 节点, 然后从快捷菜单中选择 [Set Link Order...]

[各部分说明]

(1) 文件列表显示区域

显示输入到链接器进行链接的文件列表。

(a) [File]

根据输入链接器的顺序, 显示下列文件名称列表。

- 在选定的主工程或子工程中, 通过已注册的源文件生成目标模块文件。
- 在选定的主工程或子工程中, 目标模块文件直接添加到工程树中。
- 在选定的主工程或子工程中, 库文件直接添加到工程树中。

在默认时, 链接器的输入顺序是在工程中的注册顺序。

通过改变文件显示的顺序改变文件输入的顺序。

使用 [Up] 和 [Down] 按钮或拖放文件名来改变显示顺序。

备注 1.

当鼠标指针移过文件名时, 文件的绝对路径会弹出显示。

2. 目标模块文件从最新添加的源文件中生成且最新添加的目标模块文件添加到模块文件列表的末尾。最新添加的库文件添加到列表的末尾。
3. 当文件拖放时, 相互紧邻的多个文件能一起选中。

(b) 按钮

| | |
|----|-----------|
| 向上 | 移动选中文件向上。 |
| 向下 | 移动选中文件向下。 |

备注 注意：上面的按钮在一些文件没有选择时不可用。

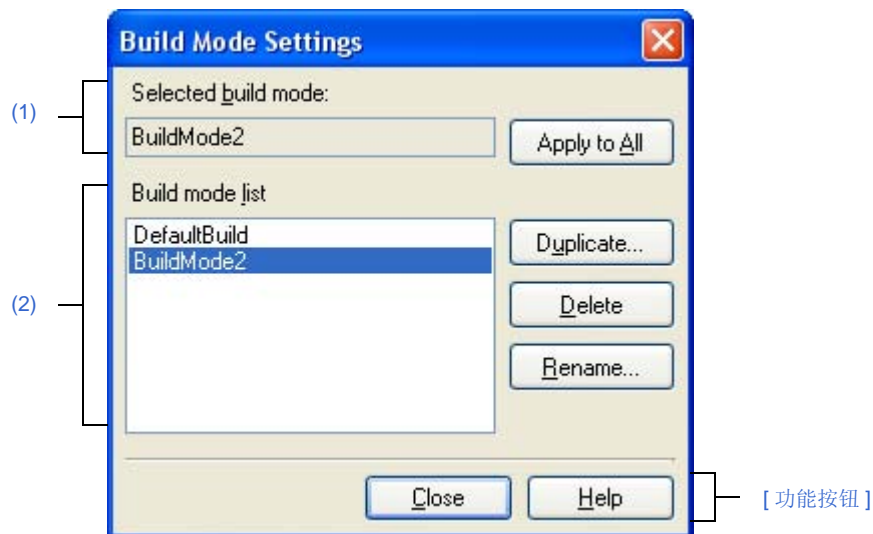
[Function buttons]

| 按钮 | 功能 |
|--------|--|
| OK | 在链接器中设置文件输入顺序作为在 文件列表显示区域 上的显示顺序并关闭此对话框。 |
| Cancel | 取消链接顺序设置并关闭此对话框。 |
| Help | 显示此对话框的在线帮助。 |

构建模式设置对话框

该对话框用于添加和删除构建模式和以批处理的方式设置当前构建模式。

图 A-29. 构建模式设置对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Build] 菜单，选择 [Batch Mode Settings...]

[各部分说明]

(1) [Selected build mode] 区

显示在 [Build mode list] 区中选择的构建模式。

(a) 按钮

| | |
|--------------|------------------------------------|
| Apply to All | 设置当前打开工程的主工程和所有子工程的构建模式到当前显示的构建模式。 |
|--------------|------------------------------------|

(2) [Build mode list] 区

在列表中显示所有存在于当前打开工程（主工程和子工程）内的构建模式。

默认选择已选择工程的当前构建模式。

用 “*” 标记显示只在主工程和子工程的部分中出现的构建模式。

注意：“DefaultBuild” 是默认构建模式并总是显示在顶部。

(a) 按钮

| | |
|-----------|--|
| 复制 ... | 复制选择的构建模式。 打开 字符串输入对话框 ，复制已输入名称的构建模式，并添加到当前打开工程的主工程和所有子工程中。 当主工程或子工程中不存在带有“*”标记的构建模式和复制构建模式时，则复制 DefaultBuild。 可添加多达 20 个构建模式。 |
| Delete | 删除选择的构建模式。 注意：不能删除 DefaultBuild。 |
| Rename... | 重命名选择的构建模式。 打开 字符串输入对话框 ，通过输入名称重命名构建模式。 |

注意事项 复制或重命名构建模式时，不能使用已有的构建模式名称。

备注 1. 构建模式名称中至多可以使用 127 个字符。当输入受到限制时，则在工具提示中显示下列消息。

| 消息 | 说明 |
|---|---|
| A build mode with the same name already exists. | 键入的构建模式名已存在。 |
| More than 127 characters cannot be specified. | 构建模式名称过长（多于 128 个字符）。 |
| The build mode name is invalid. The following characters cannot be used: ?, /, :, *, ?, " , <, >, | 键入了无效的构建模式名称。文件夹名称中不能包含字符 (\, /, :, *, ?, " , <, >,)。 |

2. 可添加多达 20 个构建模式。当输入受到限制时，则在工具提示中显示下列消息。

| 消息 | 说明 |
|---|----------------|
| The maximum number of build modes that can be set per project/subproject is 20. | 构建模式的数量超过了 20。 |

[功能按钮]

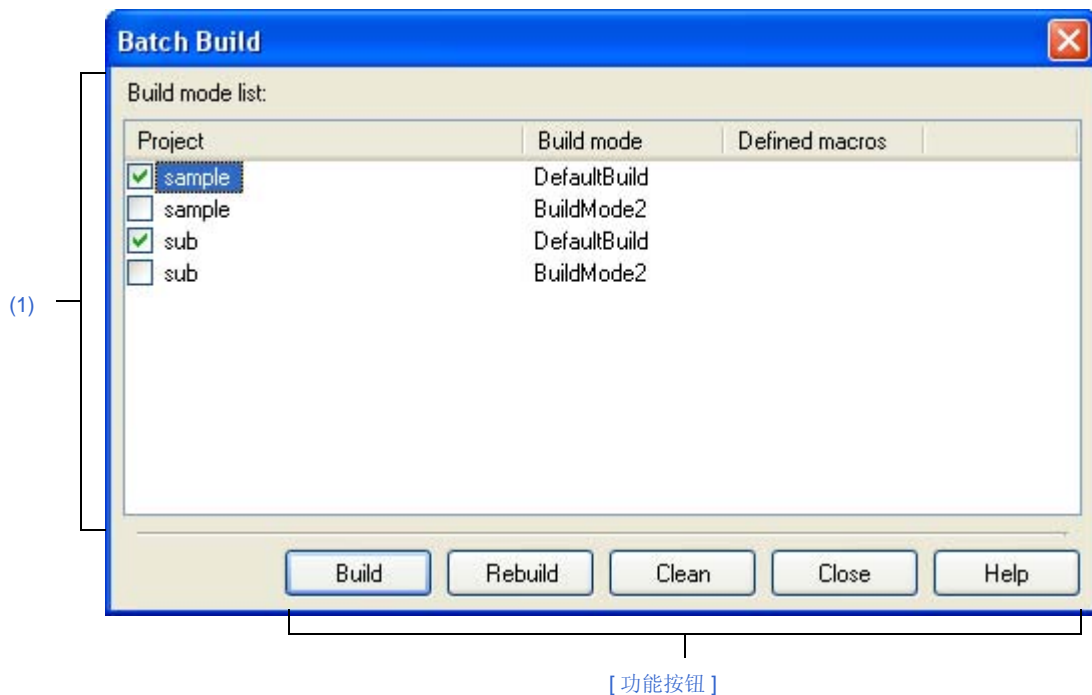
| 按钮 | 功能 |
|-------|--------------|
| Close | 关闭此对话框。 |
| Help | 显示此对话框的在线帮助。 |

批量构建对话框

该对话框被用于批量方式构建、重新构建和清理处理每个存在构建模式的工程（主工程和子工程）。

备注 批量构建的顺序按照工程的构建顺序 - 子工程在主工程之前。
在主工程或子工程中不止一个的构建模式选定时，则构建所有选定的构建模式，然后构建下一个子工程或主工程。

图 A-30. 批量构建对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Build] 菜单，选择 [Batch Build...]

[各部分说明]

(1) [Build mode list] 区

显示当前打开工程的主工程和子工程名的组合列表，以及它们所有的构建模式和定义宏。

(a) [Project]

显示当前打开工程的主工程和子工程。

选择要构建的主工程和子工程组合并选择构建模式。

在工程创建后第一次打开该对话框时，所有复选框处于未选状态。通过第二次，保留了以前的设置。

(b) [Build mode]

显示主工程和子工程中存在的构建模式。

(c) [Defined macros]

显示用 “[” 分隔的已定义的宏，在 [\[Compile Options\]](#) 标签和 [\[Assemble Options\]](#) 标签在属性面板中，配置主工程和子工程组合以及它们的构建模式。

注意：在编译选项中定义的宏出现在汇编选项中定义的宏之前，且它们之间用 “，” 分隔。

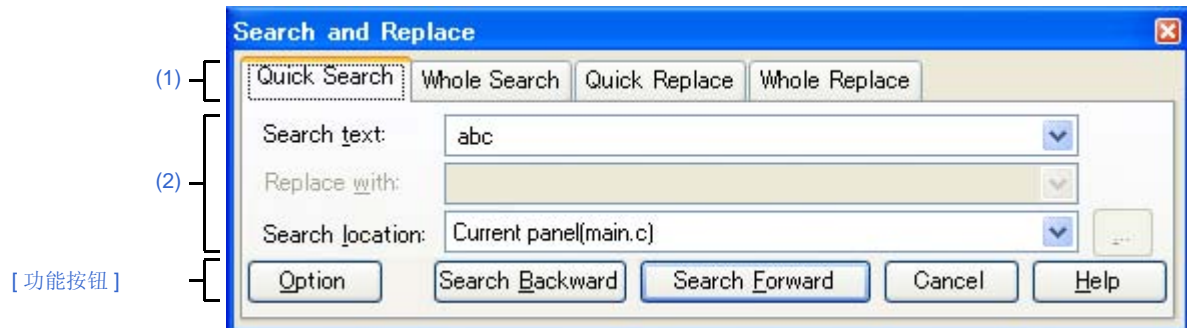
[功能按钮]

| 按钮 | 功能 |
|-------|--|
| 构建 | 关闭该对话框并对选定的有各自构建模式的工程执行批量构建。构建的执行结果显示在 输出面板 上。 在批量构建完成后，构建模式配置恢复到对话框打开之前的状态。 注意：该按钮在当一些工程没有选定时不可用。 |
| 重新构建 | 关闭该对话框并对选定的有各自构建模式的功能执行批量重新构建。重新构建的执行结果显示在 输出面板 上。 在批量重新构建完成后，构建模式的配置恢复到该对话框打开之前的状态。 注意：该按钮在当一些工程没有选定时不可用。 |
| 清除 | 关闭该对话框并删除内含各自的构建模式设置的选定工程。清除的执行结果显示在 输出面板 上。 在清除完成后，构建模式配置恢复到该对话框打开之前的状态。 注意：该按钮在当一些工程没有选定时不可用。 |
| Close | 关闭此对话框。 |
| Help | 显示此对话框的在线帮助。 |

查找和替换对话框

此对话框用于查找和替换指定字符。

图 A-31. 查找和替换对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Edit] 菜单, 选择 [Search...]
- 通过 [Edit] 菜单, 选择 [Replace...]

[各部分说明]

(1) 标签选定区域

当选择标签时, 切换查找 / 替换。
此对话框包括以下标签。

- [Quick Search] 标签
- [Whole Search] 标签
- [Quick Replace] 标签
- [Whole Replace] 标签

(2) 查找 / 替换规则设置区域

设置查找 / 替换的详细规则。
内容 / 如何设定的详情请参阅相关标签的说明。

[功能按钮]

执行查找 / 替换的按钮。
详情请参阅相关按钮的说明。

[Quick Search] 标签

此标签查找指定的字符并且移动光标从已查找的位置到正选定的位置。

注意：如果从面板访问 **查找和替换对话框** 而不是从 **输出面板 / 编辑面板** 访问时此标签不可用。

图 A-32. 查找和替换对话框：[Quick Search] 标签



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Edit] 菜单，选择 [Search...]
- 通过 [Edit] 菜单，选择 [Replace...]

[各部分说明]

(1) [Search text]

查找指定字符。

可以直接在文本框内输入字符（最大字符数：1024）或从下拉列表中的选择历史输入记录（最大历史记录数：10）。

如果通过面板已打开的对话框中选定字符，所选字符显示为默认。

如果通过编辑面板已打开的对话框，单词（变量 / 函数）在光标位置显示为默认。

(2) [Replace with]

此条目不可用。

(3) [Search location]

指定查找位置。

通过下拉列表选择下列条目之一。

| 条目 | 操作 |
|--------------|--|
| 选定区域 | 在最近一次活动且查找有效的面板中查找选定内容。 如果上一次激活的面板中没有字符，或者不能使用该面板，则此项目将不可用。 |
| 当前面板 (面板名称) | 在最近活动且能查找的面板中查找。 如果最近活动面板不能查找或面板不存在，则此条目将不可用。 |

备注 在下拉列表中记录的输入多达 10 条历史条目。

(4) [Option] 区

指定下列选项作为查找规则。

点击 [Option] 按钮，此区域就会显现（默认不显现）。

(a) [Search criteria]

通过下拉列表选择下列条目之一。

| 条目 | 操作 |
|------------|--------------------------|
| Plain text | 在 [Search text] 中查找指定字符。 |
| Wild-card | 使用下列通配符查找。 |
| | * 任意字符串 |
| | ? 任意字符。 |

(b) [File type]

此条目不可用。

(c) [Case-sensitive]

| | |
|-------------------------------------|-----------------------|
| <input checked="" type="checkbox"/> | 查找指定字符串（区分大小写）。 |
| <input type="checkbox"/> | 查找指定字符串 - 不区分大小写（默认）。 |

(d) [Word by word]

| | |
|-------------------------------------|---------------|
| <input checked="" type="checkbox"/> | 查找指定完全匹配的单词。 |
| <input type="checkbox"/> | 查找至少一个单词（默认）。 |

(e) [Open file before replacing]

此条目不可用。

[功能按钮]

| 按钮 | 功能 |
|------|--|
| 选项 | 在此标签里切换 [Option] 区的显示 / 隐藏状态。 |
| 向后查找 | 从当前光标的位置到文件顶部按指定的规则进行查找。选择已查找到的字符串并移动光标。 |

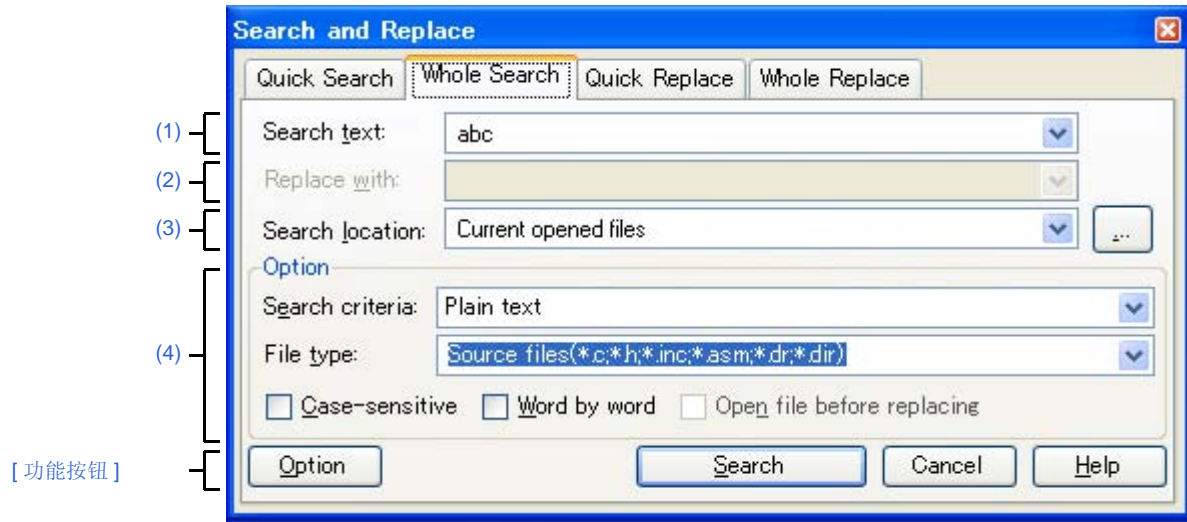
| 按钮 | 功能 |
|----------------|---|
| Search Forward | 从当前光标的位置到文件结束按照指定的规则进行查找。选择已查找到的字符串并移动光标 ^注 。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Help | 显示此对话框的在线帮助 |

注 如果指定的字符串没有查找到，会在主窗口的状态栏上显示 "Can not found Search text."。

[Whole Search] 标签

在此标签中指定的字符串进行批量的查找并查找结果显示在输出面板中。
通过双击查找结果，输出面板用于跳过相关的位置。

图 A-33. 查找和替换对话框：[Whole Search] 标签



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Edit] 菜单，选择 [Search...]
- 通过 [Edit] 菜单，选择 [Replace...]

[各部分说明]**(1) [Search text]**

查找指定字符。

可以直接在文本框内输入字符（最大字符数：1024）或从下拉列表中的选择历史输入记录（最大历史记录数：10）。

如果通过面板已打开的对话框中选定字符，所选字符显示为默认。

如果通过编辑面板已打开的对话框，单词（变量 / 函数）在光标位置显示为默认。

(2) [Replace with]

此条目不可用。

(3) [Search location]

指定查找位置。

通过下拉列表无论是选择下列条目之一还是通过键盘直接输入文件位置（最大数：10）。

| 条目 | 操作 |
|---------|---|
| 当前打开的文件 | 在所有打开着的编辑面板中查找。 如果在编辑面板中没有文件已打开，此项目不可用。 |
| 活动工程 | 查找包含在当前活动的工程中的文本文件。 当指定 [File type] 时，只查找指定的类型。 注意：当前工程不存在时，此条目不可用。 |
| 主工程及子工程 | 查找包含在主工程和子工程中的文本文件。 当指定 [File type] 时，只查找指定的类型。 注意：当前工程不存在时，此条目不可用。 |
| 文件夹名称 | 通过直接输入（最多字符数：259）路径（工程文件夹用的是相对路径）在指定文件夹中的文本文件内部查找，或打开 文件夹浏览对话框 通过在此区域单击 [...] 按钮来查找。如果文件夹没有指定，工程文件夹名称将会以默认文件夹显示在 "()" 中（如果工程不存在，当前用户文档文件夹会显示）。 当指定 [File type] 时，只查找指定的类型。 |

备注 在下拉列表中记录的输入多达 10 条历史条目。

(4) [Option] 区

点击 [Option] 按钮，此区域就会显现（默认不显现）。

指定下列选项作为查找规则。

(a) [Search criteria]

通过下拉列表选择下列条目之一。

| 条目 | 操作 |
|------------|--------------------------|
| Plain text | 在 [Search text] 中查找指定字符。 |
| Wild-card | 使用下列通配符查找。 |
| | * 任意字符串 |
| | ? 任意字符。 |

(b) [File type]

查找指定的文件类型。

通过下拉列表选择下列条目之一。

| 条目 | 操作 |
|----------------------------------|---------------|
| 源文件 (<i>扩展名</i> ^{注意}) | 查找的文件限定为源文件。 |
| *.txt | 查找的文件限定为文本文件。 |
| ** | 查找所有的文件。 |

注 显示已加入到 [工程树面板](#) 的源文件的扩展名。

注意： 通过在文本框内直接输入文件名（最大字符数：1024），限制查找规则来进行查找操作。

如果在这种情况下，能使用通配符 "*" 以及用 ";" 来分隔多个文件名称。

备注 在下拉列表中记录的输入多达 10 条历史条目。

(c) [Case-sensitive]

| | |
|-------------------------------------|-----------------------|
| <input checked="" type="checkbox"/> | 查找指定字符串（区分大小写）。 |
| <input type="checkbox"/> | 查找指定字符串 - 不区分大小写（默认）。 |

(d) [Word by word]

| | |
|-------------------------------------|---------------|
| <input checked="" type="checkbox"/> | 查找指定完全匹配的单词。 |
| <input type="checkbox"/> | 查找至少一个单词（默认）。 |

(e) [Open file before replacing]

此条目不可用。

[功能按钮]

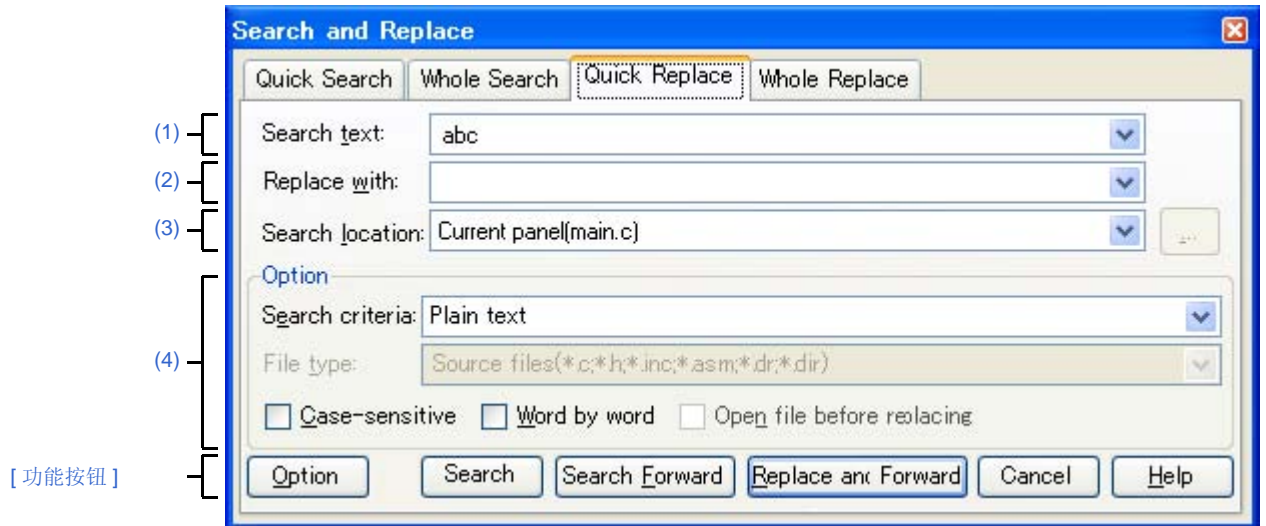
| 按钮 | 功能 |
|--------|------------------------------------|
| 选项 | 在此标签里切换 [Option] 区的显示 / 隐藏状态。 |
| Search | 批量的查找指定符合指定规则的字符并在输出面板中的列表中显示查找结果。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Help | 显示此对话框的在线帮助。 |

[Quick Replace] 标签

在此标签里指定字符的查找已完成，然后它们正替换成要替换的字符。

注意，如果从面板访问 **查找和替换对话框** 而不是从编辑面板访问时此标签将不可用。

图 A-34. 查找和替换对话框: [Quick Search] 标签



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Edit] 菜单, 选择 [Search...].
- 通过 [Edit] 菜单, 选择 [Replace...].

[各部分说明]**(1) [Search text]**

查找指定字符。

可以直接在文本框内输入字符 (最大字符数: 1024) 或从下拉列表中的选择历史输入记录 (最大历史记录数: 10)。

如果通过面板已打开的对话框中选定字符, 所选字符显示为默认。

如果通过编辑面板已打开的对话框, 单词 (变量 / 函数) 在光标位置显示为默认。

(2) [Replace with]

替换指定字符串。

可以直接在文本框内输入字符 (最大字符数: 1024) 或从下拉列表中的选择历史输入记录 (最大历史记录数: 10)。

(3) [Search location]

指定查找位置。

通过下拉列表选择下列条目之一。

| 条目 | 操作 |
|--------------|--|
| 选定区域 | 在最近一次活动且有效查找的面板中选定内容查找。 如果上一次激活的面板中没有字符，或者不能使用该面板，则此项目将不可用。 |
| 当前面板 (面板名称) | 在最近活动且能查找的面板中查找。 如果最近活动面板不能查找或面板不存在，则此条目将不可用。 |

备注 在下拉列表中记录的输入多达 10 条历史条目。

(4) [Option] 区

点击 [Option] 按钮，此区域就会显现（默认不显现）。

指定下列选项作为查找规则。

(a) [Search criteria]

通过下拉列表选择下列条目之一。

| 条目 | 操作 |
|------------|--------------------------|
| Plain text | 在 [Search text] 中查找指定字符。 |
| Wild-card | 使用下列通配符查找。 |
| | * 任意字符串 |
| | ? 任意字符。 |

(b) [File type]

此条目不可用。

(c) [Case-sensitive]

| | |
|-------------------------------------|-----------------------|
| <input checked="" type="checkbox"/> | 查找指定字符串（区分大小写）。 |
| <input type="checkbox"/> | 查找指定字符串 - 不区分大小写（默认）。 |

(d) [Word by word]

| | |
|-------------------------------------|---------------|
| <input checked="" type="checkbox"/> | 查找指定完全匹配的单词。 |
| <input type="checkbox"/> | 查找至少一个单词（默认）。 |

(e) [Open file before replacing]

此条目不可用。

[功能按钮]

| 按钮 | 功能 |
|--------|--|
| 选项 | 在此标签里切换 [Option] 区的显示 / 隐藏状态。 |
| Search | 从当前光标的位置到文件顶部按指定的规则进行查找。选择已查找到的字符串并移动光标。 |

| 按钮 | 功能 |
|---------------------|---|
| Search Forward | 从当前光标的位置到文件结束按照指定的规则进行查找。选择已查找到的字符串并移动光标 ^注 。 |
| Replace and Forward | 将选中的字符替换为替换字符，然后继续（往后）查找下一个并选中它们 ^注 。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Help | 显示此对话框的在线帮助。 |

注 如果指定的字符串没有查找到，会在主窗口的状态栏上显示 "Can not found Search text."。

[Whole Replace] 标签

在此标签里，指定字符的批量查找完成后，它们将批量替换成要替换的字符。

图 A-35. 查找和替换对话框：[Whole Replace] 标签



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Edit] 菜单，选择 [Search...]
- 通过 [Edit] 菜单，选择 [Replace...]

[各部分说明]**(1) [Search text]**

查找指定字符。

可以直接在文本框内输入字符（最大字符数：1024）或从下拉列表中的选择历史输入记录（最大历史记录数：10）。

如果通过面板已打开的对话框中选定字符，所选字符显示为默认。

如果通过编辑器打开了对话框，单词（变量 / 函数）默认显示在光标位置。

(2) [Replace with]

替换指定字符串。

可以直接在文本框内输入字符（最大字符数：1024）或从下拉列表中的选择历史输入记录（最大历史记录数：10）。

(3) [Search location]

指定查找位置。

通过下拉列表选择下列条目之一或通过键盘直接输入文件位置（最大数字：10）。

| 条目 | 操作 |
|---------|---|
| 当前打开的文件 | 在所有打开着的编辑面板中查找。 如果在编辑面板中没有文件已打开，此条目不可用。 |
| 活动工程 | 查找包含在当前活动的工程中的文本文件。 当指定 [File type] 时，只查找指定的类型。 注意：当前工程不存在时，此条目不可用。 |
| 主工程及子工程 | 查找包含在主工程和子工程中的文本文件。 当指定 [File type] 时，只查找指定的类型。 注意，如果当前工程不存在，此条目不可用。 |
| 文件夹名称 | 通过直接输入（最多字符数：259 个）路径（对工程文件夹采用的是相对路径）在指定文件夹中的文本文件中查找，或在此区域通过单击 [...] 按钮在打开的 文件夹浏览对话框 中指定。如果文件夹没有指定，工程文件夹名称将会以默认文件夹显示在 "()" 中（如果工程不存在，当前用户文档文件夹会显示）。 当指定 [File type] 时，只查找指定的类型。 |

备注 在下拉列表中记录的输入多达 10 条历史条目。

(4) [Option] 区

点击 [Option] 按钮，此区域就会显现（默认不显现）。

指定下列选项作为查找规则。

(a) [Search criteria]

通过下拉列表选择下列条目之一。

| 条目 | 操作 |
|------------|--------------------------|
| Plain text | 在 [Search text] 中查找指定字符。 |
| Wild-card | 使用下列通配符查找。 |
| | * 任意字符串 |
| | ? 任意字符。 |

(b) [File type]

查找指定的文件类型。

通过下拉列表选择下列条目之一。

| 条目 | 操作 |
|----------------------------------|---------------|
| 源文件 (<i>扩展名</i> ^{注意}) | 查找的文件限定为源文件。 |
| 文本文件 (*.txt) | 查找的文件限定为文本文件。 |
| 所有文件 (*.*) | 查找所有的文件。 |

注 显示已加入到 **工程树面板** 中的源文件扩展名。

注意： 通过在文本框内直接输入文件名（最大字符数：1024），限制查找规则来进行查找操作。

如果在这种情况下，能使用通配符 "*" 以及用 ";" 来分隔多个文件名称。

备注 在下拉列表中记录的输入多达 10 条历史条目。

(c) [Case-sensitive]

| | |
|-------------------------------------|-----------------------|
| <input checked="" type="checkbox"/> | 查找指定字符串（区分大小写）。 |
| <input type="checkbox"/> | 查找指定字符串 - 不区分大小写（默认）。 |

(d) [Word by word]

| | |
|-------------------------------------|---------------|
| <input checked="" type="checkbox"/> | 查找指定完全匹配的单词。 |
| <input type="checkbox"/> | 查找至少一个单词（默认）。 |

(e) [Open file before replacing]

| | |
|-------------------------------------|---------------------------|
| <input checked="" type="checkbox"/> | 通过编辑面板打开查找文件 / 替换字符后完成替换。 |
| <input type="checkbox"/> | 不打开查找文件 / 替换字符，替换完成（默认）。 |

[功能按钮]

| 按钮 | 功能 |
|--------|--------------------------------|
| 选项 | 在此标签里切换 [Option] 区的显示 / 隐藏状态。 |
| 替换 | 按照指定的规则批量查找字符并将已查找到的字符替换成指定字符。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Help | 显示此对话框的在线帮助。 |

跳转定位对话框

该对话框用于移动光标到指定位置。

图 A-36. 跳转定位对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [Function buttons]

[如何打开]

- 通过 [Edit] 菜单, 选择 [Move To...]

[各部分说明]

(1) [Line number] 区

确定想要移动光标的位置。

可以直接在文本框内输入字符（最大字符数：1024）或通过选择下拉列表的历史输入记录（最大历史记录数目：10）。

标识的差别取决于下列的目标面板。

| 目标面板 | 说明 |
|-------|-----------|
| 编辑器面板 | 行号 (十进制数) |

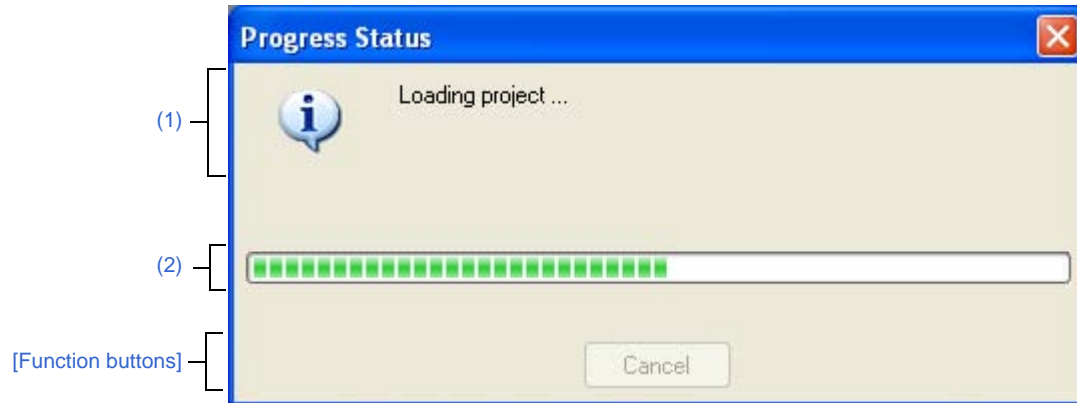
[Function buttons]

| 按钮 | 功能 |
|--------|--------------------------|
| OK | 显示目标面板顶部显示的指定位置并移动光标到那儿。 |
| Cancel | 取消规则并关闭此对话框。 |
| Help | 显示此对话框的在线帮助。 |

进度状态对话框

当发生耗时的进程时，此对话框用以显示进程的进度情况。
当进程结束后，此对话框将自动关闭。

图 A-37. 进度状态对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [Function buttons]

[如何打开]

- 当发生耗时的进程时会输出消息，此对话框自动打开。

[各部分说明]

(1) 消息显示区域

当进程在进行时显示输出消息（不允许编辑）。

(2) 进度条

进度条的长度表示当时进程的进度。

当进程 100% 全部完成（进度条到达最右边），对话框将自动关闭。

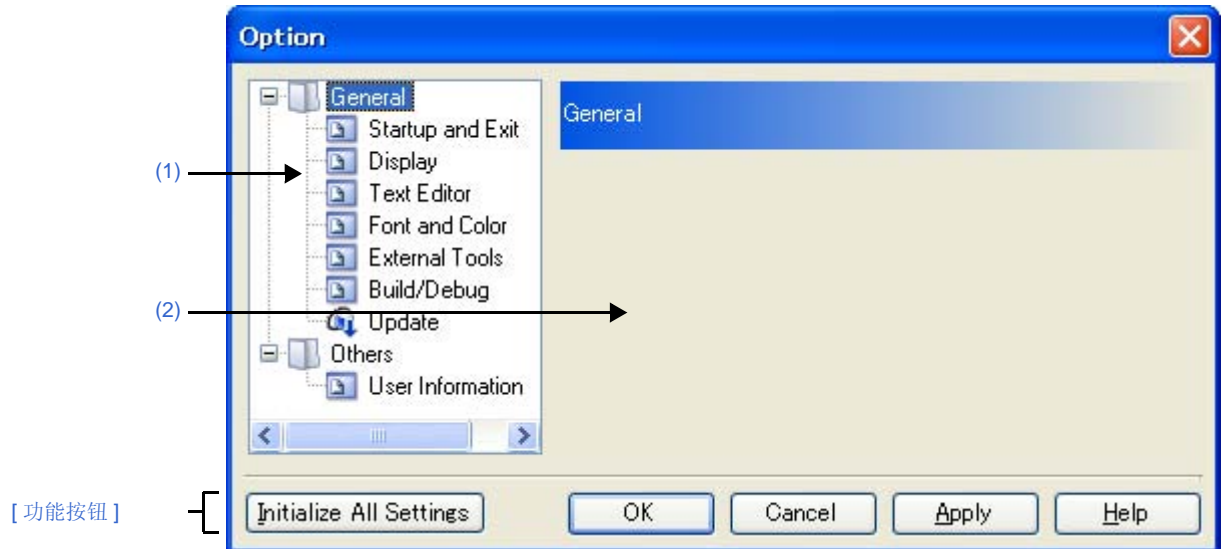
[Function buttons]

| 按钮 | 功能 |
|--------|---|
| Cancel | 取消运行中的进程并关闭对话框。 注意：如不能中止此进程，则此按钮不可用。 |

选项对话框

该对话框被用于配置 CubeSuite 环境。
通过本对话框优先保存当前用户做的所有设置。

图 A-38. 选项对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单, 选择 [Option...]

[各部分说明]

(1) 分项选择区

通过下述分项选择配置的条目。

| 分项 | 说明 |
|---------------------------------|---------------|
| [General - Startup and Exit] 分项 | 配置启动和停止。 |
| [General - Display] 分项 | 应用程序的配置消息。 |
| [General - Text Editor] 分项 | 配置文本编辑器。 |
| [General - Font and Color] 分项 | 配置每个面板的字体和颜色。 |
| [General - External Tools] 分项 | 配置外部工具的启动。 |
| [General - Build/Debug] 分项 | 配置创建和调试。 |
| [General - Update] 分项 | 配置升级 |
| [Other - User Information] 分项 | 配置用户信息。 |

(2) 设置

该区用于配置选择分项的不同选项。
要了解配置特定分项的详细信息, 参阅分项问题的章节。

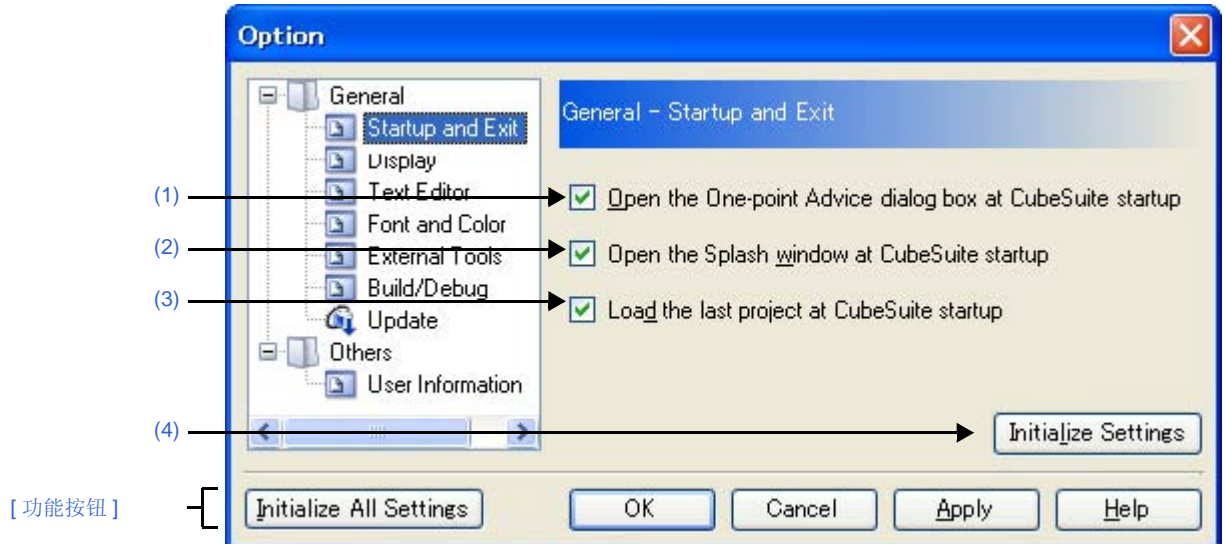
[功能按钮]

| 按钮 | 功能 |
|---------|---|
| 初始化所有设置 | 恢复此对话框上所有的设置为默认值。 注意：不会移除新进加入 [General - External Tools] 分项 的条目。 |
| OK | 应用所有的设置并关闭此对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Apply | 应用所有的设置 (不关闭此对话框)。 |
| Help | 显示此对话框的在线帮助。 |

[General - Startup and Exit] 分项

使用该分项配置与启动和关闭有关的常用设置。

图 A-39. 选项对话框 ([General - Startup and Exit] 分项)



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单，选择 [Option...]

[各部分说明]**(1) [Open the One-point Advice dialog box at CubeSuite startup]**

| | |
|-------------------------------------|------------------------------|
| <input checked="" type="checkbox"/> | 在启动中显示 FormOnePoint 对话框（默认）。 |
| <input type="checkbox"/> | 在启动中不显示 FormOnePoint 对话框。 |

(2) [Open the Splash window at CubeSuite startup]

| | |
|-------------------------------------|-----------------------|
| <input checked="" type="checkbox"/> | 在启动中显示 Splash 窗口（默认）。 |
| <input type="checkbox"/> | 在启动中不显示 Splash 窗口。 |

(3) [Load the last project at CubeSuite startup]

| | |
|-------------------------------------|---------------------|
| <input checked="" type="checkbox"/> | 启动时自动载入最近的工程文件（默认）。 |
| <input type="checkbox"/> | 启动时不自动载入最近的工程文件。 |

(4) 按钮

| | |
|-------|------------------|
| 初始化设置 | 恢复当前所有的显示设置为默认值。 |
|-------|------------------|

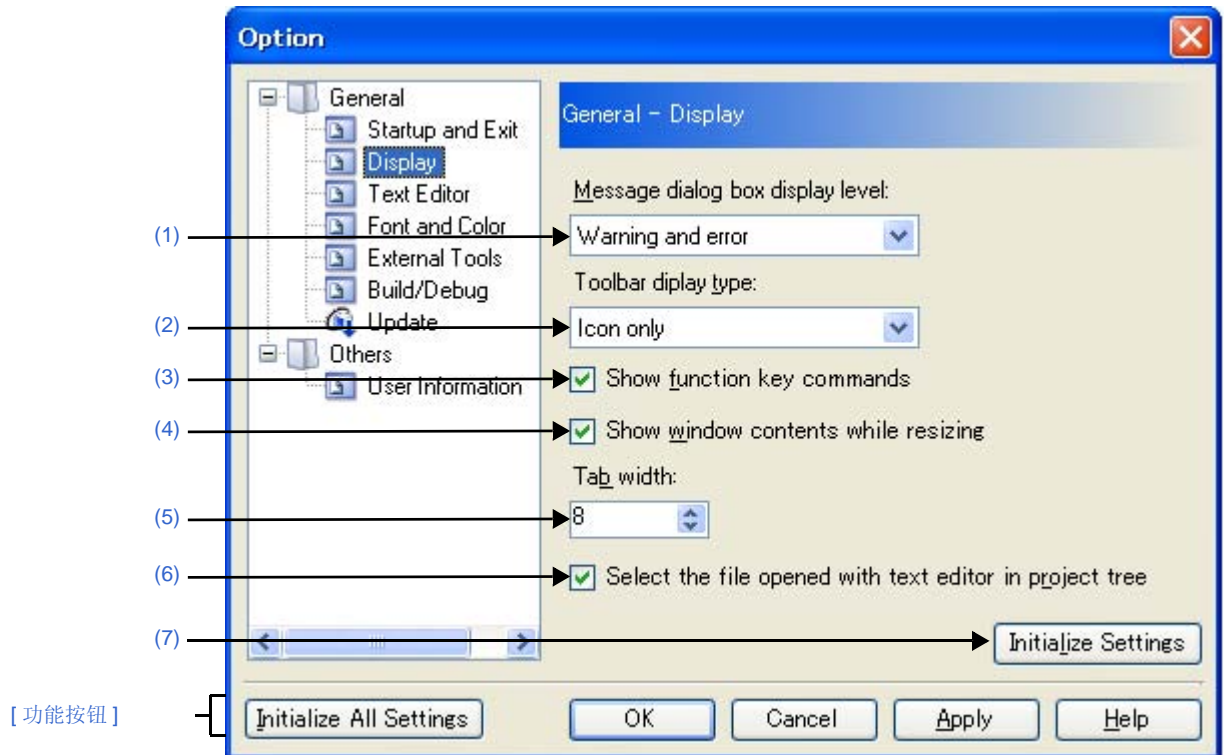
[功能按钮]

| 按钮 | 功能 |
|---------|---|
| 初始化所有设置 | 恢复此对话框上所有的设置为默认值。 注意：不会移除新进加入 [General - External Tools] 分项 的条目。 |
| OK | 应用所有的设置并关闭此对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Apply | 应用所有的设置 (不关闭此对话框)。 |
| Help | 显示此对话框的在线帮助。 |

[General - Display] 分项

使用该分项配置与程序消息有关的常用设置。

图 A-40. 选项对话框 ([General - Display] 分项)



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单, 选择 [Option...]

[各部分说明]**(1) [Message dialog box display level]**

通过下列的下拉列表选择消息对话框中的显示级别 (冗长)。

不管此设置, 所有消息显示在输出面板里。

| | |
|------------|------------------------|
| 信息、警告和出错 | 在消息对话框内显示所有消息。 |
| 警告和出错 | 在消息对话框里显示警告和出错信息 (默认)。 |
| Error only | 在消息对话框内只显示出错消息。 |
| 仅致命错误 | 在消息对话框内只显示致命错误消息。 |

(2) [Toolbar display type]

使用此区域, 通过以下的下拉列表, 选择每个面板上工具栏的显示格式。

| | |
|----------------|--------------|
| Icon only | 仅显示图标（默认）。 |
| Icon and label | 显示图标和标签（文本）。 |

(3) [Show function key commands]


| | |
|-------------------------------------|------------------|
| <input checked="" type="checkbox"/> | 在主窗口中显示功能键栏（默认）。 |
| <input type="checkbox"/> | 在主窗口中不显示功能键栏。 |

(4) [Show window contents while resizing]

| | |
|-------------------------------------|----------------------|
| <input checked="" type="checkbox"/> | 在显示区不可见之前调整窗口的大小。 |
| <input type="checkbox"/> | 在显示区可见之前调整窗口的大小（默认）。 |

(5) [Tab width]

指定标签栏的数目。

无论是通过键盘直接输入 1 到 16 之间的数字还是通过数字按钮  来设定数字。默认为 8。

(6) [Select the file opened with text editor in project tree]

| | |
|-------------------------------------|----------------------------------|
| <input checked="" type="checkbox"/> | 当编辑器面板活动时，选定工程树面板中当前正打开着的文件（默认）。 |
| <input type="checkbox"/> | 即使编辑器面板活动着，不选定工程树面板里当前正打开着的文件。 |

(7) 按钮

| | |
|-------|------------------|
| 初始化设置 | 恢复当前所有的显示设置为默认值。 |
|-------|------------------|

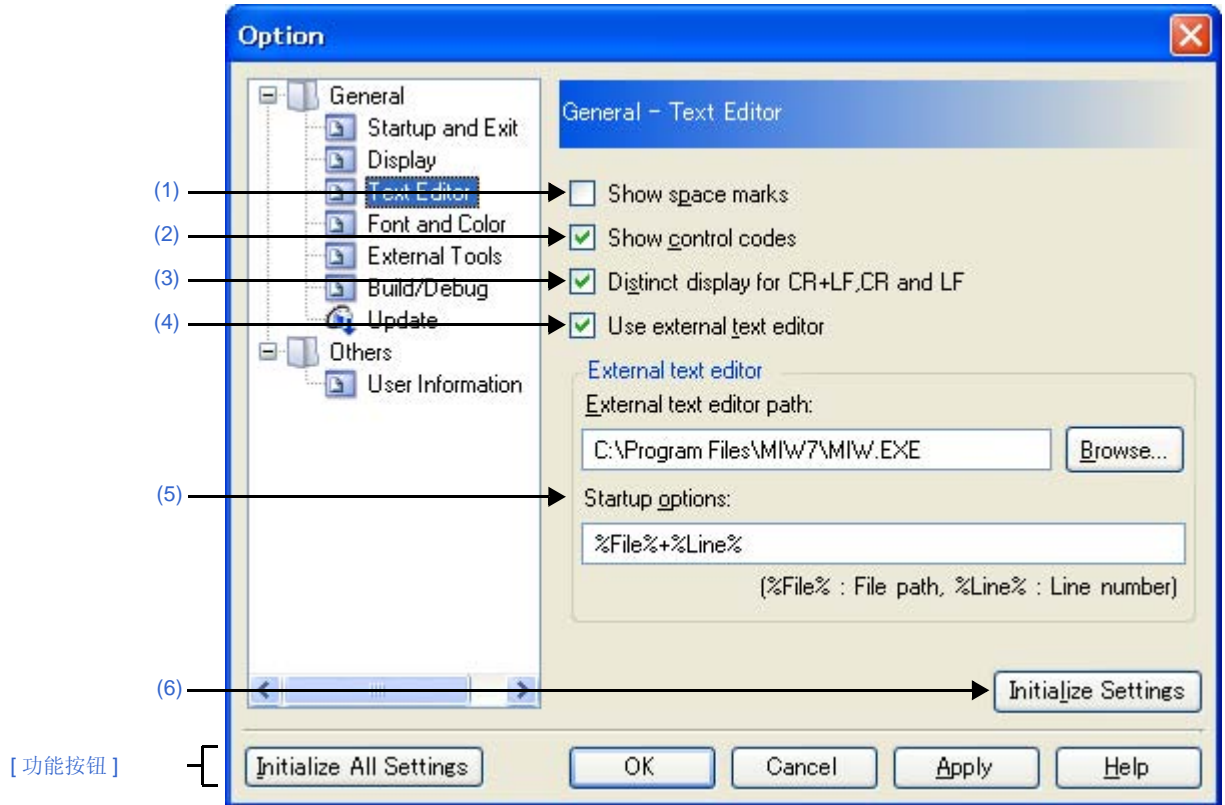
[功能按钮]

| 按钮 | 功能 |
|---------|---|
| 初始化所有设置 | 恢复此对话框上所有的设置为默认值。 注意：不会移除新进加入 [General - External Tools] 分项 的条目。 |
| OK | 应用所有的设置并关闭此对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Apply | 应用所有的设置（不关闭此对话框）。 |
| Help | 显示此对话框的在线帮助。 |

[General - Text Editor] 分项

使用该分项配置与文本编辑器相关的常用设置。

图 A-41. 选项对话框 ([General - Text Editor] 分项)



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单, 选择 [Option...]

[各部分说明]**(1) [Show space marks]**

| | |
|-------------------------------------|--------------------|
| <input checked="" type="checkbox"/> | 在文本编辑器里显示空字符 (默认)。 |
| <input type="checkbox"/> | 在文本编辑器里不显示空字符。 |

(2) [Show control codes]

| | |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | 显示控制代码 (如 :tab、换行、与 EOF) 在文本编辑器中 (默认)。 |
| <input type="checkbox"/> | 不显示控制代码 (如 :tab、换行、与 EOF) 在文本编辑器中。 |

(3) [Distinct display for CR+LF, CR and LF]

如果 [\[Show control codes\]](#) 的复选框选定时，则此条目可用。

| | |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | 在文本编辑器中从外观上区分换行符类型 (CR+LF, CR, and LF) (默认)。 |
| <input type="checkbox"/> | 在文本编辑器中从外观上不区分换行符类型 (CR+LF, CR, and LF)。 |

(4) [Use external text editor]

| | |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | 使用外部文本编辑器。 此 [External text editor] 区可用。通过它指定使用外部文本编辑器。 |
| <input type="checkbox"/> | 使用编辑面板作为文本编辑器 (默认)。 |

(5) [External text editor] 区

如果选定 [\[Use external text editor\]](#) 的复选框时，此区域才可用。

使用此区域来指定使用外部文本编辑器并设定启动选项。

(a) [External text editor path]

无论是通过键盘 (多达 259 个字符) 直接在外部文本编辑器中输入可执行文件名 (包括绝对路径) 还是单击 [\[Browse...\]](#) 按钮并在 [选择外部文本编辑器对话框](#) 定义可执行文件名。

(b) [Startup options]

为外部文本编辑器设定启动选项 (多达 256 个字符)。

启动选项变量可做如下设定。

如果想设定一个以上的选项，则需用空格来分隔。

| | |
|--------|-----------------------------|
| %File% | 传递文件名并显示在外部文本编辑器上 (使用绝对路径)。 |
| %Line% | 传递光标位置 (行号) 给外部文本编辑器。 |

(6) 按钮

| | |
|-------|------------------|
| 初始化设置 | 恢复当前所有的显示设置为默认值。 |
|-------|------------------|

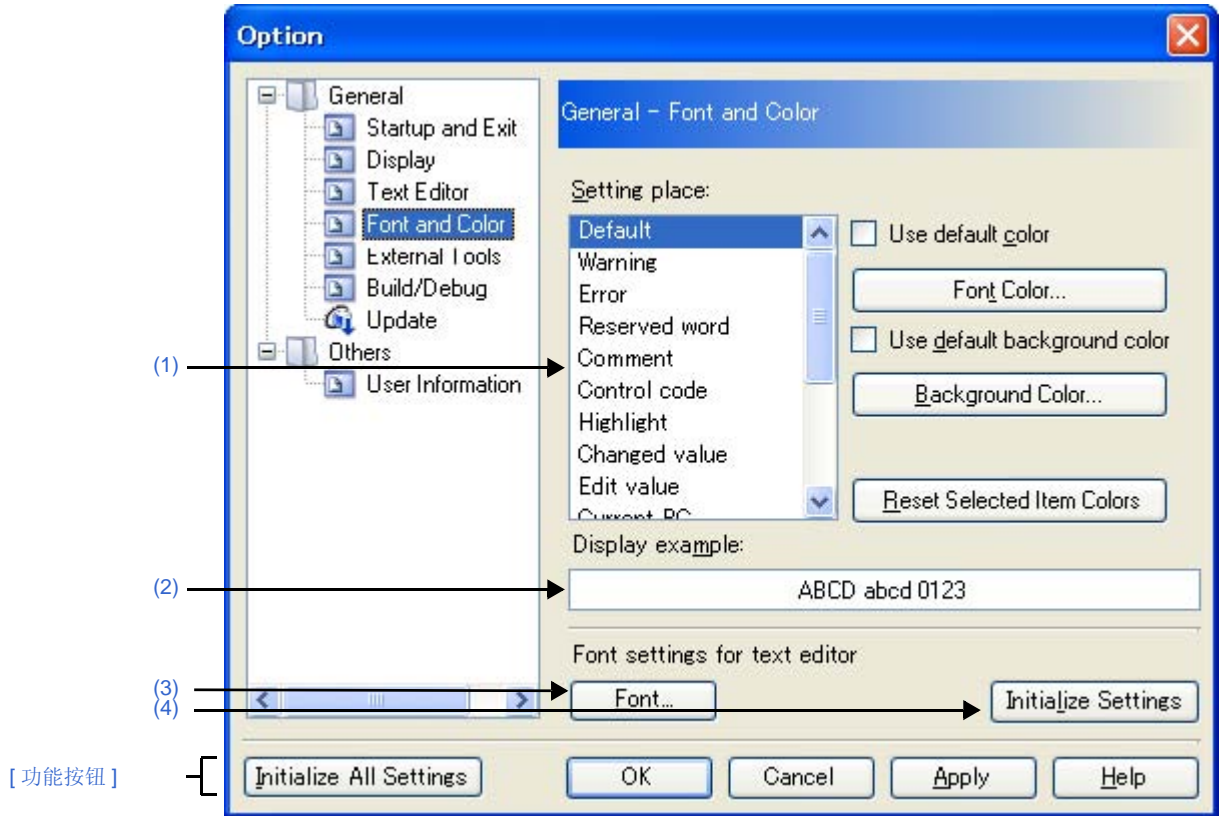
[功能按钮]

| 按钮 | 功能 |
|---------|--|
| 初始化所有设置 | 恢复此对话框上所有的设置为默认值。 注意：不会移除新加入 [General - External Tools] 分项 的条目。 |
| OK | 应用所有的设置并关闭此对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Apply | 应用所有的设置 (不关闭此对话框)。 |
| Help | 显示此对话框的在线帮助。 |

[General - Font and Color] 分项

使用该分项配置每个面板上与字体和颜色相关的常用设置。

图 A-42. 选项对话框 ([General - Font and Color] 分项)



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单, 选择 [Option...]

[各部分说明]**(1) 颜色选项区**

使用此区域配置颜色。

(a) [Setting place] 区域

设定从列表中选定位置的颜色。

列表项与默认设置颜色的关系如下。

| 条目 | 示例 | | 说明 |
|------------------|--------|------|-----------------|
| 默认 ^{注意} | AaBbCc | 字体颜色 | 黑色 |
| | | 背景颜色 | 白色 |
| | | | 所有窗口和面板的标准显示颜色。 |

| 条目 | 示例 | | 说明 | |
|---------|--------|------|------|--|
| Warning | AaBbCc | 字体颜色 | 蓝色 | 在输出面板中显示的警告消息使用此颜色并在工程树面板中显示的含警告的文件名使用此颜色。 |
| | | 背景颜色 | 默认颜色 | |
| 错误 | AaBbCc | 字体颜色 | 红色 | 在输出面板中的出错消息显示用的是这种颜色并且在工程树面板中出错的文件名显示也用的是这种颜色。 |
| | | 背景颜色 | 浅灰色 | |
| 保留字 | AaBbCc | 字体颜色 | 褐色 | 在编辑面板中显示编译 / 汇编的保留字用这种颜色。 |
| | | 背景颜色 | 默认颜色 | |
| 注释 | AaBbCc | 字体颜色 | 绿色 | 注释（在 C 语言源文件中， <code>/*...*/</code> ）在编辑面板中用这种颜色来显示。 |
| | | 背景颜色 | 默认颜色 | |
| 控制代码 | AaBbCc | 字体颜色 | 水鸭色 | 在编辑面板中显示的控制字符用的是这种颜色。 |
| | | 背景颜色 | 默认颜色 | |
| 高亮 | AaBbCc | 字体颜色 | 白色 | 类似插件等产品强调区的显示用的是这种颜色。 |
| | | 背景颜色 | 洋红色 | |
| 可变值 | AaBbCc | 字体颜色 | 棕褐色 | 在存储器面板，CPU 寄存器面板，局部变量面板，SFR 面板及监视面板中通过执行用户程序改变的值的显示用的是此颜色。 |
| | | 背景颜色 | 奶油色 | |
| 编辑值 | AaBbCc | 字体颜色 | 蓝色 | 在存储器面板，CPU 寄存器面板，局部变量面板，SFR 面板及监视面板中用户强制性修改过的变量显示用的是此颜色。 |
| | | 背景颜色 | 默认颜色 | |
| 当前 PC | AaBbCc | 字体颜色 | 黑色 | 在编辑面板中当前 PC 的行位置用此颜色显示。 |
| | | 背景颜色 | 亮橙色 | |
| 中断点 | AaBbCc | 字体颜色 | 黑色 | 用此颜色显示在编辑面板中设置有中断点的行。 |
| | | 背景颜色 | 橙红色 | |
| 更新周期 | AaBbCc | 字体颜色 | 粉红色 | 在存储面板和监视面板中实时显示更新的配置区域显示此颜色。 |
| | | 背景颜色 | 默认颜色 | |
| 读取或提取 | AaBbCc | 字体颜色 | 默认颜色 | 在跟踪面板 [IECUBE] [Simulator] 中读取和提取位置的显示用此颜色。 |
| | | 背景颜色 | 亮绿色 | |
| 写 | AaBbCc | 字体颜色 | 默认颜色 | 在跟踪面板 [IECUBE] [Simulator] 中写入位置的显示是此颜色。 |
| | | 背景颜色 | 橙色 | |

| 条目 | 示例 | | | 说明 |
|-------|---|------|------|---|
| 读取和写入 |  | 字体颜色 | 默认颜色 | 用此颜色显示在跟踪面板 [IECUBE] [Simulator] 中读取和写入的位置。 |
| | | 背景颜色 | 天蓝色 | |
| 无效 |  | 字体颜色 | 灰色 | 用此颜色表示在工程树面板的文件名和存储面板中不存在非映射存储区域。 |
| | | 背景颜色 | 默认颜色 | |

注 [Default] 文本和背景颜色由主计算机的窗口设置决定。这里使用 WINDOWS 默认设置，即黑色文本和白色背景。

(b) [Use default color]

| | |
|-------------------------------------|---|
| <input checked="" type="checkbox"/> | 通过 [Setting place] 区域使用的标准文本颜色来显示选定条目。 |
| <input type="checkbox"/> | 通过 [Setting place] 区域使用用户定义的文本颜色来显示选定条目。 [Font color...] 按钮可用。 |

(c) [Use default background color]

| | |
|-------------------------------------|---|
| <input checked="" type="checkbox"/> | 通过 [Setting place] 区域使用标准背景颜色来显示选定条目。 |
| <input type="checkbox"/> | 通过 [Setting place] 区域使用用户定义的背景颜色来显示选定条目。 [Background Color...] 按钮可用。 |

(d) 按钮

| | |
|---------------------|--|
| Font Color... | 打开颜色编辑对话框。设定从 [Setting place] 区域选定条目的文本颜色。 注意：如果选中 [Use default color] 的复选框时则此按钮不可用。 |
| Background Color... | 打开颜色编辑对话框。设定从 [Setting place] 区域选定条目的背景颜色。 注意：如果选中 [Use default background color] 的复选框时则此按钮不可用。 |
| 复原已选条目颜色 | 通过 [Setting place] 区域来恢复选定条目的颜色信息为默认。 |

图 A-43. 颜色编辑对话框

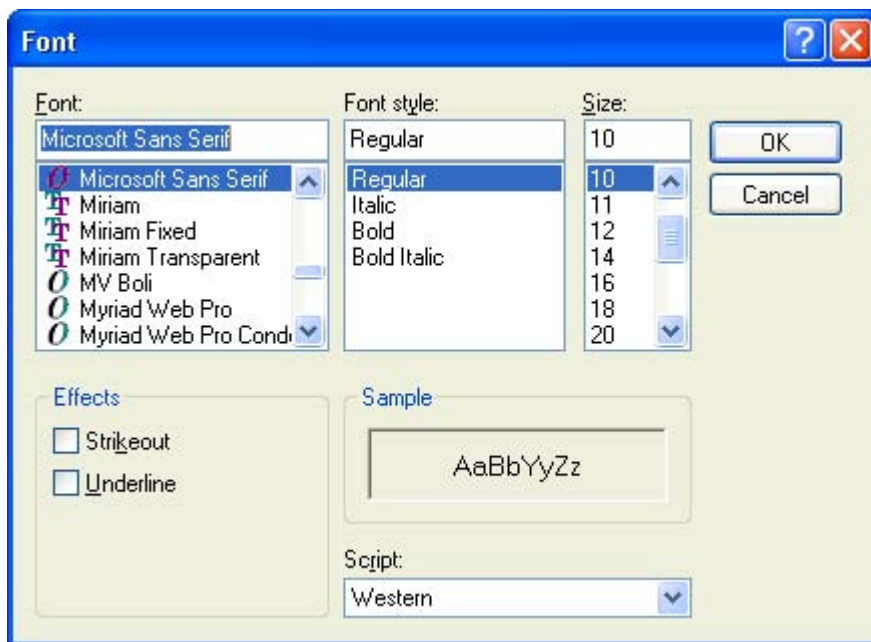
**(2) [Display example] 区**

通过颜色选项区和 [Font settings for text editor] 区显示应用颜色和字体设置的样例文本。通过默认显示字符串 "AaBbCc"，直接在文本框内可以键入任意字符串。

(3) [Font settings for text editor] 区

点击 [Font...] 按钮，打开字体对话框并为文本编辑器配置字体。

图 A-44. 字体对话框

**(4) 按钮**

| | |
|-------|------------------|
| 初始化设置 | 恢复当前所有的显示设置为默认值。 |
|-------|------------------|

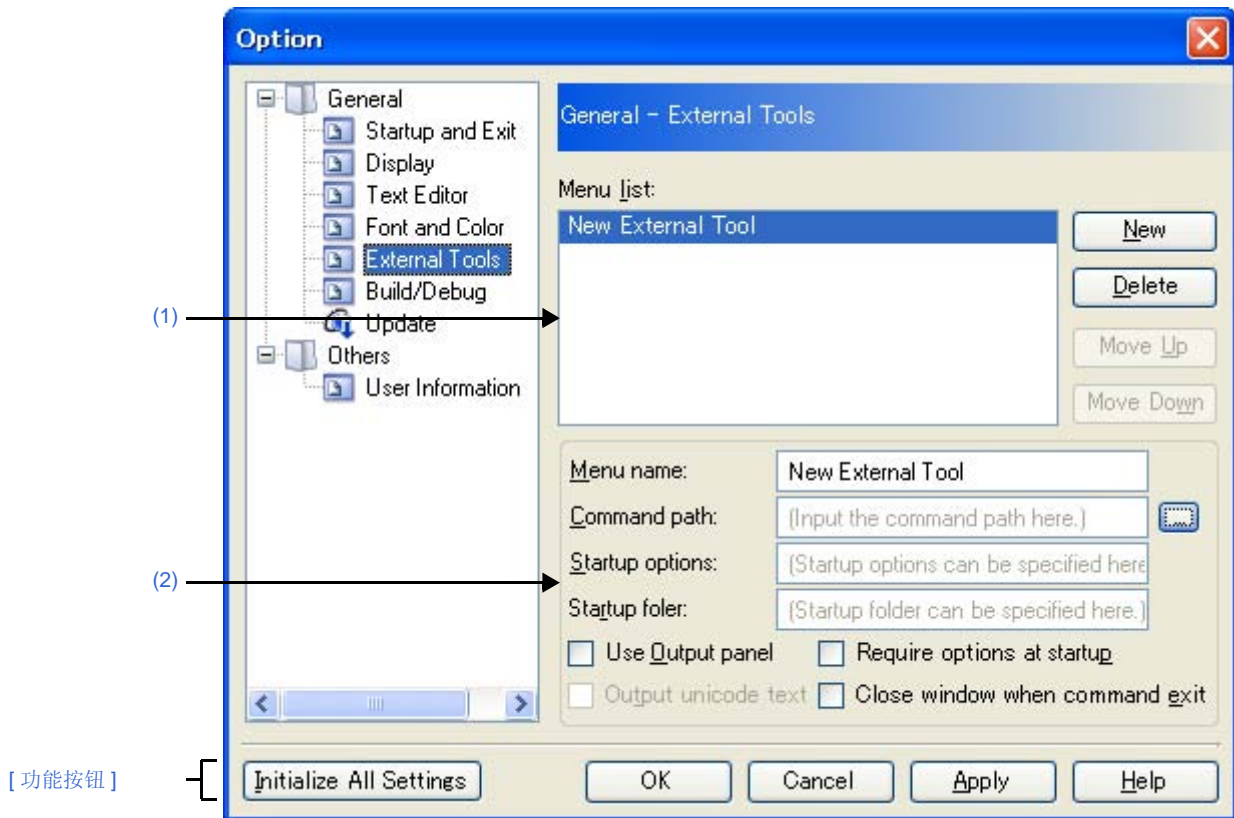
[功能按钮]

| 按钮 | 功能 |
|---------|---|
| 初始化所有设置 | 恢复此对话框上所有的设置为默认值。 注意：不会移除新进加入 [General - External Tools] 分项 的条目。 |
| OK | 应用所有的设置并关闭此对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Apply | 应用所有的设置 (不关闭此对话框)。 |
| Help | 显示此对话框的在线帮助。 |

[General - External Tools] 分项

使用该分项注册外部工具并能通过 CubeSuite 直接运行，在它们运行时配置这些外部工具。
注册外部工具并可以通过 [Tool] 菜单直接运行。

图 A-45. 选项对话框 ([General - External Tools] 分项)。



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单，选择 [Option...]

[各部分说明]

(1) 注册外部工具区域

(a) [Menu list] 区

通过此对话框中显示的菜单条目列表（已添加到 [Tool] 菜单）运行已添加的外部工具。

点击 [New] 按钮，添加新的外部工具。

当单击 [New]，"新外部工具"项目被添加到列表的下方。在这种状态下，详细配置添加在 [新建注册区域](#) 中的外部工具。

多达 8 个外部工具可以注册。

改变已注册外部工具的设置，选则外部工具的名称并修改和编辑所需设置。

在此列表中条目的顺序与菜单中的顺序相同。

(b) 按钮

| | |
|-----------|--|
| New | 添加项目 " 新建外部工具 " 到列表下方以便注册一个新的外部工具。 注意：如果已注册了 8 个外部工具，此按钮将不可用。 |
| Delete | 在列表中移除选中的外部工具。 |
| Move Up | 在列表中移动选定的外部工具至上一行。 |
| Move Down | 在列表中移动选定的外部工具至下一行。 |

(2) 新建注册区域

使用此区域配置新近添加的外部工具。

多达 8 个外部工具可以注册。

(a) [Menu name]

设定在菜单中运行的外部工具的名称（多达 100 个字符）。

设定的名称在 [Tool] 菜单中出现时，选择那个条目即可运行外部工具。

在输入菜单名称后（文本框失去焦点），此名称将自动在列表中替换成 " 新建外部工具 " 字符串。

(b) [Command path]

无论是直接通过键盘（多达 259 个字符）输入新建的外部工具的可执行文件名称（包括绝对路径）还是在 [选择程序对话框](#) 内点击 [...] 按钮，设定可执行文件的名称 ([Startup folder] 自动设置可执行文件所在位置为文件夹的绝对路径)。

(c) [Startup options]

设定新建外部工具的启动选项（多达 256 个字符）。

启动选项变量可做如下设定。

如果想设定一个以上的选项，则需用空格来分隔。

| | |
|--------------|---|
| %File% | 传送 工程树面板 中当前选中的文件名称（含绝对路径）给外部工具。如果选中了一个以上的文件，则发送文件中光标所在位置的信息。 |
| %OutputFile% | 构建 工程树面板 中当前选定的工程时（如果没有选定则不传送）并输出程序发送给外部工具的模块文件名称（包括绝对路径）。 |
| %Line% | 如果选中的文件正在编辑面板的 工程树面板 中编辑，则发送文件中光标所在行的位置（如果焦点不在编辑面板上，则发送 "1"）。 |

(d) [Startup folder]

为运行外部工具设定文件夹绝对路径（多达 254 个字符）。

启动文件夹变量可做如下设定。

如果想设定一个以上的选项，则需用空格来分隔。

| | |
|--------------|---|
| %FileDir% | 在 工程树面板 中发送存放当前选中的文件的文件夹的名称（包含绝对路径）给外部工具。如果选定一个以上的文件，则发送文件中光标所在位置的信息（如果没有选定，则不传送）。 |
| %OutputDir% | 构建 工程树面板 中当前选定的工程时（如果没有选定则不发送），输出程序发送的模块文件夹的名称（含绝对路径）。 |
| %ProjectDir% | 程序发送 工程树面板 中当前选定的工程所在文件夹的（包含绝对路径）名称（如果没有选定则不发送）。 |

(e) [Use Output panel]

| | |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | 新建的外部工具显示消息输出到 输出面板 中的 <code>stdout</code> 和 <code>stderr</code> 。消息将被输出到 输出面板 的 <code>[Tool Name]</code> 标签（“ <code>工具名称</code> ”是用 <code>[Menu name]</code> 中来指定的字符串）。 |
| <input type="checkbox"/> | 新建外部工具使用窗口命令提示的消息输出到 <code>stdout</code> 和 <code>stderr</code> （默认）。 |

(f) [Require options at startup]

| | |
|-------------------------------------|----------------------------|
| <input checked="" type="checkbox"/> | 打开输入的字符串，便于设定新建的外部工具的启动参数。 |
| <input type="checkbox"/> | 不设定新建的外部工具启动参数（默认）。 |

(g) [Output unicode text]

如果 [\[Use Output panel\]](#) 的复选框选定时，则此条目可用。

| | |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | 新建外部工具的显示消息输出到 输出面板 的 <code>stdout</code> 和 <code>stderr</code> （使用的是 Unicode 字符集）。 |
| <input type="checkbox"/> | 新建外部工具显示消息输出到 输出面板 的 <code>stdout</code> 和 <code>stderr</code> （使用的是 ASCII 字符集）。 |

(h) [Close window when command exit]

如果 [\[Use Output panel\]](#) 复选框选定时，此条目不可用。

| | |
|-------------------------------------|----------------------------|
| <input checked="" type="checkbox"/> | 有新的外部工具存在时关闭窗口命令提示。 |
| <input type="checkbox"/> | 在有新建外部工具之后打开脱离窗口命令的提示（默认）。 |

[功能按钮]

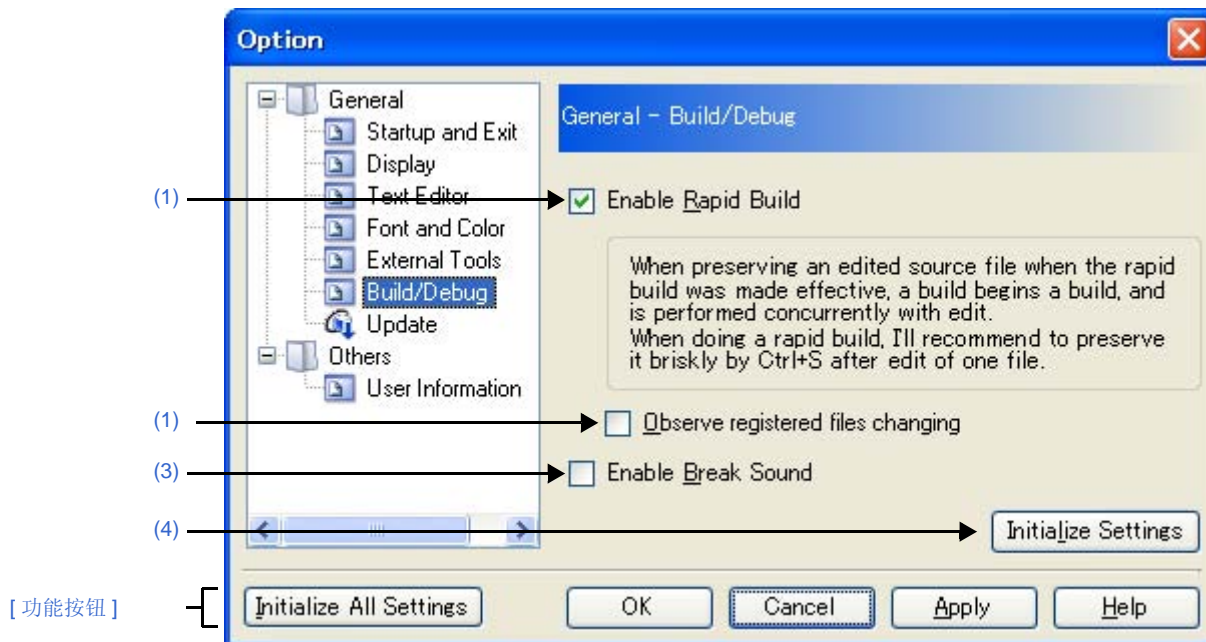
| 按钮 | 功能 |
|---------|---|
| 初始化所有设置 | 通过此对话框来恢复所有设置为默认值。 注意：不会移除新加入 [General - External Tools] 分项 的条目。 |

| 按钮 | 功能 |
|--------|----------------------|
| OK | 应用所有的设置并关闭此对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Apply | 应用所有的设置 (不关闭此对话框)。 |
| Help | 显示此对话框的在线帮助。 |

[General - Build/Debug] 分项

使用该分项配置与构建和调试相关的常用设置。

图 A-46. 选项对话框 ([General - Build/Debug] 分项)



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单, 选择 [Option...]

[各部分说明]

(1) [Enable Rapid Build]

| | |
|-------------------------------------|---------------------------------|
| <input checked="" type="checkbox"/> | 使快速 构建 ^{可用} 注意特性 (默认)。 |
| <input type="checkbox"/> | 不使用快速 构建特性。 |

注 当编辑已保存过的源文件时, 该特性自动开始构建。
此特性的作用是使编辑源文件时进行构建成为可能。
如果使用该特性, 我们建议编辑源文件后请多保存。

(2) [Observe registered files changing]

如果 [Enable Rapid Build] 的复选 框选定时, 则此条目可用。

| | |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | 使用外部文本编辑器及类似产品编辑或保存在注册在工程中的源文件时, 开启快速构建。 |
| <input type="checkbox"/> | 使用外部文本编辑器及类似产品编辑或保存在注册在工程中的源文件时, 不开启快速构建。(默认)。 |

(3) [Enable Break Sound]

| | |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | 当用户程序在执行过程中由于中断事件（硬件或软件中断）导致程序中止时发出蜂鸣声。 |
| <input type="checkbox"/> | 当用户程序在执行过程中由于中断事件（硬件或软件中断）导致程序中止时不发出蜂鸣声。（默认） |

(4) 按钮

| | |
|-------|------------------|
| 初始化设置 | 恢复当前所有的显示设置为默认值。 |
|-------|------------------|

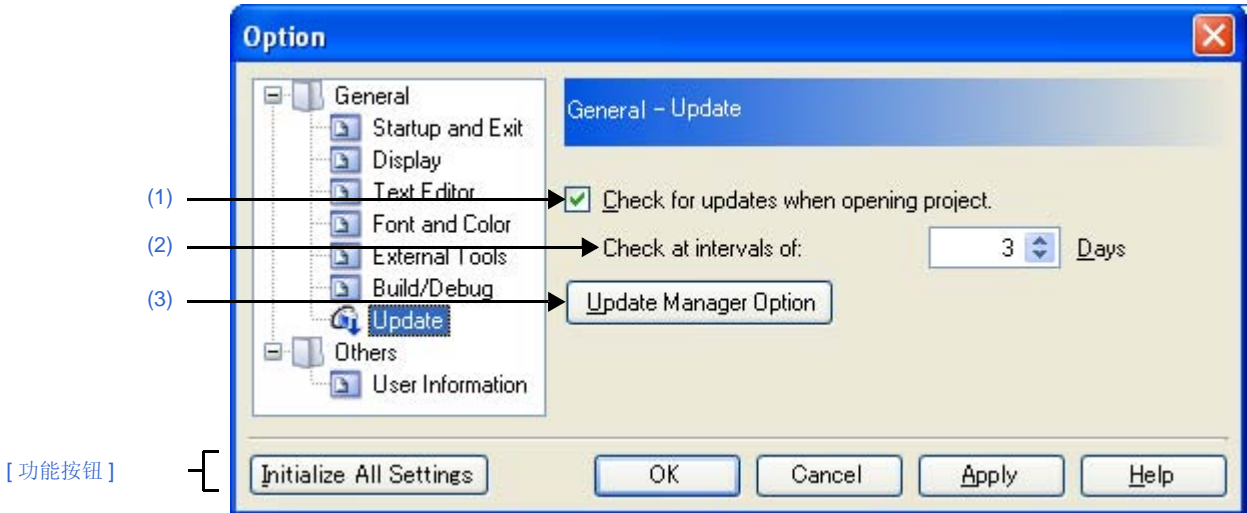
[功能按钮]

| 按钮 | 功能 |
|---------|---|
| 初始化所有设置 | 恢复此对话框上所有的设置为默认值。 注意：不会移除新进加入 [General - External Tools] 分项 的条目。 |
| OK | 应用所有的设置并关闭此对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Apply | 应用所有的设置（不关闭此对话框）。 |
| Help | 显示此对话框的在线帮助。 |

[General - Update] 分项

使用该分项去配置与更新相关的常用设置。

图 A-47. 选项对话框 (General - Update) 分项



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]


- 通过 [Tool] 菜单, 选择 [Option...]

[各部分说明]**(1) [Check for updates when opening project.]**

| | |
|-------------------------------------|----------------|
| <input checked="" type="checkbox"/> | 在工程打开时自动检查更新。 |
| <input type="checkbox"/> | 在工程打开时不自动检查更新。 |

(2) [Check at intervals of:]

设定检查更新的间隔时间。

无论是通过键盘直接输入 0 到 99 之间的数字还是通过  按钮来设定数字。

(3) 按钮

| | |
|--------|--------------|
| 更新管理选项 | 显示更新管理选项对话框。 |
|--------|--------------|

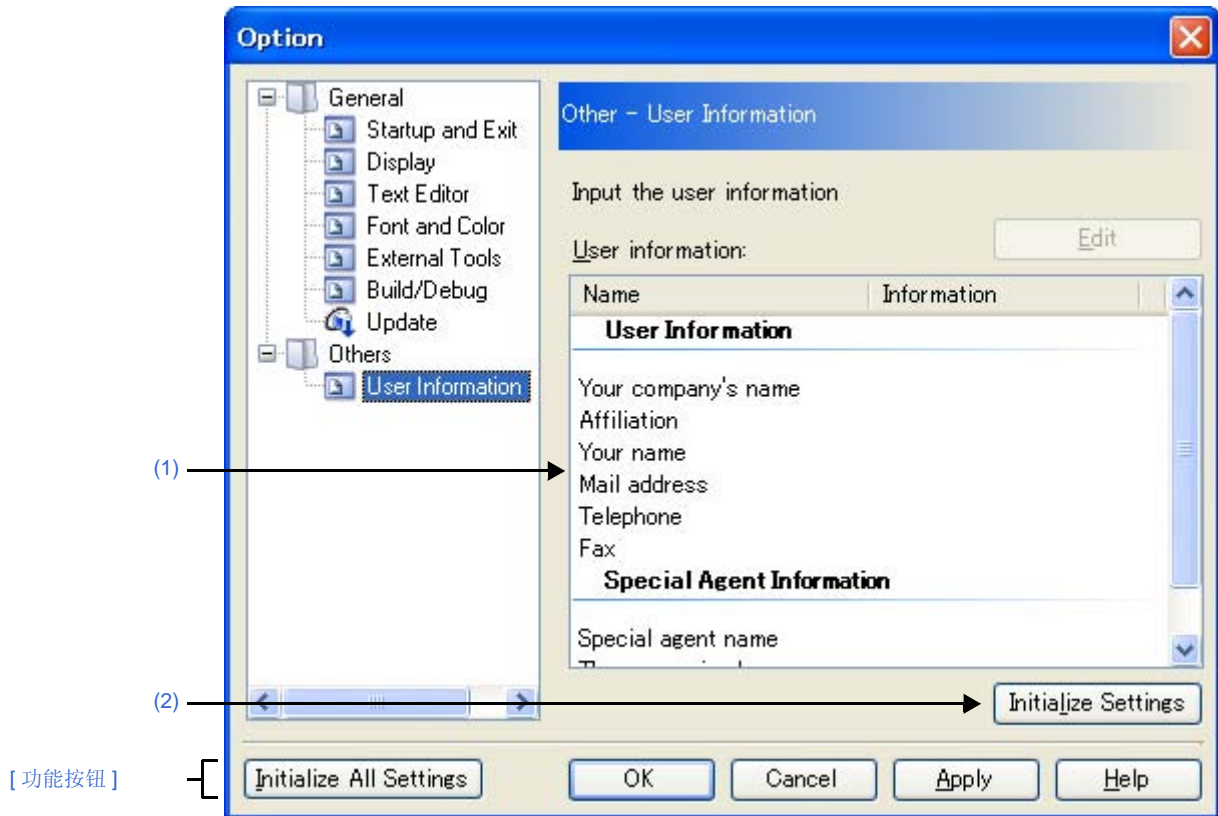
[功能按钮]

| 按钮 | 功能 |
|---------|---|
| 初始化所有设置 | 恢复此对话框上所有的设置为默认值。 注意：不会移除新进加入 [General - External Tools] 分项 的条目。 |
| OK | 应用所有的设置并关闭此对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Apply | 应用所有的设置 (不关闭此对话框)。 |
| Help | 显示此对话框的在线帮助。 |

[Other - User Information] 分项

使用该分项设置其他与用户信息相关的配置。

图 A-48. 选项对话框 ([Other - User Information] 分项)



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单，选择 [Option...]

[各部分说明]

(1) 信息区域

(a) [User information] 区域

此区域显示用户信息列表。

[Information] 字段的内容是可编辑的。编辑信息，选择列表中的条目之一，点击 [Edit] 按钮，并直接输入信息在文本框中 (多达 256 个字符)。

(b) 按钮

| | |
|------|---|
| Edit | 通过直接在文本框输入来编辑所选 [Information] 条目的内容。 如果列表中没有任何选择，此按钮是不可用的。 |
|------|---|

(2) 按钮

| | |
|-------|------------------|
| 初始化设置 | 恢复当前所有的显示设置为默认值。 |
|-------|------------------|

[功能按钮]

| 按钮 | 功能 |
|---------|---|
| 初始化所有设置 | 恢复此对话框上所有的设置为默认值。 注意：不会移除新进加入 [General - External Tools] 分项 的条目。 |
| OK | 应用所有的设置并关闭此对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Apply | 应用所有的设置 (不关闭此对话框)。 |
| Help | 显示此对话框的在线帮助。 |

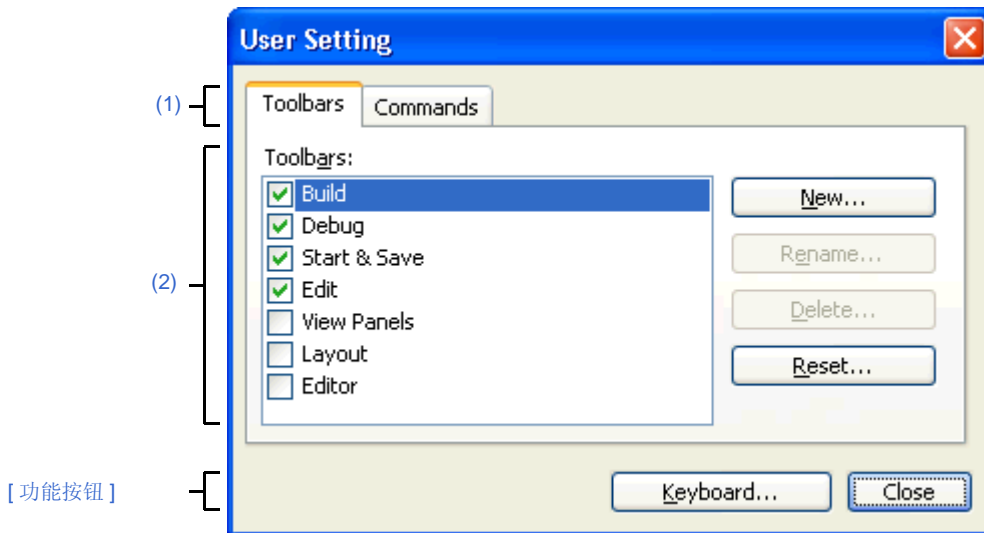
用户设置对话框

此对话框允许自定义主窗口中的工具栏和菜单。

当打开此对话框时，显示在主窗口工具条上的任何按钮或在菜单条里的任何菜单项，都可以拖动到所要的位置，用以改变按钮或原菜单项的顺序，或者删除按钮 / 菜单。

注意事项 当此对话框打开时，不能使用工具栏 / 菜单上的功能。
当关闭对话框后能照常使用这些功能。

图 A-49. 用户设置对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单，选择 [User Setting...]

[各部分说明]

(1) 标签选定区域

标签选择可以在自定义目标之间切换。

此对话框包括以下标签：

- [Toolbars] 标签
- [Commands] 标签

(2) 自定义区域

能设置详细的自定义条件。

关于显示条目 / 设置方法的详情，请参见各标签的说明。

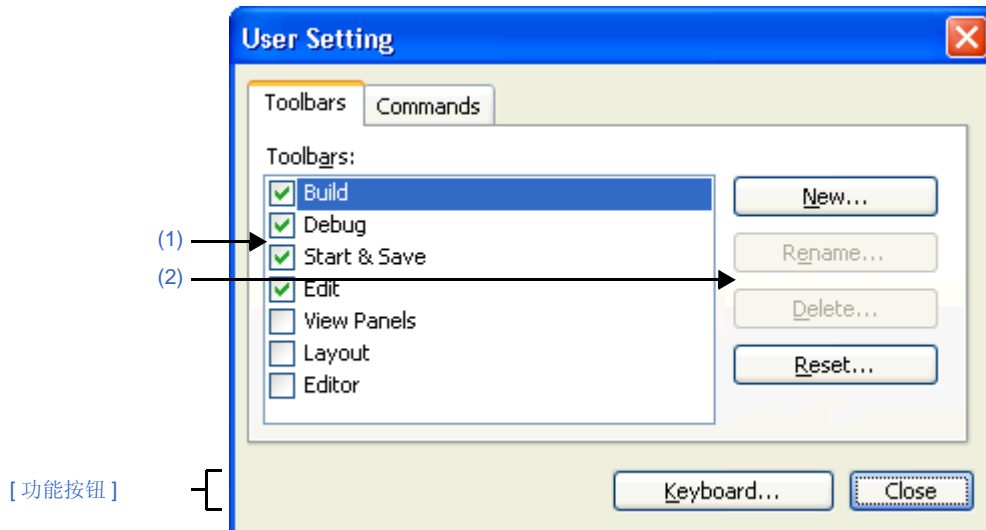
[功能按钮]

| 按钮 | 功能 |
|--------|---|
| 键盘 ... | 打开 自定义键盘对话框 在键盘的按键上分配自定义条目。 |
| Close | 取消工具条 / 菜单自定义设置并且关闭对话框。 |

[Toolbars] 标签

工具栏无论是显示还是隐藏都可以设置改变工具栏名称以及建立新工具栏。

图 A-50. 用户设置对话框 :[Toolbars] 标签



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单, 选择 [User Setting...].

[各部分说明]**(1) [Toolbars] 标签**

显示已注册工具栏的名称列表。

主窗口里当前显示工具栏名称的前面出现复选标记。通过移除复选标记, 使工具栏不显示。

(2) 按钮

| | |
|-----------|---|
| 新建 ... | 打开新建工具栏对话框建立新的工具栏。 新工具栏名称将会加入到 [Toolbars] 标签中显示的列表并有复选标记在前面。 注意: 新建的工具栏上没有按钮。使用 [Commands] 标签, 在上面注册按钮。 |
| Rename... | 打开重命名工具栏对话框改变当前选中工具栏的名称。 当工具栏不是通过用户选择而建立, 此功能无效。 |
| 删除 ... | 删除已选工具栏。 当工具栏不是通过用户选择而建立, 此功能无效。 |
| 复位 ... | 取消已选择工具栏的所有变化并让其恢复到默认状态。 当工具栏由用户选择来创建时, 此按钮使选中工具栏返回无按钮可注册在上面的状态。 |

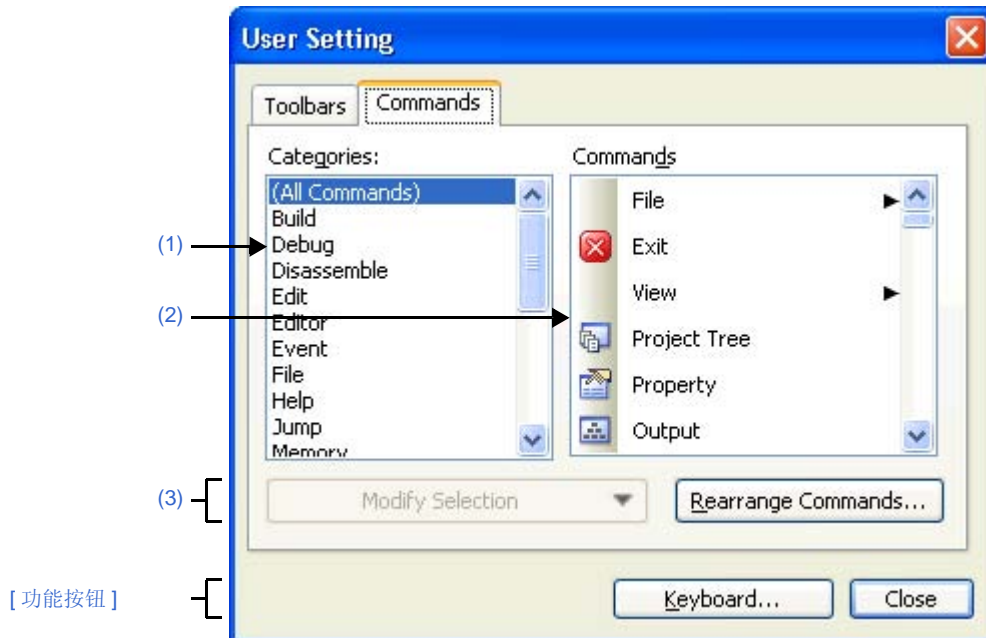
[功能按钮]

| 按钮 | 功能 |
|--------|---|
| 键盘 ... | 打开 自定义键盘对话框 在键盘的按键上分配自定义条目。 |
| Close | 取消工具栏 / 菜单自定义设置并且关闭对话框。 |

[Commands] 标签

可以在工具栏或菜单里自定义条目。

图 A-51. 用户设置对话框 :[Commands] 标签



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过 [Tool] 菜单, 选择 [User Setting...]

[各部分说明]**(1) [Categories] 区**

CubeSuite 允许显示命令分项列表。

(2) [Commands] 区

在 [Categories] 区显示属于已选分项的命令名称列表, 并显示它们的图标 (如果存在)。

当 “(All Commands)” 在 [Categories] 区被选择, 所有命令名称在 CubeSuite 中允许被显示, 包括他们的图标 (如果有的话)。

在工具栏 / 菜单上添加命令, 在此区域拖放此命令名称到当前主窗口的工具栏 / 菜单上。

(3) 按钮

| | | |
|----------|--|---|
| 修改选项 | 在 主窗口 里当前已选菜单项或工具栏上的按钮显示在相关菜单项的按钮之下。 | |
| | 菜单条目 | 说明 |
| | Reset | 恢复已选菜单项目 / 按钮。 |
| | Delete | 删除已选菜单项目 / 按钮。 |
| | Name | 显示已选菜单项目 / 按钮的名称。 |
| | 默认类型 | 校对菜单项目恢复已选菜单项目 / 按钮的显示类型到默认状态（默认）。 |
| | 仅文本（总是） | 校对菜单项目显示已选菜单项目 / 按钮仅限文本（图标将不会显示）。 |
| | 仅文本（在菜单里） | 当选中菜单条目时这为可用。校对菜单项目显示已选菜单项目 / 按钮仅限文本（图标不会显示）。 |
| | 图片和文本 | 校对菜单项目显示已选菜单项目 / 按钮仅限文本图标。 |
| | 开始一个组 | 在选中菜单项目 / 按钮之前嵌入分隔符。 |
| 最近使用过的 | 该版本里不支持此项目。 | |
| 重列命令 ... | 打开 重排命令对话框 为了改变 主窗口 里的菜单项目和工具栏按钮的布局（包含添加和删除的项目）。 | |

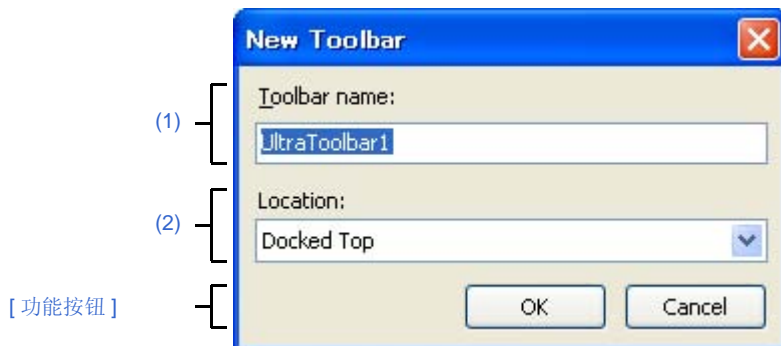
[功能按钮]

| 按钮 | 功能 |
|--------|------------------------------------|
| 键盘 ... | 打开 自定义键盘对话框 在键盘的按键上分配自定义条目。 |
| Close | 取消工具条 / 菜单自定义设置并且关闭对话框。 |

新建工具栏对话框

此对话框用于在主窗口中创建新工具栏。

图 A-52. 新建工具栏对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在用户设置对话框的 [Toolbars] 标签中，点击 [New...] 按钮。

[各部分说明]

(1) [Toolbar name]

通过键盘直接输入新建工具栏的名称。
默认名称是“UltraToolbar1”。

(2) [Location]

从下列的下拉列表中选择新建工具栏的位置。
指定新建工具栏创建后的显示位置（工具栏可自由拖放移动）。

| | |
|---------------|----------------------|
| Docked Top | 工具栏显示在主窗口的顶部（默认）。 |
| Docked Bottom | 工具栏显示在主窗口的底部。 |
| Docked Left | 工具栏显示在主窗口的左方。 |
| Docked Right | 工具栏显示在主窗口的右方。 |
| Floating | 工具栏将显示在主窗口的上方，不与其连接。 |

[功能按钮]

| 按钮 | 功能 |
|--------|--|
| OK | 按指定信息创建新工具栏，并关闭对话框。 新工具栏在选定单选框后在用户设置对话框的 [Toolbars] 标签显示。 |
| Cancel | 忽略设置并关闭此对话框。 |

重命名工具栏对话框

该对话框用于编辑由用户创建的工具栏的名称。

图 A-53. 重命名工具栏对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在用户设置对话框的 [Toolbars] 标签中，选择用户工具栏的名称，并点击 [Rename...] 按钮。

[各部分说明]

(1) [Toolbar name]

直接用键盘输入编辑工具栏名称。

当前选中工具栏的默认名称如下。

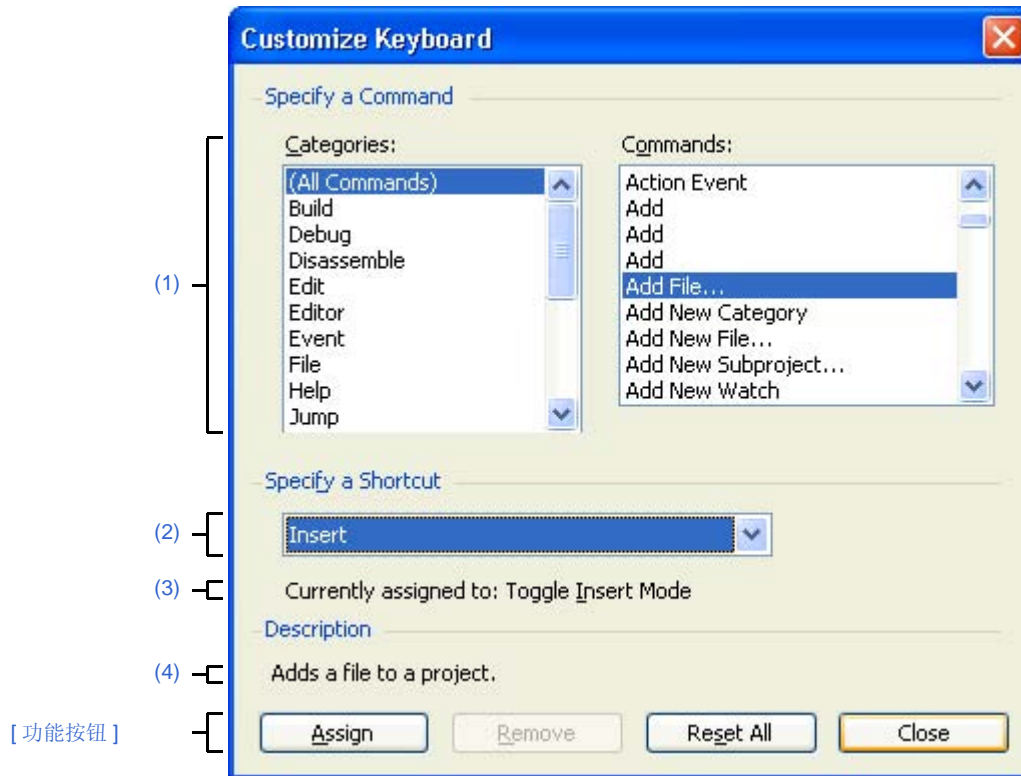
[功能按钮]

| 按钮 | 功能 |
|--------|------------------------|
| OK | 将选中的对话框改成指定的名称，并关闭对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |

自定义键盘对话框

该对话框用于分配不同命令的快捷键。

图 A-54. 自定义键盘对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在用户设置对话框的 [Toolbars] 标签中，点击 [Keyboard...] 按钮。

[各部分说明]

(1) [Specify a Command] 区

(a) [Categories]

显示由 CubeSuite 提供的命令分类列表。

(b) [Commands]

显示在 [Categories] 下选中的分项下的命令列表及与它们相关联的图标（假如有）。

如果在 [Categories] 下选择 “(All Command)”，则由 CubeSuite 提供的所有命令以及它们相对应的图标（假如有）出现。

(2) [Specify a Shortcut] 区

该区显示分配给在 [Commands] 下选定命令的当前默认快捷键（如果没有分配的键，则出现“None”）。要改变分配的快捷键，从下述列表中选择键，然后单击 [Assign] 按钮。

| | | | |
|----------------|------------------|-------------------------|----------------|
| None | Shift + F1 ~ F12 | Ctrl + Shift + 0 ~ 9 | Alt + Right |
| Insert | Ctrl + Insert | Ctrl + Shift + A ~ Z | Alt + Down |
| Delete | Ctrl + Delete | Ctrl + Shift + F1 ~ F12 | Alt + 0 ~ 9 |
| F1 ~ F12 | Ctrl + 0 ~ 9 | Alt + Backspace | Alt + F1 ~ F12 |
| Shift + Insert | Ctrl + A ~ Z | Alt + Left | |
| Shift + Delete | Ctrl + F1 ~ F12 | Alt + Up | |

(3) [Current assigned to]

该区显示分配在 [Specify a Shortcut] 区中设定的快捷键的命令（如果没有命令分配给该键，则显示“None”）。

(4) [Description] 区

该区显示弹出菜单，说明在 [Commands] 下选中的命令功能。

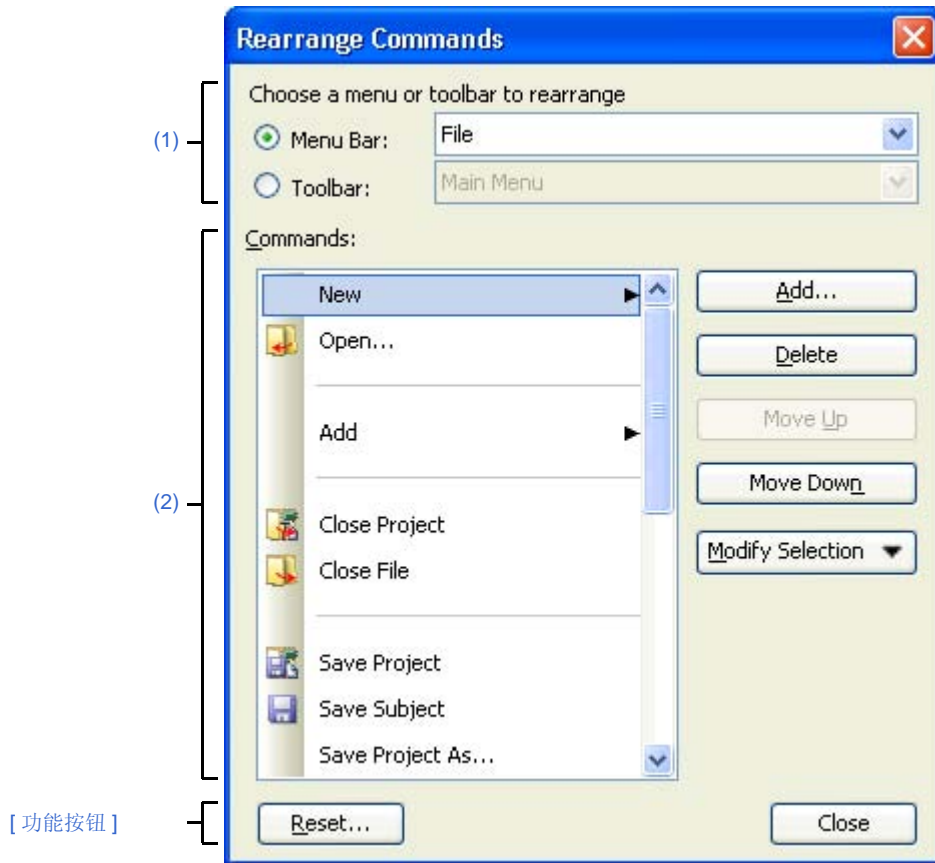
[功能按钮]

| 按钮 | 功能 |
|-----------|---|
| Assign | 分配在 [Commands] 下选择的快捷键给在 [Specify a Shortcut] 区下选定命令。 注意，如果在 [Specify a Shortcut] 区选择的键已经分配给别的命令，则该按钮失效。 |
| Remove | 移除在 [Specify a Shortcut] 区下选择的快捷键到在 [Commands] 下选择的命令的配置（“None”将出现在 [Specify a Shortcut] 区的下拉列表）。 注意，如果没有键分配给在 [Commands] 下选择的命令，则该按钮失效。 |
| Reset All | 恢复当前所有的快捷键设置为默认值。 |
| Close | 结束快捷键分配并关闭对话框。 |

重排命令对话框

该对话框用以改变主窗口中菜单条目和按钮的排列（包含加法和除法）。

图 A-55. 重排命令对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在用户设置对话框的 [Commands] 标签，点击 [Rearrange Commands...] 按钮。

[各部分说明]

(1) [Choose a menu or toolbar to rearrange] 区

该区允许用户设定想要改变位置的条目。

如果用户想使用选项按钮改变工具栏按钮或者想改变菜单条目或 [Toolbar]，首先选择 [Menu Bar] 然后通过下拉列表选择要改变的分项。

(2) [Commands] 区

该区显示属于在 [Choose a menu or toolbar to rearrange] 区（显示在主窗口）中的分项命令列表。可以在该区域中使用下述按钮改变这些命令的排放。

| | | |
|-----------|--|---|
| Add... | 通过打开 Add Command Dialog Box 选择命令，用于在该区添加当前选择的命令。 在 Add Command Dialog Box 中，显示所有能添加的命令，并根据它们的类别分组。 首先选择在 [Categories] 区中的命令分类，然后选择想要添加的命令，并按 [OK] 按钮（按 [Cancel] 按钮取消添加命令并关闭该对话框） | |
| Delete | 删除该区当前选择的命令。 | |
| Move Up | 移动该区当前选择的命令到第一行。 | |
| Move Down | 移动该区当前选择的命令到最后一行。 | |
| 修改选项 | 显示下述菜单条目，用以编辑该区当前选中的菜单条目或按钮。 | |
| | 菜单条目 | 说明 |
| | Reset | 恢复已选菜单项目 / 按钮。 |
| | Delete | 删除已选菜单项目 / 按钮。 |
| | Name | 显示已选菜单项目 / 按钮的名称。 |
| | 默认类型 | 校对菜单项目恢复已选菜单项目 / 按钮的显示类型到默认状态（默认）。 |
| | 仅文本（总是） | 校对菜单项目显示已选菜单项目 / 按钮仅限文本（图标将不会显示）。 |
| | Text Only (in Menus) | 仅当菜单条目选中时有效。校对菜单项目显示已选菜单项目 / 按钮仅限文本（图标将不会显示）。 |
| | 图片和文本 | 校对菜单项目显示已选菜单项目 / 按钮仅限文本图标。 |
| | 开始一个组 | 在选中菜单项目 / 按钮之前嵌入分隔符。 |
| 最近使用过的 | 该版本里不支持此项目。 | |

图 A-56. Add Command Dialog Box



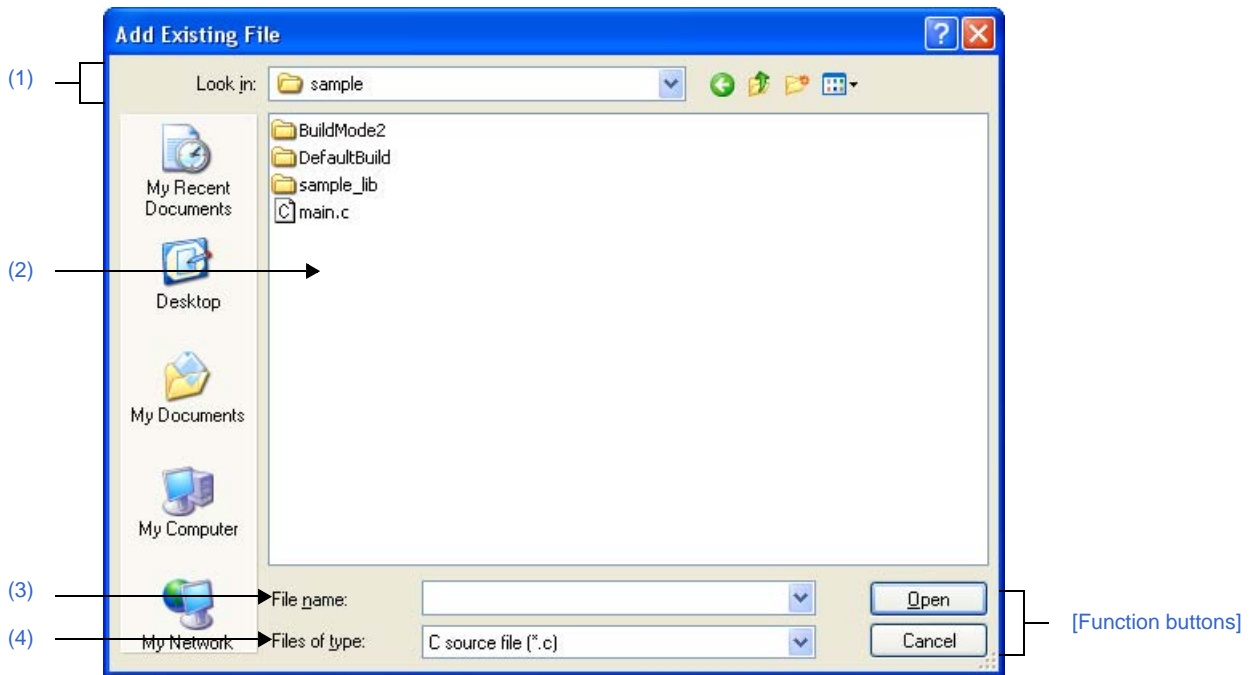
[功能按钮]

| 按钮 | 功能 |
|--------|--|
| 复位 ... | 将在 [Choose a menu or toolbar to rearrange] 区中已选中的，分类下的菜单条目或工具栏按钮布局到恢复默认状态。 |
| Close | 完成按钮的排布并关闭该对话框。 |

添加现有的文件对话框

该对话框用于选择现有的文件 e 并添加到工程中。

图 A-57. 添加现有的文件对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [Function buttons]

[如何打开]

- 通过 [File] 菜单，选择 [Add] >> [Add File...]
- 在工程树面板上选择 Project 节点、Subproject 节点、File 节点或者文件，并从 [Add File...] 快捷菜单中选择 [Add] >> 。

[各部分说明]

(1) [Look in] 区

选择添加到已有工程中的文件所在文件夹。
工程文件夹默认为选中。

(2) 文件列表区域

文件列表与 [Look in] 和 [Files of type] 里的选择项匹配如下所示。

(3) [File name] 区域

指定添加到工程的文件的文件名。

(4) [Files of type] 区域

指定添加到工程的文件的类型。

| | |
|-------------|---------|
| C 源文件 (*.c) | C 语言源文件 |
|-------------|---------|

| | |
|----------------------|-----------|
| 头文件 (*.h; *.inc) | 头文件 |
| 汇编文件 (*.asm) | 汇编语言源文件 |
| 链接指令文件 (*.dr; *.dir) | 链接指令文件 |
| 变量和函数信息文件 (*.vfi) | 变量和函数信息文件 |
| 库文件 (*.lib) | 库文件 |
| 目标文件 (*.rel) | 目标文件 |
| 文本文件 (*.txt) | 文本格式 |
| 所有文件 (*.*) | 所有格式 (默认) |

[Function buttons]

| 按钮 | 功能 |
|--------|------------|
| Open | 添加指定文件到工程。 |
| Cancel | 关闭此对话框。 |

文件夹浏览对话框

该对话框用于选择文件夹并可在调用请求下检索文件夹。

图 A-58. 文件夹浏览对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在查找和替换对话框中，点击 [Whole Search] 标签中的 [...] 按钮，或点击 [Whole Replace] 标签中的 [...] 按钮。
- 在添加文件对话框中，点击 [File location] 区域的 [...] 按钮。
- 在路径编辑对话框中，点击路径编辑区域的 [...] 按钮。
- 通过属性面板，选择下列属性，然后点击 [...] 按钮。
 - 通过 [Common Options] 标签，[Intermediate file output folder] 在 [Output File Type and Path] 分项中，[Output folder] 在 [Frequently Used Options(for Link)] 分项中，[Output folder for hex file] 在 [Frequently Used Options(for Object Convert)] 分项中，且 [Temporary folder] 在 [Others] 分项中。
 - 通过 [Link Options] 标签，[Others] 分项中的 [Output folder]。
 - 通过 [Object Convert Options] 标签，[Hex File] 分项中的 [Output folder for hex file]。
 - 通过 [Create Library Options] 标签，[Output File] 分项中的 [Output folder]。
 - 通过 [Variables/Functions Relocation Options] 标签，在 [Output File] 分项中的 [Output folder for variables/functions information file]。

[各部分说明]**(1) 消息区**

显示关于在此对话框中选择文件夹的消息。

(2) Folder location 区

选择文件夹设置在对话框的调用方中。

默认情况下，选定设置调用方的文件夹。

备注 当此区域为空，或者键入的路径不存在，则由 "C:\Documents and Settings*user name*\My Documents" 取代选择。

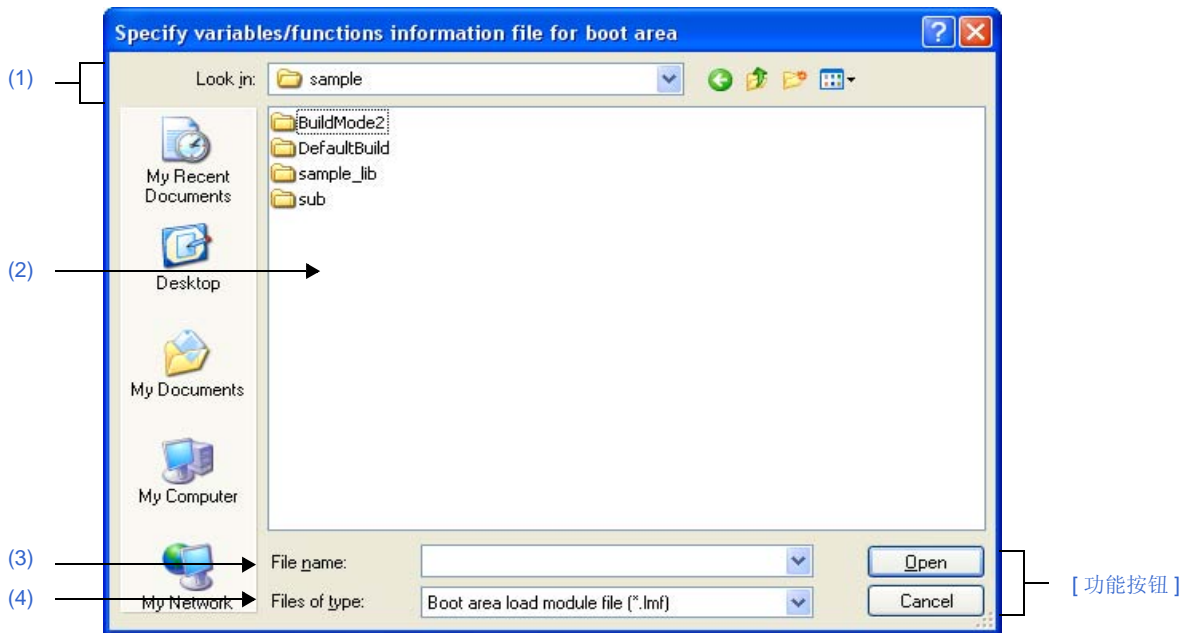
[功能按钮]

| 按钮 | 功能 |
|--------|--|
| 建立新文件夹 | 在所选文件夹的根目录创建新文件夹。 默认文件夹名为 "New category"。 |
| OK | 指定的文件夹路径设置到从对话框调用的区域。 |
| Cancel | 关闭此对话框。 |

设定启动区变量 / 函数信息文件对话框

该对话框用于选择引导区变量 / 函数信息文件并设置在对话框的调用请求中。

图 A-59. 设定启动区变量 / 函数信息文件对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在属性面板上，选择下列属性，然后点击 [...] 按钮。
- 通过 [Compile Options] 标签，在 [Variables/Functions Information File] 分项中的 [Variables/functions information file for boot area]。

[各部分说明]

(1) [Look in] 区

选择文件夹中的文件并设置在对话框中的调用方中。
工程文件夹默认为选中。

(2) 文件列表区域

显示与 [Look in] 区和 [Files of type] 区里的选择区匹配的文件列表。

(3) [File name] 区域

设定文件名并设置在对话框中的调用方上。

(4) [Files of type] 区域

设定文件类型并设置在对话框中的调用方上。

| | |
|-------------------------|------------------|
| 启动区的变量 / 函数信息文件 (*.vfi) | 引导区域的变量 / 函数信息文件 |
|-------------------------|------------------|

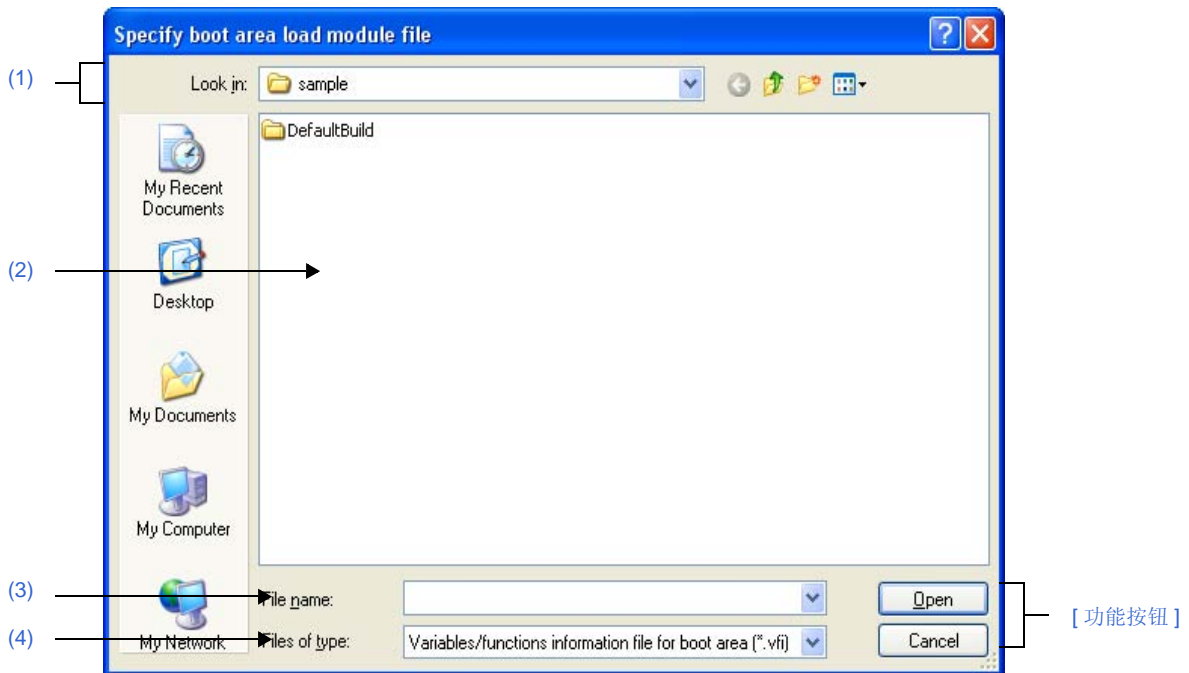
[功能按钮]

| 按钮 | 功能 |
|--------|------------------|
| Open | 设置指定文件在对话框的调用方中。 |
| Cancel | 关闭此对话框。 |

设定引导区载入模块文件对话框

该对话框用于选择引导区加载模块文件并设置在对话框的调用请求中。

图 A-60. 设定引导区载入模块文件对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 通过属性面板，选择下列属性，然后点击 [...] 按钮。
- 通过 [Link Options] 标签，在 [Device] 分项的 [Boot area load module file name]。

[各部分说明]

(1) [Look in] 区

选择文件夹中的文件并设置在对话框中的调用方中。
工程文件夹默认为选中。

(2) 文件列表区域

显示与 [Look in] 区和 [Files of type] 区里的选择区匹配的文件列表。

(3) [File name] 区域

设定文件名并设置在对话框中的调用方上。

(4) [Files of type] 区域

设定文件类型并设置在对话框中的调用方上。

| | |
|-------------------|----------------|
| 引导区载入模块文件 (*.lmf) | 引导区加载模块文件 (默认) |
| All files (*.*) | 所有格式 |

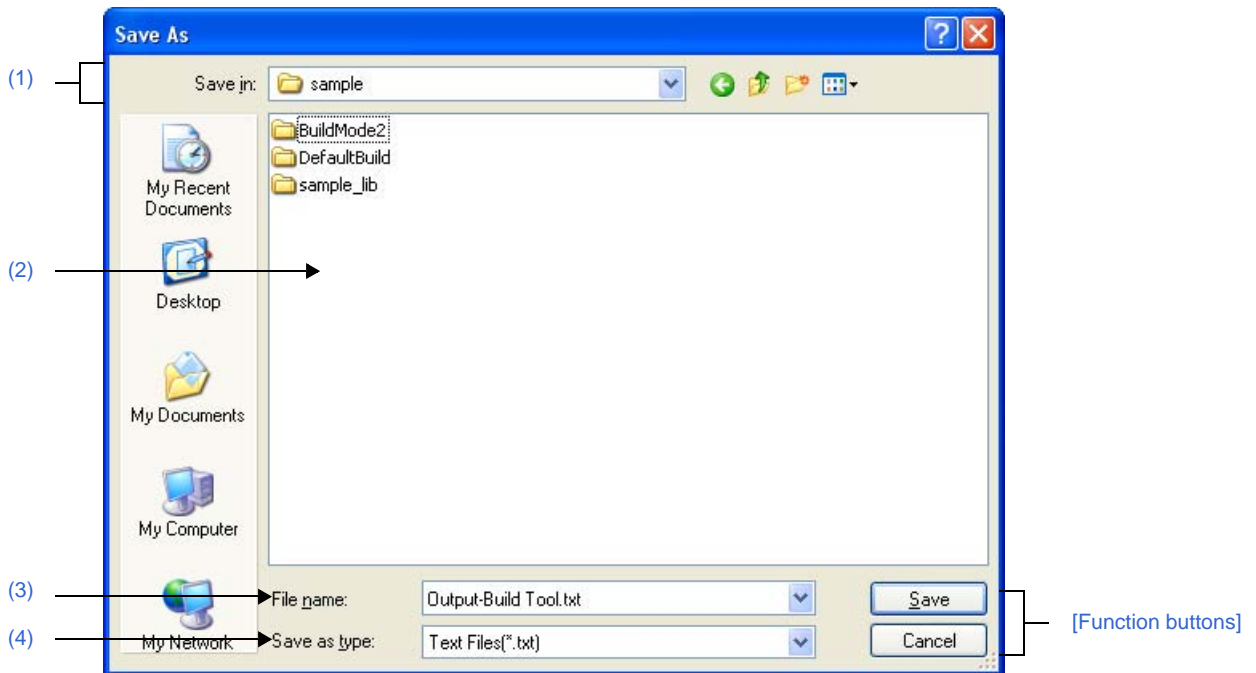
[功能按钮]

| 按钮 | 功能 |
|--------|------------------|
| Open | 设置指定文件在对话框的调用方中。 |
| Cancel | 关闭此对话框。 |

另存为对话框

该对话框用于将编辑中的文件或每个面板的内容保存到有名称的文件中。

图 A-61. 另存为对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [Function buttons]

[如何打开]

- 选中编辑器面板，然后从 [File] 菜单中选择 [Save file name As...]
- 选中输出面板，然后从 [File] 菜单中选择 [Save tab name As...]

[各部分说明]

(1) [Save in] 区

选择文件夹用以在文件中保存面板内容。

下列文件夹默认为选定。

(a) 在编辑器面板中

保存当前编辑中文件的文件夹。

(b) 在输出面板中

当文件第一次保存时，则需选定工程文件夹。在第二次后选定之前选中的文件。

(2) 文件列表区域

显示与 [Save in] 区和 [Save as type] 区内选择区相匹配的文件列表。

(3) [File name] 区域

设定要存储的文件名。

(4) [Save as type] 区**(a) 在编辑器面板中**

根据当前编辑文件的类型，显示下列文件类型。

| | |
|---------------------------|-----------|
| 文本文件 (*.txt) | 文本格式 |
| C 源文件 (*.c) | C 语言源文件 |
| 头文件 (*.h; *.inc) | 头文件 |
| 汇编文件 (*.asm) | 汇编语言源文件 |
| 链接指令文件 (*.dr; *.dir) | 链接指令文件 |
| 变量和函数信息文件 (*.vfi) | 变量和函数信息文件 |
| Map 文件 (.map) | 结构文件 |
| 符号表格文件 (.sym) | 符号表格文件 |
| Hex 文件 (.hex; .hxb; .hxf) | Hex 文件 |

(b) 在输出面板中

显示下列文件类型。

| | |
|--------------|------|
| 文本文件 (*.txt) | 文本格式 |
|--------------|------|

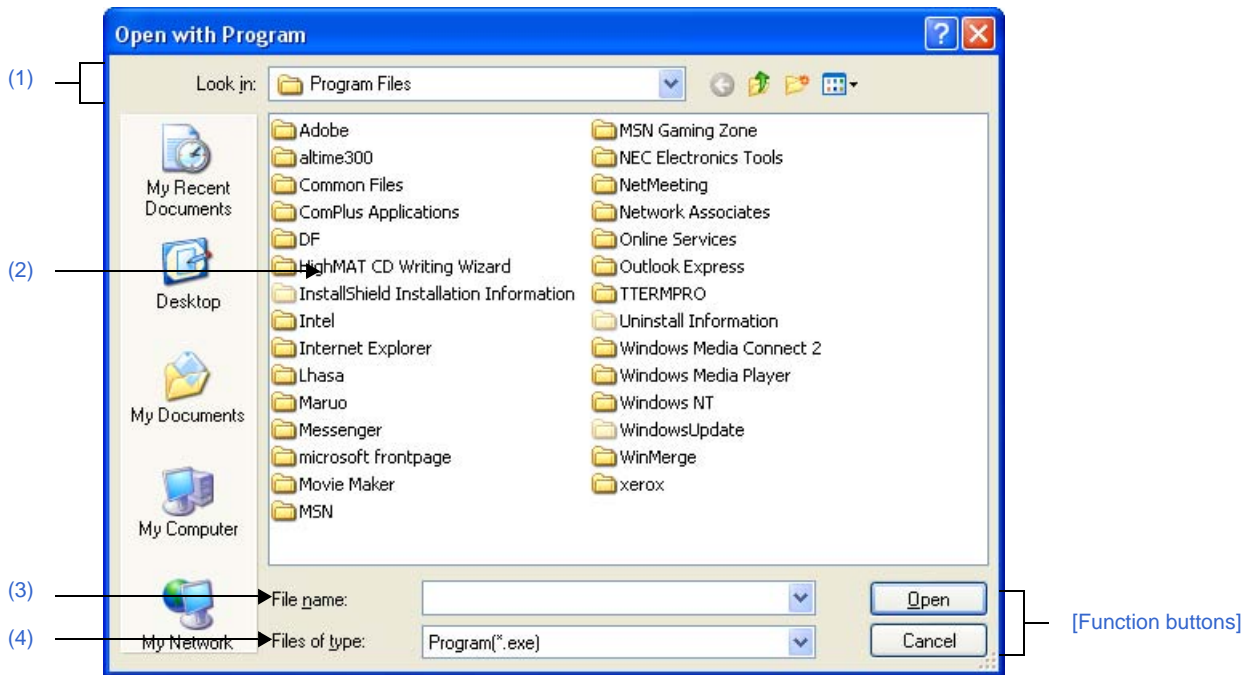
[Function buttons]

| 按钮 | 功能 |
|--------|-------------|
| Save | 以指定文件名保存文件。 |
| Cancel | 关闭此对话框。 |

打开程序对话框

该对话框用于为工程树中打开的已选文件选择应用程序。

图 A-62. 打开程序对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [Function buttons]

[如何打开]

- 在工程树面板上选择文件，然后通过快捷菜单选择 [Open with Selected Application...]

[各部分说明]

(1) [Look in] 区

选择存放打开文件的应用程序的文件夹。

程序文件夹默认选定 (用于 Windows XP, "C:\Program Files")

(2) 文件列表区域

显示与 [Look in] 区和 [Files of type] 区中选择区域匹配的文件列表。

(3) [File name] 区域

设定打开文件应用程序的可执行文件名。

(4) [Files of type] 区域

设定打开文件应用程序的可执行文件类型。

| | |
|--------------|------------|
| 程序文件 (*.exe) | 可执行格式 (默认) |
| 所有文件 (*.*) | 所有格式 |

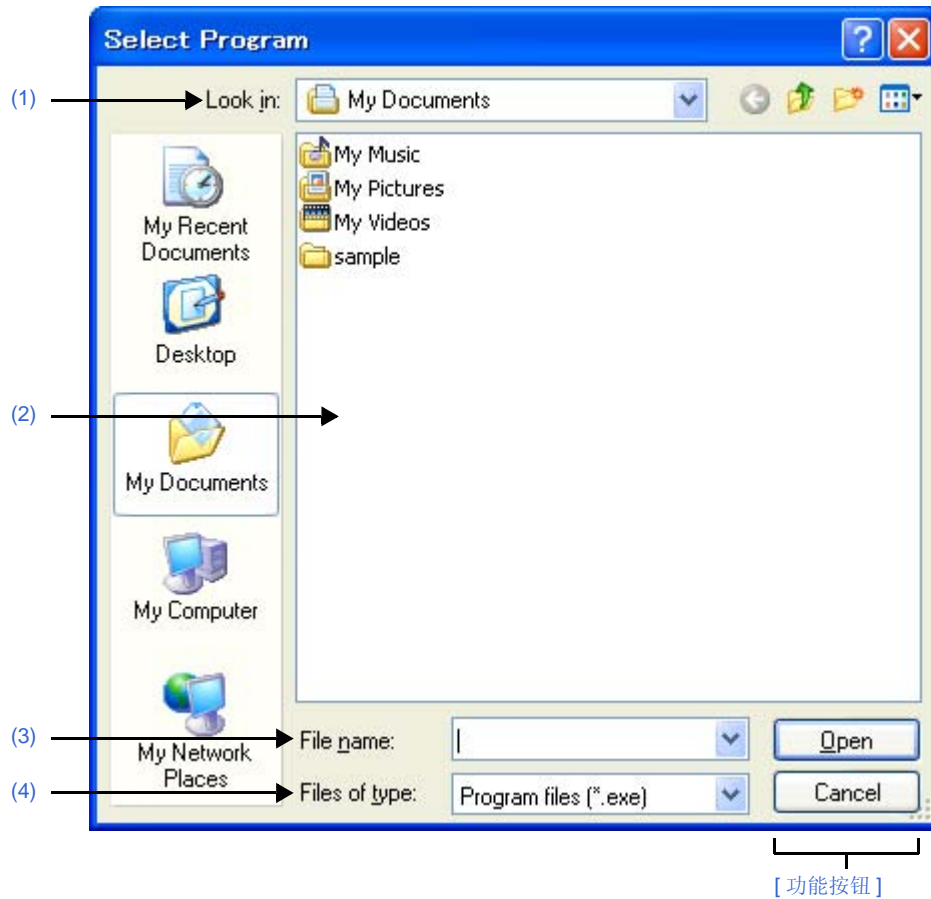
[Function buttons]

| 按钮 | 功能 |
|--------|---------------|
| Open | 用设定的应用程序打开文件。 |
| Cancel | 关闭此对话框。 |

选择程序对话框

此对话框用于选择外部工具的可执行文件。

图 A-63. 选择程序对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在选项对话框的 [General - External Tools] 分项中，在新注册区域点击 [...] 按钮。

[各部分说明]

(1) [Look in]

从下拉式列表中选择外部工具可执行工具的位置（文件夹）进行注册。

(2) List of files

此区域显示在 [Look in] 和 [Files of type] 中已选择匹配条件的文件列表。

(3) [File name]

指定外部工具可执行文件的文件名来注册。

(4) [Files of type]

从下列下拉式列表中选择外部工具可执行文件的类型进行注册。

| | |
|-----------------------|------------|
| Program files (*.exe) | 可执行格式 (默认) |
| All files(*.*) | 所有格式 |

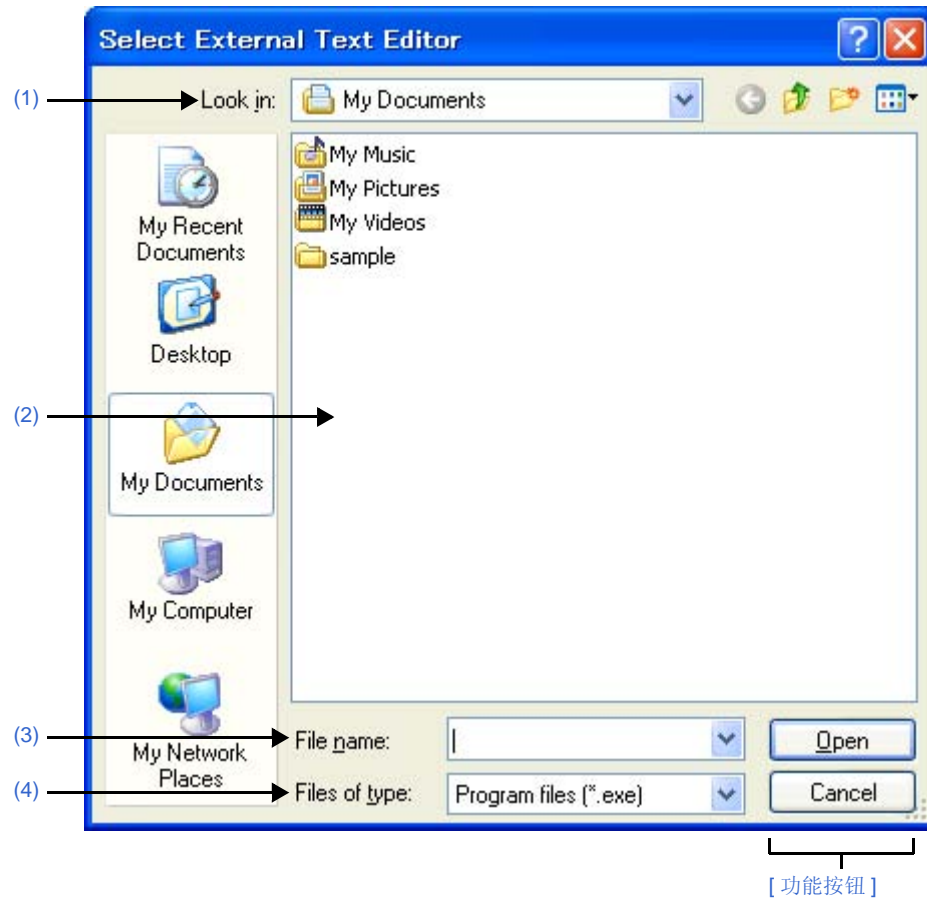
[功能按钮]

| 按钮 | 功能 |
|--------|-----------------|
| Open | 在选项对话框中设定选中的文件。 |
| Cancel | 忽略设置并关闭此对话框。 |

选择外部文本编辑器对话框

此对话框用于选择外部文本编辑器的可执行文件。

图 A-64. 选择外部文本编辑器对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在选项对话框的 [General - Text Editor] 分项下，点击 [External text editor] 区的 [Browse...] 按钮。

[各部分说明]

(1) [Look in]

从下拉式列表中选择外部文本编辑器可执行文件的位置（文件夹）进行注册。

(2) List of files

此区域显示在 [Look in] 和 [Files of type] 中已选择匹配条件的文件列表。

(3) [File name]

指定外部文本编辑器可执行文件的文件名。

(4) [Files of type]

从下列下拉式列表中选择外部文本编辑器可执行文件的类型进行注册。

| | |
|-----------------------|-----------|
| Program files (*.exe) | 可执行格式（默认） |
| All files (*.*) | 所有格式 |

[功能按钮]

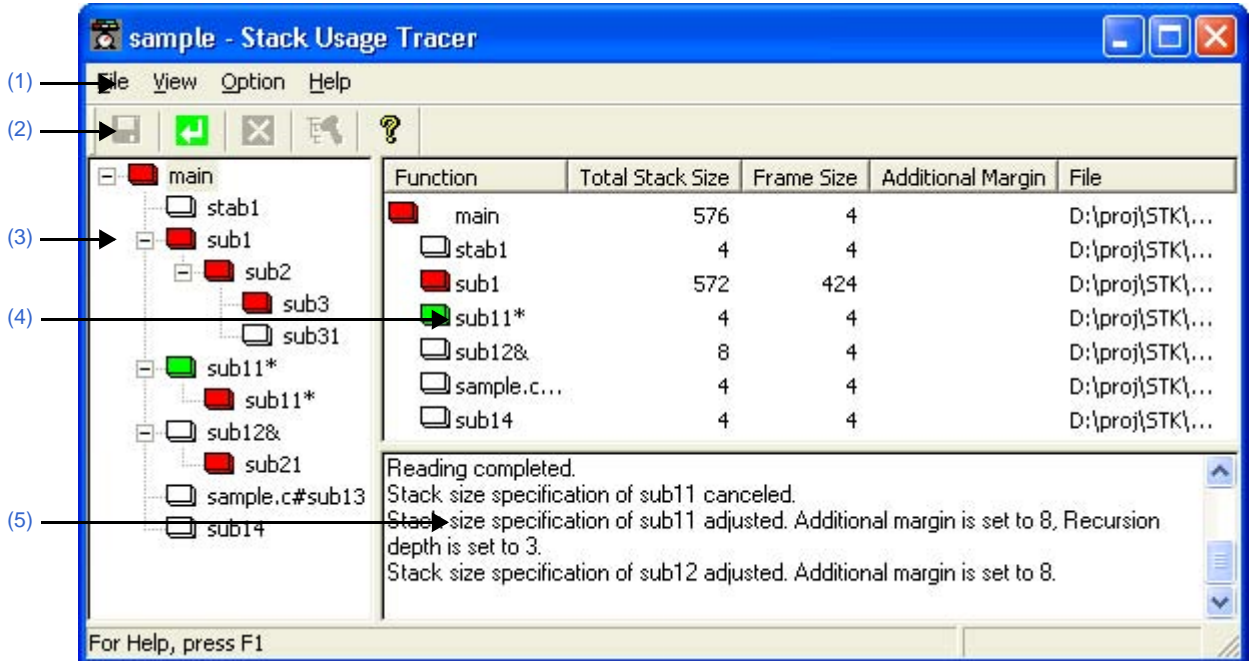
| 按钮 | 功能 |
|--------|----------------------------------|
| Open | 在 选项对话框 中设定选中文件。 |
| Cancel | 忽略设置并关闭此对话框。 |

堆栈用法跟踪窗口

当堆栈用法跟踪器被运行时首先打开此窗口。

使用此窗口去校验和修改应用在每个基础功能中的大量堆栈。

图 A-65. 堆栈用法跟踪窗口



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [Caution]

[如何打开]


- 通过 [Tool] 菜单，选择 [Startup Stack Usage Tracer]。

[各部分说明]

(1) 菜单条

这区域包含下列菜单项。




(a) [File] 菜单

| | |
|--|---|
| Save Call Chain with Maximum Stack from Selected Function... | 打开另存为对话框以将在树状显示区 / 列表显示区选中的具有最大堆栈大小调用关联保存到结果输出文件中。与此按钮有着相同规则的  功能。 |
| Save All Call Chains from Selected Function... | 打开另存为对话框将树状显示区 / 列表显示区已选功能的所有调用关联输出到结果文件中。 |
| Save Call Chain with Maximum Stack from Every Root... | 打开另存为对话框将树状显示区 / 列表显示区显示的占用最大堆栈大小的功能调用关联输出到结果文件中。 |
| Save All Call Chains from Every Root... | 打开另存为对话框以将在树状显示区的全部函数的所有调用关联输出到结果文件中。 |


| | |
|---------------------------------------|--|
| Load Stack Size Specification File... | 打开 打开对话框 并载入指定文件的堆栈大小。 |
| Save Stack Size Specification File... | 打开另存为对话框用于保存 调整堆大小对话框 产生的操作结果 (e.g. 改变的功能信息) 以及特定文件的单个堆栈大小。 |
| 退出 sk78K0R | 关闭此窗口。 |

备注 输出结果文件只能以文本格式 (*.txt) 或者 CSV 格式 (*.csv) 来保存。


(b) [View] 菜单

| | | |
|--------------|---|--|
| 重新计算堆栈大小 | 重新计算全部堆栈大小。 与此按钮有着相同规则的  功能。 | |
| 停止 | 强行停止运行堆栈用法跟踪器 (e.g. 重新计算全部堆栈大小)。 与此按钮有着相同规则的  功能。 | |
| Sort List by | 在列表显示区改变功能显示命令。 | |
| | 功能名称 | 以功能名称排序。 |
| | 图标类型 | 通过图标显示优先级排序 (高:  到低: )。 |
| | 堆栈大小 | 以总堆栈大小排序 |
| | 帧大小 | 以帧大小排序 |
| | 附加空间 | 以附加空间排序 |
| | 文件名 | 以文件名称排序。 |

(c) [Option] 菜单






| | |
|---|---|
| Stack Size Unknown / Adjusted Function Lists... | 打开 堆栈大小未知 / 校准函数列表对话框 显示一系列未知帧大小的功能, 功能信息 (附加空间, 递归深度或调用功能) 已被修改, 并且堆栈用法跟踪器的功能被强行设置了附加空间。 |
| Adjust Stack Size... | 打开此 调整堆大小对话框 改变选定的功能信息 (附加空间、递归深度以及调用功能) 在树结构显示区 / 列表显示区。 此对话框被用于改变此选定功能的信息 (附加空间、递归深度以及调用功能)。 与此按钮有着相同规则的  功能。 |
| 复位功能 | 恢复选定功能的信息 (附加空间、递归深度、以及调用功能) 为默认值。 如果所有选定功能恢复默认值, 此按钮将为设成灰色。 |
| 复位所有功能 | 恢复所有功能的信息 (附加空间、递归深度以及调用功能) 为默认值。 如果所有功能恢复默认值, 此按钮将被设成灰色。 |

(d) [Help] 菜单

| | |
|---------------|---|
| sk78k0R Help | 显示此窗口的在线帮助。 与此按钮有着相同规则的  功能。 |
| 关于 sk78k0R... | 打开堆栈用法跟踪器的版本信息对话框。 |

(2) 工具栏






这部分包含下列按钮。



| | |
|---|--|
|  | 打开另存为对话框以将在树状显示区 / 列表显示区选中的具有最大堆栈大小调用关联保存到结果输出文件中。 功能与在 [File] 菜单中选定 [Save Call Chain with Maximum Stack from Selected Function...] 是相同的。 |
|  | 重新计算全部堆栈大小。功能与在 [View] 菜单中选定 [Recalculate Stack Size] 是相同的。 |
|  | 强行停止运行堆栈用法跟踪器 (e.g. 重新计算全部堆栈大小)。 功能与在 [View] 菜单中选定 [Stop] 是相同的。 |
|  | 打开调整堆栈大小对话框改变此选定的功能信息 (附加空间、递归深度以及调用功能) 在树结构显示区 / 列表显示区。 功能与在 [Option] 菜单中选定 [Adjust Stack Size...] 是相同的。 |
|  | 显示此窗口的在线帮助。 功能与在 [Help] 菜单中选定 [sk78k0R Help] 是相同的。 |

(3) 树状显示区

功能的调用关系用树状结构格式表示。

在此表之下说明的是这些图标的含义，这些图标显示在字符串左边被用来描述有关功能名称。

| | |
|---|--|
|  | 此功能可以直接通过占有最大总堆栈大小来调用指定功能 |
|  | 信息 (附加空间、递归深度、以及调用功能) 已修改通过调整堆栈大小对话框或者特定文件的堆栈大小。 |
|  | 重现功能 |
|  | 堆栈用法跟踪器不需要获得此功能的任何堆栈信息 |
|  | 除上面之外 |

备注 图标显示优先级是从高： 到低：。

(a) 快捷菜单

在此区域选择功能，并单击鼠标右键。关于此快捷菜单的描述如下。






| | |
|----------------------|---|
| Adjust Stack Size... | 打开调整堆栈大小对话框用于改变这个选定功能的信息 (附加空间、递归深度以及调用功能)。 |
|----------------------|---|

(4) 列表显示区

以列表格式显示单个功能的堆栈信息（功能名称、总堆栈大小、帧大小、附加空间、和文件名）。



| | |
|--------|---|
| 功能 | 显示功能名称。 注意，此区域只显示第 1 层（选定功能）和第 2 层（通过选定功能直接功能调用）的功能。 |
| 全部堆栈大小 | 显示 总堆栈大小（包含调用功能的堆栈大小；用字节表示）。 |
| 帧大小 | 显示帧大小（不包含调用功能的堆栈大小；用字节表示）。 |
| 附加空间 | 显示强制占用到帧大小中的值（用字节数表示）。 |
| 文件 | 显示文件名称。 |

在此表之下说明的是这些图标的含义，这些图标显示在字符串左边用来描述有关功能名称。

| | |
|---|---|
|  | 此功能可以直接通过占有最大总堆栈大小来调用指定功能 |
|  | 信息（附加空间、递归深度、以及调用功能）已修改通过调整堆大小对话框或者特定文件的堆栈大小。 |
|  | 重现功能 |
|  | 堆栈用法跟踪器不需要获得此功能的任何堆栈信息 |
|  | 除上面之外 |

(a) 快捷菜单

在此区域选择功能，并单击鼠标右键。关于此快捷菜单的描述如下。

| | | |
|----------------------|---|---|
| Adjust Stack Size... | 打开调整堆大小对话框用于改变这个选定功能的信息（附加空间、递归深度以及调用功能）。 | |
| Sort List by | 在列表显示区改变功能显示命令。 | |
| | 功能名称 | 以功能名称排序。 |
| | 图标类型 | 通过图标显示优先级排序（高：  到低：  ）。 |
| | 堆栈大小 | 以总堆栈大小排序 |
| | 帧大小 | 以帧大小排序 |
| | 附加空间 | 以附加空间排序 |
| | 文件名 | 以文件名称排序。 |

(5) 消息显示区

显示堆栈用法跟踪器的操作日志。

[Caution]**- 汇编文件**

堆栈用法跟踪器计算总堆栈大小是通过从汇编文件输出、C 编译器的中间文件以及添加的调试信息中收集。结果为的是在应用堆栈用法跟踪器的功能层获得堆栈信息，这些信息是配置编译选项输出到 " 带有调试信息的编译文件 " 所必须的。





- 静态计时分析

堆栈用法跟踪器在启动组上进行静态分析，且在它的主窗口上显示与调用有关的功能与功能层堆栈之间的信息。因此改变与调用有关功能与功能层堆栈之间的信息（比如：添加文件、改变编译选项或修改源代码）将不会在此窗口中反应。

- 功能分析

堆栈用法跟踪器只分析带有调试信息输出的汇编文件，通过 C 编译器编译的中间文件以及通过 **build tool** 构建的库文件。因此在汇编文件中写入功能是通过用户和用户没有经过分析而建立的库文件。基于此原因，这些文件的信息必须通过使用 **调整栈堆大小对话框** 来做设置。

- 图标显示颜色

显示优先级（高： 到低：）分配给显示在窗口的树状结构显示区 / 列表显示区的图标。因此即使  图标（功能直接调用占用最大总堆栈大小的相类似的功能）显示，带有相对低优先级别的信息，如： 图标（帧大小未知）将会用 **GUI** 隐藏起来。

- 确定最大的堆栈大小

当堆栈用法跟踪器查找占用最大堆栈大小的路径时，假定功能没有被分析过并占有 0 容量的堆栈。因此，当确定最大堆栈大小时，必须确认没有任何功能在 **堆栈大小未知 / 校准函数列表对话框** 中的 **[Unknown Functions]** 之下。

- 重现功能的树状结构显示

窗口的树状显示区只显示不超过秒的返回功能调用。因此，第三级以及以下级的调用被隐藏。

- 功能库的查找、退出与排序

堆栈用法跟踪器把查找，退出与排序看作未知功能，即使它们在 **build tool** 所允许的库文件中。因此如果在使用此功能的话，必须设置相关信息（例如：递归深度和调用功能）在 **调整栈堆大小对话框** 中。

- 调用功能

堆栈用法跟踪器只允许在 **调整栈堆大小对话框** 中已添加的下列“调用功能”类型：功能包含 C 源文件，以及明确的调用功能（不被使用指针调用）。因此在 **调整栈堆大小对话框** 中的 **[All Functions]** 段只显示满足以上条件的功能。

- 复合功能的功能调用

堆栈用法跟踪器把复合功能的功能调用堆栈信息看作是唯一的。因此当这些功能有其他功能调用他们时，不可能改变堆栈信息。

示例 如果通过在树状显示区的 **func1** 选择子功能调用并打开 **调整栈堆大小对话框**，此变化也会在子功能调用 **func2** 中反应。

```
int    sub ( int i );
void   func1 ( void );
void   func2 ( void );

void main ( void ) {
    func1 ( );
    func2 ( );
}
int sub ( int i ) {
    i++;
    return ( i );
}

void func1 ( void ) {
    int ret, i = 0;
    ret = sub ( i );
}

void func2 ( void ) {
    int ret, i = 100;
    ret = sub ( i );
}
```

- 在 C 源文件中的 ASM 标记

如果 C 源程序包含 ASM 标记，则堆栈用法跟踪器会输出下列消息："W9432 : Illegal format in file (path name : line number)"。如果其出现，则通过禁止代码错误使用 #if 声明或类似有关注释修正问题。

- 间接返回功能调用

如果返回路径包含复合功能，堆栈大小可能被错误计算。

示例 假设返回功能 "func_rec1/func_rec2" 的帧大小是 8 个字节，又如果 "func_rec1/func_rec2" 的返回深度被设置为 3 在[调整栈堆大小对话框](#)中，因此 func1 的堆栈大小将被正确计算为 "(8 + 24) * 3"，func2 的堆栈大小被计算为 "8 * 3"，跳过调用 func_rec1。

```
void    func_rec1 ( int i );
void    func_rec2 ( int i );
void    func1 ( void );
void    func2 ( void );

void main ( void ) {
    func1 ( );
    func2 ( );
}

void func_rec1 ( int i ) {
    func_rec2 ( i );
}

void func_rec2 ( int i ) {
    if ( i ) {
        func_rec1 ( i - 1 );
    }
}

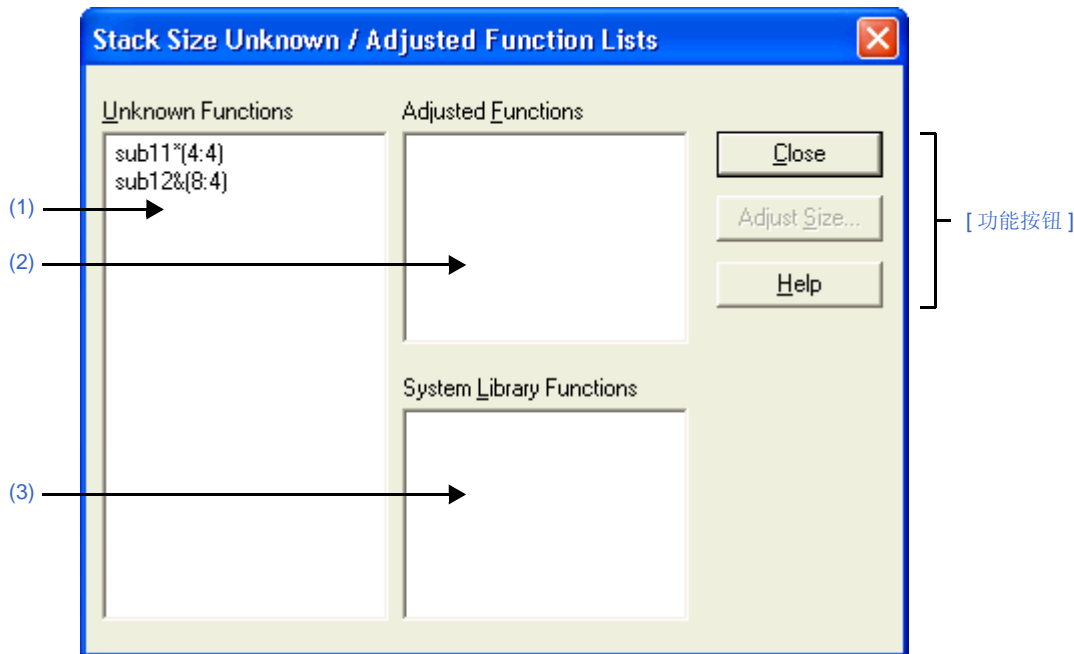
void func1 ( void ) {
    func_rec1 ( 2 );
}

void func2 ( void ) {
    func_rec2 ( 2 );
}
```

堆栈大小未知 / 校准函数列表对话框

该对话框用于显示堆栈用法跟踪程序不能获取的堆栈信息的函数列表；经过考虑地改变信息的函数列表（附加空间、递归深度和调用功能）以及堆栈用法跟踪程序强制设定附加空间的函数列表。

图 A-66. 堆栈大小未知 / 校准函数列表对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在堆栈用法跟踪窗口上，通过 [Option] 菜单选择 [Stack Size Unknown / Adjusted Function Lists...]

[各部分说明]

(1) [Unknown Functions]

显示“未知函数”列表，即堆栈用法跟踪程序不能获取堆栈信息的函数列表。该区通常以下列格式显示未知函数。

函数名（总堆栈大小：帧大小）

- 备注 1.**
1. 如果未知函数是用汇编语言写的，则删除前置附加到符号名称下的下划线（ ），然后名称用方括号（[]）括起来；并作为函数名显示。
 2. 如果未知函数是递归函数，则星号（*）附加到函数名称末尾。
 3. 如果未知函数包括使用函数指针间接调用的函数，则“&”附加到函数名称末尾。
 4. 如果未知函数是静态函数，则“file name#”附加到函数名称末尾。

(2) [Adjusted Functions]

信息显示通过调整堆大小对话框或堆栈大小设定的文件来修改信息（附加空间、递归深度以及调用功能）的函数列表。该区通常以下列格式显示修改过（“调整过”）的函数。

函数名（总堆栈大小：帧大小：附加空间）

- 备注 1.** 如果调整后的函数是用汇编语言编写，则删除前置附加到符号名称下的下划线（_），然后名称用方括号（[]）括起来；并作为函数名显示。
2. 如果调整后的函数是递归函数，则星号（*）附加到函数名称末尾。
 3. 如果调整后的函数包括使用函数指针间接调用的函数，则“&”附加到函数名称末尾。
 4. 如果调整后的函数是静态函数，则“file name#”附加到函数名称末尾。
 5. 如果仅在调整栈堆大小对话框中执行的操作添加“调用功能”，则该区的显示格式如下。
函数名 (总堆栈大小: 帧大小)

(3) [System Library Functions]

显示自动配置系统库中未知帧大小函数的函数列表，同时使用堆栈用法跟踪程序强制设置附加空间。该区通常以下列格式显示修改过的系统库函数。

函数名 (总堆栈大小:?:附加空间)

- 备注 1.** 删除前置附加到符号名称下的下划线（_），名称用方括号（[]）括起来；并作为函数名显示。
2. 适当的帧大小添加到对应的堆栈用法跟踪程序中的系统库函数内作为附加空间。

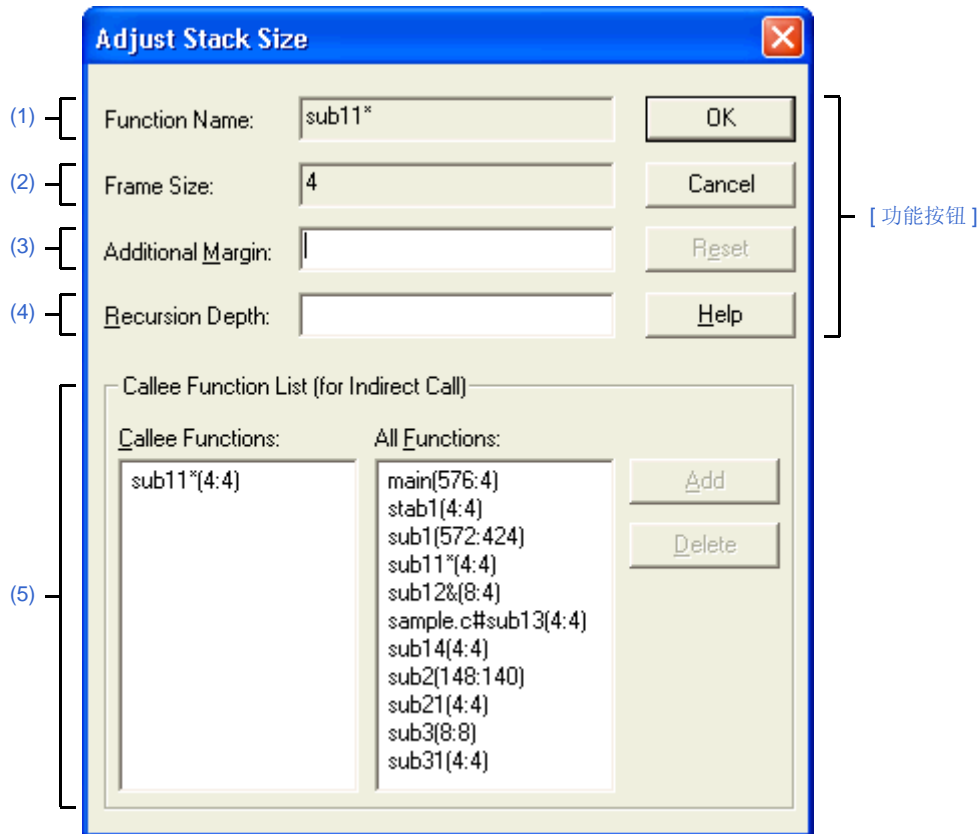
[功能按钮]

| 按钮 | 功能 |
|----------------|---|
| Close | 关闭此对话框。 |
| Adjust Size... | 打开调整栈堆大小对话框改变在 [Unknown Function]/[Adjusted Functions]/[System Library Functions] 中选择的函数信息 (附加空间, 递归深度和调用功能)。 |
| Help | 显示此对话框的在线帮助。 |

调整栈堆大小对话框

此对话框被用于改变此选定功能的信息（附加空间、递归深度以及调用功能）。


图 A-67. 调整栈堆大小对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在树结构显示区上 / 堆栈用法跟踪窗口的列表显示区，选择功能，然后通过 [Option] 菜单选择 [Adjust Stack Size...]
- 在树结构显示区上 / 堆栈用法跟踪窗口的列表显示区，选择功能，然后通过工具栏点击按钮 。
- 在树结构显示区上 / 堆栈用法跟踪窗口的列表显示区，选择功能，然后通过快捷菜单选择 [Adjust Stack Size...]
- 在 [Unknown Functions] 上 / 堆栈大小未知 / 校准函数列表对话框中的 [Adjusted Functions]/[System Library Functions], 选择功能，然后点击 [Adjust Size...] 按钮。

[各部分说明]

(1) [Function Name]

显示选定功能的功能名称。

- 备注**
1. 如果选择函数以汇编语言写入或者它是系统库函数，然后前置附加到符号名的下划线（ ）被删除，然后名称用方括号（`[]`）括起来；并作为函数名显示。
 2. 如果选择函数是递归函数，则星号（*）附加到函数名末尾。
 3. 如果选择函数包含使用函数指针间接调用的函数，则“&”附加到函数名末尾。
 4. 如果选择函数是静态函数，则“file name#”附加到函数名末尾。

(2) [Frame Size]

显示选择函数的帧大小（不包含调用函数的堆栈大小，以字节方式）。

备注 如果帧大小未知，则显示问号（?），如果其超过最大限制，则显示“SIZEOVER”。

(3) [Additional Margin]

设定强制添加到选择函数（以字节方式）的值为以“0x”或“0X”起始的十进制数或十六进制数。

(4) [Recursion Depth]

设定递归深度为以“0x”或“0X”起始的十进制数或十六进制数。

备注 如果选择的函数不是递归函数，则该条目无效。

(5) [Callee Function List (for Indirect Call)] 区

(a) [Callee Functions]

显示选择函数（使用函数指针或相同操作直接调用的函数）调用的函数的列表。

该区通常以下列格式显示调用函数。

函数名（总堆栈大小：帧大小：附加空间）

- 备注 1.**
1. 如果调用函数以汇编语言写入或者它是系统库函数，然后前置附加到符号名的下划线（ ）被删除，然后名称用方括号（`[]`）括起来；并作为函数名显示。
 2. 如果调用函数是递归函数，则星号（*）附加到函数名末尾。
 3. 如果调用函数包含使用函数指针间接调用的函数，则“&”附加到函数名末尾。
 4. 如果调用函数是静态函数，则“file name#”附加到函数名末尾。
 5. 通过单击 [Add] 按钮从 [All Functions] 主观添加的函数以在函数名末尾增加（+）符显示。

(b) [All Functions]

显示能够作为选择函数所调用的函数的列表（“调用函数”）。

该区通常以下列格式显示可以添加的函数。

函数名（总堆栈大小：帧大小：附加空间）

- 备注 1.**
1. 如果可以添加的函数以汇编语言写入或者它是系统库函数，然后前置附加到符号名的下划线（ ）删除，然后名称用方括号（`[]`）括起来；并作为函数名显示。
 2. 如果可添加的函数是递归函数，则星号（*）附加到函数名末尾。
 3. 如果可以添加的函数包含使用函数指针间接调用的函数，则“&”附加到函数名末尾。
 4. 如果可以添加的函数是静态函数，则“file name#”附加到函数名末尾。

(c) 按钮区

| | |
|--------|--|
| Add | <p>添加在 [All Functions] 中选择的函数到 [Callee Functions]。 如果没有在 [All Functions] 中选择函数，则该按钮无效。</p> |
| Delete | <p>从 [Callee Functions] 中删除在 [Callee Functions] 中选择的函数。 如果没有在 [Callee Functions] 中选择函数，则该按钮无效。</p> |

备注 如果函数名以 (+) 结束，则函数仅可以从 [Callee Functions] 中删除（函数从 [All Functions] 中通过主观单击 [Add] 添加）

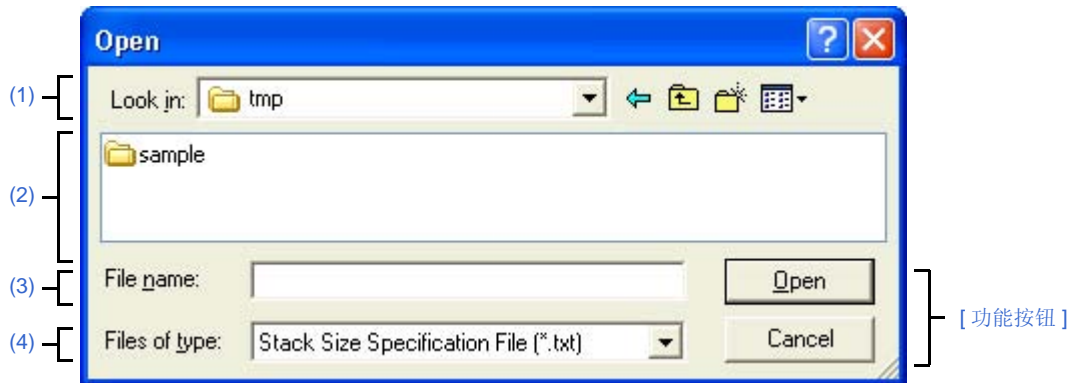
[功能按钮]

| 按钮 | 功能 |
|--------|--|
| OK | 在堆栈用法跟踪窗口中反应设置 / 并保存它们到目标文件 (*.prj)，然后关闭对话框。 |
| Cancel | 忽略设置并关闭此对话框。 |
| Reset | 恢复选定功能的信息 (附加空间、递归深度、以及调用功能) 为默认值。 如果所有选定功能恢复默认值，此按钮将为设成灰色。 |
| Help | 显示此对话框的在线帮助。 |

打开对话框

此对话框用于打开存在的堆栈大小说明文件。

图 A-68. 打开对话框



下列条目说明如下。

- [如何打开]
- [各部分说明]
- [功能按钮]

[如何打开]

- 在堆栈用法跟踪窗口上，通过 [File] 菜单选择 [Load Stack Size Specification File...]

[各部分说明]

(1) [Look in] 区

选择包括堆栈大小说明文件的文件夹。

(2) List of files

此区域显示在 [Look in] 区和 [Files of type] 区域中已选择匹配条件的文件列表。

(3) [File name] 区域

设定已打开堆栈大小说明文件的名称。

(4) [Files of type] 区域

选择文件类型去打开。

| | |
|------------------|------|
| 堆栈大小说明文件 (*.txt) | 文本格式 |
|------------------|------|

[功能按钮]

| 按钮 | 功能 |
|--------|--------------|
| Open | 打开指定文件。 |
| Cancel | 忽略设置并关闭此对话框。 |

附录 B 参考命令

本章节描述了包含在 `build tool` 里每条命令的详细说明。

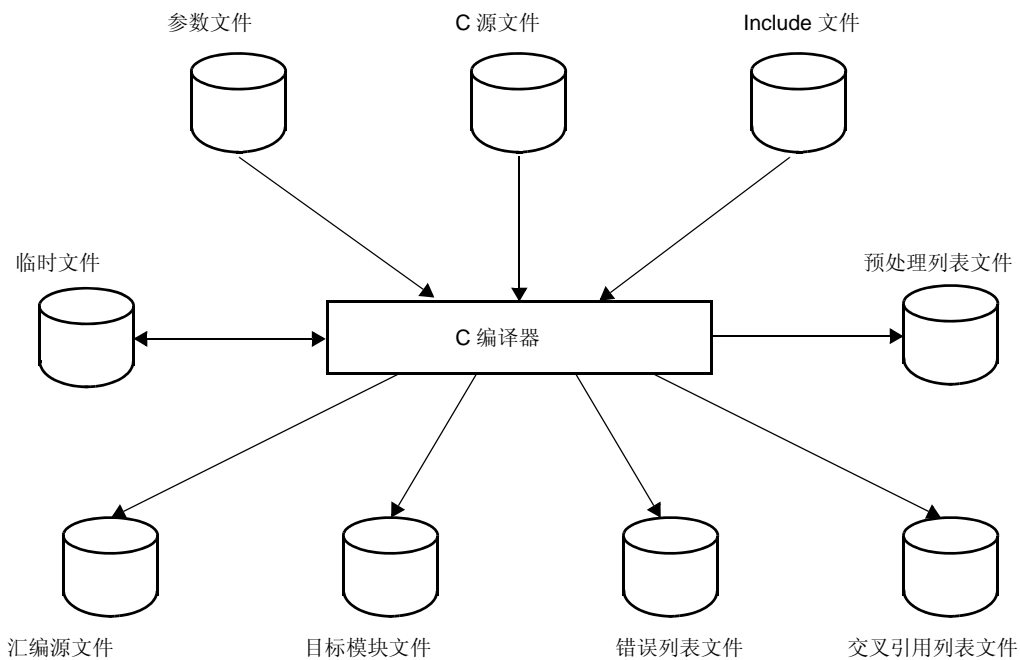
B.1 C 编译器

C 编译器输入由 C 语言编写的 C 源文件，转换它们为机器语言，并输出目标模块文件。经过编译后，输出汇编源文件以使用户使用汇编语言对其进行校对和修改。

根据编译选项，输出列表文件例如预处理列表、交叉引用列表和错误列表。

如果存在编译器错误，则输出错误信息到控制台和错误列表文件中。如果产生错误，不能输出除错误列表文件外的其余各类文件。

图 B-1. C 编译器的 I/O 文件



备注 如果存在编译错误，无法输出除了错误列表文件和交叉引用文件外的各类文件。编译结束没有错误产生时，给临时文件重新命名为适当的名称。如果编译结束有错误产生，则会删除临时文件。

B.1.1 I/O 文件

C 编译器的 I/O 文件如下所示。

关于输出文件的详细信息，请参阅“[3.1 C 编译器](#)”。

表 B-1. C 编译器的 I/O 文件

| 类型 | 文件名 | 说明 | 默认文件类型 |
|------|------------|---|--------|
| 输入文件 | C 源文件 | - C 语言编写的源文件（用户创建的文件） | .c |
| | Include 文件 | - 从 C 源文件引用的文件 - C 语言编写的文件（用户创建的文件） | .h |
| | 参数文件 | - 当用户想设定多重命令时，用户创建的文件在 C 编译器运行时不能在命令行中设定。 | .pcc |

| 类型 | 文件名 | 说明 | 默认文件类型 |
|--------|----------|--|-----------------------------|
| 输出文件 | 目标模块文件 | - 二进制映像文件包含机器语言信息, 以及与机器语言地址定位相关的重定位信息和符号信息。 | .rel |
| | 汇编源文件 | - 由编译器输出目标代码的 ASCII 映像文件 | .asm |
| | 预处理列表文件 | - 通过预处理指令例如 #include 输出列表文件 - ASCII 映像文件 | .ppl |
| | 交叉引用列表文件 | - 列表文件包含在 C 源文件中使用的函数名和变量名信息 | .xrf |
| | 错误列表文件 | - 列表文件包含源文件和编译器错误消息 ^{注意 1} | .cer .her .er .ecc |
| I/O 文件 | 临时文件 | - 编译时的中间文件 - 编译结束后没有产生错误时, 用适当的名称重新命名文件, 而编译结束后产生错误时则会删除文件。 | \$nn (固定的文件名) |

注 下列 4 种文件类型可用于错误列表文件。

| 文件类型 | 说明 |
|------|--|
| .cer | 含 C 源代码的错误列表文件相当于 *.c' 文件 (通过设定 -se 选项输出) |
| .her | 含 C 源代码的错误列表文件相当于 *.h' 文件 (通过设定 -se 选项输出) |
| .er | 含 C 源代码的错误列表文件相当于上述以外的文件 (通过设定 -se 选项输出) |
| .ecc | 含 C 源代码的错误列表文件相当于全部的源文件 (通过设定 -e 选项输出) |

B.1.2 功能

(1) 优化方法

在 CA78K0 中，实现创建有效的目标模块文件的优化方法。

支持的优化方法显示如下。

表 B-2. 优化方法

| 相位 | | 内容 | 示例 |
|-------|-----|------------------|--|
| 语法分析器 | (a) | 在编译过程中执行常量计算 | <code>a = 3 * 5 ;</code> <code>-> a = 15 ;</code> |
| | (b) | 基于逻辑表达式部分求值的真假判定 | <code>0 && (a b) -> 0</code> <code>1 (a && b) -> 1</code> |
| | (c) | 指针、数组等的偏移量计算。 | 在编译过程中计算偏移量。 |
| 代码生成器 | (d) | 寄存器管理 | 有效地利用未使用的寄存器。 |
| | (e) | 使用目标 CPU 的专用指令 | <code>a = a + 1 ;</code> <code>-></code> 使用 <code>inc</code> 指令 使用移动指令代替数组元素。 |
| | (f) | 使用短指令 | 如果存在相同操作的指令，使用较少字节的指令。 <code>mov a, #0</code> <code>-></code> <code>clrb a</code> |
| | (g) | 改变长跳跃指令为短跳跃指令 | 中间代码在预处理时输出。 |

| 相位 | 内容 | 示例 | |
|------|-----|----------------------------|--|
| 优化程序 | (h) | 删除公用的部分表达式。 | <code>a = b + c ; -> a = b + c ;</code> <code>d = b + c + e ; d = a + e ;</code> |
| | (i) | 出现指令循环 | <code>for (i = 0 ; i < 10 ; i++)</code> <code>{</code> <code> :</code> <code> a = b + c ;</code> <code> :</code> <code>}</code> <code> ?</code> <code>a = b + c ;</code> <code>for (i = 0 ; i < 10 ; i++)</code> <code>{</code> <code> :</code> <code>}</code> |
| | (j) | 删除不用的指令 | <code>a = a ; -> Delete</code> After “a = b;”, “a” is not referenced -> Delete (“a” 是自动变量) |
| | (k) | 删除备份 | <code>a = b ; -> c = b + d ;</code> <code>c = a + d ;</code> “a” 不再引用 (a 是自动变量)。 |
| | (l) | 在表达式中改变计算顺序 | 操作结果保留在寄存器内，并首先执行有效操作。 |
| | (m) | 配制存储设备 (临时变量) | 局部使用的变量分配到寄存器。 |
| | (n) | 窥孔优化 | 特定模式的替换 示例： <code>a * 1 -> a</code> , <code>a + 0 -> a</code> |
| | (o) | 降低计算的强度 | 示例： <code>a * 2 -> a + a</code> , <code>a << 1</code> |
| | (p) | 分配存储设备 (寄存器变量) | 快速分配数据到可使用的存储器。 示例：寄存器， <code>saddr</code> (仅当 <code>-qr</code> 选项被设定时) |
| | (q) | 跳转优化 (-qj 选项) | 整合连续的跳转指令为一条指令。 |
| | (r) | 寄存器分配 (-qv/-qr/-rd/-rs 选项) | 变量自动分配到寄存器。 |

备注 (a) ~ (g), (n) 以及 (o) 执行可忽略优化选项说明。
 (h) ~ (m), (q) 以及 (r) 中的优化，只在优化选项设定时执行。
 (p) 当 C 源代码中存在寄存器声明时执行。然而，当 - 设定 `-qr` 选项时，仅分配 `saddr` 区。
 关于优化选项，参阅 “[优化说明](#)”。

(2) ROM 化功能

ROM 化是指在系统执行时，复制存储在 ROM 中的外部变量的初始值到 RAM 中的处理过程。
 CA78K0 提供带 ROM 化处理过程的程序为启动程序。使用启动程序，消除启动中描述 ROM 化进程的问题。
 参阅 CubeSuite 78K0 编码用户使用手册中有关启动程序的内容。

(a) 如何在 ROM 上存储程序

在链接期间，链接启动程序、目标模块文件和库。启动程序初始化目标程序。

<1> s0r*.rel

这些是启动程序 (当在 ROM 中存储时)

包含初始化数据的复制程序，并标识初始数据的起始位置。在起始地址上添加标号“_@cstart”（符号）。

<2> cl0r*.lib

这些是附加在 CA78K0 中的库。

这些文件包含以下库。

- 运行时间库
附加“@@”作为运行时间库名称的起始符号。
- 标准库
附加“_”作为标准库名的起始符号。

<3> *.lib

这些库是由用户创建。

添加“_”到符号头部。

注意事项 CA78K0 提供不同种类的启动程序和库。参阅 CubeSuite 78K0 编码用户使用手册中关于启动程序和库的内容。

B.1.3 操作方法

(1) C 编译器启动

以下两种方法能用于启动 C 编译器。

(a) 从命令行启动

```
X:[path-name]>cc78k0r[Δoption]Csource-file-name [Δoption]
```

| | |
|---------|--|
| X | 当前驱动器名称 |
| 路径名 | 当前文件夹名 |
| cc78k0 | C 编译器的命令名称 |
| 选项 | 输入 C 编译器操作的详细指令。 当设定两个或两个以上编译器选项时，使用空格来分隔选项。在编译选项中不含空白（例如空格）时，设定文件名或子选项。编译选项中不区分大、小写字符。关于编译选项的详细信息，参阅“B.1.4 选项”。 带有空格的路径附在一对双引号（"）中。 |
| C 源文件名称 | 已编译的源文件名 带空格的文件名路径附在一对双引号（"）中。 |

示例 基于代码大小优先方式的输出汇编源文件（prime.asm）及执行优化，说明如下：

```
cc78k0r -cf1166a0 prime.c -aprime.asm -qx3
```

(b) 从参数文件启动

当启动 C 编译器所需数据没有放置在命令行上时，或者每次编译执行时重复指定了相同的编译选项，则可使用参数文件。

通过参数文件启动汇编程序，在命令行上设定参数文件选项（-f）。

通过下列参数文件启动 C 编译器：

```
X>cc78k0r[ΔCsource-file-name]Δ-fparameter-file-name
```

| | |
|-------|-------------------|
| -f | 参数文件说明选项 |
| 参数文件名 | 包含启动 C 编译器所需数据的文件 |

备注 使用编辑器创建参数文件。

参数文件内容的书写规则如下：

```
[[[Δ]option[Δoption] ... [Δ]Δ]] ...
```

- 如果 C 源文件名在命令行中遗漏，则在参数文件中仅可以设定一个 C 源文件名。
- C 源文件名也可以在选项设定后写入。
- 在参数文件中写入所有的编译选项以及在命令行中设定的输出文件名。

示例 使用编辑器创建参数文件 k0rmain.pcc，然后启动 C 编译器。

```
; parameter file
-cf1166a0 k0rmain.c -e -a
```

```
C>cc78k0r -fk0rmain.pcc
```

(2) 执行开始和结束消息

(a) 执行开始消息

当 C 编译器启动时，屏幕上显示执行启动消息。

```
78K0R C Compiler Vx.xx [xx xxx xxxxx]
Copyright(C) NEC Electronics Corporation xxxx
```

(b) 执行结束消息

如果从 C 编译器中没有检测到编译错误结果，则 C 编译器输出下列消息到屏幕上并返还控制权给主操作系统。

```
Target chip : uPD78F1166_A0
Device file : Vx.xx

Compilation complete, 0 error(s) and 0 warning(s) found.
```

如果从 C 编译器中检测到编译错误结果，则 C 编译器输出错误编号到屏幕并返还控制权给主操作系统。

```
prime.c(18) : CC78K0R warning W0745 : Expected function prototype
prime.c(20) : CC78K0R warning W0745 : Expected function prototype
prime.c(26) : CC78K0R warning W0622 : No return value
prime.c(37) : CC78K0R warning W0622 : No return value
prime.c(44) : CC78K0R warning W0622 : No return value

Target chip : uPD78F1166_A0
Device file : Vx.xx

Compilation complete, 0 error(s) and 5 warning(s) found.
```

如果在编译执行期间，C 编译器检测到致命错误致使编译无法继续，则 C 编译器输出消息到屏幕，取消编译并返还控制权给主操作系统。

示例 设定了不存在的编译选项。

```
C>cc78k0r k0rmain.c -s
```

```
78K0R C Compiler Vx.xx [xx xxx xxxx]
  Copyright(C) NEC Electronics Corporation xxxx

CC78K0R error F0018 : Option is not recognized '-s'
Please enter 'CC78K0R--' , if you want help messages.
程序终止。
```

在上述例子中，设定了不存在编译选项。产生错误，C 编译器则中止编译。

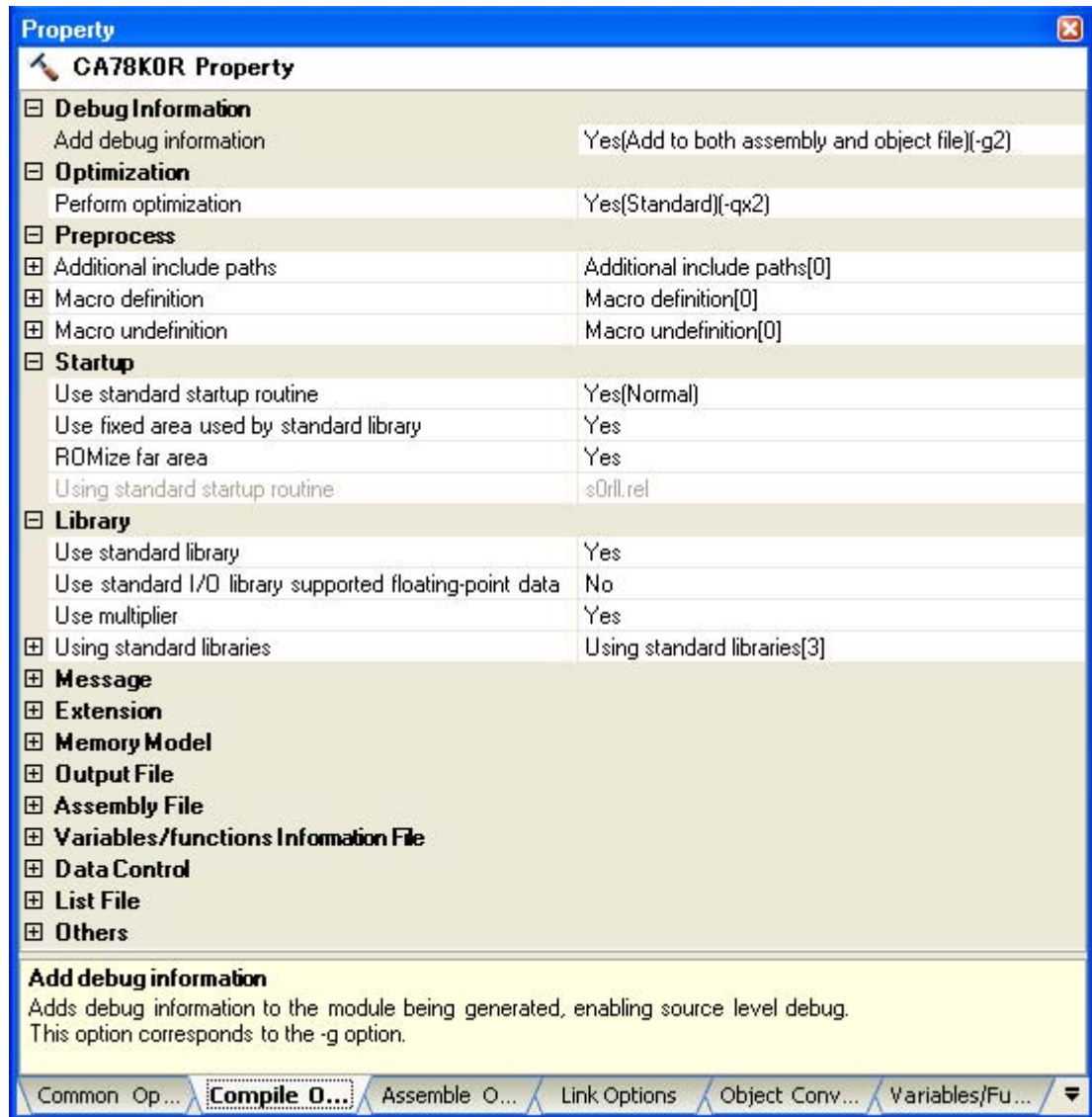
(3) 在 CubeSuite 中设置选项

本章节描述了通过 CubeSuite 如何设置编译选项。

通过 CubeSuite 的[工程树面板](#)，选择 Build Tool 节点。然后通过 [View] 菜单选择 [Property]。打开[属性面板](#)。再选择 [\[Compile Options\]](#) 标签。

可以通过在该标签中设定需要的属性来设置不同的编译选项。

图 B-2. 属性面板: [Compile Option] 标签



B.1.4 选项

(1) 类型

编译选项是 C 编译器操作的详细指令。

编译选项的类型和注解如下所示。

表 B-3. 编译选项

| 分类 | 选项 | 说明 |
|--------------|-----|--|
| 设备类型说明 | -c | 指定目标设备的类型。 |
| 目标模块文件创建设定 | -o | 指定目标模块文件的输出。 |
| | -no | |
| 存储分配说明 | -r | 设定如何分配程序到存储器。 |
| | -nr | |
| | -rd | 设定自动分配外部变量 / 内部静态变量到 <code>saddr</code> 区。 |
| | -nr | |
| | -rs | 设定自动分配静态自动变量到 <code>saddr</code> 区。 |
| | -nr | |
| 优化说明 | -q | 设定优化类型。 |
| | -nq | |
| 调试信息输出说明 | -g | 设定输出 C 源代码级别的调试信息。 |
| | -ng | |
| 预处理列表文件创建说明 | -p | 设定预处理列表文件的输出。 |
| | -k | 设定预处理列表的处理。 |
| 预处理说明 | -d | 执行宏定义。 |
| | -u | 使宏定义作废。 |
| | -i | 从指定文件夹中读取 <code>include</code> 文件。 |
| 汇编源文件创建说明 | -a | 设定汇编源文件的输出。 |
| | -sa | |
| 错误列表文件创建说明 | -e | 设定错误列表文件的输出。 |
| | -se | |
| 交叉引用列表文件创建说明 | -x | 设定交叉参考列表文件的输出。 |
| 列表格式说明 | -lw | 设定每个列表文件每行的字符数。 |
| | -ll | 设定每个列表文件每页的行数。 |
| | -lt | 设定每个列表文件中标签的扩展字符数。 |
| | -lf | 在每个列表文件的末尾插入换页码。 |
| | -li | 添加 <code>include</code> 文件中的 C 源代码到带 C 源代码注释的汇编源文件中。 |
| 警告输出说明 | -w | 设定是否输出警告消息到控制台。 |
| 执行状态显示设定 | -v | 设定是否输出编译的执行状态到控制台。 |
| | -nv | |
| 参数文件说明 | -f | 通过指定文件输入选项和输入文件名。 |
| 临时文件创建文件夹说明 | -t | 在指定驱动器和文件夹中创建临时文件。 |

| 分类 | 选项 | 说明 |
|---------------|---------|------------------|
| 函数扩展说明 | -z | 允许对扩展函数处理。 |
| | -nz | |
| 设备文件查找路径说明 | -y | 设定查找设备文件的路径。 |
| 存储类型说明 | -m | 为编译指定使用的存储模式。 |
| 镜像空间说明 | -mi | 在镜像区域设定段的可分配目标。 |
| 公用目标设定 | -common | 设定 78K0 公共对象的输出。 |
| 变量 / 函数信息文件说明 | -ma | 设定变量 / 函数信息文件 |
| 帮助说明 | -- | 在屏幕上输出帮助信息。 |
| | -? | |
| | -h | |

(2) 优先级

在下列表格中显示有关编译选项、优先级的说明，同时设定两个或以上垂直安放和水平安放的选项。

表 B-4. 编译选项的优先级

| | -no | -p | -np | -d | -u | -a | -e | -x | -sa |
|-----|-----|----|-----|----|----|----|----|----|-----|
| -r | NG | | | | | | | | |
| -q | NG | | | | | | | | |
| -g | NG | | | | | | | | |
| -k | | Δ | NG | | | | | | |
| -d | | | | | OK | | | | |
| -u | | | | OK | | | | | |
| -sa | | | | | | NG | | | |
| -lw | | Δ | | | | Δ | Δ | Δ | |
| -ll | | Δ | | | | Δ | Δ | Δ | |
| -lt | | Δ | | | | Δ | Δ | Δ | |
| -lf | | Δ | | | | Δ | Δ | Δ | |
| -li | | | | | | | | | Δ |

- 标有 NG 的位置

如果设定选项在横坐标上，则此选项在纵坐标上无效。

示例 -rd 和 -g 选项无效。

```
C>cc78k0r -cf1166a0 -e sample.c -no -rd -g
```

- 标有 Δ 的位置

如果没有设定水平轴上的选项，则此选项在垂直轴上无效。

示例 -p 选项被设定，因此 -k 选项有效。

```
C>cc78k0r -cf1166a0 -e sample.c -p -k
```

- 标识 OK 的位置

设定在水平或垂直轴上最后的选项具有优先级。

示例 最后设定的是 `-d` 选项，因此 `-u` 选项无效并且 `-d` 选项具有优先级。

```
C>cc78k0r -cf1166a0 -e sample.c -utest -dtest=1
```

- 空白区域

如果设定水平轴上的选项，则此选项在垂直轴上无效。

指定 `-o/-no` 选项时，如果同时设定了两个可在选项前加“n”的选项，则最近指定的选项是有效的。

示例 在 `-o` 选项后设定 `-no` 选项时，`-o` 选项是无效的，而 `-no` 选项是有效的。

```
C>cc78k0r -cf1166a0 -e sample.c -o -no
```

没有在“表 B-4. 编译选项的优先级”描述的选项不受其它选项的影响。然而，如果设定了帮助说明选项（`--/?-h`），则所有其他选项说明变成无效。

设备类型说明

设备类型的选项说明如下。

- **-c**

-c

[Description format]

```
-cdevice-type
```

- 省略时解释
不能省略。

[功能]

- **-c** 选项为执行编译设定目标设备。

[Application]

- 确保设定了 **-c** 选项。CA78K0 为目标设备执行编译并为该设备生成目标代码。

[说明]

- 参阅 CubeSuite 关于可以通过 **-c** 选项（该选项相当于设备类型）设定目标设备的操作注意事项。
- 当使用 CA78K0 时，需要设备文件。

[Cautions]

- 不能忽略 **-c** 选项。

[Example of use]

- 在命令行中设定 uPD78F1166_A0 为目标设备，说明如下：

```
C>cc78k0r -cf1166a0 prime.c
```

目标模块文件创建设定

目标模块文件创建设定选项如下。

- `-o/-no`

`-o/-no`

[Description format]

```
-o[output-file-name]  
-no
```

- 省略时解释
-o.rel

[功能]

- `-o` 选项指定输出目标模块文件。该选项同时设定输出位置和文件名。
- `-no` 选项设定不输出目标模块文件。

[Application]

- 使用 `-o` 选项设定输出目标模块文件的路径或改变文件名的路径。
- 当执行编译时设定 `-no` 选项仅输出汇编源文件。该操作将缩短编译时间。

[说明]

- 如果设定 `-o` 选项时省略了文件名，则输出文件名为 "*input-file-name.rel*"。
- 如果设定 `-o` 选项时省略了输出文件名的扩展名，则输出文件名为 "*output-file-name.rel*"。
- 当编译错误产生时，即使设定了 `-o` 选项，目标模块文件也不能输出。
- 如果设定 `-o` 选项时省略了驱动器名，则会输出目标模块文件到当前驱动器。
- 如果同时设定 `-o` 选项和 `-no` 选项，最后设定的选项具有优先级。

[Cautions]

- 在使用 CubeSuite 时改变输出目标，只要在属性面板上，通过 [\[Link Options\]](#) 标签，在 [\[Output File\]](#) 分项中设定输出目标即可。
- 当设定独立编译选项时，改变输出文件的文件名也是可能的，可以通过 [\[Individual Compile Options\]](#) 标签，在 [\[Output File\]](#) 分项中设定文件名即可。

[Example of use]

- 忽略先设定的 `-no` 选项，最后设定的 `-o` 选项有效，因此目标模块文件 (`prime.rel`) 将输出。

```
C>cc78k0r -cf1166a0 prime.c -no -o
```

存储分配说明

存储分配选项说明如下。

- r/-nr
- rd/-nr
- rs/-nr

-r/-nr

[Description format]

```
-rprocess-type (two or more types can be specified)
-nr
```

- 省略时解释
- nr

[功能]

- r 选项设定如何分配程序到存储器。
- nr 选项使 -r 选项失效。

[Application]

- 使用 -r 选项设定如何分配程序到存储器。

[说明]

- 通过 -r 选项设定处理的类型，如下所示：
不能省略处理类型设定。如果省略设定，会出现致命错误（F0012）。

| 处理类型 | 功能 |
|--------------------------|--|
| a | 在 1 字节单元里进行间接引用。 |
| b | 从最高有效位（MSB）分配位字段。 |
| d[n][m] (n = 1, 2, 4) | 自动分配外部变量 / 外部静态变量（除了常变量）到 <code>saddr</code> 区，无论是否有 <code>sreg</code> 声明。 参阅“-rd/-nr”获取详细信息。 |
| s[n][m] (n = 1, 2, 4) | 自动分配静态自动变量到 <code>saddr</code> 区，无论是否有 <code>sreg</code> 声明。 参阅“-rs/-nr”获取详细信息。 |
| c | 在 1 字节单元里进行间接引用。 打包结构体并安放结构体成员在 1 个字节中。 |
| f | 在 <code>far</code> 区域分配 ROM 数据。 |
| n | 在 <code>near</code> 区域分配 ROM 数据。 |

备注 可以设定两个或多个处理类型。

- 如果 -nr 选项设定，处理类型说明如下。

| 处理类型 | 功能 |
|------|---|
| a | 在 1 字节单元里不进行间接引用。 |
| b | 从最低有效位 (LSB) 分配位字段。 |
| d | 不自动分配任何变量到 <code>saddr</code> 区。 |
| s | 不自动分配任何变量到 <code>saddr</code> 区。 |
| c | 在 1 字节单元里不进行间接引用。 不打包结构体，不安放结构体成员在 1 个字节中。 |

[Example of use]

- 自动分配外部变量或外部静态变量和静态自动变量到 `saddr` 区，无论是否有 `sreg` 声明，描述如下：

```
C>cc78k0r -cf1166a0 -rds
```

-rd/-nr**[Description format]**

```
-rd[n][m] (n = 1, 2, 4)
-nr
```

- 省略时解释
- nr

[功能]

- rd 选项设定自动分配外部变量 / 内部静态变量自动到 **saddr** 区。
- nr 选项使 -rd 选项失效。

[Application]

- 使用 -rd 选项自动分配外部变量 / 外部静态变量（除了常变量）到 **saddr** 区，无论是否有 **sreg** 声明。

[说明]

- 分配变量，并依据 *n* 的值和 “*m*” 的设定来做改变。

| <i>n</i> 的设定, “ <i>m</i> ” | 分配不同类型到 saddr 区 |
|----------------------------|--|
| <i>n</i> | <ul style="list-style-type: none"> - 当 <i>n</i> = 1: 字符, unsigned 字符 - 当 <i>n</i> = 2: 字符、unsigned 字符、短型、unsigned 短型、整型、unsigned 整型、enum 型、near 指针时 - 当 <i>n</i> = 4: 字符、unsigned 字符、短型、unsigned 短型、整型、unsigned 整型、enum 型、长型、unsigned 长型, 指针时 |
| <i>m</i> | 结构, 联合体, 数组 |
| Omitted | 所有变量 |

- **sreg** 声明变量分配到 **saddr** 区，无论 -rd 选项是否设定。
- 处理通过外部声明引用的变量并分配到 **saddr** 区。
- 通过设定该选项分配到 **saddr** 区的变量执行与 **sreg** 变量相同的操作。

[Example of use]

- 自动分配 **char** 或 **unsigned char** 类型外部变量或外部静态变量和静态自动变量到 **saddr** 区，无论是否有 **sreg** 声明，描述如下：

```
C>cc78k0r -cf1166a0 -rd1
```

-rs/-nr**[Description format]**

```
-rs[n][m] (n = 1, 2, 4)
-nr
```

- 省略时解释
- nr

[功能]

- rs 选项设定分配静态自动变量自动到 `saddr` 区。
- nr 选项使 -rs 选项失效。

[Application]

- 使用 -rs 选项自动分配静态自动变量到 `saddr` 区，无论是否有 `sreg` 声明。

[说明]

- 分配变量，并依据 `n` 的值和 “`m`” 的设定来做改变。

| <code>n</code> 的设定, “ <code>m</code> ” | 分配不同类型到 <code>saddr</code> 区 |
|--|--|
| <code>n</code> | <ul style="list-style-type: none"> - 当 <code>n = 1</code>: 字符, unsigned 字符 - 当 <code>n = 2</code>: 字符、unsigned 字符、短型、unsigned 短型、整型、unsigned 整型、enum 型、near 指针时 - 当 <code>n = 4</code>: 字符、unsigned 字符、短型、unsigned 短型、整型、unsigned 整型、enum 型、长型、unsigned 长型, 指针时 |
| <code>m</code> | 结构, 联合体, 数组 |
| Omitted | 所有变量 |

- `sreg` 声明变量分配到 `saddr` 区，无论 -rs 选项是否设定。
- 通过设定该选项分配到 `saddr` 区的变量执行了与 `sreg` 声明自动变量相同的操作。

[Example of use]

- 自动分配 `char` 或无符号的 `char` 类型静态自动变量到 `saddr` 区，无论是否有 `sreg` 声明，描述如下：

```
C>cc78k0r -cf1166a0 -rs1
```

优化说明

优化设定选项如下。

-q/-nq

-q/-nq

[Description format]

```
-q[optimization-type] (two or more types can be specified)
```

```
-nq
```

- 省略时解释
- qx2 (-qcjlvw)

[功能]

- q 选项设定通过优化程序生成高效的目标文件。
- nq 选项使 -q 选项失效。

[Application]

- 使用 -q 选项提高目标文件的执行速率以及减小代码大小。
- 当 -q 选项设定时，如果想同时执行多重优化，需要连续地设定优化类型。参阅 [Description] 获取详细信息。

[说明]

- 通过 -q 选项设定优化类型显示如下：

| 优化类型 | 过程说明 | | | | | | | | | | | | | | |
|-----------------------------------|--|------|------|----------------------------------|---------------|--------------------------------|---------------|---------------------|---------|-----------------------------------|---------------|----------------------------|---------|--------------------------------|---------------|
| 无说明 | 假定 -qx2 (-qcjlvw) 选项已经设定。 | | | | | | | | | | | | | | |
| u | 将无限定的字符当成 unsigned 字符处理可以提高代码效率。 | | | | | | | | | | | | | | |
| c | 对无符号扩展的字符进行计算。 <table border="1" data-bbox="545 1503 1442 1839"> <thead> <tr> <th>计算目标</th> <th>计算结果</th> </tr> </thead> <tbody> <tr> <td>unsigned 字符类型变量和 unsigned 字符类型变量</td> <td>unsigned 字符类型</td> </tr> <tr> <td>unsigned 字符类型变量和 signed 字符类型变量</td> <td>unsigned 字符类型</td> </tr> <tr> <td>有符号字符类型变量和有符号字符类型变量</td> <td>有符号字符类型</td> </tr> <tr> <td>从 -128 到 255 的常量和 unsigned 字符类型变量</td> <td>unsigned 字符类型</td> </tr> <tr> <td>从 -128 到 127 的常量和有符号字符类型变量</td> <td>有符号字符类型</td> </tr> <tr> <td>带前缀 U 的从 0 到 255 的常量和有符号字符类型变量</td> <td>unsigned 字符类型</td> </tr> </tbody> </table> | 计算目标 | 计算结果 | unsigned 字符类型变量和 unsigned 字符类型变量 | unsigned 字符类型 | unsigned 字符类型变量和 signed 字符类型变量 | unsigned 字符类型 | 有符号字符类型变量和有符号字符类型变量 | 有符号字符类型 | 从 -128 到 255 的常量和 unsigned 字符类型变量 | unsigned 字符类型 | 从 -128 到 127 的常量和有符号字符类型变量 | 有符号字符类型 | 带前缀 U 的从 0 到 255 的常量和有符号字符类型变量 | unsigned 字符类型 |
| 计算目标 | 计算结果 | | | | | | | | | | | | | | |
| unsigned 字符类型变量和 unsigned 字符类型变量 | unsigned 字符类型 | | | | | | | | | | | | | | |
| unsigned 字符类型变量和 signed 字符类型变量 | unsigned 字符类型 | | | | | | | | | | | | | | |
| 有符号字符类型变量和有符号字符类型变量 | 有符号字符类型 | | | | | | | | | | | | | | |
| 从 -128 到 255 的常量和 unsigned 字符类型变量 | unsigned 字符类型 | | | | | | | | | | | | | | |
| 从 -128 到 127 的常量和有符号字符类型变量 | 有符号字符类型 | | | | | | | | | | | | | | |
| 带前缀 U 的从 0 到 255 的常量和有符号字符类型变量 | unsigned 字符类型 | | | | | | | | | | | | | | |
| r | 添加寄存器变量到寄存器并且分配该变量到 saddr 区。 | | | | | | | | | | | | | | |
| j | 优化分支指令。 | | | | | | | | | | | | | | |

| 优化类型 | 过程说明 |
|---------------------|--|
| x[n] (n = 1 - 3) | 根据速度 / 代码大小的优先级自动分配优化选项。 分配选项根据 n 的值改变, 如下。当省略 n 时, 则认为 n = 2。 - 当 n = 1: 速度优先。假定 -qjvw 选项已经设定。 - 当 n = 2: 默认。假定 -qjlvw 选项已经设定。 - 当 n = 3: 代码大小优先。假定 -qjlvw 选项已经设定。 |
| w | 进行进一步优化。 调整表达式内部的执行次序。 |
| v | 将参数和自动变量自动分配到寄存器或 saddr 区域。 |
| [n] (n = 1 - 3) | 进行基于代码大小优先的优化, 并替换库中的标准代码格式。如果没有指定这个类型, 则以基于速度优先的方式来优化代码。 使用库替代作用域, 依据 n 的不同的值, 如下。如果省略 n, 则认为 n = 1。 - 当 n = 1: 库中无内容替换。 - 当 n = 2: 只替换库中前处理和后处理函数。 - 当 n = 3: 除 2 以外, 使用低级库, 且在子程序中替换公共代码。 |
| -t | 用 switch 语句产生分支表, 且相关分支的功能按 far 区域分配。 如果分支间距超过 64KB, 则输出错误信息 (F0924)。 |
| g | 指定调试的优先级。 |

- 可以设定多重优化类型。
- 如果省略 -q 选项或优化类型, 当设定 -qjlvw 选项时, 优化为统一类型。
- 删除默认选项的一部分, 设定除了想删除选项以外的选项 (例如: 设定 -qr 为 -> 删除 -qjlvw)。
- 如果目标模块文件和汇编源模块文件均没有输出, 除 -qu 以外的 -q 选项均为无效。
- 如果同时设定 -q 选项和 -nq 选项, 最后设定的选项有效。
- 如果同时设定两个或两个以上的 -q 选项, 最后设定的选项有效。
- 实时 OS 不支持 -qr 选项。

[Example of use]

- 将无限定的 char 当成无符号的 char 处理用以提高代码效率, 描述如下:

```
C>cc78k0r -cf1166a0 prime.c -qu
```

- 忽略首先设定的 -qc 选项, 最后设定的 -qr 选项有效, norec 的自变量, 自动变量和寄存器变量被分配到 saddr 区。

```
C>cc78k0r -cf1166a0 prime.c -qc -qr
```

- 使 -qc 和 -qr 选项生效, 描述如下:

```
C>cc78k0r -cf1166a0 prime.c -qcr
```

调试信息输出说明

调试信息输出选项说明如下。

-g/-ng

-g/-ng**[Description format]**

```
-g[n] (n = 1, 2)
-ng
```

- 省略时解释
-g2

[功能]

- g 选项设定调试信息添加到目标模块文件。
- ng 选项禁止 -g 选项。

[Application]

- 如果没有设定 -g 选项，输入到调试器中的目标模块文件所需要的行号和符号信息将不输出。因此，在源代码级别的调试，所有的模块通过设定 -g 选项进行链接编译。

[说明]

- 不同的操作依据 *n* 的值如下。

| <i>n</i> 的值 | 功能 |
|-------------|--|
| 无说明 | 假定 <i>n</i> 选项已经设定。 |
| 1 | 仅将调试信息（从 \$DGS 或 \$DGL 开始的信息）添加到目标模块文件。没有调试信息添加到汇编源文件。 该选项使引用汇编文件更为简单。 因为调试信息添加到可用的目标文件的源代码调试。 |
| 2 | 添加调试信息到目标模块文件和汇编源模块文件。 |

- 如果同时设定了 -g 和 -ng 选项，则最后设定的选项无效。
- 如果目标模块文件和汇编源模块文件均没有输出，-g 选项无效。

[Example of use]

- 添加汇编源代码调试信息到目标模块文件（prime.rel），描述如下：

```
C>cc78k0r -cf1166a0 prime.c -g
```

预处理列表文件创建说明

预处理列表文件创建选项的说明如下。

- p
- k

-p

[Description format]

```
-p[output-file-name]
```

- 省略时解释
无（无文件输出）

[功能]

- p 选项设定预处理列表文件的输出。该选项同时设定输出位置和文件名。如果省略 -p 选项，没有预处理列表文件输出。

[Application]

- 使用 -p 选项在根据 -k 操作类型选项执行的预处理操作后输出源文件，或者改变输出目的地或预处理列表文件的输出文件名。

[说明]

- 当设定 -p 选项时省略输出文件名，输出文件名将为 “*input-file-name.ppl*”。
- 当设定 -p 选项时省略输出文件名的扩展名，输出文件名将为 “*output-file-name.ppl*”。
- 如果当设定 -p 选项时，省略驱动器名，则预处理列表文件将输出到当前驱动器。

[Cautions]

- 当使用 CubeSuite 时，可以改变输出文件的文件名。

[Example of use]

- 输出预处理列表文件（*sample.ppl*），描述如下：

```
C>cc78k0r -cf1166a0 prime.c -psample.ppl
```

-k**[Description format]**

```
-k[process-type] ( 可以设定两个或以上的类型 )
```

- 省略时解释
- kfln

[功能]

- -k 选项设定预处理列表的处理。

[Application]

- 当输出预处理列表文件时，使用 -k 选项删除注释和引用定义扩展。

[说明]

- 通过 -k 选项设定处理的类型如下所示：

| 处理类型 | 功能 |
|------|---------------------------------|
| 无说明 | 假定 -kfln 选项已经设定。 |
| c | 删除注释。 |
| d | 扩展 #define。 |
| f | 执行 #if, #ifdef 和 #ifndef 的条件编译。 |
| i | 扩展 #include。 |
| l | 执行 #line 操作。 |
| n | 执行行号和分页处理。 |

备注 可以设定两个或多个处理类型。

- 如果没有设定 -p 选项， -k 选项无效。
- 如果同时设定两个或两个以上的 -k 选项，最后设定的选项有效。

[Example of use]

- 当预处理列表文件（prime.ppl）输出时，删除注释并执行行号和分页处理。

```
C>cc78k0r -cf1166a0 prime.c -p -kcn
```

输出显示如下。

```
/*
78K0R C Compiler Vx.xx Preprocess List          Date:xx xxx xxxx   Page:  1

Command   : -cf1166a0 prime.c -p -kcn
In-file   : prime.c
PPL-file  : prime.ppl
Para-file:
*/

    1 : #define TRUE      1
    2 : #define FALSE    0
    3 : #define SIZE     200
    4 :
    5 : char    mark [ SIZE + 1 ] ;
    6 :
    7 : main ( )
    8 : {
        :
/*
Target chip : uPD78F1166_A0
Device file : Vx.xx
*/
```

预处理说明

预处理选项说明如下。

- -d
- -u
- -i

-d

[Description format]

```
-dmacro-name[=definition-name][,macro-name[=definition-name]] ... ( 可以设定两个或两个以上的类型 )
```

- 省略时解释
仅 C 源文件内的宏定义有效。

[功能]

- -d 选项设定了与 C 源文件内 `#define` 语句相同的宏定义。

[Application]

- 使用 -d 选项，使特殊宏定义有效。

[说明]

- 在宏定义之间插入 “,” 可以同时实现多达 30 个宏定义。
- 在 “=” 和 “,” 的前后不能加入空格。
- 如果省略定义名，编译器会假设：“*macro-name=1*” 定义。
- 如果在 -d 和 -u 选项设定了相同的宏名，则最后的设定有效。

[Example of use]

- 下述代码在 C 源文件（`prime.c`）中定义。

```
#define TEST 1  
#define TIME 10
```

```
C>cc78k0r -cf1166a0 prime.c -dTEST,TIME=10
```

-u

[Description format]

```
-umacro-name[,macro-name] ... (可以设定两个或两个以上的宏名)
```

- 省略时解释
使用 **-d** 设定的宏定义有效。

[功能]

- **-u** 选项使与 C 源文件中 **#undef** 语句相同的宏定义无效。

[Application]

- 使用 **-u** 选项使由 **-d** 选项定义的宏名无效。

[说明]

- 在宏定义之间插入 “,” 可以同时使多达 30 个宏定义失效。
在 “,” 的前后不能加入空格。
- 可以通过 **-u** 选项设定失效的宏定义是由 **-d** 选项定义过的。
通过 C 源文件中的 **#define** 定义的宏名或 CA78K0 中的系统宏名不能通过 **-u** 选项设定失效。
- 如果在 **-d** 和 **-u** 选项设定了相同的宏名, 则最后的设定有效。

[Example of use]

- 忽略首先设定的 **-d** 选项以及最后设定的 **-u** 选项有效, 则 **TEST** 的宏定义失效。

```
C>cc78k0r -cf1166a0 prime.c -dTEST,TIME=10 -uTEST
```

-i**[Description format]**

```
-ifolder[,folder] ... (可以设定两个或两个以上的文件夹)
```

- 省略时解释
假定以下的文件夹已经设定。

(1) 包含源文件的文件夹^{注意 1}(2) 通过环境变量设定的文件夹 **INC78K0R**(3) **C:\Program Files\NEC Electronics CubeSuite\CubeSuite\CA78K0\Vx.xx\inc78k0**^{注意 2}**[功能]**

- 为了从指定的文件夹输入，将 **-i** 选项设定由 C 源文件中的 **#include** 语句设定的 **include** 文件。

[Application]

- 使用 **-i** 选项从特定的文件夹中查找 **include** 文件。

[说明]

- 在文件夹名之间插入 “,” 可以同时实现多达 64 个文件夹设定。
在 “,” 的前后不能加入空格。
- 如果设定两个或两个以上的文件夹通过下述的 **-i** 选项，或者如果设定两个或两个以上的 **-i** 选项，通过 **#include** 设定的文件以指定的顺序查找。
- 查找顺序如下。

(1) 包含源文件的文件夹^{注意 1}(2) 通过 **-i** 选项设定的文件夹(3) 通过环境变量 **INC78K0** 设定的文件夹(4) **C:\Program Files\NEC Electronics CubeSuite\CubeSuite\CA78K0\Vx.xx\inc78k0**^{注意 2}

注 1. 如果在 **#include** 语句中设定 **include** 文件名带 “ ” (双引号)，首先查找包含源文件的文件夹。如果 **include** 文件名 <> 设定，则不执行查找。

2. 这是 C 编译器安装到 **C:\Program Files\NEC Electronics CubeSuite\CubeSuite\CA78K0\Vx.xx** 时的例子。

[Example of use]

- 从文件夹 **D:** 和 **D:\sample** 将由在 C 源文件 (**prime.c**) 中的 **# include** 语句所设定的 **include** 文件输入，描述如下：

```
C>cc78k0r -cf1166a0 prime.c -iD:.,D:\sample
```

汇编源文件创建说明

汇编源文件创建选项说明如下。

- **-a**
- **-sa**

-a

[Description format]

```
-a[output-file-name]
```

- 省略时解释
没有汇编源文件输出。

[功能]

- **-a** 选项设定汇编源文件的输出。该选项同时设定输出位置和文件名。

[Application]

- 使用 **-a** 选项设定汇编源文件的输出位置或改变其文件名。

[说明]

- 当设定 **-a** 选项时省略输出文件名，输出文件名将为 “*input-file-name.asm*”。
- 当设定 **-a** 选项时省略输出文件名的扩展名，输出文件名将为 “*output-file-name.asm*”。
- 如果当设定 **-a** 选项时省略驱动器名，汇编源文件将输出到当前驱动器。
- 如果同时设定 **-a** 选项和 **-sa** 选项，则忽略 **-sa** 选项。

[Cautions]

- 当使用 CubeSuite 时，可以改变输出文件的文件名。

[Example of use]

- 输出汇编源文件（*sample.asm*），描述如下：

```
C>cc78k0r -cf1166a0 prime.c -asample.asm
```

-sa

[Description format]

```
-sa[output-file-name]
```

- 省略时解释
没有汇编源文件输出。

[功能]

- **-sa** 选项添加 C 源作为汇编源文件的注释。
这也同时指定输出文件名和路径。

[Application]

- 使用 **-sa** 选项同时输出汇编源文件和 C 源文件。

[说明]

- 当设定 **-sa** 选项时省略输出文件名，输出文件名将为 “*input-file-name.asm*”。
- 当设定 **-sa** 选项时省略输出文件名的扩展名，输出文件名将为 “*output-file-name.asm*”。
- 如果当设定 **-sa** 选项时省略驱动器名，汇编源文件将输出到当前驱动器。
- 如果同时设定 **-a** 选项和 **-sa** 选项，则忽略 **-sa** 选项。
- 包含文件中的 C 源没有添加到输出汇编源文件的注释。然而，如果设定 **-li** 选项，**include** 文件中的 C 源代码添加到注释。

[Cautions]

- 当使用 **CubeSuite** 时，可以改变输出文件的文件名。

[Example of use]

- 添加 C 源文件（**prime.c**）作为汇编源文件（**prime.asm**）的注释，描述如下：

```
C>cc78k0r -cf1166a0 prime.c -sa
```

输出显示如下。

```

; 78K0 C Compiler Vx.xx Assembler Source           Date:xx xxx xxxx  Time:xx:xx:xx

; Command   : -cf1166a0 prime.c -sa
; In-file   : prime.c
; Asm-file  : prime.asm
; Para-file :

$PROCESSOR ( f1166a0 )
$DEBUG
$NODEBUGA
$KANJI CODE SJIS
$TOL_INF    03FH , 100H , 00H , 00H , 00H

$DGS  FIL_NAM , .file ,      037H , 0FFFEH , 03FH , 067H , 01H , 00H
$DGS  AUX_FIL , prime.c
$DGS  MOD_NAM , prime ,      00H , 0FFFEH , 00H , 077H , 00H , 00H
      :
      EXTRN  _@RTARG0
      EXTRN  @@isrem
      PUBLIC _printf
      PUBLIC _putchar
      PUBLIC _mark
      PUBLIC _main
      :
@@CODEL CSEG
_main :
$DGL  1 , 19
      push  hl                      ; [ INF ] 1 , 1
      subw  sp , #08H                ; [ INF ] 2 , 1
      movw  hl , sp                  ; [ INF ] 3 , 1
??bf_main :
; line  9 :   int i , prime , k , count ;
; line 10 :
; line 11 :   count = 0 ;
$DGL  0 , 4
      clrw  ax                      ; [ INF ] 1 , 1
      movw  [ hl ] , ax              ; count ; [ INF ] 1 , 1
; line 12 :
; line 13 :   for ( i = 0 ; i <= SIZE ; i++ )
$DGL  0 , 6
      movw  [ hl + 6 ] , ax ; i      ; [ INF ] 2 , 1
?L0003 :
      movw  ax , [ hl + 6 ] ; i      ; [ INF ] 2 , 1
      cmpw  ax , #0C8H              ; 200 ; [ INF ] 3 , 1
      orl   CY , a.7                ; [ INF ] 2 , 1
      skc                                     ; [ INF ] 2 , 1
      bnz  $?L0004                  ; [ INF ] 2 , 4

```

```

:

; *** Code Information ***
;
; $FILE C:\Program Files\NEC Electronics CubeSuite\CubeSuite\CA78K0R\Vx.xx\smp78k0r\cc78k0r\
prime.c
;
; $FUNC main ( 8 )
;     bc = ( void )
;     CODE SIZE = 117 bytes , CLOCK_SIZE = 86 clocks , STACK_SIZE = 16 bytes
;
; $CALL printf ( 18 )
;     bc = ( pointer : ax , int : [ sp + 2 ] )
;
; $CALL putchar ( 18 )
;     bc = ( pointer : ax , int : [ sp + 2 ] )
;
; $CALL putchar ( 20 )
;     bc = ( int : ax )
;
; $CALL printf ( 25 )
;     bc = ( pointer : ax , int : [ sp + 2 ] )
;
; $FUNC printf ( 31 )
;     bc = ( pointer s : ax , int i : [ sp + 4 ] )
;     CODE SIZE = 22 bytes , CLOCK_SIZE = 20 clocks , STACK_SIZE = 14 bytes
;
; $FUNC putchar ( 41 )
;     bc = ( char c : x )
;     CODE SIZE = 16 bytes , CLOCK_SIZE = 16 clocks , STACK_SIZE = 6 bytes

; Target chip : uPD78F1166_A0
; Device file : Vx.xx
```

错误列表文件创建说明

错误列表文件创建选项说明如下。

- **-e**
- **-se**

-e

[Description format]

```
-e[output-file-name]
```

- 省略时解释
没有错误列表文件输出。

[功能]

- **-e** 选项指定错误列表文件的输出。该选项同时设定输出位置和文件名。

[Application]

- 使用 **-e** 选项设定错误列表文件的输出位置或改变其文件名。

[说明]

- 当设定 **-e** 选项时省略输出文件名，输出文件名将为 “*input-file-name.ecc*”。
- 当设定 **-e** 选项时省略输出文件名的扩展名，输出文件名将为 “*output-file-name.ecc*”。
- 如果当 **-e** 选项设定时驱动器名省略，错误列表文件将输出到当前驱动器。
- 如果设定 **-w0** 选项，不能输出警告消息。

[Cautions]

- 当使用 CubeSuite 时，可以改变输出文件的文件名。

[Example of use]

- 输出错误列表文件（*prime.ecc*），描述如下：

```
C>cc78k0r -cf1166a0 prime.c -e
```

输出显示如下。

```
prime.c(18) : CC78K0R warning W0745 : Expected function prototype
prime.c(20) : CC78K0R warning W0745 : Expected function prototype
prime.c(26) : CC78K0R warning W0622 : No return value
prime.c(37) : CC78K0R warning W0622 : No return value
prime.c(44) : CC78K0R warning W0622 : No return value
```

```
Target chip : uPD78F1166_A0
```

```
Device file : Vx.xx
```

```
Compilation complete, 0 error(s) and 5 warning(s) found.
```

-se

[Description format]

```
-se[output-file-name]
```

- 省略时解释
没有错误列表文件输出。

[功能]

- **-se** 选项添加 C 源文件到错误列表文件。该选项同时设定输出位置和文件名。

[Application]

- 使用 **-se** 选项同时输出错误列表文件和 C 源文件。

[说明]

- 当设定 **-se** 选项时省略输出文件名，输出文件名将为 “*input-file-name.cer*”。
- 当设定 **-se** 选项时省略输出文件名的扩展名，输出文件名将为 “*output-file-name.cer*”。
- 如果当设定 **-se** 选项时，省略驱动器名，则错误列表文件将输出到当前驱动器。
- 不能给 **include** 文件设定文件夹和文件名。
如果 **include** 文件的文件类型是 “H”，类型为 “her” 的错误列表文件被输出到当前驱动器。如果 **include** 文件的文件类型是 “C”，类型为 “cer” 的错误列表文件输出。在其他情况下，输出类型为 “er” 的错误列表文件。
- 如果没有错误出现，C 源代码不添加。在这种情况下，不为 **include** 文件创建错误列表文件。
- 如果设定 **-w0** 选项，不能输出警告消息。

[Cautions]

- 当使用 CubeSuite 时，可以改变输出文件的文件名。

[Example of use]

- 添加 C 源文件（*prime.c*）到错误列表文件（*prime.cer*），描述如下：

```
C>cc78k0r -cf1166a0 prime.c -se
```

输出显示如下。

```
/*
78K0R C Compiler Vx.xx Error List                               Date:xx xxx xxxx  Time:xx:xx:xx

Command   : -cf1166a0 prime.c -se
In-file   : prime.c
Err-file  : prime.cer
Para-file:
*/

#define TRUE  1
#define FALSE 0
#define SIZE  200

char  mark [ SIZE + 1 ] ;

void main ( void ) {
    :
    prime = i + i + 3 ;
    printf ( "%6d" , prime ) ;
*** CC78K0R warning W0745: Expected function prototype
    count++ ;
    if ( ( count%8 ) == 0 ) putchar ( '\n' ) ;
*** CC78K0R warning W0745: Expected function prototype
    for ( k = i + prime ; k <= SIZE ; k += prime )
        :
}
}
```

交叉引用列表文件创建说明

交叉引用列表文件创建选项说明如下。

-x

-x

[Description format]

```
-x[output-file-name]
```

- 省略时解释
没有交叉引用列表文件输出。

[功能]

- x 选项设定交叉引用列表文件的输出。该选项同时设定输出位置和文件名。交叉引用列表文件具有检查引用频率、定义和符号引用指针的作用。

[Application]

- 使用 -x 选项输出交叉引用列表文件并设定交叉引用列表文件的输出位置或改变其文件名。

[说明]

- 当设定 -x 选项时省略输出文件名，输出文件名将为 “*input-file-name.xrf*”。
- 当设定 -x 选项时省略输出文件名的扩展名，输出文件名将为 “*output-file-name.xrf*”。
- 即使除 CO101 以外的内部错误或错误编号 F0024（或从 E 开始编号）的编译错误发生，交叉引用列表文件仍然创建。然而，文件的内容不能保证。

[Cautions]

- 当使用 CubeSuite 时，可以改变输出文件的文件名。

[Example of use]

- 输出交叉引用列表文件（*prime.xrf*），描述如下：

```
C>cc78k0r -cf1166a0 prime.c -x
```

输出显示如下。

| 78K0R C Compiler Vx.xx Cross reference List | | | | Date:xx xxx xxxx | | | | Page: 1 | |
|---|--------|---------|---------------|------------------|-----------|----|----|---------|----|
| Command : -cf1166a0 prime.c -x | | | | | | | | | |
| In-file : prime.c | | | | | | | | | |
| Xref-file : prime.xrf | | | | | | | | | |
| Para-file: | | | | | | | | | |
| ATTRIB | MODIFY | TYPE | SYMBOL | DEFINE | REFERENCE | | | | |
| EXTERN | NEAR | | array mark | 5 | 29 | 31 | 37 | | |
| EXTERN | FAR | func | printf | 7 | 33 | 40 | | | |
| REG1 | | pointer | s | 7 | 13 | | | | |
| PARAM | | | | | | | | | |
| REG1 | | int | i | 7 | 12 | | | | |
| PARAM | | | | | | | | | |
| REG1 | | int | j | 9 | 12 | | | | |
| REG1 | | pointer | ss | 10 | 13 | | | | |
| EXTERN | FAR | func | putchar | 16 | 35 | | | | |
| REG1 | | char | c | 16 | 19 | | | | |
| PARAM | | | | | | | | | |
| REG1 | | char | d | 18 | 19 | | | | |
| EXTERN | FAR | func | main | 22 | | | | | |
| REG1 | | int | i | 24 | 28 | 28 | 28 | 29 | 30 |
| | | | | | | | 30 | 31 | 32 |
| | | | | | | | 36 | | |
| REG1 | | int | prime | 24 | 32 | 33 | 36 | 36 | |
| REG1 | | int | k | 24 | 36 | 36 | 36 | 37 | |
| REG1 | | int | count | 24 | 26 | 34 | 35 | 40 | |
| | | | #define TRUE | 1 | 29 | | | | |
| | | | #define FALSE | 2 | 37 | | | | |
| | | | #define SIZE | 3 | 5 | 28 | 30 | 36 | |
| Target chip : uPD78F1166_A0 | | | | | | | | | |
| Device file : Vx.xx | | | | | | | | | |

列表格式说明

列表格式选项说明如下。

- `-lw`
- `-ll`
- `-lt`
- `-lf`
- `-li`

-lw**[Description format]**

```
-lw[number-of-characters]
```

- 省略时解释
- `-lw132`（控制器输出为 80 个字符）

[功能]

- `-lw` 选项设定每个类型列表文件每行的字符数。

[Application]

- 使用 `-lw` 选项改变每个类型列表文件每行的字符数。

[说明]

- 可以由 `-lw` 选项设定的字符数为 72 ~ 132，其中不包含端接器（CR，LF）。
- - 如果字符数省略，每行的字符数为 132（控制器输出为 80 个字符）。
- 如果没有设定列表文件，`-lw` 选项无效。

[Example of use]

- 设定 72 作为交叉引用列表文件（`prime.xrf`）每行的字符数，描述如下：

```
C>cc78k0r -cf1166a0 prime.c -x -lw72
```

-ll

[Description format]

```
-ll[number-of-lines]
```

- 省略时解释
无分页符

[Function]

- ll 选项设定每个类型列表文件每页的行数。

[Application]

- 使用 ll 选项改变每个类型列表文件每页的行数。

[说明]

- ll 选项可指定的行数范围为 20~65535。
- 如果设定 -ll0，将没有分页符建立。
- 如果省略行数，将不会生成分页符。
- 如果没有设定列表文件，ll 选项无效。

[Example of use]

- 设定 20 作为交叉引用列表文件（prime.xrf）每页的行数，描述如下：

```
C>cc78k0r -cf1166a0 prime.c -x -ll20
```

-lt

[Description format]

```
-lt[number-of-characters]
```

- 省略时解释
- lt8

[功能]

- lt 选项设定字符的基号用于在源文件中输出水平列表（HT）代码，使用几个空格（空格键）在每个列表中替换字符基号。

[Application]

- 使用 -lt 选项通过减小每个 HT 代码的空格数来减小每行的字符数，例如通过 -lw 选项设定较小的行字符数。

[说明]

- lt 选项可指定的字符数范围为 0~8。
- 如果 -lt0 设定，列表进程将不执行，列表代码将输出。
- 如果省略字符数，制表位的扩展字符数为 8。
- 如果没有设定列表文件，-lt 选项无效。

[Example of use]

- 如果省略 -lt 选项，编译器假设设定 -lt8 选项并由 HT 代码键入的空格数设为 8。

```
C>cc78k0r -cf1166a0 prime.c -p
```

- 设定通过 HT 代码键入的空格，描述如下：

```
C>cc78k0r -cf1166a0 prime.c -p -lt1
```

-lf

[Description format]

```
-lf
```

- 省略时解释
没有换页码插入。

[功能]

- lf 选项在每个列表文件的最后插入换页码。

[说明]

- 如果没有设定列表文件，-lf 选项无效。

[Example of use]

- 在汇编源文件（prime.asm）的最后插入换页码，描述如下：

```
C>cc78k0r -cf1166a0 prime.c -a -lf
```

-li

[Description format]

```
-li
```

- 省略时解释
没有 C 源代码添加到 include 文件中。

[功能]

- li 选项将 include 文件中的 C 源代码添加到包含 C 源代码注释的汇编源文件中。

[说明]

- 如果设定 -sa 选项， -li 选项无效。

[Example of use]

- 将 include 文件中的 C 源代码添加到包含 C 源代码注释的汇编源文件（prime.asm）中，描述如下：

```
C>cc78k0r -cfl166a0 prime.c -sa -li
```

警告输出说明

警告输出选项说明如下。

- **-w**

-w

[Description format]

```
-w[level]
```

- 省略时解释
-w1

[功能]

-w 选项设定是否输出警告消息到控制台。

[Application]

- 使用 -w 选项设定是否输出警告消息到控制台。
详细消息也能输出。

[说明]

- 警告消息的级别如下。

| 级别 | 说明 |
|----|------------|
| 0 | 没有警告消息输出。 |
| 1 | 标准的警告消息输出。 |
| 2 | 详细的警告消息输出。 |

- 如果设定 -e 或 -se 选项，警告消息同时输出到错误列表文件。
- 如果设定级别 0，警告消息不输出到控制台和错误列表文件（当设定 -e 或 -se 时）。

[Example of use]

- 如果省略 -w 选项，编译器假设 -w1 选项设定并输出标准警告消息。

```
C>cc78k0r -cf1166a0 prime.c
```

执行状态显示设定

执行状态显示设定选项如下。

- `-v/-nv`

`-v/-nv`

[Description format]

```
-v  
-nv
```

- 省略时解释
-nv

[功能]

- `-v` 选项输出当前编译的执行状态到控制台。
- `-nv` 选项使 `-v` 选项失效。

[Application]

- 使用 `-v` 选项检查编译的执行状态。

[说明]

- 处理中的阶段名和函数名输出。
- 如果同时设定 `-v` 选项和 `-nv` 选项，最后设定的选项具有优先级。

[Example of use]

- 输出当前编译的执行状态到控制台，描述如下：

```
C>cc78k0r -cf1166a0 prime.c -v
```

参数文件说明

参数文件选项说明如下。

-f

-f

[Description format]

```
-f file-name
```

- 省略时解释
选项和输出文件名仅能从命令行输入。

[功能]

- f 选项从指定文件输入选项和输入文件名。

[Application]

- 当需要启动 CA78K0 的信息由于在编译过程中两个或两个以上选项输入而无法由命令行输入时使用 -f 选项。
- 当每次执行编译时重复设定选项，在参数文件中描述选项和设定 -f 选项。

[说明]

- 参数文件嵌套是禁止的。
- 能在参数文件中描述的字符数是没有限定的。
- 使用空格键或 tab 键分离选项或输入文件名。
- 参数文件中的选项和输入文件名可以在命令行设定的参数文件位置扩展。
- 最后设定的扩展选项具有优先级。
- 在每行末端位置之前，下列字符 “;” 或 “#” 全部假定为注释。

[Example of use]

- 参数文件（prime.pcc）的内容

```
; parameter file  
prime.c -cf1166a0 -aprime.asm -e -x
```

使用参数文件（prime.pcc）执行编译。

```
C>cc78k0r -fprime.pcc
```

临时文件创建文件夹说明

临时文件创建文件夹选项说明如下。

-t

-t

[Description format]

```
-tfolder
```

- 省略时解释

在通过环境变量 `TMP` 设定的驱动器文件夹内创建临时文件。如果没有设定环境变量 `TMP`，在当前驱动器和当前文件夹内创建临时文件。

[功能]

-t 选项设定创建临时文件的驱动器和文件夹。

[Application]

- 使用 -t 选项设定临时文件的创建位置。

[说明]

- 即使之前创建的临时文件存在，如果文件没有保护，其将在下次重写。

- 只要需要的存储大小可用，临时文件将在存储器中扩展。

如果需要的存储大小不再可用，在指定文件夹创建的临时文件以及存储内容写入文件。访问后来的临时文件就是访问不在存储器中的文件。

- 当编译结束时，临时文件删除。通过 `[CTRL] + [C]` 按键退出编译时，临时文件也删除。

[Example of use]

- 输出临时文件到文件夹 `C:\tmp`，描述如下：

```
C>cc78k0r -cf1166a0 prime.c -ttmp
```

函数扩展说明

函数扩展选项说明如下。

-z/-nz

-z/-nz

[Description format]

-z 类型 (可以设定两个或两个以上类型)

-nz

- 省略时解释

-nz

[功能]

- z 选项允许扩展函数的使用。
- nz 选项使 -z 选项失效。
- 不能省略类型。如果省略设定，会出现致命错误 (F0012)。

[Application]

- 通过下述类型设定的用于处理的函数对于 78K0 扩展函数是可用的。

[说明]

- z 选项的类型设定如下。

| 类型说明 | 说明 |
|------|---------------------------------------|
| p | 字符插入在 “//” 后换行码之前用作注释。 |
| c | 嵌套注释是禁止的。 |
| s 注意 | 在注释中插入 kanji 代码作为 SJIS。 |
| e 注意 | 在注释中插入 kanji 代码作为 EUC。 |
| n 注意 | 说明注释不包含 kanji 代码。 |
| b | char-/unsigned char 类型自变量和返回值不是整型扩展的。 |

| 类型说明 | 说明 |
|------|--|
| a | <p>不在 ANSI 标准内的函数是无效的。在 ANSI 标准内函数的部分是有效的。</p> <p>下述任务逐一地执行。</p> <ul style="list-style-type: none"> - 下述不再是保留字。 <code>callt/sreg/bit/boolean/#asm/#endasm</code> - 三维图序列（3 字符表达式）是有效的。 - 编译器定义的宏 <code>__STDC__</code> 当作是 1。 - 通过执行 3 字节的 <code>far</code> 指针关系表达式，可将数据分配到 64KB 边界区域的最后一个字节。 - 下述的警告是整数型位字段的输出。 (CC78K0 警告 W0787：位字段类型不是整数) - 如果设定 <code>-w2</code> 针对 <code>-qc</code>，<code>-zp</code>，<code>-zc</code> 选项，则输出下列警告。 (CC78K0 警告 W0029：'-QC' 选项不能改变) (CC78K0 警告 W0031：'-ZP' 选项不能改变) (CC78K0 警告 W0032：'-ZC' 选项不能改变) - 如果设定 <code>-w2</code> 针对每个 <code>#pragma</code> 语句，则输出下述警告。 (CC78K0 警告 W0849：<code>#pragma</code> 语句不能改变) - 如果设定 <code>-w2</code> 针对 <code>__asm</code> 语句，会输出下述警告，同时执行汇编输出。 (CC78K0 警告 W0850：<code>Asm</code> 语句不能改变) - 如果设定 <code>-w2</code> 针对 <code>#asm</code> 到 <code>#endasm</code> 的块，输出下述错误。 (CC78K0 错误 E0801：未定义的控制等) |
| f | 输出 flash 的目标。 |
| t 地址 | 设定 Flash 区域分支表的起始地址。 |
| z 地址 | 设定 Flash 区域的起始地址。 |

注 不能同时设定 s,e 和 n。

[Example of use]

- 字符作为注释插入在 C 源代码文件（prime.c）的“//”之后和换行码之前。同时，嵌套注释是禁止的。

```
C>cc78k0r -cf1166a0 prime.c -zpc
```

设备文件查找路径说明

设备文件查找路径选项说明如下。

-y

-y

[Description format]

`-yfolder`

- 省略时解释
仅一般的查找路径

备注 一般查找路径如下。

(1) `<..\..\dev >` (该路径用于启动 `cc78k0.exe`)

(2) 该路径用于启动 `cc78k0.exe`

(3) 当前文件夹

(4) 环境变量 `PATH`

[功能]

-y 选项首先查找设定为读取设备文件的查找路径。如果其不存在，则查找一般路径。

[Application]

- 如果设备文件没有安装在标准查找路径，而在指定的文件夹，则通过该选项设定路径。

[Cautions]

- 当使用 `CubeSuite` 时，文件夹通过在工程创建时选择的微控制器确定，因此当使用该编译器设定选项时，不需要设定该选项。

[Example of use]

- 查找 “`C:\tmp\dev`” 读取设备文件，描述如下：

```
C>cc78k0r -cf1166a0 -yC:\tmp\dev
```

存储类型说明

存储类型选项说明如下。

-m

-m

[Description format]

-m 类型

- 省略时解释
-mm

[功能]

- 使用 -m 选项，为编译指定使用的存储模式。
- 不可以设定两个或多个处理类型。
- 不能省略类型。如果省略设定，会出现致命错误（F0012）。

[Application]

- 使用 -m 选项，是否按照设定的存储模式在 near 和 far 区域分配每个函数和变量。
- 如果在 C 源文件中修饰符 __near 或 __far 用来描述函数和变量，则由 __near 或 __far 修饰符设定的 near 和 far 区域说明优先。

[说明]

-m 选项的类型说明如下。

| 类型说明 | 存储模块 | 说明 |
|------|------|---|
| s | 小模块 | 该存储应该包含代码部分 64KB(最大) 和 64KB 数据部分(最大)，总共 128KB, 并设定 near 和 far 区域。 |
| m | 中等模块 | 该存储应该包含代码部分 1MB(最大) 和 64KB 数据部分(最大)，总共 1MB, 并设定 near 和 far 区域。 |
| l | 大型模块 | 该存储应该包含代码部分 1MB(最大) 和 1MB 数据部分(最大)，总共 1MB, 并设定 near 和 far 区域。 |

注意事项 即使存储模式设定了包含 64KB 的数据部分或代码部分，指定带有 __far 修饰符的函数和变量能分配到 1MB 的空间（最大）。

存储模块说明指定在未使用 __near/ __far 修饰符设定的函数或变量位置上。

[Example of use]

- 在编译期间，使用存储模式为小模块，描述如下：

```
C>cc78k0r -cfl166a0 prime.c -ms
```

镜像空间说明

镜像空间选项说明如下。

-mi

-mi

[Description format]

```
-mi0 or -mi1
```

- 省略时解释
-mi0

[功能]

-m 选项，在镜像区域设定段的可分配目标。

[Application]

- 使用 -m 选项，在镜像区域设定段的可分配目标。

[说明]

- 如果设定了 -mi0, 当 MAA=0 时，则段分配在镜像空间中。如果设定了 -mi1, 当 MAA=1, 则段分配在镜像空间中。
有关镜像空间的详细内容，参阅设备的用户手册。

[Example of use]

- 当 MAA=1 时，在镜像空间里分配段，描述如下：

```
C>cc78k0r prime.c -mi1
```

公用目标设定

公用目标设定选项如下。

- `-common`

-common

[Description format]

```
-common
```

- 省略时解释
输出指定设备目标。

[功能]

- `-common` 选项设定 78K0 公用对象的输出。

[Application]

- 使用 `-common` 选项生成对象，该对象可以在 78K0 中共享，不需考虑设备类型设定选项 (`-c`)。

[说明]

- 指定生成可以在 78K0 中公用对象的选项。

[Example of use]

- 生成可以在 78K0 中公用的对象，描述如下：

```
C>cc78k0r prime.c -cf1166a0 -common
```

变量 / 函数信息文件说明

变量 / 函数信息文件选项说明如下。

- **ma**

-ma

[Description format]

```
-mafile-name[ -mafile-name ]  
-mafile-name[ ,file-name ]
```

- 省略时解释
不使用变量 / 函数信息文件。

[功能]

- **ma** 选项设定使用的变量 / 函数信息文件。

[Application]

- 使用 **ma** 选项，使用变量 / 函数信息文件有效分配变量和函数。

[说明]

- 设定最多可达 2 个文件名。
- 使用变量 / 函数信息文件设定属性，把变量和函数和 C 源代码分开。
有关变量 / 函数信息文件的详细内容，参阅 "B.7 变量 / 函数信息文件生成器"。
- 如果同时设定 **ma** 和 **rd** 选项，则将忽略 **rd** 选项并输出警告信息。
- 如果同时设定 **ma** 和 **rs** 选项，分配时则 **ma** 将取消，因为 **rs** 具有更高优先级。

[Example of use]

- 由使用变量 / 函数信息文件 (k0r.vfi) 分配变量 / 函数，描述如下：

```
C>cc78k0r prime.c -cf1166a0 -mak0r.vfi
```

帮助说明

帮助选项如下。

- --/?/-h

--/?/-h

[Description format]

```
--/?/-h
```

- 省略时解释
无显示

[功能]

- --, -? 和 -h 选项显示选项的主要说明和帮助信息例如在控制台上的默认选项。

注意事项 该选项不能通过 **CubeSuite** 设定。

[Application]

- 显示选项及其说明。当执行编译时参阅这些内容。

[说明]

- 当设定 --, -? 或 h 选项时，所有其他选项无效。
- 查看显示帮助信息的后续内容，按 [Enter] 键。按 [Enter] 键以外的任意键可退出显示，然后按 [Enter] 键即可。

[Example of use]

- 在控制台上输出帮助消息。

```
C>cc78k0r -h
```

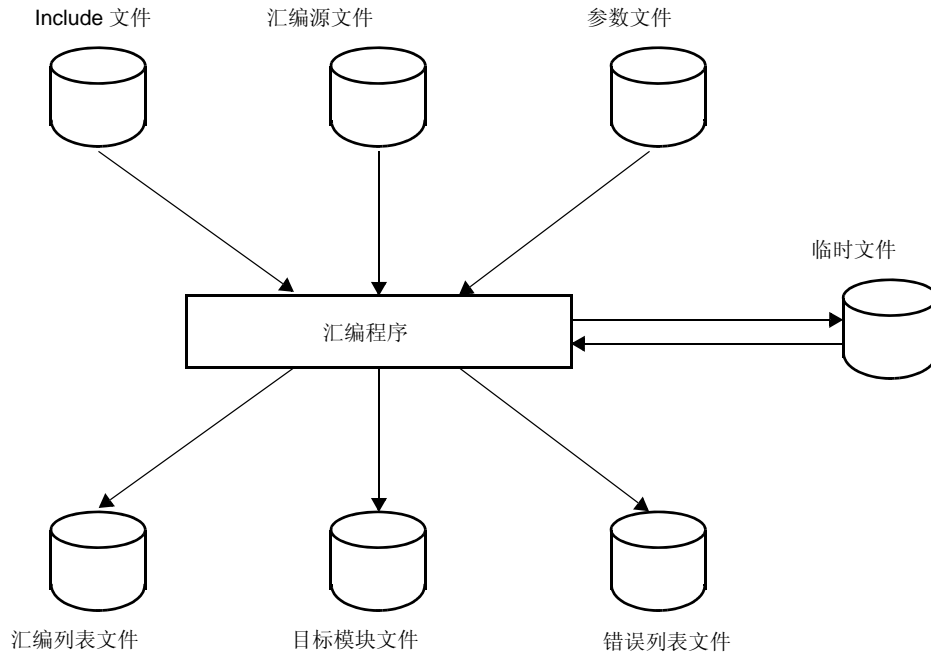
B.2 汇编程序

汇编程序将输入的 78K0 微控制器汇编语言源文件，转化为机器语言代码，并将它们输出为目标模块文件。

汇编程序也会输出表文件，比如汇编表文件和错误表文件。

如果汇编出错，会将错误信息输出到汇编表文件，错误表文件就是为了说明这些错误的原因。

图 B-3. 汇编程序的 I/O 文件



B.2.1 I/O 文件

汇编程序的 I/O 文件显示如下。

详细的输出文件见 "3.2 汇编程序"。

表 B-5. 汇编程序的 I/O 文件

| 类型 | 文件名 | 说明 | 默认文件类型 |
|------|------------|--|--------|
| 输入文件 | 汇编源文件 | - 78K0 微控制器使用的汇编语言源文件（用户创建的文件） | .asm |
| | Include 文件 | - 汇编源文件引用的文件。 - 78K0 微控制器使用的汇编语言文件（用户创建的文件） | None |
| | 参数文件 | - 包含执行程序所需参数的文件（用户创建的文件） | .pra |
| 输出文件 | 目标模块文件 | - 包含与机器语言和机器语言地址定位相关的重定位信息与符号信息的二进制文件 | .rel |
| | 汇编列表文件 | - 包含如汇编表和交叉参考表等汇编信息的文件 | .prn |
| | 错误列表文件 | - 包含汇编期间产生错误信息的文件 | .era |

| 类型 | 文件名 | 说明 | 默认文件类型 |
|--------|------|---------------------------------------|---------------------------|
| I/O 文件 | 临时文件 | - 由汇编程序自动产生的用于汇编的文件 临时文件在汇编结束后会删除。 | RAxxxx.\$\$n (n = 1~4) |

B.2.2 功能

(1) 将汇编语言转化为机器语言

汇编程序读取源文件，将它们从汇编语言转化为机器语言并保存。

B.2.3 操作方法

(1) 汇编程序启动

可以用以下两种方法启动汇编程序。

(a) 从命令行启动

```
X:[path-name]>ra78k0r[Δoption] ... source-file-name [Δoption] ... [Δ]
```

| | |
|--------|--|
| X | 当前驱动器名称 |
| 路径名 | 当前文件夹名 |
| ra78k0 | 汇编程序的命令名 |
| 选项 | 为汇编程序的执行输入详细的指令。 当指定两个或更多汇编选项时，用空格分隔这些选项。汇编选项不区分大小写。详细的汇编选项见 "B.2.4 选项"。 带有空格的路径附在一对双引号 (") 中。 |
| 源文件名 | 将要汇编的文件名 带空格的文件名路径附在一对双引号 (") 中。 |

示例 对输出错误表文件 k0rmain.era，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -e -np
```

(b) 从参数文件启动

当数据要求不需要配合命令行，或当每次执行汇编都要重复指定同样的汇编属性时，使用参数文件启动汇编程序。

通过参数文件启动汇编程序，在命令行上设定参数文件选项 (-f)。

按如下方法用参数文件启动汇编程序：

```
X>ra78k0r[ΔSource-file]Δ-fparameter-file-name
```

| | |
|-------|-------------|
| -f | 参数文件说明选项 |
| 参数文件名 | 包含汇编需要数据的文件 |

备注 使用编辑器创建参数文件。

参数文件内容的书写规则如下：

```
[[[Δ]Option[ΔOption] ... [Δ]Δ]] ...
```

- 如果命令行省略源文件名，参数文件中仅可以指定 1 个源文件名。
- 源文件名也可以在加在选项后。
- 在参数文件中，所有的汇编选项和输入文件都要在命令行后指定。

示例 使用编辑器创建参数文件 k0rmain.pra，之后起订汇编程序。

```
; parameter file
k0rmain.asm -osample.rel
-psample.prn
```

```
C>ra78k0r -fk0rmain.pra
```

(2) 执行开始和结束消息

(a) 执行开始消息

当汇编程序启动时，在显示器上会显示执行启动消息。

```
78K0R Assembler Vx.xx [xx xxx xxxx]
Copyright(C) NEC Electronics Corporation xxxx
```

(b) 执行结束消息

如果检测到没有汇编错误，则汇编程序会在显示器上输出如下消息，同时将控制权返回给主操作系统。

```
PASS1 Start
PASS2 Start

Target chip : uPD78xxx
Device file : Vx.xx

Assembly complete, 0 error(s) and 0 warning(s) found.
```

如果检测到有汇编错误，则汇编程序会在显示器上输出错误数目，同时将控制权返回给主操作系统。

```
PASS1 Start
k0rmain.asm ( 12 ) : RA78K0R error E2201 : Syntax error
PASS2 Start
k0rmain.asm ( 12 ) : RA78K0R error E2201 : Syntax error
k0rmain.asm ( 29 ) : RA78K0R error E2407 : Undefined symbol reference 'CONVAH'
k0rmain.asm ( 29 ) : RA78K0R error E2303 : Illegal expression

Target chip : uPD78xxx
Device file : Vx.xx

Assembly complete, 3 error(s) and 0 warning(s) found.
```

如果汇编程序检测到在汇编过程中有致命错误，导致汇编不能进行，则汇编程序输出在显示器上输出消息，取消汇编，同时将控制权返回给主操作系统。

示例 1. 指定的源文件不存在。

```
C>ra78k0r sample.asm
```

```
78K0R Assembler Vx.xx [xx xxx xxxx]
      Copyright(C) NEC Electronics Corporation xxxx

RA78K0R error F2006 : File not found 'sample.asm'
程序终止。
```

在上面的例子中，指定的源文件是不存在的。产生错误且汇编程序终止汇编。

2. 指定的汇编选项不存在。

```
C>ra78k0r k0rmain.asm -z
```

```
78K0R Assembler Vx.xx [xx xxx xxxx]
      Copyright(C) NEC Electronics Corporation xxxx

RA78K0R error F2012 : Missing parameter '-z'
Please enter 'RA78K0R--' , if you want help messages.
程序终止。
```

在上面的例子中，指定的汇编选项是不存在的。产生错误且汇编程序终止汇编。

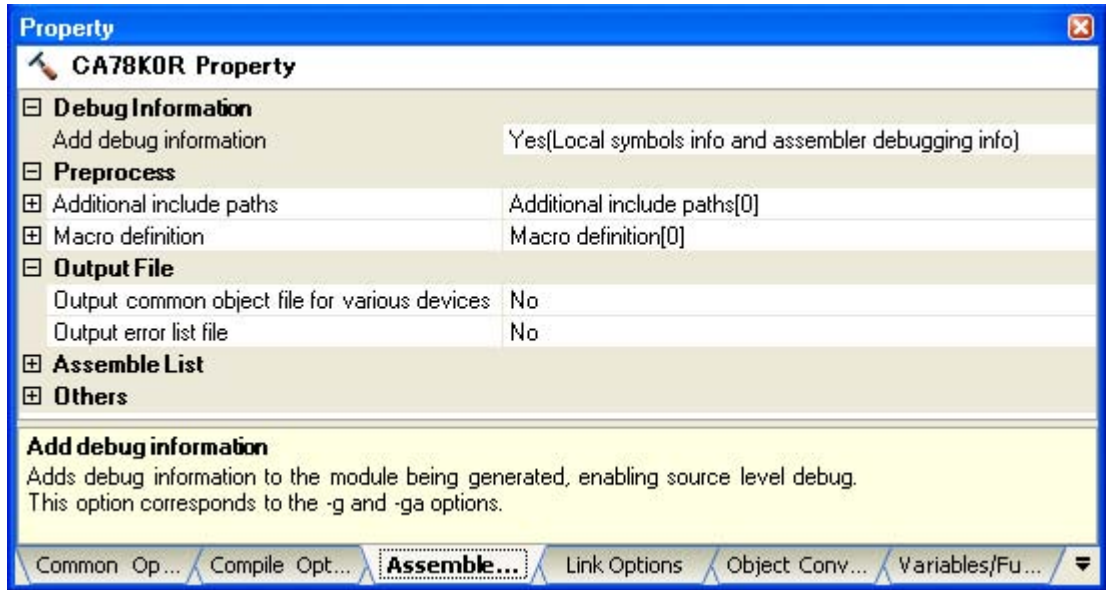
(3) 在 CubeSuite 中设置选项

这部分描述如何通过 CubeSuite 设置汇编选项。

通过 CubeSuite 的 [工程树面板](#)，选择 **Build Tool** 节点。然后通过 **[View]** 菜单选择 **[Property]**。打开 [属性面板](#)。再选择 [\[Assemble Options\]](#) 标签。

可以通过该属性页给各种汇编选项设置必要的值。

图 B-4. 属性面板 : [Assemble Option] 标签



B.2.4 选项

(1) 类型

汇编选项是汇编程序的执行的详细指令。

汇编选项的类型和解释显示如下：

表 B-6. 汇编选项

| 分类 | 选项 | 说明 |
|--------------|------|---------------------------|
| 设备类型说明 | -c | 指定目标设备的类型。 |
| 目标模块文件输出指定 | -o | 指定目标模块文件的输出。 |
| | -no | |
| 强制指定目标模块文件输出 | -j | 强制目标模块文件的输出。 |
| | -nj | |
| 调试信息输出说明 | -g | 指定将调试信息（局部符号信息）加到目标模块文件中。 |
| | -ng | |
| | -ga | 指定将汇编源文件调试信息加到目标模块文件中。 |
| | -nga | |
| 包含文件读取路径指定 | -i | 从指定的路径中读取包含文件。 |
| 汇编表文件输出指定 | -p | 指定汇编表文件的输出。 |
| | -np | |
| 汇编表文件信息指定 | -ka | 将汇编表输出到汇编表文件。 |
| | -nka | |
| | -ks | 将符号表输出到汇编表文件。 |
| | -nks | |
| | -kx | 输出交叉参考表到汇编表文件。 |
| | -nkx | |

| 分类 | 选项 | 说明 |
|-------------------|-----------|---------------------------------|
| 汇编表文件格式指定 | -lw | 改变汇编表文件中每行字符数。 |
| | -ll | 改变汇编表文件中每页行数。 |
| | -lh | 将标题中的特殊字符串输出到汇编表文件。 |
| | -lt | 指定 tab 键的填充字符数。 |
| | -lf | 在汇编表文件的最后插入换页码。 |
| | -nlf | |
| 错误表文件输出指定 | -e | 输出错误列表文件 |
| | -ne | |
| 参数文件说明 | -f | 通过指定文件输入选项和输入文件名。 |
| 临时文件创建路径设定 | -t | 在指定的路径建立临时文件。 |
| Kanji 指令（双字节指令）设定 | -zs | 用 Shift-JIS 编码解释注释部分的 Kanji 描述。 |
| | -ze | 用 EUC 编码解释注释部分的 Kanji 描述。 |
| | -zn | 不将注释部分的说明字符认为时 Kanji。 |
| 设备文件查找路径说明 | -y | 从指定的路径中读取器件文件。 |
| 符号定义设定 | -d | 定义符号。 |
| 公用目标设定 | -common | 指定输出目标模块文件和 78K0 公共的部分作为输出。 |
| 78K0 兼容宏 | -compati | 可以汇编由 78K0 汇编程序生成的汇编程序源文件。 |
| | -ncompati | |
| 帮助说明 | -- | 在屏幕上输出帮助信息。 |

(2) 优先级

对下表所列的汇编选项，说明当同时指定行或列中多个选项时的优先级。

表 B-7. 汇编选项的优先级

| | -no | -np | -nka | -nks | -kx | -nkx | -- |
|-----|-----|-----|------|------|-----|------|----|
| -j | NG | | | | | | NG |
| -g | NG | | | | | | NG |
| -p | | | △ | △ | | △ | NG |
| -ka | | NG | | | | | NG |
| -ks | | NG | | | NG | | NG |
| -kx | | NG | | | | | NG |
| -lw | | NG | | | | | NG |
| -ll | | NG | | | | | NG |
| -lh | | NG | | | | | NG |
| -lt | | NG | | | | | NG |
| -lf | | NG | | | | | NG |

- 标有 NG 的位置

如果设定选项在横坐标上，则此选项在纵坐标上无效。

示例 -lw 和 -lf 选项是无效的。

```
C>ra78k0r -cf1166a0 k0rmain.asm -np -lw80 -lf
```

- 标有 Δ 的位置

如果同时指定行中的所有三个选项，那么列选项是无效的。

示例 如果同时指定 `-nka`、`-nks` 和 `-nkx` 选项，则 `-p` 选项时无效的。

```
C>ra78k0r -cf1166a0 k0rmain.asm -p -nka -nks -nkx
```

- 空白区域

如果设定水平轴上的选项，则此选项在垂直轴上无效。

指定 `-o/-no` 选项时，如果同时设定了两个可在选项前加“n”的选项，则最近指定的选项是有效的。

示例 在 `-o` 选项后设定 `-no` 选项时，`-o` 选项是无效的，而 `-no` 选项是有效的。

```
C>ra78k0r -cf1166a0 k0rmain.asm -o -no
```

没有在“表 B-7. 汇编选项的优先级”描述的选项不受其它选项的影响。然而，如果指定了帮助选项 (`--`)，所有其它的选项都是无效的。

设备类型说明

设备类型的选项说明如下。

- **c**

-c

[Description format]

```
-cdevice-type
```

- 省略时解释
不能省略。

[功能]

- **c** 选项指定汇编的目标器件。

[Application]

- 确保设定了 **-c** 选项。汇编程序为目标器件执行汇编且为指定的器件生成目标代码。

[说明]

- **c** 选项可以指定器件请参考 *CubeSuite Operating Precautions*。

[Cautions]

- **c** 选项不能省略。如果在源文件的开头说明了与 **-c** 选项相同功能的控制指令（`$PROCESSOR`）的话，命令行中可以不指定此选项。

```
Δ$ΔPROCESSORΔ(Δdevice-typeΔ)
Δ$ΔPCA(Δdevice-typeΔ) ; Abbreviated form
```

[Example of use]

- 要指定 `uPD78F1166_A0` 作为目标器件，说明如下：

```
C>ra78k0r -cf1166a0 main.asm
```

目标模块文件输出指定

目标模块文件输出指定如下：

- `-o/-no`

`-o/-no`

[Description format]

```
-o[output-file-name]  
-no
```

- 省略时解释
- `-oinput-file-name.rel`

[功能]

- `-o` 选项指定输出目标模块文件。该选项同时设定输出位置和文件名。
- `-no` 选项使 `-o`、`-j`、`-g` 和 `-ga` 选项无效。

[Application]

- 使用 `-o` 选项设定输出目标模块文件的路径或改变文件名的路径。
当汇编程序仅输出汇编表文件时，指定 `-no` 选项。这会缩短汇编时间。

[说明]

- 当发生严重错误时，即使指定 `-o` 选项，也不会输出目标模块文件。
- 如果设定 `-o` 选项时省略了驱动器名，则会输出目标模块文件到当前驱动器。
- 如果设定 `-o` 选项时省略了文件名，则输出文件名为 "`input-file-name.rel`"。
- 如果同时指定了 `-o` 和 `-no` 选项，则最后指定的选项无效。

[Example of use]

- 输出 Hex 文件 (`sample.rel`)，说明如下：

```
C>ra78k0 -cf1166a0 k0rmain.asm -osample.rel
```

强制指定目标模块文件输出

强制目标模块文件输出指定如下：

- /-nj

- /-nj

[Description format]

```
-j  
-nj
```

- 省略时解释
-nj

[功能]

- j 选项指定即使发生严重错误也输出目标模块文件。
- nj 选项禁止 -j 选项。

[Application]

- 通常，当发生严重错误时，不输出目标模块文件。如需要在发生严重错误时执行程序，则设定 -j 选项输出目标模块文件。

[说明]

- 当指定 -j 选项时，即使发生严重错误也输出目标模块文件。
- 如果同时指定了 -j 和 -nj 选项，则最后指定的选项无效。
- 如果指定 -no 选项时，-j 选项无效。

[Example of use]

- 即使发生严重错误也输出模块文件 (k0rmain.rel)，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -j
```

调试信息输出说明

调试信息输出选项说明如下。

- g/-ng
- ga/-nga

-g/-ng**[Description format]**

```
-g
-ng
```

- 省略时解释
- g

[功能]

- g 选项指定将调试信息（局部符号信息）加到目标模块文件中。
- ng 选项禁止 -g 选项。

[Application]

- 当执行包含局部符号数据的符号调试时，使用 -g 选项。
- 在下面的例子中使用 -ng 选项。

- (1) 仅对全局符号执行符号调试
- (2) 无符号调试
- (3) 当仅需要目标时（用 PROM 计算等）

[说明]

- 如果同时设定了 -g 和 -ng 选项，则最后设定的选项无效。
- 如果同时指定 -g/-ng 和 -ga/-nga 时，-ga/-nga 选项是无效的，不管它处在什么位置。
- 如果指定 -no 选项时，-g 选项无效。

[Cautions]

- 具有与 -g 和 -ng 相同功能的控制代码（DEBUG/NODEBUG 或 DG/NODG）可以在源代码开始时声明。声明格式如下所示。

```
Δ$ΔDEBUG
Δ$ΔDG      ; Abbreviated form
Δ$ΔNODEBUG
Δ$ΔNODG    ; Abbreviated form
```

[Example of use]

- 将调试信息（局部符号信息）加入到目标模块文件 (k0rmain.rel)，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -g
```

-ga/-nga**[Description format]**

```
-ga
-nga
```

- 省略时解释
- ga

[功能]

- ga 选项指定将汇编源文件调试信息加到目标模块文件中。
- nga 选项禁止 -g 和 -ga 选项。

[Application]

- 当执行源代码级汇编调试时使用 -ga 选项。执行源代码级汇编调试，需要单独可用集成调试器。
- 在下面的例子中使用 -nga 选项。

(1) 无汇编源代码调试**(2) 当仅需要目标时（用 PROM 计算等）****(3) 在 C 编译器源文件级调试****[说明]**

- 如果同时指定了 -ga 和 -nga 选项，则最后指定的选项无效。
- 如果同时指定 -g/-ng 和 -ga/-nga 时，-ga/-nga 选项是无效的，不管它处在什么位置。
- 如果指定 -no 选项时，-ga 选项无效。

[Cautions]

- 具有与 -ga 和 -nga 相同功能的控制代码（DEBUG/NODEBUG）可以在源代码开始时声明。声明格式如下所示。

```
Δ$ΔDEBUG
Δ$ΔNODEBUGA
```

[Example of use]

- 将汇编源文件调试信息加入到目标模块文件 (k0rmain.rel)，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -ga
```

包含文件读取路径指定

指定包含文件读取路径选项如下：

-i

-i

[Description format]

```
-ipath-name[,path-name] ... (可以设定两个或以上的路径)
```

- 省略时解释
按如下顺序查找包含文件。

(1) 源代码所在路径

(2) 环境变量 (INC78K0) 指定的路径

[功能]

- i 选项指定的在源文件中通过 “\$include” 指定的包含文件从指定的路径读取。

[Application]

- 使用 -i 选项在指定的路径查找。

[说明]

- 两个或多个路径名可用通过分隔符 “,” 指定。
- 在 “,” 前后不能有空格。
- 按如下顺序查找由 “\$include” 指定的包含文件。

(1) 两个或两个以上的路径名 -i 选项设定，包含文件依照指定顺序查找。

(2) 如果 -i 选项设置，包含文件查找具有较低的优先级设置。

(3) 在查找由 -i 选项设定的路径后，包含文件以与选项省略时相同的顺序查找。

- 除在 -i 选项后设定路径名或路径名省略的情况外，均会发生异常中断错误。
- 当 -i 选项指定 65 个或更多时，会产生终止错误。

[Example of use]

- 按 C:\sample1 和 C:\sample2 这样的顺序查找并读取包含文件，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -iC:\sample1,C:\sample2
```

- 从 C:\Program Files\NEC Electronics Tools\include 读取包含文件，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -i" C:\Program Files\NEC Electronics Tools\include files"
```

汇编表文件输出指定

模块汇编表文件输出指定如下：

- `-p/-np`

`-p/-np`

[Description format]

```
-p[output-file-name]  
-np
```

- 省略时解释
- `-pinput-file-name.prn`

[功能]

- `-p` 选项指定汇编表文件的输出。
这也同时指定输出文件名和路径。
- `-np` 选项禁止 `-p`、`-ka`、`-ks`、`-kx`、`-lw`、`-ll`、`-lh`、`-lt` 和 `-lf` 选项。

[Application]

- 使用 `-p` 选项指定输出汇编表文件的路径或改变它的文件名。
- 当汇编程序仅输出目标模块文件时，指定 `-np` 选项。这会缩短汇编时间。

[说明]

- 如果指定 `-p` 选项时省略了文件名，则输出文件名为 "`input-file-name.prn`"。
- 如果指定 `-p` 选项时省略了驱动器名，则会将输出汇编表文件到当前驱动器上。
- 如果同时指定了 `-p` 和 `-np` 选项，则最后指定的选项无效。

[Example of use]

- 输出汇编表文件 (`sample.prn`)，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -psample.prn
```

汇编表文件信息指定

模块汇编表文件信息指定如下：

- ka/-nka
- ks/-nks
- kx/-nkx

-ka/-nka

[Description format]

```
-ka  
-nka
```

- 省略时解释
-ka

[功能]

- ka 选项输出汇编表到汇编表文件。
- nka 选项禁止 -ka 选项。

[Application]

- 使用 -ka 选项输出汇编表。

[说明]

- 如果同时指定了 -ka 和 -nka 选项，则最后指定的选项无效。
- 如果同时指定了 -nka、-nks 和 -nkx 选项，不会输出汇编表文件。
- 如果指定 -np 选项时，-ka 选项无效。

[Example of use]

- 将汇编表文件输出到汇编表文件 (k0rmain.prn)，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -ka
```

k0rmain.prn 文件内容如下：

```

Assemble list

ALNO  STNO  ADRS   OBJECT  M I  SOURCE STATEMENT
 1     1
 2     2                                NAME    SAMPM
 3     3                                ;*****
 4     4                                ;
 5     5                                ;    HEX -> ASCII Conversion Program
 6     6                                ;
 7     7                                ;          main-routine
 8     8                                ;
 9     9                                ;*****
10    10
11    11                                PUBLIC  MAIN , START
12    12                                EXTRN   CONVAH
13    13                                EXTRN   @_STBEG
14    14
15    15 -----                        DATA   DSEG    AT 0FFE20H
16    16 FFE20                        HD TSA:  DS      1
17    17 FFE21                        STASC:  DS      2
18    18
19    19 -----                        CODE    CSEG    AT 0H
20    20 00000 R0000                    MAIN:   DW      START
21    21
22    22 -----                        CSEG
23    23 00000                        START :
24    24
25    25                                ; chip initialize
26    26 00000 RCBF80000 MOVW    SP , #_@STBEG
27    27
28    28 00004 CD201A  MOV      HD TSA , #1AH
29    29 00007 3620FE                                MOVW   HL , #LOWW ( HD TSA ) ; set hex 2-code
在 HL 寄存器中的数据
:
```

-ks/-nks**[Description format]**

```
-ks
-nks
```

- 省略时解释
- nks

[功能]

- ks 选项将符号表附在汇编表后输出到汇编表文件。
- nks 选项禁止 -ks 选项。

[Application]

- 使用 -ks 选项输出符号表。

[说明]

- 如果同时指定了 -nka、-nks 和 -nkx 选项，不会输出汇编表文件。
- 如果同时指定 -ks 和 -kx 选项，则忽略 -ks 选项。
- 如果同时指定了 -ks 和 -nks 选项，则最后指定的选项无效。
- 如果指定 -np 选项时，-ks 选项无效。

[Example of use]

- 将符号表附在汇编表文件后输出到汇编表文件 (k0rmain.prn)，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -ks
```

k0rmain.prn 文件内容如下：

| Symbol Table List | | | | Symbol Table List | | | |
|-------------------|------|------|--------|-------------------|------|------|----------|
| VALUE | ATTR | RTYP | NAME | VALUE | ATTR | RTYP | NAME |
| | | CSEG | ?CSEG | | | CSEG | CODE |
| -----H | | EXT | CONVAH | | | DSEG | DATA |
| FFE20H | ADDR | | HDTSA | 0H | ADDR | PUB | MAIN |
| | MOD | | SAMPM | 0H | ADDR | PUB | START |
| FFE21H | ADDR | | STASC | -----H | | EXT | ._@STBEG |

-kx/-nkx**[Description format]**

```
-kx
-nkx
```

- 省略时解释
- nkx

[功能]

- -kx 选项将交叉参考表附在汇编表后输出到汇编表文件。
- -nka 选项禁止 -kx 选项。

[Application]

- 当想知道源文件中定义的所有符号都使用在什么地方和使用的程度，或想知道汇编表的哪一行包含特定的符号这类信息时，可以使用 -kx 选项输出交叉参考表。

[说明]

- 如果同时指定了 -nka、-nks 和 -nkx 选项，不会输出汇编表文件。
- 如果同时指定 -ks 和 -kx 选项，则忽略 -ks 选项。
- 如果同时指定了 -kx 和 -nkx 选项，则最后指定的选项无效。
- 如果指定 -np 选项时，-kx 选项无效。

[Cautions]

- 具有与 -kx 和 -nkx 相同功能的控制代码（XREF/NOXREF 或 XR/NOXR）可以在源代码开始时声明。声明格式如下所示。

```
Δ$ΔXREF
Δ$ΔXR      ; abbreviated form
Δ$ΔNOXREF
Δ$ΔNOXR    ; abbreviated form
```

[Example of use]

- 将交叉参考表附在汇编表文件后输出到汇编表文件 (k0rmain.prn)，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -kx
```

k0rmain.prn 文件内容如下：

| Cross-Reference List | | | | | | | | | | |
|----------------------|--------|----|------|------|---------|-------|--------|--------|----|---------|
| NAME | VALUE | R | ATTR | RTYP | SEGNAME | XREFS | | | | |
| ?CSEG | | | | | CSEG | ?CSEG | 9.98kg | | | |
| CODE | | | | | CSEG | CODE | 8.62kg | | | |
| CONVAH | -----H | | E | | EXT | 12@ | 31 | | | |
| DATA | | | | | DSEG | DATA | 6.80kg | | | |
| HD TSA | FFE20H | | | ADDR | DATA | 16# | 28 | 29 | | |
| MAIN | | 0H | | ADDR | PUB | CODE | 11@ | 9.07kg | | |
| SAMPM | | | | | MOD | | 2# | | | |
| START | | 0H | | R | ADDR | PUB | ?CSEG | 11@ | 20 | 10.43kg |
| STASC | FFE21H | | | ADDR | DATA | 17# | 33 | | | |
| __@STBEG | -----H | | E | | EXT | 13@ | 26 | | | |

汇编表文件格式指定

模块汇编表文件格式指定如下：

- `-lw`
- `-ll`
- `-lh`
- `-lt`
- `-lf/-nlf`

-lw**[Description format]**

```
-lw[number-of-characters]
```

- 省略时解释
- `-lw132` （显示器输出为 80 个字符）

[功能]

- `-lw` 选项指定表文件中每行的字符数。

[Application]

- 使用 `-lw` 选项改变表文件中每行的字符数。

[说明]

- `-lw` 参数可指定的字符数范围为 72~2046，（显示器输出为 80 个字符）。
如果数值超过此范围或指定的不是数字，会产生终止错误。
- 如果没有指定，默认值为“132”。
当汇编表输出到显示器时，假定值为 80。
- 指定的字符数不包含终止符（CR，LF）。
- 如果指定 `-np` 选项时，`-lw` 选项无效。

[Cautions]

- 具有与 `-lw` 选项相同功能的控制代码（WIDTH）可以在源代码开始时声明。
声明格式如下所示。

```
Δ$ΔWIDTH
```

[Example of use]

- 指定汇编表文件每行的字符数为 80(k0rmain.prn)，描述如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -lw80
```

汇编表文件 k0rmain.prn 内容如下：

```

      Assemble list

ALNO  STNO  ADRS   OBJECT   M I  SOURCE STATEMENT

   1    1
   2    2
   3    3           ;*****
   4    4           ;
   5    5           ;   HEX -> ASCII Conversion Program
   6    6           ;
   7    7           ;           main-routine
   8    8           ;
   9    9           ;*****
  10   10
  11   11           PUBLIC  MAIN , START
  12   12           EXTRN   CONVAH
  13   13           EXTRN   @_STBEG
  14   14
  15   15  -----   DATA   DSEG   AT 0FFE20H
  16   16  FFE20     HDTSA:  DS     1
  17   17  FFE21     STASC:  DS     2
  18   18
  19   19  -----   CODE    CSEG   AT 0H
  20   20 00000 R0000   MAIN:   DW     START
  21   21
  22   22  -----           CSEG
  23   23 00000           START :
  24   24
  25   25           ; chip initialize
  26   26 00000 RCBF80000 MOVW   SP , @_STBEG
  27   27
  28   28 00004 CD201A  MOV    HDTSA , #1AH
  29   29 00007 3620FE  MOVW   HL , #LOWW ( HDTSA ) ; set hex 2-code data in
HL 寄存器
      :
```

-ll

[Description format]

```
-ll[number-of-lines]
```

- 省略时解释
- ll0 (No page breaks)

[Function]

- ll 选项指定汇编表文件中每页行数。

[Application]

- 使用 ll 选项改变汇编表文件中每页行数。

[说明]

- ll 选项可指定的行数范围为 20~32767。
- 如果数值超过此范围或指定的不是数字，会产生终止错误。
- 如果没有指定，默认值为 0。
- 如果行数指定为 0，将不会生成分页符。
- 如果指定 -np 选项时，ll 选项无效。

[Cautions]

- 具有与 ll 选项相同功能的控制代码（LENGTH）可以在源代码开始时声明。
声明格式如下所示。

```
Δ$ΔLENGTH
```

[Example of use]

- 指定汇编表文件每页行数为 20(k0rmain.prn)，描述如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -ll20
```

k0rmain.prn 文件内容如下：

```

78K0R Assembler Vx.xx                               Date:xx xxx xxxx Page: 1

Command: -cf1166a0 k0rmain.asm -l120
Para-file:
In-file: k0rmain.asm
Obj-file: k0rmain.rel
Prn-file: k0rmain.prn

Assemble list
-----

78K0R Assembler Vx.xx                               Date:xx xxx xxxx Page: 2

ALNO  STNO  ADRS  OBJECT  M I  SOURCE STATEMENT

  1    1
  2    2                                NAME    SAMPM
  3    3                                ;*****
  4    4                                ;
  5    5                                ;    HEX -> ASCII Conversion Program
  6    6                                ;
  7    7                                ;    main-routine
  8    8                                ;
-----

78K0R Assembler Vx.xx                               Date:xx xxx xxxx Page: 3

ALNO  STNO  ADRS  OBJECT  M I  SOURCE STATEMENT

  9    9                                ;*****
 10   10
 11   11                                PUBLIC MAIN , START
 12   12                                EXTRN  CONVAH
 13   13                                EXTRN  _@STBEG
 14   14
 15   15 -----                                DATA  DSEG    AT 0FFE20H
 16   16 FFE20                                HDTSA: DS     1
:

```

-lh**[Description format]**

| |
|----------------------------------|
| <code>-lhcharacter-string</code> |
|----------------------------------|

- 省略时解释
None

[功能]

- `-lh` 选项指定汇编表文件头标题栏中打印的字符串。

[Application]

- 使用 `-lh` 选项显示汇编表文件简单说明的标题。
- 通过在每页头上打印标题，汇编表文件的内容就会一目了然。

[说明]

- 标题最多可以指定 60 个字符。字符串不能含有空格。
- 如果指定字符串的长度大于 61，则只有开始的 60 个字符时有效的，且不会由错误信息输出。
kanji 和 hiragana 字符（双字节字符）作为两个字符。
如果每行对多字符数时 199 或更少，有效字符串长度会改变如下：
有效长度 = （每行最多字符数） - 60
- 如果没有指定字符串，会产生终止错误。
- 如果指定 `-np` 选项时，`-lh` 选项无效。
- 如果省略 `-lh` 选项，汇编表的标题栏为空。
- 可以在标题栏中使用的字符编码说明如下：

| 字符 | 在命令行中 | 在参数文件中 |
|--|---------------------------------|--|
| * ? > < | 如果用 "" 将其括起来，则可以描述。 | 可以描述。 即使用 "" 括起来，也同命令行一样解释。 |
| ; | 如果用 "" 将其括起来，则可以描述。 | 不可以描述。 (假设为注释) |
| # | 可以描述。 | 不可以描述。 (假设为注释) |
| " (双引号标记) | 不能描述为有效字符。 | 不能描述为有效字符。 |
| 00H | 不可以描述。 | 可以描述。 因此，它可以解释为字符串的末尾。 |
| 03H, 06H, 08H, 0DH, 0EH, 10H, 15H, 17H, 18H, 1BH, 7FH | 不可以描述。 | 可以描述。 因此，这些将以 “!” 出现在汇编表文件中。 (单独的 0DH 将不输出到表。) |
| 01H, 02H, 04H, 05H, 07H, 0BH, 0CH, 0FH, 11H, 12H, 13H, 14H, 16H, 19H, 1CH, 1DH, 1EH, 1FH | 可以描述。 因此，这些将以 “!” 出现在汇编表文件中。 | 可以描述。 因此，这些将以 “!” 出现在汇编表文件中。 |

| 字符 | 在命令行中 | 在参数文件中 |
|------|-------------------------------|------------------|
| 1AH | 可以描述。 因此，这些将以“！”出现在汇编表文件中。 | 不可以描述。 (文件末尾) |
| 字母字符 | 大小写字符按原样输入。 | 大小写字符按原样输入。 |
| 其他 | 可以描述。 | 可以描述。 |

备注 如果启动行上的星号 (*) 不是匹配符扩展的目标，即使不封闭在" "中，它也可以写入。

[Cautions]

- 具有与 -llh 选项相同功能的控制代码 (TITLE 或 TT) 可以在源代码开始时声明。
声明格式如下所示。

```
Δ$ΔTITLEΔ(Δ'character-string'Δ)
Δ$ΔTTΔ(Δ'character-string'Δ) ; abbreviated form
```

[Example of use]

- 打印标题“RA78K0_MAINROUTINE”在汇编表文件 (k0rmain.prn) 的文件头，描述如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -lhRA78K0R_MAINROUTINE
```

k0rmain.prn 文件内容如下：

```
78K0R Assembler Vx.xx RA78K0R_MAINROUTINE Date:xx xxx xx Page:1
|
| Title
|
Command: -cf1166a0 k0rmain.asm -lhRA78K0R_MAINROUTINE
Para-file:
In-file: k0rmain.asm
Obj-file: k0rmain.rel
Prn-file: k0rmain.prn

Assemble list

ALNO STNO ADRS OBJECT M I SOURCE STATEMENT
1 1
2 2 NAME SAMPM
3 3 ;*****
4 4 ;
5 5 ; HEX -> ASCII Conversion Program
6 6 ;
```

```
7 7 ; main-routine  
:
```

-lt

[Description format]

```
-lt[number-of-characters]
```

- 省略时解释
- lt8

[功能]

- lt 选项设定字符的基号用于在源文件中输出水平列表（HT）代码，使用几个空格（空格键）在每个列表中替换字符基号。

[Application]

- 使用 -lt 选项通过减小每个 HT 代码的空格数来减小每行的字符数，例如通过 -lw 选项设定较小的行字符数。

[说明]

- lt 选项可指定的字符数范围为 0~8。
- 如果数值超过此范围或指定的不是数字，会产生终止错误。
- 如果没有指定，默认值为“8”。
- 如果 -lt0 设定，列表进程将不执行，列表代码将输出。
- 如果指定 -np 选项时，-lt 选项无效。

[Cautions]

- 具有与 -lt 选项相同功能的控制代码（TAB）可以在源代码开始时声明。
声明格式如下所示。

```
Δ$ΔTABΔnumber-of-tabs
```

[Example of use]

- 当 -lt 选项省略时，引用汇编表文件（sample.prn）。

```
C>ra78k0r -cf1166a0 sample.asm
```

sample.prn 文件内容如下：

```

Assemble list

ALNO  STNO  ADRS   OBJECT   M I  SOURCE STATEMENT

  1     1
  2     2                                NAME   SAMPM
  3     3                                ;*****
  4     4                                ;
  5     5                                ;   HEX -> ASCII Conversion Program
  6     6                                ;
  7     7                                ;           main-routine
  8     8                                ;
  9     9                                ;*****
10    10
11    11                                PUBLIC MAIN , START
12    12                                EXTRN  CONVAH
13    13                                EXTRN  @_STBEG
14    14
15    15 -----                        DATA  DSEG   AT 0FFE20H
16    16 FFE20                        HD TSA: DS    1
17    17 FFE21                        STASC: DS    2
18    18
19    19 -----                        CODE   CSEG   AT 0H
20    20 00000 R0000                    MAIN:  DW    START
:

```

- 设定通过 HT 代码键入的空格，描述如下：

```
C>ra78k0r -cf1166a0 sample.asm -lt1
```

sample.prn 文件内容如下：

```

Assemble list

ALNO  STNO  ADRS   OBJECT   M I  SOURCE STATEMENT

  1     1
  2     2                                NAME   SAMPM
  3     3                                ;*****
  4     4                                ;
  5     5                                ;   HEX -> ASCII Conversion Program
  6     6                                ;
  7     7                                ;           main-routine
  8     8                                ;
  9     9                                ;*****
10    10
11    11                                PUBLIC MAIN , START
12    12                                EXTRN  CONVAH

```

| | | | | | |
|----|----|-------------|--------|---------|------------|
| 13 | 13 | | EXTRN | _@STBEG | |
| 14 | 14 | | | | |
| 15 | 15 | ----- | DATA | DSEG | AT 0FFE20H |
| 16 | 16 | FFE20 | HDTSA: | DS | 1 |
| 17 | 17 | FFE21 | STASC: | DS | 2 |
| 18 | 18 | | | | |
| 19 | 19 | ----- | CODE | CSEG | AT 0H |
| 20 | 20 | 00000 R0000 | MAIN: | DW | START |
| | | : | | | |

备注 由 HT 代码键入的空格数为 1。

-lf/-nlf

[Description format]

```
-lf  
-nlf
```

- 省略时解释
-nlf

[功能]

- lf 选项在汇编表文件的最后插入换页码。
- nlf 选项禁止 -lf 选项。

[Application]

- 如果在打印的汇编表文件内容后添加分页符，使用 -lf 选项插入换页码。

[说明]

- 如果指定 -lf 选项时，-lt 选项无效。
- 如果同时指定了 -lf 和 -nlf 选项，则最后指定的选项具有较高的优先级。

[Cautions]

- 具有与 -lf 和 -nlf 相同功能的控制代码（FORMFEED/NOFORMFEED）可以在源代码开始时声明。声明格式如下所示。

```
Δ$ΔFORMFEED  
Δ$ΔNOFORMFEED
```

[Example of use]

- 插入换页码在汇编表文件（k0rmain.prn）的末尾，描述如下：

```
C>ra78k0r -cfl166a0 k0rmain.asm -p -lf
```

错误表文件输出指定

错误表文件输出指定如下：

-e/-ne

-e/-ne

[Description format]

```
-e[output-file-name]
-ne
```

- 省略时解释
-ne

[功能]

- e 选项指定错误表文件的输出。该选项同时设定输出位置和文件名。
- ne 选项使 -e 选项失效。

[Application]

- 使用 -e 选项将错误消息保存为文件。
- 使用 -e 选项设定错误列表文件的输出位置或改变其文件名。

[说明]

- 当 -e 选项设定时输出文件名省略，输出文件名将为 “input-file-name.era”。
- 如果当 -e 选项设定时驱动器名省略，错误列表文件将输出到当前驱动器。
- 如果 -e 选项和 -ne 选项同时设定，最后设定的选项有效。

[Example of use]

- 对输出错误表文件 k0rmain.era，说明如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -ek0rmain.era
```

```
78K0R Assembler Vx.xx [xx xxx xxxx]
    Copyright(C) NEC Electronics Corporation xxxx
PASS1 Start
k0rmain.asm(31) : RA78K0 error E2202: llegal operand
PASS2 Start
k0rmain.asm(26) : RA78K0 error E2312: Operand out of range ( byte )
k0rmain.asm(31) : RA78K0 error E2202: llegal operand

Target chip : uPD78F1166_A0
Device file : Vx.xx

Assembly complete, 3 error(s) and 0 warning(s) found.
```

k0rmain.era 文件内容如下：

```
PASS1 Start
```

```
k0rmain.asm(31) : RA78K0R error E2202: illegal operand
```

```
PASS2 Start
```

```
k0rmain.asm(26) : RA78K0R error E2312: Operand out of range ( byte )
```

```
k0rmain.asm(31) : RA78K0R error E2202: illegal operand
```

参数文件说明

参数文件选项说明如下。

`-f`

`-f`

[Description format]

```
-f file-name
```

- 省略时解释
选项和输出文件名仅能从命令行输入。

[功能]

- `-f` 选项从指定文件输入选项和输入文件名。

[Application]

- 当启动汇编程序的信息不能适用于命令行时，使用 `-f` 选项
- 当每次执行汇编时重复设定选项，在参数文件中描述选项和设定 `-f` 选项。

[说明]

- 如果文件名省略，产生异常中断错误。
- 参数文件嵌套是禁止的。如果 `-f` 选项在参数文件范围内设定，则产生异常中断错误。
- 能在参数文件中描述的字符数是没有限制的。
- 使用空格键、`tab` 键或者换行码（LF）分离选项或输入文件名。
- 参数文件中的选项和输入文件名可以在命令行设定的参数文件位置扩展。
- 最后设定的扩展选项具有优先级。
- 下述字符“;”或“#”都假设为注释，相当于换行码（LF）或 EOF。
- 如果 `-f` 选项设定两次或两次以上时产生异常中断错误

[Example of use]

- 使用参数文件执行汇编。
设置参数文件（`k0rmain.pra`）的内容如下：

```
; parameter file
k0rmain.asm -osample.rel -g -cf1166a0
-psample.prn
```

从命令行输入下述内容。

```
C>ra78k0r -fk0rmain.pra
```

临时文件创建路径设定

临时文件创建路径设定选项如下。

-t

-t

[Description format]

-t *path-name*

- 省略时解释
环境变量 TMP 指定的路径
如果没有指定路径，则为当前路径。

[功能]

- t 选项设定创建临时文件的路径。

[Application]

- 使用 -t 选项设定临时文件的创建位置。

[说明]

- 仅路径可以设定作为路径名。
- 路径名不能省略。
- 即使之前创建的临时文件存在，如果文件没有保护，它将重写。
- 只要需要的存储大小可用，临时文件将在存储器中扩展。
如果没有足够的内存，临时文件的内容将写入磁盘。
这些临时文件可以在之后通过保存的磁盘文件访问。
- 当汇编结束时，临时文件删除。当通过 [CTRL] + [C] 按键退出汇编时，临时文件也删除。
- 依照下述次序确定创建临时文件的路径。

(1) 由 -t 选项设定的路径

(2) 通过环境变量 TMP 设定的路径（当 -t 选项省略时）

(3) 当前路径（当 TMP 没有设定时）

注意事项 当（1）或（2）设定时，如果临时文件不能在指定的路径创建，则发生异常中断错误。

[Example of use]

- 输出临时文件到文件夹 C:\tmp，描述如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -tC:\tmp
```

- 从 C:\Program Files\NEC Electronics Tools\include 输出临时文件，描述如下：

```
C>ra78k0r -cf1166a0 k0rmain.asm -t" C:\Program Files\NEC Electronics Tools\temporary files"
```

Kanji 指令（双字节指令）设定

Kanji 指令（双字节指令）设定选项如下。

- [-zs/-ze/-zn](#)

-zs/-ze/-zn**[Description format]**

```
-zs
-ze
-zn
```

- 省略时解释
-zs

[功能]

- 在注释中描述的 Kanji 指令（2 位字符）作为指定的 Kanji 指令（双字节指令）插入。
- Kanji 指令根据选项解释如下。
 - zs: Shift-JIS code
 - ze: EUC code
 - zn: Not interpreted as kanji

[Application]

- 使用这些选项设定注释行中 kanji 指令的解释。

[说明]

- 如果 -zs, -ze 和 -zn 选项同时设定，最后设定的选项有效。
- 具有与 -zs, -ze 和 -zn 选项相同功能的控制指令（control）可以在源代码开始时声明。声明格式如下所示。

```
Δ$ΔKANJICODEΔSJIS
Δ$ΔKANJICODEΔEUC
Δ$ΔKANJICODEΔNONE
```

- Kanji 代码也可以通过环境变量 LANF78K 设定。

[Example of use]

- 使用 EUC code 解释 kanji 代码，描述如下：

```
C>ra78k0r_k0rmain.asm -cf1166a0 -ze
```

设备文件查找路径说明

设备文件查找路径选项说明如下。

-y

-y

[Description format]

`-ypath-name`

- 省略时解释

依照下述次序确定读取设备文件的路径。

- (1) 在设备文件的安装文件中的注册路径
- (2) 该路径用于启动 **ra78k0.exe**
- (3) 当前文件夹
- (4) 环境变量 **PATH**

[功能]

-y 选项从指定路径中读取设备文件。

[Application]

- 使用 -y 选项设定设备文件存在的路径。

[说明]

- 如果不是路径名的内容在 -y 选项后设定，则会产生异常中断错误。
- 如果路径名在 -y 选项后省略，则会产生异常中断错误。
- 依照下述次序确定读取设备文件的路径。

- (1) 由 -y 选项设定的路径
- (2) 在设备文件的安装文件中的注册路径
- (3) 该路径用于启动 **RA78K0**
- (4) 当前文件夹
- (5) 环境变量 **PATH**

[Example of use]

- 设定设备文件为文件夹 C:\78k0\dev，描述如下：

```
C>ra78k0r k0rmain.asm -cf1166a0 -yC:\78k0r\dev
```

- 设定设备文件的路径为 C:\Program Files\NEC Electronics Tools\device files，描述如下：

```
C>ra78k0r k0rmain.asm -cf1166a0 -y" C:\Program Files\NEC Electronics Tools\device files"
```

符号定义设定

符号定义设定选项如下。

-d

-d

[Description format]

```
-dsymbol-name[=value][,symbol-name[=value] ... ]
```

- 省略时解释
None

[功能]

-d 选项定义符号。

[Application]

- 使用 -d 选项定义符号。

[说明]

- 给定的符号值为二进制，八进制，十进制或者十六进制。如果没有指定，默认值为“1”。
 - 使用逗号作为分隔符可以设定多至 30 个的符号。
 - 多达 31 个的字符可以用于描述符号名。
 - 如果复制名称设定，最后设定的符号有效。
 - 符号名区分字母大小写。
- 使用 -d 定义的符号取代 EQU/\$SET/\$RESET。如果 -d 设定的符号名也在源代码中定义，则发生异常中断错误。

[Example of use]

- 设定 2 作为符号定义，描述如下：

```
C>ra78k0r k0rmain.asm -cf1166a0 -dSYM=2
```

公用目标设定

公用目标设定选项如下。

- `-common`

`-common`

[Description format]

```
-common
```

- 省略时解释
指定设备的目标文件输出。

[功能]

- `-common` 选项设定目标模块文件到 78K0 的输出

[Application]

- 使用 `-common` 选项生成目标代码，该代码可以在 78K0 中共享，不需考虑设备类型设定选项（`-c`）。
输出模块文件可以和 78K0 中指定不同设备的目标文件相链接。

[说明]

- 指定生成可以在 78K0 中共用目标代码的选项。

[Cautions]

- 即使当指定 `-common` 选项时，指定设备类型的选项（`-c`）或与之由相同功能的控制指令也不可省略。
如果为需要链接的所有输入目标模块文件指定共用目标选项（`-common`），则会产生终止错误。

[Example of use]

- 生成可以在 78K0 中公用的目标代码，描述如下：

```
C>ra78k0r k0rsub.c -cf1166a0 -common
```

78K0 兼容宏

78K0 兼容宏选项如下。

- compati/-ncompati

-compati/-ncompati

[Description format]

```
-compati  
-ncompati
```

- 省略时解释
-ncompati

[功能]

- compati 选项可以汇编由 78K0 汇编程序生成的汇编程序源文件。
- ncompati 选项使 -compati 选项无效。

[Application]

- 对汇编程序源文件进行汇编，该文件包含 78K0 的指令，如果不能正常使用于 78K0R，则产生致命错误（E2337）。
设定 -compati 选项在不改变以下 78K0 指令（不用于 78K0R）的情况下，汇编汇编程序源代码。
DIVUW/ROR4/ROL4/ADJBA/ADJBS/CALLF/DBNZ

[说明]

- 当设定 -compati 选项，汇编程序包含文件 "..\inc78k0\compati.inc"(通过该路径，运行 ra78k0r.exe) 并使用 78K0 指令进行（不用作 78K0R）宏转换。

[Example of use]

- 使用 78K0 指令进行宏转换，从 78K0R 中移除，描述如下：

```
C>ra78k0r k0rsub.asm -cf1166a0 -compati
```

帮助说明

帮助选项如下。

--

--

[Description format]

--

- 省略时解释
无显示

[功能]

- 选项在显示器上输出帮助消息。

[Application]

- 帮助消息是汇编选项说明的列表。当执行汇编时参阅这些内容。

[说明]

- 当 -- 选项设定时，所有其他选项无效。
- 按回车键读取帮助消息的下一部分。
退出帮助显示，按 [Enter] 键以外的任意键，然后按返回键即可。

注意事项 该选项不能通过 **CubeSuite** 设定。

[Example of use]

- 在显示器上输出帮助消息，描述如下：

```
C>ra78k0r --
```

```

78K0R Assembler Vx.xx [xx xxx xxxx]

  Copyright(C) NEC Electronics Corporation xxxx

usage : ra78k0r [option[...]] input-file [option[...]]
The option is as follows ([ ] means omissible).

-cx          :Select target chip. (x = f1166a0 etc.) * Must be specified.
-o[file]/-no :Create the object module file [with the specified name] / Not.
-e[file]/-ne :Create the error list file [with the specified name] / Not.
-p[file]/-np :Create the print file [with the specified name] / Not.
-ka/-nka    :Output the assemble list to print file / Not.
-ks/-nks    :Output the symbol table list to print file / Not.
-kx/-nkx    :Output the cross reference list to print file / Not.
-lw[width]  :Specify print file columns per line.
-ll[length] :Specify print file lines per page.
-lf/-nlf    :Add Form Feed at end of print file / Not.
-lt[n]      :Expand TAB character for print file(n=1 to 8) / Not expand(n=0).
-lhstring   :Print list header with the specified string.
-g/-ng      :Output debug information to object file / Not.
-j/-nj      :Create object file if fatal error occurred / Not.
-idirectory[,directory ...] :Set include search path.
-tdirectory :Set temporary directory.
-ydirectory :Set device file search path.
-ffile      :Input option or source module file name from specified file.
-ga/-nga    :Output assembler source debug information to object file / Not.
-dname[=data][,name[=data][...]] :Define name [with data].
-common     :Create the common object module file for 78k0r.
-self       :Use Self-programming.
-zs/-ze/-zn :Change source regulation.
             -zs: SJIS code usable in comment.
             -ze: EUC code usable in comment.
             -zn: no multibyte code in comment.
-compati/-nocompati :Use macro for DIVUW,ROR4,ROL4,ADJBA,ADJBS,CALLF,DBNZ / Not.
--          :Show this message.
DEFAULT ASSIGNMENT :
             -o -ne -p -ka -nks -nkx -lw132 -ll0 -nlf -lt8 -g -nj -ga

```

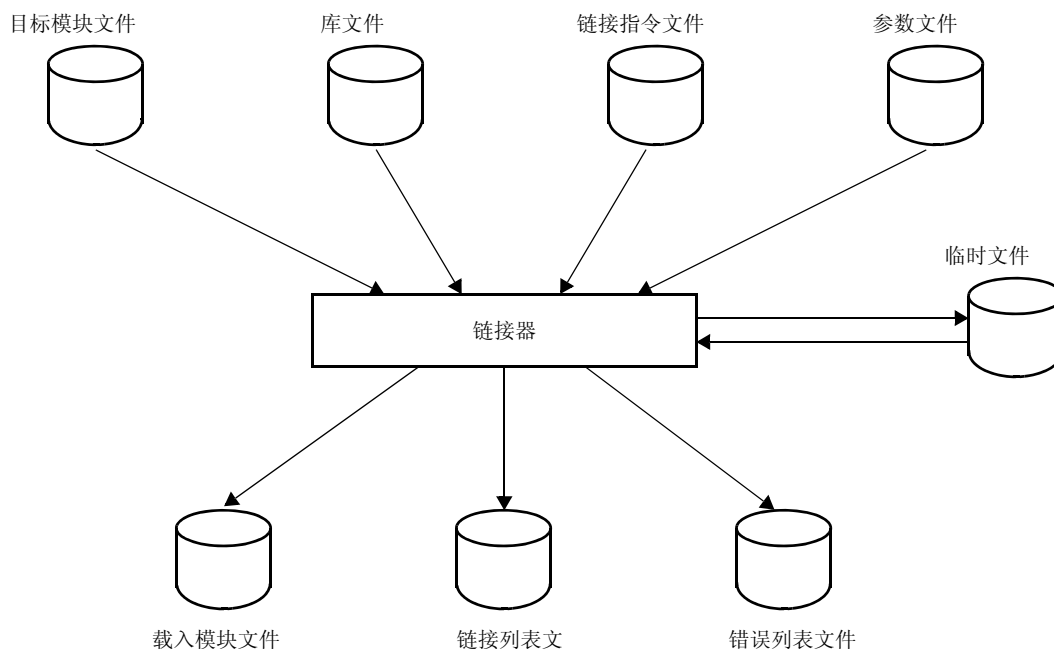
B.3 链接器

链接器输入了多个通过 78K0 汇编输出的对象模块文件，决定定位地址并且作为单个载入模块文件输出。

链接器同样输出类似链接列表文件和错误列表文件的列表文件。

如果出现链接错误，出错信息将输出到出错列表文件用以说明导致出错的原因。当错误发生时，将不会载入模块文件。

图 B-5. 链接器的 I/O 文件



B.3.1 I/O 文件

链接器 I/O 文件如下显示。

详细的输出文件见 "3.3 链接器"。

表 B-8. 链接器的 I/O 文件

| 类型 | 文件名 | 说明 | 默认文件类型 |
|--------|--------|---|----------------------------|
| 输入文件 | 目标模块文件 | - 包含与机器语言和机器语言地址定位相关的重定位信息与符号信息的二进制文件 - 通过汇编输出文件 | .rel |
| | 库文件 | - 包含两个或两个以上的对象模块文件的文件 - 通过库管理程序输出文件 | .lib |
| | 链接指令文件 | - 包含用于链接器的链接指令的文件（用户创建文件） | .dr |
| | 参数文件 | - 包含执行程序所需参数的文件（用户创建的文件） | .plk |
| 输出文件 | 载入模块文件 | - 包含链接结果创建的全部信息的二进制映像文件 此文件输入至目标转换器。 | .lmf |
| | 链接列表文件 | - 显示链接结果的列表文件 | .map |
| | 错误列表文件 | - 包含链接中产生的错误信息的文件 | .elk |
| I/O 文件 | 临时文件 | - 由链接器根据链接目的自动创建的文件。 当链接结束时删除临时文件。 | LKxxxxx.\$n (n = 1 ~ 3) |

B.3.2 功能

(1) 输入段的合并

链接器决定和控制各段的定位地址。

链接器鉴别相同的段并与其一起合并入单个段，即使它们各自处于不同的目标模块文件里。

(2) 输入模块判定

当指定用于输入的库文件时，输入对象模块文件查询的模块从库中得到回应，并且作为输入模块处理。

(3) 用于输入段的定位地址的判定。

链接器判定用于输入模块各段的定位地址。如果在源文件中指定用于段的定位属性，则段的定位由那些属性决定。链接器同样可以在链接器的链接指令文件中指定定位属性。

(4) 目标代码的修正

当定位地址埋在目标代码中时，链接器根据上述 (3) 指定的定位地址修正目标代码。

B.3.3 操作方法

(1) 启动链接器

以下两种方法可用于启动链接器。

(a) 从命令行启动

```
X:[path-name]>lk78k0R[Δoption] ... object-module-file-name[Δoption] ... [Δ]
```

| | |
|-------------------------|--|
| X | 当前驱动器名称 |
| 路径名 | 当前文件夹名 |
| lk78k0 | 链接器命令名称 |
| 选项 | 键入链接器操作的详细指令。 当设定两个或两个以上链接选项时，以空格分隔选项。连接选项不区分大小写字符。关于链接选项的详细情况，参见“B.3.4 选项”。 带有空格的路径附在一对双引号 (") 中。 |
| object-module-file-name | 链接目标模块的文件名 作为输入模块可输入多达 1024 个项目。 带空格的文件名路径附在一对双引号 (") 中。 |

示例 要添加调试信息载入模块文件 (k0r.lmf), 描述为：

```
C>lk78k0r k0rmain.rel k0rsub.rel -ok0r.lmf -g
```

(b) 从参数文件启动

当启动链接器时需要的数据无法适用于命令行时，或者当相同的链接选项重复指定，每次链接执行时，使用参数文件。

通过参数文件启动汇编程序，在命令行上设定参数文件选项 (-f)。

通过参数文件启动链接器如下：

```
X>lk78k0r[Δobject-module-file]Δ-fparameter-file-name
```

| | |
|-------|-----------------|
| -f | 参数文件说明选项 |
| 参数文件名 | 包含启动链接器所需要数据的文件 |

备注 使用编辑器创建参数文件。

参数文件内容的书写规则如下：

```
[[[Δ]Option[ΔOption] ... [Δ]Δ]] ...
```

- 如果命令行省略源文件名，参数文件中仅可以指定 1 个源文件名。
- 源文件名也可以在加在选项后。
- 在参数文件中写入所有的链接选项以及在命令行中定义的输出文件名。

示例 使用编辑器创建参数文件 kOr.plk，然后启动链接器。

```
; parameter file
k0rmain.rel k0rsub.rel -ok0r.lmf -pk0r.map -e
-tC:\tmp
```

```
C>lk78k0r -fk0r.plk
```

(2) 执行开始和结束消息

(a) 执行开始消息

当链接器启动时，执行启动消息出现在显示器上。

```
78K0R Linker Vx.xx [xx xxx xxxxx]
Copyright(C) NEC Electronics Corporation xxxxx
```

(b) 执行结束消息

如果没有检测到链接器链接错误，则链接器输出下述消息到显示器并返回控制权到主操作系统。

```
Target chip : uPD78xxx
Device file : Vx.xx

Link complete, 0 error(s) and 0 warning(s) found.
```

如果检测到链接器链接错误，则链接器输出错误编号到显示器并返回控制权到主操作系统。

```
Target chip : uPD78xxx
Device file : Vx.xx

Link complete, 1 error(s) and 0 warning(s) found.
```

如果在链接执行期间链接器检测到致命错误致使链接无法继续，则链接器输出消息到显示器，取消链接并返回控制权到主操作系统。

- 设定不存在的目标模块文件。

```
C>lk78k0r sampl.rel samp2.rel
```

```

78K0R Linker Vx.xx [xx xxx xxxx]
    Copyright(C) NEC Electronics Corporation xxxx

RA78K0R error F3006 : File not found 'saml.rel'
RA78K0R error F3006 : File not found 'samp2.rel'
程序终止。

```

在上述例子中，不存在的目标模块文件设定。出错产生并且链接器中止链接。

- 设定不存在的链接选项。

```
C>l78k0r k0rmain.rel k0rsub.rel -z
```

```

78K0R Linker Vx.xx [xx xxx xxxx]
    Copyright(C) NEC Electronics Corporation xxxx

RA78K0 error F3018 : Option is not recognized '-z'
Please enter 'LK78K0R --' , if you want help messages.
程序终止。

```

在上述例子中，不存在链接选项设定。出错产生并且链接器中止链接。

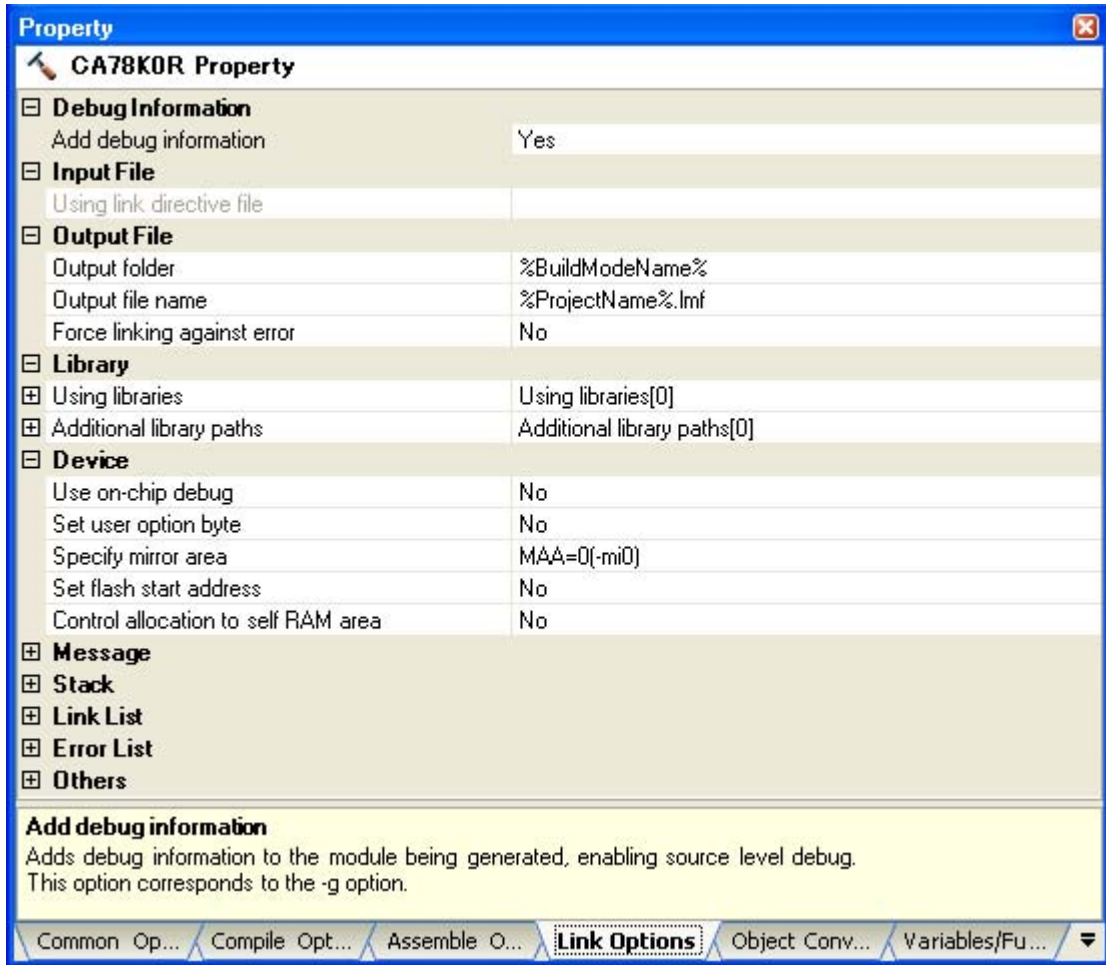
(3) 在 CubeSuite 中设置选项

本章节描述了通过 CubeSuite 怎样设置链接选项。

通过 CubeSuite 的 [工程树面板](#)，选择 Build Tool 节点。然后通过 [View] 菜单选择 [Property]。打开 [属性面板](#)。再选择 [\[Link Options\]](#) 标签。

可以通过在该列表中设定需要的属性设置不同的链接选项。

图 B-6. 属性面板: [Link Option] 标签



B.3.4 选项

(1) 类型

链接选项是链接器操作的详细指令。

链接选项的类型和注解如下。

表 B-9. 链接选项

| 分类 | 选项 | 说明 |
|--------------|-----|--------------------|
| 载入模块文件输出规格 | -o | 设定载入模块文件的输出。 |
| | -no | |
| 强制载入模块文件输出规格 | -j | 强制载入模块文件的输出。 |
| | -nj | |
| 调试信息输出说明 | -g | 指定的调试信息添加到载入模块文件中。 |
| | -ng | |
| 堆栈判定符号生成说明 | -s | 自动生成用于堆栈判定的公用符号。 |
| | -ns | |
| 链接指令文件选项 | -d | 输入指定文件作为链接指令文件。 |
| 链接列表文件输出指定 | -p | 设定链接列表文件的输出。 |
| | -np | |

| 分类 | 选项 | 说明 |
|---------------------------|--------|----------------------------|
| 链接列表文件信息指定 | -km | 输出映像列表至链接列表文件。 |
| | -nkm | |
| | -kd | 输出链接指令文件到链接列表文件。 |
| | -nkd | |
| | -kp | 输出公用符号列表至链接列表文件。 |
| | -nkp | |
| | -kl | 输出局部符号列表至链接列表文件。 |
| -nkl | | |
| 链接列表文件格式说明 | -ll | 在链接列表文件的每页打印变更的行数。 |
| | -lf | 在链接列表文件的最后插入换页码。 |
| | -nlf | |
| 错误表文件输出指定 | -e | 输出错误列表文件 |
| | -ne | |
| 库文件设定 | -b | 输入指定文件作为库文件。 |
| 库文件读取路径的设定 | -i | 从指定路径读取库文件。 |
| 参数文件说明 | -f | 通过指定文件输入选项和输入文件名。 |
| 临时文件创建路径设定 | -t | 在指定的路径建立临时文件。 |
| 设备文件查找路径说明 | -y | 从指定的路径中读取器件文件。 |
| 警告消息输出设定 | -w | 设定是否输出警告消息到控制台。 |
| 引导区域 ROM 程序链接设定在内置的闪存产品中。 | -zb | 设定 Flash 存储区域的起始地址。 |
| 片上调试设定 | -go | 指定是否使用片上调试。 |
| 安全 ID 设定 | -gi | 设定安全标识。 |
| 用户选项字节说明 | -gb | 为用户选项字节设置指定数值。 |
| 镜像空间说明 | -mi | 在镜像区域设定段的可分配目标。 |
| 64 KB 边界分配说明 | -ccza | 设定是否分配段给 64KB 边界区域的最后一个字节。 |
| | -nccza | |
| 控制自由 RAM 空间的分配 | -self | 设定是否限制自有 RAM 空间的分配。 |
| | -selfw | |
| 帮助说明 | -- | 在屏幕上输出帮助信息。 |

(2) 优先级

在下列表格中列出的链接选项，优先级通过以下方式解释，其中沿垂直轴的两个或两个以上选项以及沿水平轴的选项同时设定。

表 B-10. 编译选项的优先级

| | -no | -ng | -np | -nkm | -nkp | -nkl | -- |
|-----|-----|-----|-----|------|------|------|----|
| -j | NG | | | | | | NG |
| -g | NG | | | | | | NG |
| -p | | | | Δ | Δ | Δ | NG |
| -km | | | NG | | | | NG |
| -kd | | | NG | NG | | | NG |
| -kp | | NG | NG | | | | NG |
| -kl | | NG | NG | | | | NG |
| -ll | | | NG | | | | NG |
| -lf | | | NG | | | | NG |

- 标有 NG 的位置

如果设定选项在横坐标上，则此选项在纵坐标上无效。

示例 -km 选项无效。

```
C>lk78k0r k0rmain.rel k0rsub.rel -np -km
```

- 标有 Δ 的位置

如果同时指定行中的所有三个选项，那么列选项是无效的。

示例 如果同时设定 -nkm、-nkp 和 -nkl 选项，由 -p 选项无效。

```
C>lk78k0r k0rmain.rel k0rsub.rel -p -nkm -nkp -nkl
```

- 空白区域

如果设定水平轴上的选项，则此选项在垂直轴上无效。

指定 -o/-no 选项时，如果同时设定了两个可在选项前加“n”的选项，则最近指定的选项是有效的。

示例 在 -o 选项后设定 -no 选项时，-o 选项是无效的，而 -no 选项是有效的。

```
C>lk78k0r k0rmain.rel k0rsub.rel -o -no
```

没有在“表 B-10. 编译选项的优先级”描述的选项不受其它选项的影响。然而，如果指定了帮助选项（--），所有其它的选项都是无效的。

载入模块文件输出规格

载入模块文件输出设定选项如下。

-o/-no

-o/-no

[Description format]

```
-o[output-file-name]  
-no
```

- 省略时解释
-o.lmf

[功能]

- o 选项设定载入模块文件的输出。
这也同时指定输出文件名和路径。
- no 选项使 -g、-j 和 -g 选项失效。

[Application]

- 使用 -o 选项设定载入模块文件的输出位置或改变其文件名。
- 当执行链接时设定 -no 选项仅输出链接列表文件。该操作将缩短链接时间。

[说明]

- 当致命错误产生时，即使 -o 选项设定，载入模块文件也不能输出。
- 如果 “output-file-name” 当 -o 选项设定时省略，载入模块文件 “input-file-name.lmf” 将输出到当前文件夹。
- 如果仅在 “output-file-name” 中设定了路径名，“input-file-name.lmf” 将输出到指定路径。
- 如果同时指定了 -o 和 -no 选项，则最后指定的选项无效。

[Example of use]

- 要输出载入模块文件 (k0r.lmf)，描述为：

```
C>lk78k0r k0rmain.rel k0rsub.rel -ok0r.lmf
```

强制载入模块文件输出规格

强制载入模块文件输出设定选项如下。

- /-nj

-j/-nj

[Description format]

```
-j  
-nj
```

- 省略时解释
-nj

[功能]

- j 选项用以设定即使出现致命错误，仍可输出载入模块文件。
- nj 选项禁止 j 选项。

[Application]

- 通常当出现致命错误时，不能输出载入模块文件。
在出现致命错误提示时去执行命令，设定 j 选项来输出载入模块文件。

[说明]

- 当设定 j 选项时，即使出现致命错误，仍将输出载入模块文件。
- 如果同时指定了 j 和 nj 选项，则最后指定的选项无效。
- 如果指定 no 选项时，j 选项无效。

[Example of use]

- 即使出现致命错误时仍输出载入模块文件 (k0rsub.lmf)，描述如下：

```
C>l k78k0r k0rmain.rel k0rsub.rel -j
```

调试信息输出说明

调试信息输出选项说明如下。

-g/-ng

-g/-ng

[Description format]

```
-g  
-ng
```

- 省略时解释
-g

[功能]

- g 选项设定调试信息（局部符号列表）添加到载入模块文件。
- ng 选项使 -g、-kp 和 -kl 选项失效。

[Application]

- 当执行源代码调试中的符号调试时，确保使用 -g 选项。

[说明]

- 如果设定 -ng 选项，不能输出公用符号列表和局部符号列表。
- 如果同时设定了 -g 和 -ng 选项，则最后设定的选项无效。
- 如果指定 -no 选项时，-g 选项无效。

[Example of use]

- 要添加调试信息载入模块文件 (k0rsub.lmf), 描述为：

```
C>lk78k0r k0rmain.rel k0rsub.rel -g
```

堆栈判定符号生成说明

堆栈判定符号生成规格选项如下。

- [-s/-ns](#)

-s/-ns

[Description format]

```
-s[area-name]  
-ns
```

- 省略时解释
- ns

[功能]

- **-s** 选项生成堆栈判定公用符号 "`_@STBEG`" 和 "`_@STEND`"。
- **-ns** 选项使 **-s** 选项失效。

[Application]

- 使用 **-s** 选项保留堆栈区域。

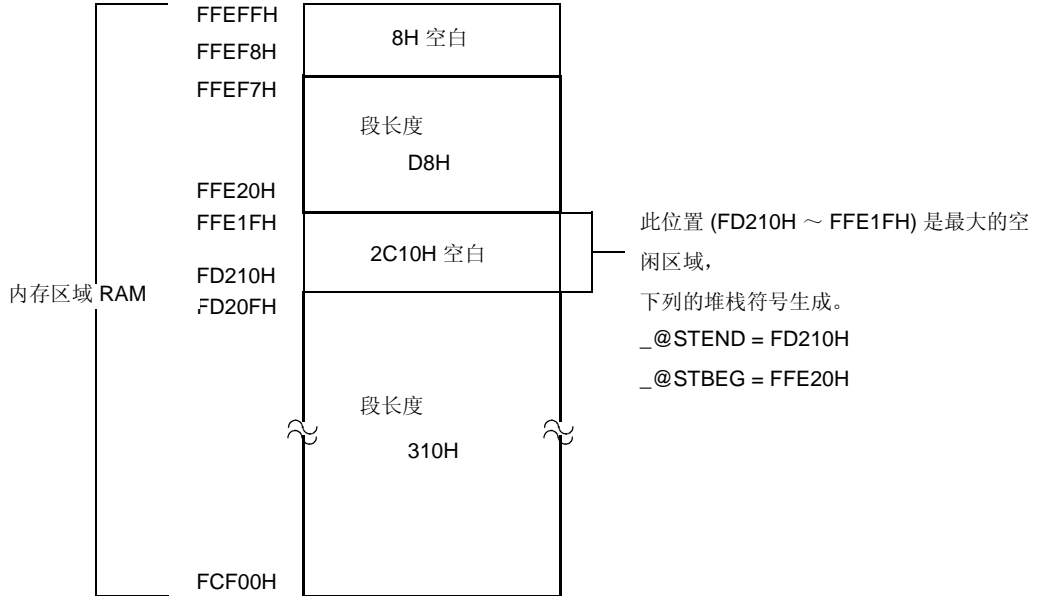
[说明]

- 由用户自定义指定存储区名称或定义为存储区默认名称默认 *area-name*。
- 使用 *area-name* 须区分大写字母和小写字母。
- **-s** 选项设定链接器查找存储区域中最大没有段分配的闲置区域。链接器随后生成公用符号 "`_@STEND`"，保持它的值为最大闲置区域的开始地址，和生成公用符号 "`_@STBEG`"，保持它的值为结束地址 +1。这些符号作为公共定义 **NUMBER** 属性符号的句柄，并注册到链接符号表的最后。当这些符号输出到链接列表文件中时，模块名称列是空白的。
- 如果最大空闲区域为 10 个字节或更小的话，警告信息输出。
- 如果没有空闲区域存在，警告消息输出且 "`_@STEND`" 和 "`_@STBEG`" 在地址末尾 +1 作为它们的值。
- 如果 *区域名称* 省略，则其指定为 "RAM"。
- 如果 **-s** 选项和 **-ns** 选项同时设定，最后设定的选项有效。

[Example of use]

- 在内存区域 RAM 中保留堆栈区域，说明如下：
然而此链接器在 RAM 区域分配段长度为 310H 及在 saddr 区域分配段长度为 D8H 的输入区域。

```
C>lk78k0r k0rmain.rel k0rsub.rel -s
```



链接指令文件选项

链接指令文件设定选项如下。

-d

-d

[Description format]

```
-dfile-name
```

- 省略时解释
None

[功能]

- 设定 -d 选项用于指定文件将作为链接指令文件输入。

[Application]

- 定义新的存储区域时，重新定义默认的存储区域时，或分配段到特定的地址或存储区域时，将会需要建立链接指令文件。使用 -d 选项输入链接指令文件到链接器。

[说明]

- 如果文件名省略，产生异常中断错误。
- 链接指令文件嵌套是禁止的。
- 在无限制链接指令文件中多个字符能标识。
- 如果 -d 选项设定两次或更多次，另如果两个或更多文件指定，则产生中止性错误。
- 关于链接指示文件的详细内容可以参阅 CubeSuite 78K0 代码用户手册。

[Example of use]

- 重定义默认内存区域 ROM/RAM。

链接指令文件 (k0r.dr) 内容如下：

```
MEMORY ROM : ( 0H , 4000H )  
MEMORY RAM : ( 0FCF00H , 3000H )
```

链接链接指令文件 (k0r.dr)，描述如下：

```
C>l k78k0r k0rmain.rel k0rsub.rel -dk0r.dr
```

链接列表文件输出指定

链接列表文件输出指定如下：

-p/-np

-p/-np

[Description format]

```
-p[output-file-name]  
-np
```

- 省略时解释
-p.map

[功能]

- p 选项设定链接列表文件的输出。该选项同时设定输出位置和文件名。
- np 选项禁止 -p、-km、-kd、-kp、-kl、-ll 和 -lf 选项。

[Application]

- 使用 -p 选项指定输出链接列表文件的路径或改变它的文件名。
- 当进行链接仅输出载入模块文件时，指定 -np 选项。该操作将缩短链接时间。

[说明]

- 如果 “*output-file-name*” 当 -p 选项设定时省略，链接列表文件 “*input-file-name.map*” 将输出到当前文件夹。
- 如果仅指定路径 “*output-file-name*”，“*input-file-name.map*” 将输出。
- 如果同时指定了 -p 和 -np 选项，则最后指定的选项无效。

[Example of use]

- 创建链接列表文件 (k0r.map)，描述如下：

```
C>lk78k0r k0rmain.rel k0rsub.rel -pk0r.map
```

链接列表文件信息指定

链接列表文件信息指定如下：

- `-km/-nkm`
- `-kd/-nkd`
- `-kp/-nkp`
- `-kl/-nkl`

-km/-nkm**[Description format]**

```
-km
-nkm
```

- 省略时解释
- km

[功能]

- `-km` 选项输出映像列表至链接列表文件。
- `-nkm` 选项使 `-kd` 和 `-km` 选项失效。

[Application]

- 使用 `-km` 选项输出映像列表至链接列表文件。

[说明]

- 如果 `-nkm`, `-nkp`, 和 `-nkl` 全部设定, 则链接列表文件不能输出。
- 如果 `-nkm` 选项设定, 则链接指令文件不能输出到链接列表文件中。
- 如果同时指定了 `-km` 和 `-nkm` 选项, 则最后指定的选项无效。
- 如果指定 `-np` 选项时, `-km` 选项无效。

[Example of use]

- 输出映像列表文件到链接列表文件中 (`k0r.map`), 描述如下:

```
C>lk78k0r k0rmain.rel k0rsub.rel -s -pk0r.map -km
```

`k0r.map` 文件内容如下:

```

78K0R Linker Vx.xx                                     Date:xx xxx xxxx Page: 1

Command:   k0rmain.rel k0rsub.rel -s -pk0r.map -km
Para-file:
Out-file:  k0rmain.lmf
Map-file:  k0r.map
Direc-file:
Directive:

*** Link information ***

    4 output segment(s)
    5FH byte(s) real data
    41 symbol(s) defined

*** Memory map ***

SPACE=REGULAR

MEMORY=ROM
BASE ADDRESS=00000H  SIZE=40000H

      OUTPUT   INPUT   INPUT   BASE   SIZE
      SEGMENT   SEGMENT   MODULE   ADDRESS
      CODE
          CODE   SAMPM   00000H   00002H   CSEG AT
* gap *
      ?CSEGOB0   000C0H   00004H   CSEG OPT_BYTE
      ?CSEG
          ?CSEG   SAMPM   000C4H   00059H   CSEG
          ?CSEG   SAMPS   000DBH   00042H
* gap *
          0011DH   3FEE3H

MEMORY=LRAM
BASE ADDRESS=FCF00H  SIZE=03100H

      OUTPUT   INPUT   INPUT   BASE   SIZE
      SEGMENT   SEGMENT   MODULE   ADDRESS
* gap *
          DATA   00003H   DSEG AT
          DATA   SAMPM   FFE20H   00003H
* gap *
          FFE23H   000DDH
* gap (Not Free Area) *
          FFF00H   00100H

```

-kd/-nkd**[Description format]**

```
-kd
-nkd
```

- 省略时解释
- kd

[功能]

- -kd 选项输出链接指令到链接列表文件中。
- -nkd 选项使 -kd 选项失效。

[Application]

- 使用 -kd 选项输出链接指令到链接列表文件中。

[说明]

- 如果 -nkm, -nkp, 和 -nkl 全部设定, 则链接列表文件不能输出。
- 如果 -nkm 选项设定, 则链接指令文件不能输出到链接列表文件中。
- 如果同时指定了 -kd 和 -nkd 选项, 则最后指定的选项有效。
- 如果指定 -np 选项时, -kd 选项无效。

[Example of use]

- 输出链接指令文件到链接列表文件中 (k0r.map), 描述如下:

```
C>l78k0r k0rmain.rel k0rsub.rel -s -dk0r.dr -pk0r.map -kd
```

k0r.map 文件内容如下:

```
78K0R Linker Vx.xx                               Date:xx xxx xxxx Page: 1

Command:    k0rmain.rel k0rsub.rel -s -dk0r.dr -pk0r.map -kd
Para-file:
Out-file:   k0rmain.lmf
Map-file:   k0r.map
Direc-file: k0r.dr                                <- Link directive file name
Directive:  MEMORY ROM : ( 0H , 0ED800H ) <- Contents of link directive file
           MEMORY RAM : ( 0FCF00H , 1100H )
           MEMORY RAM : ( 0FE000H , 1F00H )

*** Link information ***

        6 output segment(s)
       9DH byte(s) real data
       40 symbol(s) defined
```

```
*** Memory map ***
```

```
SPACE=REGULAR
```

```
MEMORY=ROM
```

```
BASE ADDRESS=00000H  SIZE=ED800H
```

| OUTPUT | INPUT | INPUT | BASE | SIZE |
|---------|---------|--------|---------|----------------|
| SEGMENT | SEGMENT | MODULE | ADDRESS | |
| CODE | | | 00000H | 00002H CSEG AT |
| : | | | | |

-kp/-nkp**[Description format]**

```
-kp
-nkp
```

- 省略时解释
- nkp

[功能]

- -kp 选项输出公用符号列表到链接列表文件。
- -nkp 选项使 -kp 选项失效。

[Application]

- -kp 选项输出公用符号列表到链接列表文件。

[说明]

- 如果 -nkm, -nkp, 和 -nkl 全部设定, 则链接列表文件不能输出。
- 如果设定 -ng 选项, 则公用符号列表不能输出。
- 如果同时指定了 -kp 和 -nkp 选项, 则最后指定的选项有效。
- 如果 -np 选项设定时, 则 -kp 选项无效。

[Example of use]

- 输出公用符号列表文件到链接列表文件中 (k0r.map), 描述如下:

```
C>lk78k0r k0rmain.rel k0rsub.rel -s -g -pk0r.map -kp
```

k0r.map 文件内容如下:

```
78K0R Linker Vx.xx                               Date:xx xxx xxxx Page: 1

Command:    k0rmain.rel k0rsub.rel -s -g -pk0r.map -kp
Para-file:
Out-file:   k0rmain.lmf
Map-file:   k0r.map
Direc-file:
Directive:

*** Link information ***

    6 output segment(s)
    9DH byte(s) real data
    40 symbol(s) defined

*** Memory map ***
```

```
SPACE=REGULAR
```

```
MEMORY=ROM
```

```
BASE ADDRESS= 00000H  SIZE=40000H
```

```
:
```

```
-----  
78K0R Linker Vx.xx
```

```
Date:xx xxx xxxx Page: 2
```

```
*** Public symbol list ***
```

```
MODULE  ATTR  VALUE NAME
```

```
SAMPM
```

```
ADDR  00000H  MAIN
```

```
ADDR  000D2H  START
```

```
SAMPS
```

```
ADDR  000E9H  CONVAH
```

```
NUM   FFE20H  _@STBEG
```

```
NUM   FCF00H  _@STEND
```

```
Target chip : uPD78xxx
```

```
Device file : Vx.xx
```

-kl/-nkl**[Description format]**

```
-kl
-nkl
```

- 省略时解释
- nkl

[功能]

- -kl 选项输出局部符号列表到链接列表文件。
- -nkl 选项使 -kl 选项失效。

[Application]

- -kl 选项输出局部符号列表到链接列表文件。

[说明]

- 如果 -nkm, -nkp, 和 -nkl 全部设定, 则链接列表文件不能输出。
- 如果设定 -ng 选项, 则局部符号列表不能输出。
- 如果同时指定了 -kl 和 -nkl 选项, 则最后指定的选项有效。
- 如果指定 -np 选项时, -kl 选项无效。

[Example of use]

- 输出局部符号列表文件到链接列表文件中 (k0r.map), 描述如下:

```
C>lk78k0r k0rmain.rel k0rsub.rel -s -g -pk0r.map -kl
```

k0r.map 文件内容如下:

```
78K0R Linker Vx.xx                               Date:xx xxx xxxx Page: 1

Command:    k0rmain.rel k0rsub.rel -s -g -pk0r.map -kl
Para-file:
Out-file:   k0rmain.lmf
Map-file:   k0r.map
Direc-file:
Directive:

*** Link information ***

    6 output segment(s)
    9DH byte(s) real data
    40 symbol(s) defined

*** Memory map ***
```

SPACE=REGULAR

:

78K0R Linker Vx.xx

Date:xx xxx xxxx Page: 2

*** Local symbol list ***

| MODULE | ATTR | VALUE | NAME |
|--------|------|--------|-------|
| SAMPM | | | |
| | MOD | | SAMPM |
| | DSEG | | DATA |
| | ADDR | FFE20H | HDTSA |
| | ADDR | FFE21H | STASC |
| | CSEG | | CODE |
| | CSEG | | ?CSEG |
| SAMPS | | | |
| | MOD | | SAMPS |
| | CSEG | | ?CSEG |
| | ADDR | 0015CH | SASC |
| | ADDR | 00162H | SASC1 |

Target chip : uPD78xxx

Device file : Vx.xx

链接列表文件格式说明

链接列表文件格式设定选项说明如下：

- `-ll`
- `-lf/-nlf`

-ll

[Description format]

```
-ll[number-of-lines]
```

- 省略时解释
- ll0 (No page breaks)

[Function]

- ll 选项设定列表文件中每页行数。

[Application]

- 使用 ll 选项改变链接列表文件中每页行数。

[说明]

- ll 选项可指定的行数范围为 20~32767。
- 如果数值超过此范围或指定的不是数字，会产生终止错误。
- 如果没有指定，默认值为 0。
- 如果行数指定为 0，将不会生成分页符。
- 如果指定 -np 选项时，ll 选项无效。

[Example of use]

- 设定链接列表文件每页行数为 20(k0rmap)，描述如下：

```
C>l k78k0r k0rmain.rel k0rsub.rel -s -pk0r.map -ll20
```

k0r.map 文件内容如下：

```
78K0R Linker Vx.xx                               Date:xx xxx xxxx Page: 1

Command:    k0rmain.rel k0rsub.rel -s -pk0r.map -km -ll20
Para-file:
Out-file:   k0rmain.lmf
Map-file:   k0r.map
Direc-file:
Directive:

*** Link information ***
```

```

4 output segment(s)
5FH byte(s) real data
41 symbol(s) defined

```

78K0R Linker Vx.xx

Date:xx xxx xxxx Page: 2

*** Memory map ***

SPACE=REGULAR

MEMORY=ROM

BASE ADDRESS=00000H SIZE=40000H

| OUTPUT | INPUT | INPUT | BASE | SIZE | |
|----------|---------|--------|---------|--------|---------------|
| SEGMENT | SEGMENT | MODULE | ADDRESS | | |
| CODE | | | 00000H | 00002H | CSEG AT |
| | CODE | SAMPM | 00000H | 00002H | |
| * gap * | | | 00002H | 000BEH | |
| ?CSEGOB0 | | | 000C0H | 00004H | CSEG OPT_BYTE |
| ?CSEG | | | 000C4H | 00059H | CSEG |
| | ?CSEG | SAMPM | 000C4H | 00017H | |

78K0R Linker Vx.xx

Date:xx xxx xxxx Page: 3

| | | | | | |
|---------|-------|-------|--------|--------|--|
| | ?CSEG | SAMPS | 000DBH | 00042H | |
| * gap * | | | 0011DH | 3FEE3H | |

MEMORY=RAM

BASE ADDRESS=FCF00H SIZE=03100H

| OUTPUT | INPUT | INPUT | BASE | SIZE | |
|-------------------------|---------|--------|---------|--------|---------|
| SEGMENT | SEGMENT | MODULE | ADDRESS | | |
| * gap * | | | FCF00H | 02F20H | |
| DATA | | | FFE20H | 00003H | DSEG AT |
| | DATA | SAMPM | FFE20H | 00003H | |
| * gap * | | | FFE23H | 000DDH | |
| * gap (Not Free Area) * | | | FFF00H | 00100H | |

Target chip : uPD78xxx

Device file : Vx.xx

-lf/-nlf

[Description format]

```
-lf  
-nlf
```

- 省略时解释
- nlf

[功能]

- lf 选项在链接列表文件的最后插入换页码 (FF)。
- nlf 选项禁止 -lf 选项。

[Application]

- 如果在打印的链接列表文件内容后添加分页符，使用 -lf 选项插入换页码。

[说明]

- 如果指定 -lf 选项时，-lt 选项无效。
- 如果同时指定了 -lf 和 -nlf 选项，则最后指定的选项有效。

[Example of use]

- 插入换页码在链接列表文件（k0rmap）的末尾，描述如下：

```
C>lk78k0r k0rmain.rel k0rsub.rel -pk0r.map -lf
```

错误表文件输出指定

错误表文件输出指定如下：

-e/-ne

-e/-ne

[Description format]

```
-e[file-name]  
-ne
```

- 省略时解释
-ne

[功能]

- e 选项指定错误表文件的输出。该选项同时设定输出位置和文件名。
- ne 选项使 -e 选项失效。

[Application]

- 使用 -e 选项设定错误列表文件的输出位置或改变其文件名。

[说明]

- 当 -e 选项设定时输出文件名省略，输出文件名将为 “*input-file-name.eoc*”。
- 如果当 -e 选项设定时驱动器名省略，错误列表文件将输出到当前驱动器。
- 如果 -e 选项和 -ne 选项同时设定，最后设定的选项有效。

[Example of use]

- 创建错误列表文件 k0r.elk，描述如下：

```
C>lk78k0r k0rmain.rel k0rsub.rel -dk0r.dr -ek0r.elk
```

在链接指令文件的内容中产生错误 (k0r.dr).
错误列表文件 (k0r.dr) 的内容如下：

```
k0r.dr(3) : RA78K0R error E3102: Directive syntax error
```

库文件设定

库文件设定选项说明如下。

- **b**

-b

[Description format]

```
-bfile-name
```

- 省略时解释
None

[功能]

- 设定 **-b** 选项用于指定文件将作为库文件输入。

[Application]

- 链接器通过从库文件中的输入模块以及数据模块的链接模块中恢复模块引用。
- 库文件的作用是在单一的文件中注册两个或更多的模块。
- 通过建立能在许多程序中通用的库文件，则文件管理和操作变得更容易和更有效。使用 **-b** 选项输入库文件到链接器。

[说明]

- 文件名不能省略。
- 如果文件名包含指定的路径名称，则库文件将从那个路径输入。如果在指定路径上不存在库文件，则错误产生。
- 如果文件名称不包含指定的路径名称，则库文件将通过 **-i** 选项从指定的默认查找路径中来输入。
- 如果两个或以上 **-b** 选项设定，库文件将在指定的队列中输入。多达 **64-b** 选项可以设定。
- 从“[B.5 库管理程序](#)”参阅关于建立库文件方法的更详细说明。

[Example of use]

- 输入库文件 (**k0r.lib**)，说明如下：
k0rsub.rel 注册到库文件中。

```
C>lk78k0r k0rmain.rel -bk0r.lib
```

库文件读取路径的设定

库文件读取路径设定选项如下：

`-i`

`-i`

[Description format]

`-i path-name [, path-name] ...` (可以设定两个或以上的路径)

- 省略时解释
- 通过环境变量设定路径 (LIB78K0)
- 如果环境变量 (LIB78K0) 没有设定，则为当前路径。

[功能]

- `-i` 选项用于设定库文件从指定路径输入。

[Application]

- 使用 `-i` 选项从指定的路径上查找库文件。

[说明]

- 当库文件名称通过 `-b` 选项 (不包含路径名称) 来设定，则只有 `-i` 选项有效。
- 两个或以上的 `-i` 选项能设定。两个或多个路径名可用通过分隔符 “,” 指定。在 “,” 前后不能有空格。
- 可以设定每个链接多达 64 个路径名称。如果两个或以上的路径名称设定，库文件将在指定的队列中查找。
- 即使在指定路径上不存在库文件，错误也不会产生。
- 如果路径名称省略，产生中止性错误。
- 如果库文件通过 `-b` 选项设定不包含路径名称，此链接器将在下列队列中查找路径。

(1) 由 `-i` 选项设定路径

(2) 通过环境变量设定路径 (LIB78K0)

(3) 当前路径

注意事项 如果指定名称的库文件在任何这些路径中不存在，则会产生错误。

[Example of use]

- 按 C:\LIB1 和 C:\LIB2 这样的顺序查找并读取库文件，说明如下：

```
C>lk78k0r k0rmain.rel k0rsub.rel -bk0r.lib -iC:\lib1,C:\lib2
```

- 从 C:\Program Files\NEC Electronics Tools\library files 文件夹中读取库文件，说明如下：

```
C>lk78k0r k0rmain.rel k0rsub.rel -bk0r.lib -i" C:\Program Files\NEC Electronics  
Tools\library files"
```

参数文件说明

参数文件选项说明如下。

-f

-f

[Description format]

`-f file-name`

- 省略时解释
选项和输出文件名仅能从命令行输入。

[功能]

- f 选项从指定文件输入选项和输入文件名。

[Application]

- 当启动链接器所必须的信息不能出现在命令行上时，使用 -f 选项。
- 当每次执行链接时重复设定选项时，则在参数文件中描述选项并设定为 -f 选项。

[说明]

- 如果文件名省略，产生异常中断错误。
- 参数文件嵌套是禁止的。如果 -f 选项在参数文件范围内设定，则产生异常中断错误。
- 能在参数文件中描述的字符数是没有限制的。
- 使用空格键、**tab** 键或者换行码（**LF**）分离选项或输入文件名。
- 参数文件中的选项和输入文件名可以在命令行设定的参数文件位置扩展。
- 最后设定的扩展选项有效。
- 下述字符“;”或“#”都假设为注释，相当于换行码（**LF**）或 **EOF**。
- 如果两个或两个以上 -f 选项设定时产生异常中断错误。

[Example of use]

- 使用参数文件 (k0r.plk) 执行链接。
参数文件 (k0r.plk) 的内容如下。

```
; parameter file
k0rmain.rel k0rsub.rel -ok0r.lmf -pk0r.map -e
-tC:\tmp -g
```

从命令行输入下述内容。

```
C>l1k78k0r -fk0r.plk
```

临时文件创建路径设定

临时文件创建路径设定选项如下。

-t

-t

[Description format]

-t *path-name*

- 省略时解释
环境变量 TMP 指定的路径
如果没有指定路径，则为当前路径。

[功能]

- t 选项设定创建临时文件的路径。

[Application]

- 使用 -t 选项设定临时文件的创建位置。

[说明]

- 仅路径可以设定作为路径名。
- 路径名不能省略。
- 即使之前创建的临时文件存在，如果文件没有保护，它将重写。
- 只要需要的存储大小可用，临时文件将在存储器中扩展。
如果没有足够的内存，临时文件的内容将写入磁盘。
这些临时文件可以在之后通过保存的磁盘文件访问。
- 当链接结束时，临时文件删除。当通过 [CTRL] + [C] 按键退出链接时，它们也会删除。
- 依照下述次序确定创建临时文件的路径。

(1) 由 -t 选项设定的路径

(2) 通过环境变量 TMP 设定的路径（当 -t 选项省略时）

(3) 当前路径（当 TMP 没有设定时）

注意事项 当（1）或（2）设定时，如果临时文件不能在指定的路径创建，则发生异常中断错误。

[Example of use]

- 输出临时文件到文件夹 C:\tmp，描述如下：

```
C>l k78k0r k0rmain.rel k0rsub.rel -tC:\tmp
```

- 从 C:\Program Files\NEC Electronics Tools\include 输出临时文件，描述如下：

```
C>lk78k0r k0rmain.rel k0rsub.rel -t" C:\Program Files\NEC Electronics Tools\temporary files"
```

设备文件查找路径说明

设备文件查找路径选项说明如下。

-y

-y

[Description format]

`-ypath-name`

- 省略时解释

依照下述次序确定读取设备文件的路径。

- (1) 在设备文件的安装文件中的注册路径
- (2) 路径用于启动 lk78k0.exe
- (3) 当前文件夹
- (4) 环境变量 PATH

[功能]

-y 选项从指定路径中读取设备文件。

[Application]

- 使用 -y 选项设定设备文件存在的路径。

[说明]

- 如果不是路径名的内容在 -y 选项后设定，则会产生异常中断错误。
- 如果路径名在 -y 选项后省略，则会产生异常中断错误。
- 依照下述次序确定读取设备文件的路径。

- (1) 由 -y 选项设定的路径
- (2) 在设备文件的安装文件中的注册路径
- (3) 路径用于启动 LK78K0
- (4) 当前文件夹
- (5) 环境变量 PATH

[Example of use]

- 设定设备文件为文件夹 C:\78k0\dev，描述如下：

```
C>lk78k0r k0rmain.rel k0rsub.rel -yC:\78k0r\dev
```

- 设定设备文件的路径为 C:\Program Files\NEC Electronics Tools\device files，描述如下：

```
C>lk78k0r k0rmain.rel k0rsub.rel -y" C:\Program Files\NEC Electronics Tools\device files"
```

警告消息输出设定

警告信息输出设定选项如下。

- [-w](#)

-w

[Description format]

```
-w[level]
```

- 省略时解释
-w1

[功能]

- -w 选项设定是否输出警告消息到控制台。

[Application]

- 使用 -w 选项设定层，警告信息就输出到那。

[说明]

- 如果在 -w 选项之后除层以外的东西设定，则会产生中止性错误。
- 只有层 0, 1 和 2 可以设定。
- 输出层的说明如下。
 - 0: 没有警告信息输出。
 - 1: 一般的警告信息输出。
 - 2: 详细的警告信息输出。

[Example of use]

- 输出详细的警告信息，说明如下：

```
C>l k78k0r k0rmain.rel k0rsub.rel -w2
```

引导区域 ROM 程序链接设定在内置的闪存产品中。

引导区域 ROM 程序链接设定在内置的闪存产品中，说明如下。

- `-zb`

`-zb`

[Description format]

`-zbaddress`

- 省略时解释
分配范围无限制

[功能]

- 设定闪存存储区域的起始地址用 `-zb` 选项。

[说明]

- 引导区域 ROM 程序链接设定在内置的闪存产品中，且设定闪存存储区域的起始地址。
- 可以用于填充值的范围是 `0H ~ 0FFFFH`。
- 如果此地址省略，则会产生错误。
- 不能分配代码在比设定地址更高的地址上。

注意事项 不要为不具有闪存存储区域自编程函数的设备设定该选项。

[Example of use]

- 设定闪存存储区域的起始地址为 `2000H`，说明如下：

```
C>lk78k0r k0rmain.rel -zb2000h
```

片上调试设定

片上调试设定选项如下。

- `-go`

-go

[Description format]

```
-gocontrol-value, start-address[, size]
```

- 省略时解释
片上调试不能使用。
地址 `C3H` 是指定设备文件的默认值。

[Function]

- `-go` 选项设定片上调试是否可用。

[Application]

- 使用 `-go` 选项改变 片上调试操作的控制值，调试监视区域的起始地址和大小。

[说明]

- 为了控制值，指定片上调试操作控制值。
如果指定的控制值没有设定数值，则产生终止错误。
有关控制值的详细说明，参见 `QB-MINI2` 片上调试仿真器可编程功能用户手册 (U18371EJ)。
- 为起始地址，设定调试监视区的可分配起始地址。
可以用于设定起始地址的范围是 `0 to 0FFFFFFH`。
如果省略了起始地址，假设已经设定 (内部 ROM 结束地址 - `1024`) + 1。
有关起始地址的详细说明，参见 `QB-MINI2` 片上调试仿真器可编程功能用户手册 (U18371EJ)。
- 为大小，设定调试监视区域的大小。
可以用于设定的大小范围为 `88` 到 `1024`。
如果没有指定大小，默认值为 “`1024`”。
有关程序大小的详细说明，参见 `QB-MINI2` 片上调试仿真器可编程功能用户手册 (U18371EJ)。
- 如果除数值以外的东西用来设置控制值，起始地址或大小，则产生终止错误。
- 如果设定 `-go` 选项，则将分配控制值在地址 `C3H` 上。
段不能分配在地址 `2H`, `3H`, 和 `CEH` 到 `D7H`, 使用 `-go` 选项也不能在地址的起始位置上指定区域的大小。
通过设定 `-gb` 选项，保留地址 `C0` 到 `C2H` 作为用户选项设置区域。
- 即使没有设定 `-go` 选项，用户代码也不能分配在地址 `C3H` 上，因为该地址为保留地址。
- 在汇编程序源文件中，通过以下显示的重定义属性定义的段，也可以设定分配在地址 `C3H` 上的控制值。然而，总共使用 4 字节定义段，包含从地址 `C0H` 开始的用户选项字节。

| [Any segment name] | CSEG | OPT_BYTE |
|--------------------|------|---------------------|
| | DB | 0FDH ; Address 0xC0 |
| | DB | 0FEH ; Address 0xC1 |
| | DB | 0FFH ; Address 0xC2 |
| | DB | 04H ; Address 0xC3 |

如果指定的汇编程序源文件及指定的该选项复制，则该选项优先。

- 有关设备、设置片上调试选项字节以及用户选项字节，请参阅用户手册。

[Example of use]

- 嵌入 04H 在地址 C3H 上作为控制值。
- 从起始地址（地址 02FC00H）开始的保留区域，最高达 1024 字节用作调试监视区域。

```
C>lk78k0r k0rmain.rel -go04H,02FC00H,1024
```

安全 ID 设定

安全 ID 设定选项如下。

-gi

-gi

[Description format]

```
-gisecurity-ID
```

- 省略时解释
安全 ID 没设置。

[功能]

-gi 选项来设定安全 ID。

[Application]

- 使用 -gi 选项来设置安全 ID。

[说明]

- 指定结尾带 "H" 的十六进制数值。如果另外的数值省略，则会产生中止性错误。
- 设定安全 ID 在 10 个字节以内。如果指定值小于 10 个字节，则高比特位填充为 0。
- 安全 ID 设置在地址 C4H 到 CDH。如果安全 ID 设置，没有段可以分配在地址 C4H 到 CDH。
- 如果该选项为不具片上调试功能的设备设定，则会产生错误。
- 在汇编源文件中通过定义重定位属性显示在下面的段安全 ID 也能设定。然而确定设定 SECUR_ID 作为段的重定位属性。

| [Any segment name] | CSEG | SECUR_ID | |
|--------------------|------|----------|----------------|
| | DB | 11H | ; Address 0xC4 |
| | DB | 22H | ; Address 0xC5 |
| | DB | 33H | ; Address 0xC6 |
| | DB | 44H | ; Address 0xC7 |
| | DB | 55H | ; Address 0xC8 |
| | DB | 66H | ; Address 0xC9 |
| | DB | 77H | ; Address 0xCA |
| | DB | 88H | ; Address 0xCB |
| | DB | 99H | ; Address 0xCC |
| | DB | 0AAH | ; Address 0xCD |

如果指定的汇编程序源文件及指定的该选项复制，则该选项优先。

[Cautions]

- 如果该选项没有为具有安全 ID 功能的设备而设定，则任何代码可以分配。

[Example of use]

- 设定相同的 "112233445566778899aah" 作为在上述的汇编程序源文件中的设置，说明如下：

```
C>l k78k0r k0rmain.rel -gil12233445566778899aah
```

用户选项字节说明

用户选项字节说明选项如下：

- `-gb`

-gb**[Description format]**

```
-gbuser-option-byte-value
```

- 省略时解释
设定设备文件的默认值

[功能]

- `-gb` 选项为用户选项字节设置指定数值。

[Application]

- 使用 `-gb` 选项设置用户选项字节值。

[说明]

- 可以用于设定用户选项字节的范围是 `0 to 0FFFFFFH`。
- 如果指定的用户选项字节没有设定值，则产生终止错误。
- 指定结尾带 "H" 的十六进制数值。如果另外的数值省略，则会产生中止性错误。
- 在地址 `C0H` 到 `C2H` 上设定用户选项字节。
- 如果没有设定 `-gb` 选项，则用户代码不可以分配在 `C0` 到 `C2H` 上，因为这些地址为保留地址。
- 设定安全 ID 在 3 个字节以内。如果指定值小于 3 个字节，则高比特位填充为 0。
- 在汇编程序源文件中，通过以下显示的重定义属性定义的段，也可以设定分配在地址 `C0H` 到 `C2H` 上的用户选项字节值。然而，用总共 4 个字节定义段，包含在地址 `C3H` 上的控制值。

| [Any segment name] | CSEG | OPT_BYTE | |
|--------------------|------|----------|----------------|
| | DB | 0FDH | ; Address 0xC0 |
| | DB | 0FEH | ; Address 0xC1 |
| | DB | 0FFH | ; Address 0xC2 |
| | DB | 04H | ; Address 0xC3 |

如果指定的汇编程序源文件及指定的该选项复制，则该选项优先。

- 有关设备、设置片上调试选项字节以及用户选项字节，请参阅用户手册。

[Example of use]

- 在地址 `C0H` 上设定 `FDH`，地址 `C1H` 上设定 `FEH`，地址 `C2H` 上设定 `FFH`，作为用户选项字节值，描述如下：

```
C>lk78k0r k0rmain.rel -gb0FDFEFFH
```

镜像空间说明

镜像空间选项说明如下。

-mi

-mi

[Description format]

```
-mi0 or -mi1
```

- 省略时解释
-mi0

[功能]

-m 选项，在镜像区域设定段的可分配目标。

[Application]

- 使用 -m 选项，在镜像区域设定段的可分配目标。

[说明]

- 通过连接器设定的 CSEG MIRRORP 重定向属性，分配段的目标。
- 如果设定了 -mi0，当 MAA=0 时，则段分配在镜像空间中。如果设定了 -mi1，当 MAA=1，则段分配在镜像空间中。
- 有关镜像空间的详细内容，参阅设备的用户手册。
- 将产生公共符号 "_@MAA"。这是 NUMBER 属性符号，当设定 -mi0，则保持为 "0"，且在设定 -mi1 时，则保持为 "1"。

[Example of use]

- 当 MAA=1 时，在镜像空间里分配段，描述如下：

```
C>lk78k0r k0rmain.rel -mi1
```

在 F1000H 或更高地址保留为设备镜像区域，在地址 11000H 和更高地址上分配带 CSEG MIRRORP 的段。使用由编译器提供的启动程序的例子如下。

```
MOVW    PMC , #_@MAA
```

在这种情况下，在 PMC 中存储 "1"。

64 KB 边界分配说明

64KB 边界分配说明选项如下。

- `-ccza/-nccza`

-ccza/-nccza**[Description format]**

```
-ccza  
-nccza
```

- 省略时解释
- `-ccza` (只有在输入文件为汇编输出文件时)
- `-nccza` (当在输入文件中包含编译程序的输出文件时)

[功能]

- `-ccza` 选项设定是否分配段给 64KB 边界区域的最后一个字节。(xFFFFH 注意)

注 x: 0H 到 EH

[Application]

- 使用 `-ccza` 选项设定是否分配段给 64KB 边界区域的最后一个字节。

[说明]

- 如果只使用汇编程序进行开发，没有必要设定此选项，因为自动分配段到每个 64KB 边界区域的最后一个字节。
- 如果从编译器中输出的目标模块文件输入到链接器中，假设链接器自动设置 `-nccza` 选项，则段不会分配到 64KB 边界区域的最后一个字节上。
- 如果在编译器中设定 `-za` 选项，允许分配段在 64KB 边界区域的最后一个字节，同样设定 `-ccza` 选项。
- 有关分配在 64KB 边界区域的最后字节的段的详细内容，参阅 CubeSuite 编码用户手册。

控制自由 RAM 空间的分配

自由 RAM 空间分配的控制选项如下。

- `-self/-selfw`

-self/-selfw

[Description format]

```
-self  
-selfw
```

- 省略时解释
自有 RAM 空间当作内部 RAM.

[功能]

- `-self` 和 `-selfw` 选项设定是否限制自有 RAM 空间的分配。

[Application]

- 使用 `-self` 和 `-selfw` 选项设定是否限制自有 RAM 空间的分配。

[说明]

- 如果设定了 `-self`，在自有 RAM 空间禁止分配段，否则产生错误。
- 如果设定了 `-selfw`，在自有 RAM 空间可以分配段，但会输出警告信息。
- 如果设定了 `-self` 或 `-selfw` 选项，在 `saddr` 区域以外分配堆栈区域。

帮助说明

帮助选项如下。

--

--

[Description format]

--

- 省略时解释
无显示

[功能]

- 选项在显示器上输出帮助消息。

[Application]

- 帮助消息是链接选项说明的列表。当执行链接时参阅这些。

[说明]

- 当 -- 选项设定时，所有其他选项无效。
- 按回车键读取帮助消息的下一部分。退出帮助显示，按除回车键外的任意键，然后按回车键。

注意事项 该选项不能通过 **CubeSuite** 设定。

[Example of use]

- 在显示器上输出帮助消息，描述如下：

```
C>l k78k0r --
```

```

78K0R Linker Vx.xx [xx xxx xx]

Copyright(C) NEC Electronics Corporation xxxx

usage : lk78K0r [option[...]] input-file [option[...]]
The option is as follows ([ ] means omissible).

-ffile           :Input option or input-file name from specified file.
-dfile           :Read directive file from specified file.
-bfile           :Read library file from specified file.
-idirectory[,directory...] :Set library file search path.
-o[file]/-no     :Create load module file [with specified name] / Not.
-p[file]/-np     :Create link map file [with specified name] / Not.
-e[file]/-ne     :Create error list file [with specified name] / Not.
-tdirectory     :Set temporary directory.
-km/-nkm        :Output map list to link map file / Not.
-kd/-nkD        :Output directive file image to link map file / Not.
-kp/-nkp        :Output public symbol list to link map file / Not.
-kl/-nkl        :Output local symbol list to link map file / Not.
-ll[page length] :Specify link map file lines per page.
-lf/-nlf        :Add Form Feed at end of the link map file / Not.
-s[memory area]/-ns :Create stack symbol [in specified memory area] / Not.
-g/-ng          :Output symbol information to load module file / Not.
-ydirectory     :Set device file search path.
-j/-nj          :Create load module file if fatal error occurred / Not.
-w[n]           :Change warning level(n=0 to 2).
-zbaddress      :Create Boot file (address:flash start address).
-godata,address[,size] :更改片上调试选项字节, 起始地址, 大小(大小=88到1024)。
-giid           :Set Security ID.
-gbdata         :Set User Option Bytes.
-mi[0 or 1]     :Select allocation for MIRRORP segment.
-self[-selfw]   :Allocate user code to BRCROSS / Not.
-ccza/-nccza    :Allocate user code to nFFFFH / Not.
--              :Show this message.

DEFAULT ASSIGNMENT: -o -p -ne -km -kd -nkp -nkl -ll0 -nlf -ns -g -nj -wl

directive file usage:
MEMORY memory-area-name:(origin-value, size)[/memory-space-name]
MERGE segment-name:[location-type-definition][merge-type-definition]
      [=memory-area-name][/memory-space-name]

example: MEMORY ROM:(0H,4000H)
          MEMORY RAMA:(0FEF00H,100H)
          MERGE CSEG1:=OM
          MERGE DSEG1:AT(0FF000H)

```

B.3.5 Boot-flash relink 功能

(1) 重链接功能

一些系统装备了闪存区或可分离 ROM。

升级程序的版本，闪存区的内容可以重写或可分离 ROM 可以替换为新的 ROM。

即使部分改变此程序，从根本上工程本身重组或 "rebuilt"。然而，如果用于升级的可分配区域限定在闪存区域或外部 ROM 中且如果重组工程不是必须的，那就会变得便利。引导区安置在内部 ROM 中。如果功能在可重写的闪存区域和引导区调用，且如果由于在闪存区域修改此功能的结果，此功能的起始地址改变，此功能不能正确调用。

"boot-flash relink function"(后指 "relink function") 用于防止其且使功能正确调用。

对此功能做如下说明：

(a) "branch table" 在可写的闪存区域的中分支此功能的指令已经在闪存中准备完毕。

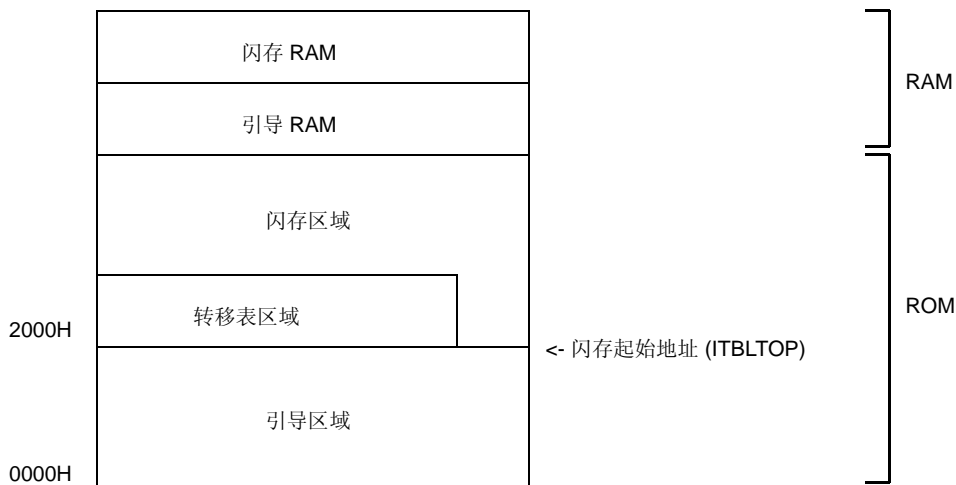
(b) 当在闪存区域中的功能从引导区域调用，在此闪存区域中执行跳跃到这个分支表，并且然后这个用于分支预订功能的指令执行且产生跳转。

用户所能认识的机制。如果 "relink function" 使用，这能相对容易的完成。

使用此功能，然而在引导区域建立时，此功能在闪存区域调用必须确定。此机制用于从引导区域调用功能即使此功能在闪存区域修改。

在复位期间的操作说明如下：

- RESET 中断向量 (引导区域)
- > _@cstart (引导区域)
- > _boot_main 功能 (引导区域)
- > ITBLTOP 地址 (闪存区域)
- > _@cstart (闪存区域)
- > _main 功能 (闪存区域)



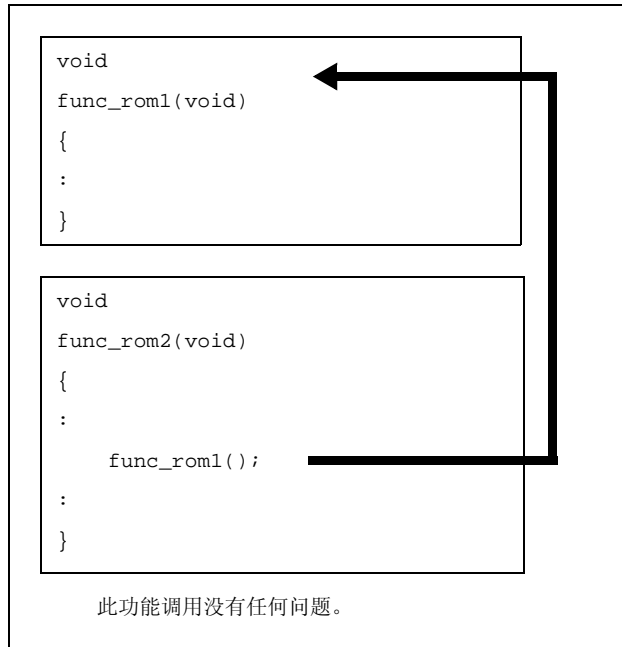
(2) 重链接功能的映像

当重链接功能使用时，功能调用显示如下。

(a) 从引导区域调用功能在引导区域

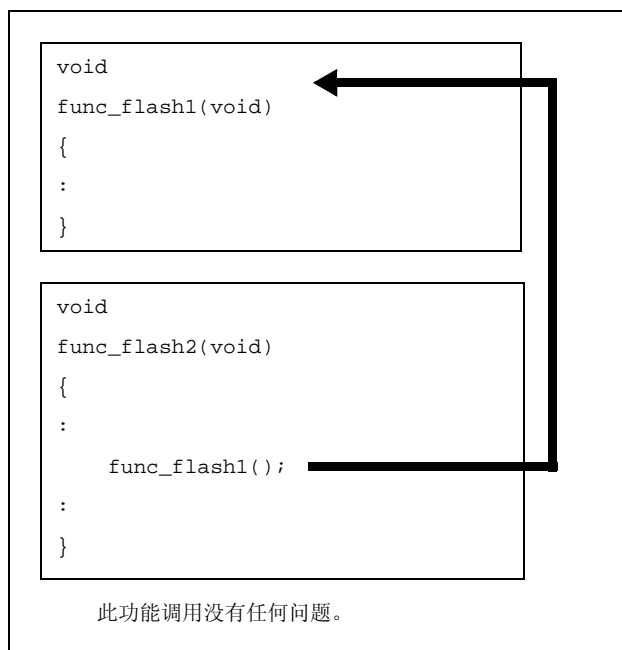
此功能调用没有任何问题因为在他们在引导区域编程之前地址已经确定。

图 B-7. 在引导区域

**(b) 从闪存区域调用功能在闪存区域**

此功能调用没有任何问题因为地址已经在闪存区域确定。

图 B-8. 在闪存区域

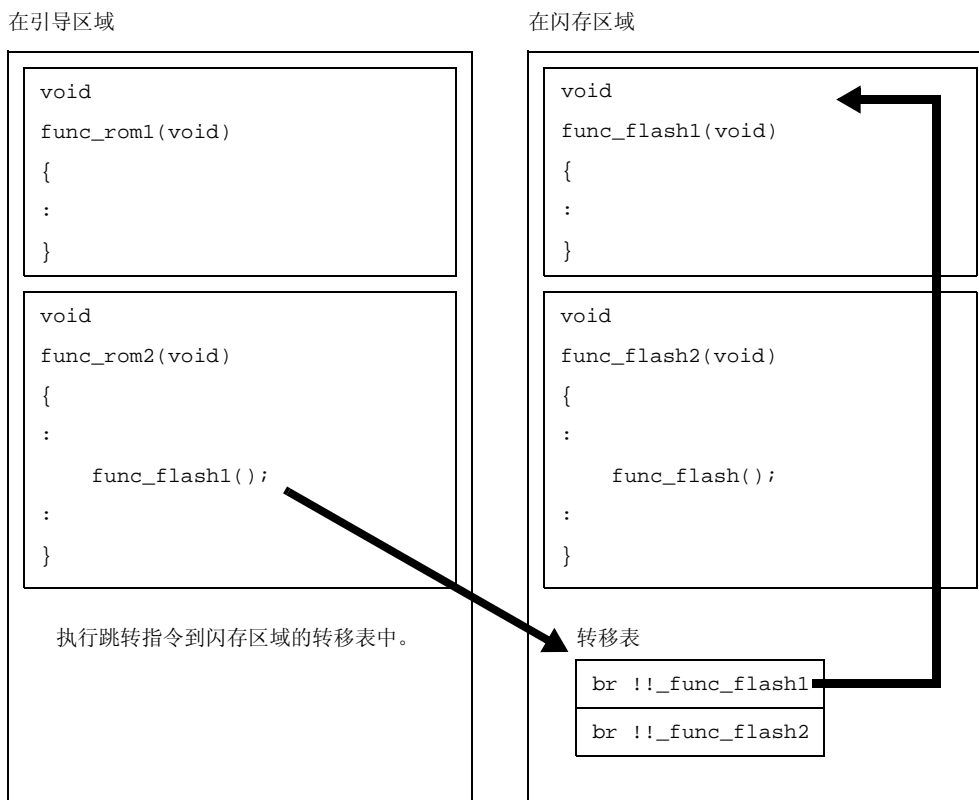


(c) 从引导区域调用功能在引导区域

当在闪存区域的功能从引导区域调用，从引导区域此功能的地址不能确定因为功能的大小，等等，在闪存区域已经改变。换句话说，在闪存区域的功能不能直接调用。解决该问题，在闪存区域执行跳跃指令到转移表中。

从那个表中执行相关功能的跳跃指令且跳跃到预定的功能上。

图 B-9. 从引导区域到闪存区域



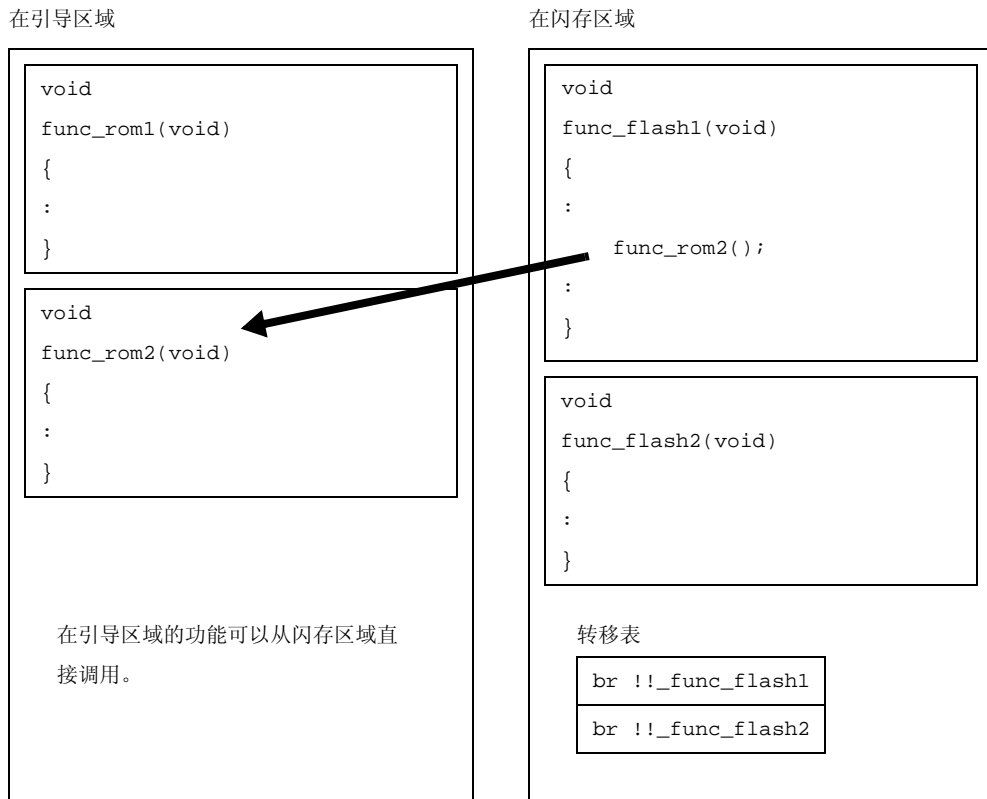
在与功能相同的方法中，这是与引用外部变量有关。

在闪存区域定义的全局变量不能从引导区域引用。因此，相同名称的外部变量不能定义在引导区域和闪存区域中。这些外部变量的只能从各自的区域引用。

(d) 从闪存区域调用在引导区域的功能

当在引导区域的功能从闪存区域调用，则此引导区域的内容不能改变。因此，在引导区域的功能可以从闪存区域直接调用。

图 B-10. 从闪存区域到引导区域



在具有相同方式的功能中，这是关于引用外部变量的。在引导区域中定义的全局变量不能从闪存区域引用。

(3) 认识重链接功能

这部分特意描述如何认识重链接功能。

(a) CubeSuite 功能

认识重链接功能，引导区域和闪存区域必须分别地建立。这意味着在引导区域创建后只有闪存区域修改（程序存储在 ROM 中）。当在 CubeSuite 中创建工程，因此，按照下列所述进行分割此工程。

- 工程分配到引导区域
- 工程分配到闪存区域（工程可以在未来修改）

除此之外，分别为每个工程准备启动事务和链接指令文件。

(b) #pragma ext_func 指令

当从引导区域调用在闪存区域中的功能，此功能的名称调用（标签名称）且 ID 编号通过使用 #pragma ext_func 指令分配到引导区域中。#pragma ext_func 指令的格式如下：

```
#pragma ext_func function-name ID-number
```

设定正数作为 ID 编号。不同的 ID 编号一定不能指定相同功能名称或相同 ID 编号一定不能指定不同功能名称。

当在闪存区域的功能名称通过使用 #pragma ext_func 指令在引导区域设定，则转移表建立。此转移表的地址由用户设定。

设定地址如下，通过使用编译选项 "-zt"，当引导区域的载入模块和闪存区域的载入模块创建时。

```
-zt start-address-of-branch-table
```

当对功能的主体执行转移时，实际功能地址是通过从已建立的转移表的起始位置引用 ID 编号的偏移量得来的，然后执行转移。

输出示例如下。

```
func_flash0()
func_flash1()
```

如果上述两个 C 函数分配到闪存区域且它们从引导区域调用，对引导区域的 C 源文件中描述如下：

```
#pragma ext_func func_flash0 1
#pragma ext_func func_flash1 2
```

建议在文件中说明这些 `#pragma ext_func` 指令以及通过使用 `#include` 指令在所有源文件中包含此文件，为了防止丢失说明或不一致的出现，也就是防止为同函数名指定不同 ID 编号或为不同函数名指定相同 ID 编号。

重链接函数映像显示如下：

<1> 引导区域的 C 源文件

```
#include "ext_def.h"

int boot_a = 0x12;
int boot_b = 0x34;
extern int func_flash1( int );
extern int func_flash2( int );

void boot_main( )
{
    :
}

void func( void )
{
    int k;
    boot_a = func_flash1(boot_a);
    boot_b = func_flash2(boot_b);
}
```

<2> 闪存区域的 C 源文件

```

#include "ext_def.h"

extern void func( void );

void main( void )
{
    func();
}

void func_flash1( )
{
    :
}

void func_flash2( )
{
    :
}

```

<3> ext_def.h

```

#pragma ext_func func_flash1 1
#pragma ext_func func_flash2 2

```

(c) 启动程序

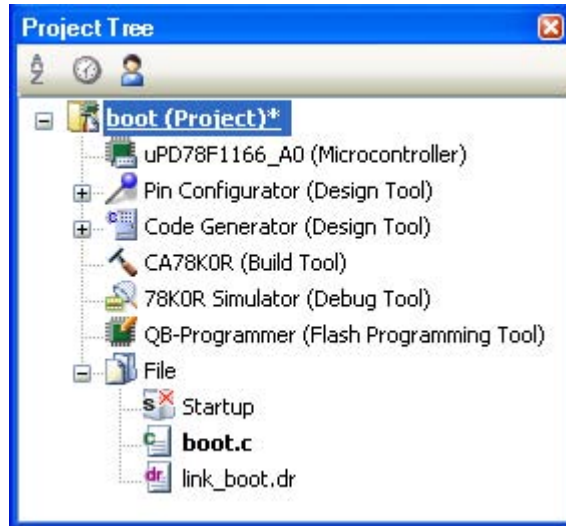
分别为引导区域和闪存区域准备启动程序。启动程序为引导区域和闪存区域做准备通过 CA78K0。每个启动程序必须进行下列的处理。

- 为引导区域进行处理 - 初始化已使用过的 RAM 区域
- 从引导区域转移到闪存区域的启动程序。
- 对闪存区域进行处理 - 初始化已使用过的 RAM 区域
- 移动到闪存区域的进程

(d) 如果建立指定的工程。**<1> 创建引导区工程**

在引导区创建工程并添加构建目标文件到工程中。

图 B-11. 引导区域工程



<2> 为引导区域工程设置构建选项。

在工程树上选择 build tool 节点并设置每个构建选项在 [属性面板](#)上。

<3> 设置变量 / 函数重新配置选项

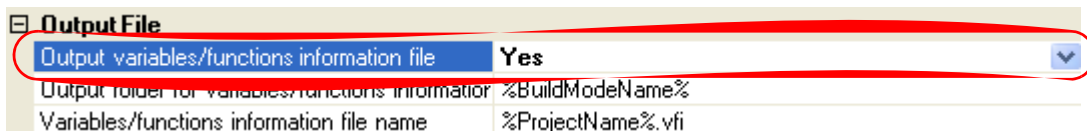
设置变量 / 函数重新配置选项，生成变量 / 函数信息文件并使用它来分配变量和函数。

选择 [\[Variables/Functions Relocation Options\]](#) 标签。

在 [\[Output File\]](#) 分项中，设置 [\[Output variables/functions information file\]](#) 属性为 [\[Yes\]](#)，生成空的变量 / 函数信息文件，并添加到工程中（它也将出现在工程树的 [File](#) 节点中）。输出目标通过在 [\[Output folder for variables/functions information file\]](#) 属性和 [\[Variables/functions information file name\]](#) 属性中进行设置。

备注 如果存在具有相同名称的变量 / 函数信息文件，则使用它配置构建。

图 B-12. 在引导区域的 [\[Output folder for variables/functions information file\]](#) 属性



设置 [\[Output folder for variables/functions information file\]](#) 属性和 [\[Variables/functions information file name\]](#) 属性来改变变量 / 函数信息文件的输出文件夹和文件名称。如果更改了 [\[Variables/functions information file\]](#) 属性，则生成空的变量 / 函数信息文件并添加到工程中（它也将出现在工程树的 [File](#) 节点中）。

<4> 设置编译选项

选择 [\[Compile Options\]](#) 标签。

在 [\[Memory Model\]](#) 分项中的 [\[Output objects for flash\]](#) 属性上选择 [\[No\]](#)。

另外，配置 [\[Start address of flash area\]](#) 属性和 [\[Start address of flash area branch table\]](#) 属性。

可以用于设定两个值的范围是 0 ~ FFFF。

备注 在 [\[Start address of flash area branch table\]](#) 属性中设定地址，它是在 flash 区域中的地址。

图 B-13. 在引导区域中的 [Memory Model] 分项

| Memory Model | |
|--|--|
| Memory model type | Medium model(Code 1M Byte, Data 24K Byte)(x-rom) |
| Output objects for flash | No |
| Start address of flash area | HEX 2000 |
| Start address of flash area branch table | HEX 2000 |
| Specify mirror area | M&A=0(-mi0) |

接下来，在 [Startup] 分项中的 [Use standard startup routine] 上选择 [Yes(For boot area)]。

图 B-14. 在引导区域的 [Use standard startup routine] 属性

| Startup | |
|---|---------------------------|
| Use standard startup routine | Yes(For boot area) |
| Use fixed area used by standard library | Yes |
| ROMize far area | Yes |
| Using standard startup routine | sOrlib.rel |

<5> 设置链接选项

选择 [Link Options] 标签。

在 [Device] 分项中，如果在 [Set flash start address] 属性上选择 [Yes(-zb)], [Flash start address] 显示。

在属性中，与在 [Compile Options] 标签的 [Memory Model] 分项上的 [Start address of flash area] 属性值配置相同值。

图 B-15. 在引导区域中的 [Device] 分项

| Device | |
|-------------------------------------|-----------------|
| Use on-chip debug | No |
| Set user option byte | No |
| Specify mirror area | M&A=0(-mi0) |
| Set flash start address | Yes(-zb) |
| Flash start address | HEX 2000 |
| Boot area load module file name | |
| Control allocation to self RAM area | No |

<6> 设置目标转换选项

选择 [Object Convert Options] 标签。

在 [[Hex File] 分项中 (默认) 的 [Split hex file] 属性上选择 [No]。

图 B-16. 在引导区域的 [Split hex file] 属性

| Hex File | |
|----------------------------|----------------------------------|
| Output hex file | Yes |
| Output folder for hex file | %BuildModeName% |
| Hex file name | %ProjectName%.hex |
| Hex file format | Intel expanded hex format (.ihx) |
| Split hex file | No |

<7> 运行引导区工程构建。

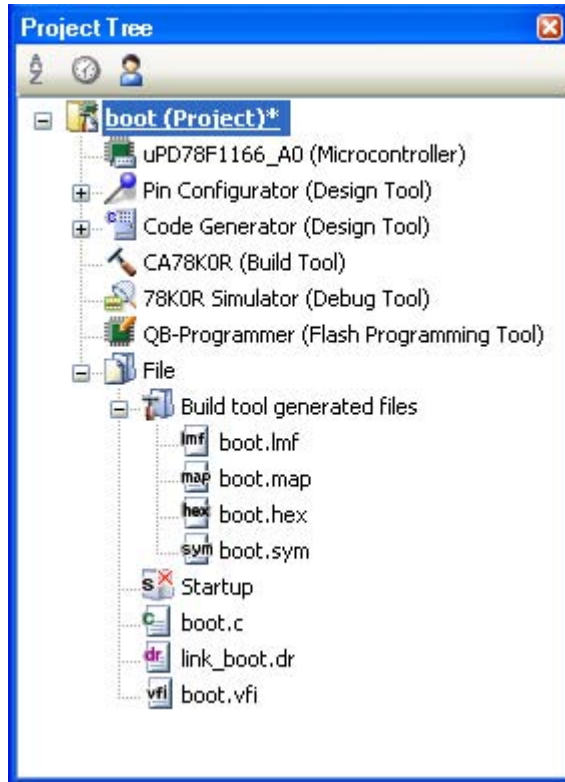
当执行引导区工程的构建时，创建载入模块文件。

也创建 hex 文件。

如果生成变量 / 函数信息文件，它将会自动输入到编译器中，并且会再次进行构建。

备注 通过运行构建，会覆盖变量 / 函数信息文件。如果改变输出文件夹或文件名，则指定的文件将生成新的。

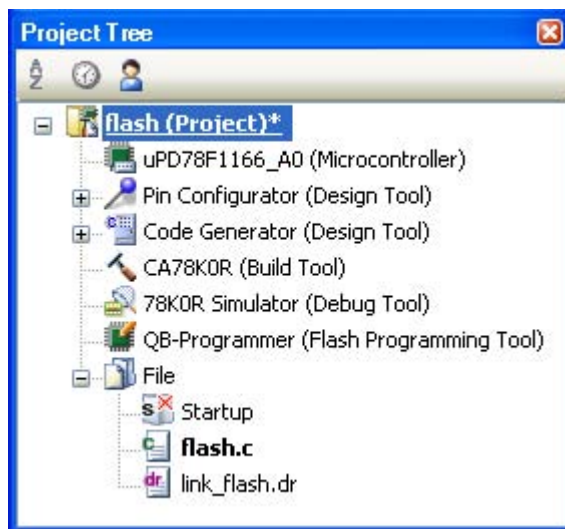
图 B-17. 为引导区域创建文件



<8> 创建闪存区工程

在引导区创建工程并添加构建目标文件到工程中。

图 B-18. 闪存区工程



<9> 为闪存区工程设置构建选项

在工程树上选择 build tool 节点并在属性面板上设置每个构建选项。

<10> 设置变量 / 函数重新配置选项

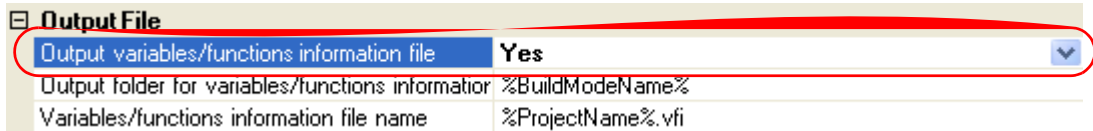
设置变量 / 函数重新配置选项，生成变量 / 函数信息文件并使用它来分配变量和函数。

选择 [Variables/Functions Relocation Options] 标签。

在 [Output File] 分项中，设置 [Output variables/functions information file] 属性为 [Yes]，生成空的变量 / 函数信息文件，并添加到工程中（它也将出现在工程树的 File 节点中）。输出目标通过在 [Output folder for variables/functions information file] 属性和 [Variables/functions information file name] 属性中进行设置。

备注 如果存在具有相同名称的变量 / 函数信息文件，则使用它配置构建。

图 B-19. 在 Flash 区域的 [Output folder for variables/functions information file] 属性



设置 [Output folder for variables/functions information file] 属性和 [Variables/functions information file name] 属性来改变变量 / 函数信息文件的输出文件夹和文件名称。如果更改了 [Variables/functions information file] 属性，则生成空的变量 / 函数信息文件并添加到工程中（它也将出现在工程树的 File 节点中）。

<11> 设置编译选项

选择 [Compile Options] 标签。

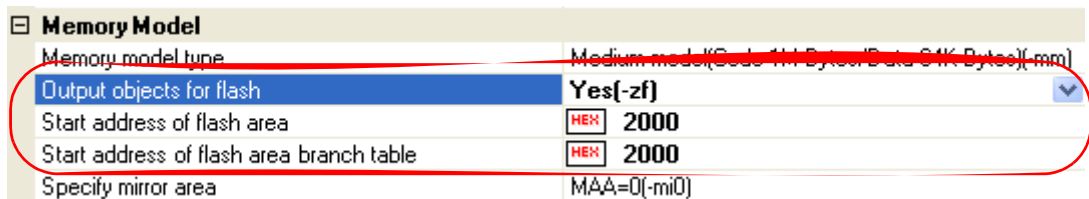
在 [Memory Model] 分项中的 [Output objects for flash] 属性上选择 Yes(-zf)]。

另外，配置 [Start address of flash area] 属性和 [Start address of flash area branch table] 属性。

可以用于设定两个值的范围是 0 ~ FFFF。

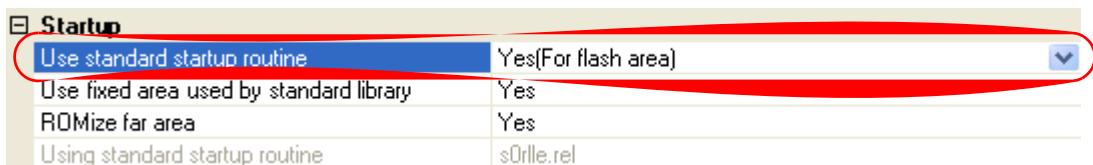
备注 在 [Start address of flash area branch table] 属性中设定的地址与在引导区域工程中设定的地址相同。

图 B-20. 在 Flash 区域中的 [Memory Model] 分项



接下来，在 [Startup] 分项中的 [Use standard startup routine] 属性上选择 [Yes(For flash area)]。

图 B-21. 在闪存区域的 [Use standard startup routine] 属性



然后，添加由启动区域创建的变量 / 函数信息文件到 flash 区域工程中。在 [Variable and Function Information File] 分项中的 [Variables/functions information file for boot area] 属性上设置启动区域的变量 / 函数信息文件。

图 B-22. 在 Flash 区域的 [Variables/functions information file for boot area] 属性

| Variables/functions Information File | |
|--|---------------------------|
| Using variables/functions information file | DefaultBuild\asm\boot.vfi |
| Variables/functions information file for boot area | ..\DefaultBuild\boot.vfi |

<12> 设置链接选项

添加已创建引导区的载入模块文件到闪存区的工程。选择 [\[Link Options\]](#) 标签。

在 [Device] 分项中的 [Boot area load module file name] 属性上设定引导区域载入模块文件。

图 B-23. 在闪存区的 [Boot area load module file name] 属性

| Device | |
|---------------------------------------|--------------------------|
| Use on-chip debug | No |
| Set user option byte | No |
| Specify mirror area | MMA=0(-mi0) |
| Set flash start address | No |
| Boot area load module file name | ..\DefaultBuild\boot.lmf |
| Control allocation to self flash area | No |

<13> 设置目标转换选项

选择 [\[Object Convert Options\]](#) 标签。

在 [[Hex File] 分项中的 [Split hex file] 属性上选择 [Yes(-zf)]。

图 B-24. 在闪存区域的 [Split hex file] 属性

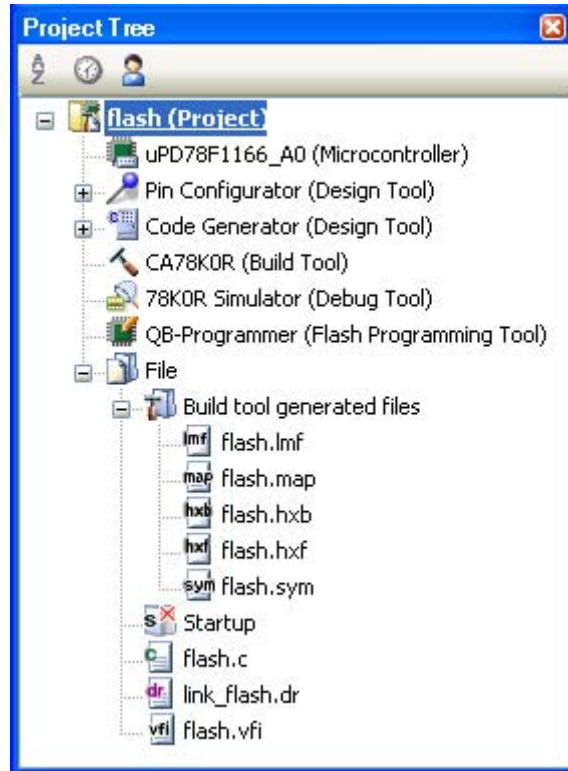
| Hex File | |
|----------------------------|----------------------------------|
| Output hex file | Yes |
| Output folder for hex file | %BuildModeName% |
| Hex file name | %ProjectName%.hex |
| Hex file format | Intel expanded hex format (.ihx) |
| Split hex file | Yes(-zf) |

<14> 运行闪存区的工程构建

当运行闪存区工程的构建时，具有重链接功能接口的载入模块文件创建。

引导区 hex 文件（通过构建引导区工程与其相同内容的文件创建）和闪存区 hex 文件也创建。

图 B-25. 为闪存区域创建文件

**(e) 如何改变转移表地址**

当设置转移表的起始位置除 2000H 以外，也改变中断向量处理按如下的方式。

- 在 vect.inc 中改变地址值 "ITBLTOP EQU 2000H"

vect.inc 文件的默认安装位置如下所述：

NEC Electronics CubeSuite\CubeSuite\CA78K0\Vx.xx\src\cc78k0\src

- ..\bat\repvect.bat

..\bat\mkstup.bat

在 DOS 提示上及升级启动程序和库，复制到 ..\..\lib78k0 并使用为链接。

(f) 描述链接指令文件

在使用链接指令文件时以下几点需要注意。

- 如果段地址放置在启动区域和闪存区域的重叠 RAM 区中，则链接器输出错误。因为 RAM 区必须同时在引导区域和闪存区域中引用，所以地址必须限定为了它们不重复。
- 与转移表相关的链接指令文件不会标识。它自动分配通过链接选项设定的地址。

然而，以下选项必须注意。

- 如果通过 -zt 在指定地址上设定分支表的空闲区域大小，则分配的链接指令文件如下。其他段没有影响。
- 如果通过 -zt 不在指定地址上设定分支表的空闲区域大小，则产生错误。

(g) 库

如果库函数从引导区域或闪存区域中调用，则在调用时库链接到目标。举个例子，即使库链接到闪存区域，相同的库链接到引导区域如果相同库函数从引导区域调用。当库函数调用，因此，在引导区域和闪存区域的转移不会发生。

(h) 中断处理程序

说明部分：在存在中断处理程序的地址区域中调用中断处理程序。

在下列例子中，中断处理程序功能名称必须也指定通过 `#pragma` 中断指令。

- 中断处理程序地址是在引导区域中。
- 中断处理程序主体是在闪存区域中。

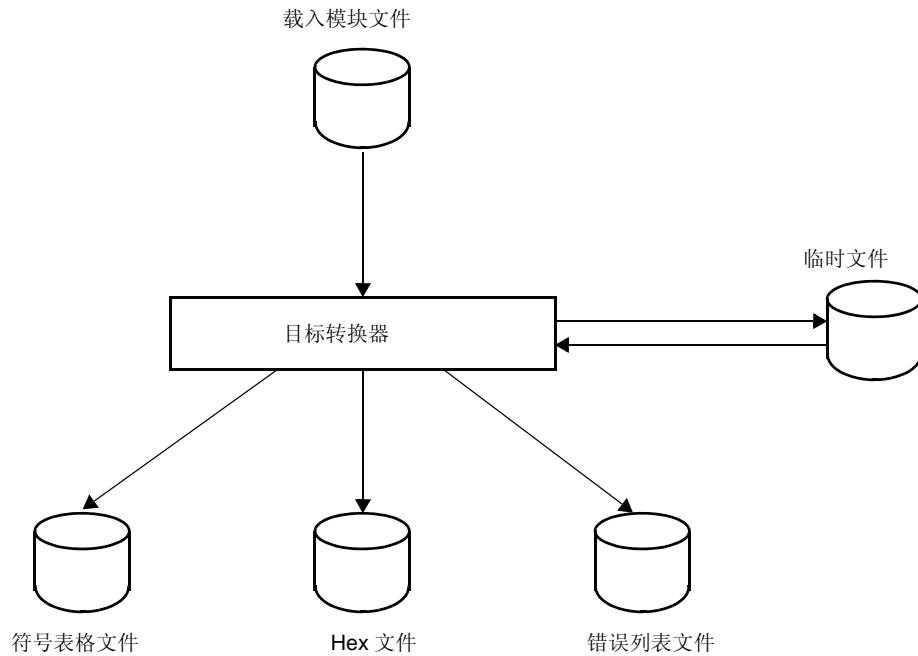
B.4 目标转换器

目标转换器输入通过 CA78K0 链接器输出的加载模块文件（所有引用地址数据必须在此断点确定）然后将数据转换为十六进制格式并将其作为目标模块文件输出。

目标转换器还将符号调试中使用的符号信息作为符号表格文件输出。

如果出现目标转换器错误，将显示错误消息用以说明导致错误的原因。

图 B-26. 目标转换器的 I/O 文件



B.4.1 I/O 文件

目标转换器的 I/O 文件如下所列：

表 B-11. 目标转换器的 I/O 文件

| 类型 | 文件名 | 说明 | 默认文件类型 |
|------|--------|---|--------|
| 输入文件 | 载入模块文件 | - 目标代码的二进制映像文件作为链接操作的结果输出。 - 通过链接器输出文件 | .lmf |
| | 参数文件 | - 包含执行命令参数的文件（用户创建文件） | .poc |
| 输出文件 | Hex 文件 | - 通过转换加载模块文件为十六进制目标格式所创建的文件 在屏蔽 ROM 进展和 PROM 程序使用时，使用这些文件。 | .hex |
| | 符号表格文件 | - 包含在输入文件的每个模块中所包含的符号信息的文件 | .sym |
| | 错误列表文件 | - 包含在目标转换过程汇总产生的错误信息的文件。 | .eoc |

B.4.2 功能

(1) Flash 存储器自改写模式支持

当 Flash 存储器自改写模式可用时，目标转换器可以在定位于 Flash 存储器的代码所在的引导区和 Flash 区创建分离的 hex 文件。为了输出分离 hex 文件，设定目标转换选项（-zf）。文件类型如下：

表 B-12. 当 -zf 选项设定时的文件类型

| 文件 | 文件类型 |
|----------------------|------|
| 在引导区 ROM 程序分区输出文件 | .hxb |
| 在程序分区而不是引导区 ROM 输出文件 | .hxf |

(2) Hex 文件

由目标转换器输出的 hex 文件可以输入到十六进制加载程序，例如 PROM 程序设计器或调试器。下述是示例程序的 hex 文件。

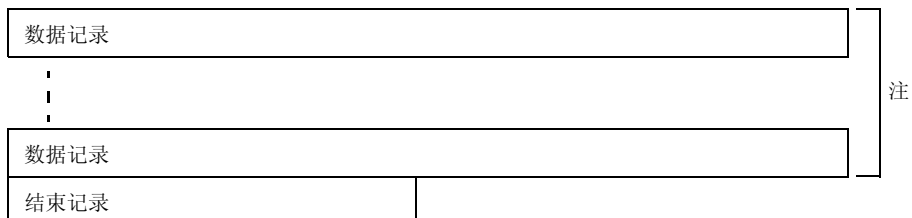
```

: 0200000080007E
: 1000800011201A1620FE9A93001421FE63958462B3
: 1000900095FAFE617131809AA40073617131809A82
: 0D00A000A40072AF4D8D020D070D30AFA8
: 00000001FF

```

(a) Intel 标准 hex 文件格式

图 B-27. Intel 标准格式



注 这里重复数据记录。

- 数据记录

| | | | | | |
|-----|-----|------|-----|------|-----|
| : | 02 | 0000 | 00 | 8000 | 7E |
| (1) | (2) | (3) | (4) | (5) | (6) |

| 条目号 | 说明 |
|-----|--|
| (1) | 记录标记 说明记录开始。 |
| (2) | 码号（2位） 存储在记录中的代码字节数目 最大存储字节数为 16。 |
| (3) | 定位地址（偏移量） 记录中的代码起始地址（偏移量）以 4 位十六进制格式显示。 |
| (4) | 记录类型 在 00 固定。 |

| 条目号 | 说明 |
|-----|---|
| (5) | 代码（最大：32 位） 目标代码每次以字节高 4 位与低 4 位分离的形式显示。 代码最大显示字节为 16 字节。 |
| (6) | 校验和（2 位） 输入的数值从零开始减去，按照从地区代码到编码的倒计时顺序。 |

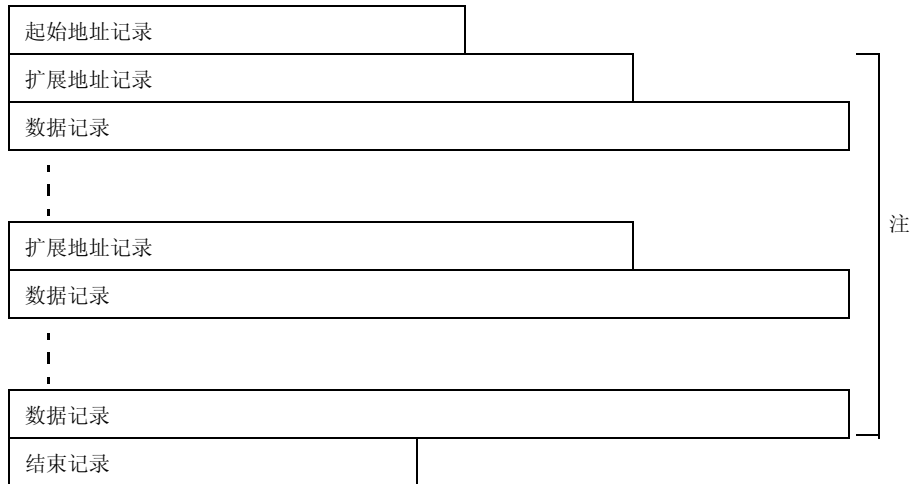
- 结束记录

| | | | | |
|-----|-----|------|-----|-----|
| : | 00 | 0000 | 01 | FF |
| (1) | (2) | (3) | (4) | (5) |

| 条目号 | 说明 |
|-----|------------------|
| (1) | 记录标记 说明记录开始。 |
| (2) | 码号 在 00 固定。 |
| (3) | 在 0000 固定。 |
| (4) | 记录类型 在 01 固定。 |
| (5) | 校验和 在 FF 固定。 |

(b) Intel 扩展 hex 文件格式

图 B-28. Intel 扩展格式



注 扩展地址记录和数据记录在这里重复。

- 扩展地址记录

| | | | | | |
|-----|-----|------|-----|------|-----|
| : | 02 | 0000 | 02 | XXXX | SS |
| (1) | (2) | (3) | (4) | (5) | (6) |

| 条目号 | 说明 |
|-----|--|
| (1) | 记录标记 说明记录开始。 |
| (2) | 码号 在 02 固定。 |
| (3) | 在 0000 固定。 |
| (4) | 记录类型 在 02 固定。 |
| (5) | 片断的图形值 段值以 4 位十六进制格式显示。 |
| (6) | 校验和 (2 位) 该值为从 0 中减去中从码号到地址高 8 位值的倒数。 |

- 数据记录

| | | | | | |
|-----|-----|------|-----|-------|-----|
| : | 02 | 0000 | 00 | 80000 | 7E |
| (1) | (2) | (3) | (4) | (5) | (6) |

| 条目号 | 说明 |
|-----|---|
| (1) | 记录标记 说明记录开始。 |
| (2) | 码号 (2 位) 存储在记录中的代码字节数目 最大存储字节数为 16。 |
| (3) | 定位地址 (偏移量) 记录中的代码起始地址 (偏移量) 以 4 位十六进制格式显示。 |
| (4) | 记录类型 在 00H 固定。 |
| (5) | 代码 (最大: 32 位) 目标代码每次以字节高 4 位与低 4 位分离的形式显示。 代码最大显示字节为 16 字节。 |
| (6) | 校验和 (2 位) 输入的数值从零开始减去, 按照从地区代码到编码的倒计时顺序。 |

- 起始地址记录

| | | | | | | |
|-----|-----|------|-----|------|------|-----|
| : | 04 | 0000 | 03 | 0000 | 0000 | F9 |
| (1) | (2) | (3) | (4) | (5) | (6) | (7) |

| 条目号 | 说明 |
|-----|------------------|
| (1) | 记录标记 说明记录开始。 |
| (2) | 码号 在 04 固定。 |
| (3) | 在 0000 固定。 |
| (4) | 记录类型 在 03 固定。 |

| 条目号 | 说明 |
|-----|-----------------|
| (5) | 在 0000 固定。 |
| (6) | 在 0000 固定。 |
| (7) | 校验和 在 F9 固定。 |

- 结束记录

| | | | | |
|-----|-----|------|-----|-----|
| : | 00 | 0000 | 01 | FF |
| (1) | (2) | (3) | (4) | (5) |

| 条目号 | 说明 |
|-----|------------------|
| (1) | 记录标记 说明记录开始。 |
| (2) | 码号 在 00 固定。 |
| (3) | 在 0000 固定。 |
| (4) | 记录类型 在 01 固定。 |
| (5) | 在 FF 固定。 |

(c) 扩展 tektronix hex 文件格式

Hex 文件包含下列三种类型的块。

- 数据块
- 符号块（这是不使用的块。符号信息使用符号表格文件。）
- 结束块

每个块由包含公用 6 字符的头文件开启，并由字符串的结束行结束。

每个块的最大长度为 255，其中不包含起始的字符%和结束行。

公用头文件的格式如下所示。

表 B-13. 扩展 Tektronix 头文件

| 条目 | ASCII 字符数 | 说明 |
|----|-----------|-------------------------|
| % | 1 | 百分号设定块为扩展 tektronix 格式。 |

| 条目 | ASCII 字符数 | 说明 |
|-----|-----------|---|
| 块长 | 2 | 这是 2 位十六进制数，用以说明块中的字符数。 该字符数不包含字符%和结束行。 |
| 块类型 | 1 | 6 = 数据块 3 = 符号块 8 = 结束块 |
| 校验和 | 2 | 这是 2 位十六进制数，是块中所有字符 ^注 的总和（除了起始字符%，校验和和结束行）除以 256 后的余数。 |

注 用于校验和计算的字符值

| 字符 | 值（十进制） |
|----------------|---------|
| 0 ~ 9 | 0 ~ 9 |
| A ~ Z | 10 ~ 35 |
| \$ | 36 |
| % | 37 |
| . (period) | 38 |
| _ (underscore) | 39 |
| a ~ z | 40 ~ 65 |

- 数据块

数据块的格式如下所示。

表 B-14. 扩展 Tektronix 的数据块格式

| 区段 | ASCII 字符数 | 说明 |
|------|-----------|-----------------------|
| 起始段 | 6 | 标准头文件 块类型 = 6 |
| 加载地址 | 2 ~ 17 | 目标代码加载的地址。 字符数为变量。 |
| 目标代码 | 2n | 字节数 n，作为 2 位十六进制数显示 |

注意事项 在扩展 Tektronix 中，指定区段的字符数在 2 ~ 17 之间（实际数为 1 ~ 16）。在该变量区段的首字符是用于说明区段长度的十六进制数。此变量区段的首字符是用于说明区段长度的十六进制数。字符串的长度因此为 1 ~ 16 个字符，并且变量长度区的长度包含字符串长度为 2 ~ 17。

| | | | | | | |
|-----|-----|-----|-----|-----|-----|--------------|
| % | 15 | 6 | 1C | 3 | 100 | 020202020202 |
| (1) | (2) | (3) | (4) | (5) | (6) | (7) |

| 条目号 | 说明 |
|-----|----------------|
| (1) | 起始段字符 |
| (2) | 块长 15H = 21 |
| (3) | 块类型 6 |
| (4) | 校验和 1CH |
| (5) | 加载地址的位数 |
| (6) | 加载地址 100H |
| (7) | 目标代码 6 字节 |

- 结束块

结束块的格式如下所示。

表 B-15. 扩展 Tektronix 的结束块格式

| 区段 | ASCII 字符数 | 说明 |
|------|-----------|-----------------------|
| 起始段 | 6 | 标准头文件 块类型 = 8 |
| 加载地址 | 2 ~ 17 | 程序执行的起始地址。 字符数为变量。 |

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| % | 08 | 8 | 1A | 2 | 80 |
| (1) | (2) | (3) | (4) | (5) | (6) |

| 条目号 | 说明 |
|-----|-------------|
| (1) | 起始段字符 |
| (2) | 块长 8H |
| (3) | 块类型 8 |
| (4) | 校验和 1AH |
| (5) | 加载地址的位数 |
| (6) | 加载地址 80H |

- 符号块（不使用）

扩展技术符号块为用于符号调试的数据。该块可以假定具有下述特征。

表 B-16. Symbo 扩展 Tektronix 的符号块特征

| 条目 | 特征 |
|-----------|--|
| 符号 | 1 ~ 16 个大写和小写字母，数字，小数点和下划线的组合。 数字不允许作为起始字符。 |
| 值 | 上限为 64 位（16 位十六进制数） |
| 类型 | 地址或标量（标量为不是地址的任意数字值）。 地址划分为代码地址（指定地址）和数据地址（数据条目地址）。 |
| 全局 / 局部设定 | 说明符号是全局（外部参考也是允许的）适用还是局部适用的。 |
| 片断成员关系 | 片断可以认为是给定存储名称的域。 程序中的每个地址属于至少一个片断。标量不属于任何片断。 |

符号块的格式如下所示。

表 B-17. 扩展 Tektronix 的符号块格式

| 区段 | ASCII 字符数 | 说明 |
|------|-----------|--|
| 起始段 | 6 | 标准头文件 块类型 = 3 |
| 片断名 | 2 ~ 17 | 包含在块中定义的符号的片断的名称。字符数为变量。 |
| 片断定义 | 5 ~ 35 | 该字段必须在每个片断的某一符号块显示。该字段可以放置于任意符号定义字符段的前面或后面。 关于此格式，参阅“表 B-18. 扩展 Tektronix 的符号块片断定义字符段”。 |
| 符号定义 | 5 ~ 35 每个 | 这是大于 0 的符号定义字段。 关于此格式，参阅“表 B-19. 扩展 Tektronix 的符号块符号定义字符段”。 |

包含在程序中的符号转换成符号块。每个符号块包含片断名和属于该片断的符号列表。（如果需要，可以在任意片断中同时包含标量。）

在相同片断里的符号可以放置在一个或更多的块当中。

片断定义字符段的格式和符号块中符号定义字符段的格式显示如下。

表 B-18. 扩展 Tektronix 的符号块片断定义字符段

| 区段 | ASCII 字符数 | 说明 |
|-----|-----------|---|
| 0 | 1 | 0 设定字符段为片断定义字符段。 |
| 基地址 | 2 ~ 17 | 这是片断起始地址。 字符数为变量。 |
| 长 | 2 ~ 17 | 说明片断长度。 字符数为变量并通过下述方法计算： $1 - (\text{高位地址} - \text{基地址})$ |

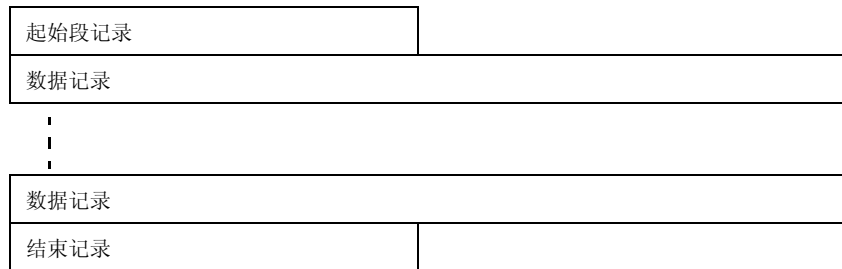
表 B-19. 扩展 Tektronix 的符号块符号定义字符段

| 区段 | ASCII 字符数 | 说明 |
|-----|-----------|---|
| 类型 | 1 | 显示 1 位十六进制数说明全局 / 局部符号设定以及值的类型。 1 = 全局地址 2 = 全局标量 3 = 全局代码地址 4 = 全局数据地址 5 = 局部地址 6 = 局部标量 7 = 局部代码地址 8 = 局部数据地址 |
| 符号 | 2 ~ 17 | 说明符号长度。这是变量。 |
| 数字值 | 2 ~ 17 | 这是与符号对应的值。字符数为变量。 |

(d) Motorola S- 类型格式

改变生成的具有三种记录类型和五个记录的 hex 文件。文件全部的结构如下图所示。

图 B-29. Motorola S- 类型格式



记录类型显示如下。

表 B-20. Motorola Hex 文件的记录类型

| 类型 | 记录类型 |
|-------------|---------------------------|
| 起始段记录 (可选的) | S0 |
| 数据记录 | S2 (标准 24 位) S3 (32 位) |
| 结束记录 | S8 (标准 24 位) S7 (32 位) |

Motorola 十六进制格式文件分成标准的 24 位地址和 32 位地址。标准地址包括记录 S0, S2 和 S8。32 位地址包括记录 S0, S3 和 S7。起始段记录 S0 是可选的并不输出。CR 字符放置在每个 S 记录的末尾。通用格式和它们对于每个记录每个字符段所具有的意义如下所示。

表 B-21. 每个记录的通用格式

| 记录类型 | 通用格式 |
|------|----------------------|
| S0 | S0XXYY ... YYZZZ |
| S2 | S2XXWWWWWDD ... DDZZ |
| S3 | S3XXWWWWWDD ... DDZZ |

| 记录类型 | 通用格式 |
|------|----------------|
| S7 | S7XXWWWWWWWWZZ |
| S8 | S8XXWWWWWWWWZZ |

表 B-22. 字符段的意义

| 区段 | 含义 |
|---------------|--|
| Sn | 记录类型 |
| XX | 数据记录的长度 地址，十六进制数据和校验和的字节数。 |
| YY ?c YY | 文件名称 输入文件名的 ASCII 代码表示为十六进制数 |
| WWWWWW [WW] | 24th [32th] 位地址 |
| DD ?c DD | 十六进制数据 数据的 1 个字节表示为 2 位十六进制数。 |
| ZZ | 校验和 记录长度、地址和十六进制数据的每个字节的和的补码 1 的低 1 字节表示为 1 个 2 位十六进制数。 |

| | | | | |
|-----|-----|--------|----------|-----|
| S2 | 08 | 00FF11 | D4520A20 | A0 |
| (1) | (2) | (3) | (4) | (5) |

| 条目号 | 说明 |
|-----|---------------|
| (1) | 记录类型 |
| (2) | 记录长度 |
| (3) | 加载地址 (24 位地址) |
| (4) | 十六进制数据 |
| (5) | 校验和 |

- S0 记录

| | | | |
|-----|-----|----------|-----|
| S0 | XX | YYYYYYYY | ZZ |
| (1) | (2) | (3) | (4) |

| 条目号 | 说明 |
|-----|--------------------------------------|
| (1) | 记录类型 |
| (2) | 记录长度 这是将 (3) 中的字节数加上 (4) 中字节数的结果。 |
| (3) | 文件名称 |
| (4) | 校验和 |

- S2 记录

| | | | | |
|-----|-----|--------|----------|-----|
| S2 | XX | WWWWWW | DD ?c DD | ZZ |
| (1) | (2) | (3) | (4) | (5) |

| 条目号 | 说明 |
|-----|---|
| (1) | 记录类型 |
| (2) | 记录长度 这是将 (3) 中的字节数加上 (4) 中字节数在加上 (5) 中的字节数的结果。 |
| (3) | 加载地址 这是范围在 0H ~ FFFFFFFH 内的 (4) 中的数据 24 位加载地址。 |
| (4) | 数据 这是加载的数据本身。 |
| (5) | 校验和 |

- S3 记录

| | | | | |
|-----|-----|---------|----------|-----|
| S3 | XX | WWWWWWW | DD ?c DD | ZZ |
| (1) | (2) | (3) | (4) | (5) |

| 条目号 | 说明 |
|-----|--|
| (1) | 记录类型 |
| (2) | 记录长度 这是将 (3) 中的字节数加上 (4) 中字节数在加上 (5) 中的字节数的结果。 |
| (3) | 加载地址 这是范围在 0H ~ FFFFFFFFH 内的 (4) 中的数据 32 位加载地址。 |
| (4) | 数据 这是加载的数据本身。 |
| (5) | 校验和 |

- S7 记录

| | | | |
|-----|-----|---------|-----|
| S7 | XX | WWWWWWW | ZZ |
| (1) | (2) | (3) | (4) |

| 条目号 | 说明 |
|-----|---|
| (1) | 记录类型 |
| (2) | 记录长度 这是将 (3) 中的字节数加上 (4) 中字节数的结果。 |
| (3) | 登录地址 这是范围在 0H ~ FFFFFFFFH 内的 32 位登录地址。 |
| (4) | 校验和 |

- S8 记录

| | | | |
|-----|-----|---------|-----|
| S8 | XX | WWWWWWW | ZZ |
| (1) | (2) | (3) | (4) |

| 条目号 | 说明 |
|-----|------|
| (1) | 记录类型 |

| 条目号 | 说明 |
|-----|--|
| (2) | 记录长度 这是将 (3) 中的字节数加上 (4) 中字节数的结果。 |
| (3) | 登录地址 这是范围在 0H ~ FFFFFFFH 内的 24 位登录地址。 |
| (4) | 校验和 |

(3) 符号表格文件

通过目标转换器输出的符号表格文件输入到调试器。

下述是示例程序的符号表格文件。

```
#05
; FF PUBLIC
01000E9CONVAH
0100000MAIN
01000D2START
00FFE20_@STBEG
00FCF00_@STEND
; FF SAMPM
<02FFE20HDTSA
02FFE21STASC
; FF SAMPS
<010015CSASC
0100162SASC1
=
```

图 B-30. 符号表格文件格式

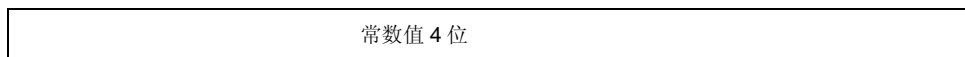
| | | | | | | | | |
|------------|------|------|-------|----|--------|----|----|-----------|
| 符号表格的起始段 | # | 05 | CR | LF | | | | |
| 公用符号的起始段 | ; | FF | 5 个空格 | | PUBLIC | CR | LF | |
| | 注 -> | 符号属性 | 符号值 | | 公用符号名称 | CR | LF | 公用符号 |
| | | : | 5 个空格 | | : | : | : | |
| | ; | FF | 5 个空格 | | 模块名 1 | CR | LF | 每个模块的局部符号 |
| 局部符号的起始段 | < | 符号属性 | 符号值 | | 局部符号名称 | CR | LF | |
| | | 符号属性 | 符号值 | | 局部符号名称 | CR | LF | |
| | | : | : | | : | : | : | |
| | ; | FF | 5 个空格 | | 模块名 2 | CR | LF | |
| 在目标模块单元中重复 | | : | : | | : | : | : | |
| 符号表格的结束标记 | = | CR | LF | | | | | |

注 符号特征的值如下所示。
参阅下述关于符号值格式的图

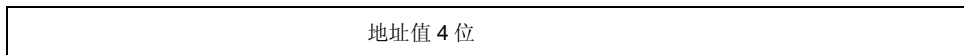
| 值 | 符号特征 |
|----|---------------|
| 00 | 由 EQU 指示定义的常量 |
| 01 | 在代码片断内的标签 |
| 02 | 在数据片断内的标签 |
| 03 | 位符号 |
| FF | 模块名称 |

图 B-31. 符号值的格式

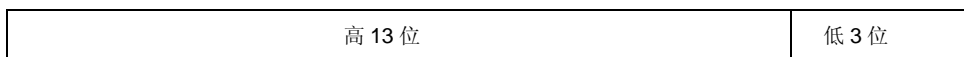
- 当符号特征为 NUMBER 时



- 当符号特征为 LABEL 时



- 当符号特征为位符号] 时



高 13 位：从 0FE00H 开始的相关地址

低 3 位：位位置 (0 ~ 7)

B.4.3 操作方法

(1) 目标转换器启动

以下两种方法可以用于启动目标转换器。

(a) 从命令行启动

```
X:[path-name]>oc78k0r[Δoption] ... load-module-file-name[Δoption] ... [Δ]
```

| | |
|---------|---|
| X | 当前驱动器名称 |
| 路径名 | 当前文件夹名 |
| oc78k0 | 目标转换器的命令名 |
| 选项 | 输入目标转换器操作的详细指令。 当设置两个或两个以上目标转换器选项时，以空格分隔选项。大写字母和小写字母在目标转换器选项中不予区分。关于目标转换选项的详细信息，参阅“B.4.4 选项”。 带有空格的路径附在一对双引号 (") 中。 |
| 加载模块文件名 | 加载模块文件名转换 带空格的文件名路径附在一对双引号 (") 中。 |

示例 输出 hex 文件 (sample.hex)，描述如下：

```
C>oc78k0r k0r.lmf -osamle.hex
```

(b) 从参数文件启动

当启动目标转换器时需要的数据无法适用于命令行时，或者当相同的目标转换选项重复指定，每次目标转换执行时，使用参数文件。

通过参数文件启动汇编程序，在命令行上设定参数文件选项 (-f)。

通过参数文件启动目标转换器如下：

```
X>oc78k0r[Δload-module-file]Δ-fparameter-file
```

| | |
|-------|-------------------|
| -f | 参数文件说明选项 |
| 参数文件名 | 包含启动目标转换器所需要数据的文件 |

备注 使用编辑器创建参数文件。

参数文件内容的书写规则如下：

```
[[[Δ]option[Δoption] ... [Δ]Δ]] ...
```

- 如果载入模块名在命令行遗漏，则在参数文件中仅有载入模块名可以设定。
- 加载模块文件名也可以在选项后写入。
- 在参数文件中写入所有的编译选项以及在命令行中定义的输出文件名。

示例 使用编辑器创建参数文件 k0r.poc，然后启动目标转换器。

```
; parameter file
k0r.lmf -osample.hex
-ssample.sym -r
```

```
C>ra78k0r -fk0rmain.pra
```

(2) 执行开始和结束消息

(a) 执行开始消息

当目标转换器启动时，执行启动消息出现在显示器上。

```
78K0R Object Converter Vx.xx [xx xxx xxxxx]
Copyright(C) NEC Electronics Corporation xxxxx
```

(b) 执行结束消息

如果没有检测到目标转换器目标转换错误，则目标转换器输出下述消息到显示器并返回控制权到主操作系统。

```
Target chip : uPD78xxx
Device file : Vx.xx

Object Conversion Complete, 0 error(s) and 0 warning(s) found.
```

如果检测到目标转换器目标转换错误，则目标转换器输出错误号码到显示器并返回控制权到主操作系统。

```
Target chip : uPD78xxx
Device file : Vx.xx

Object Conversion Complete, 3 error(s) and 0 warning(s) found.
```

如果在目标转换执行期间目标转换器检测到致命错误致使目标转换无法继续，则目标转换器输出消息到显示器，取消目标转换并返回控制权到主操作系统。

示例 1. 设定不存在的加载模块文件。

```
C>oc78k0r sample.lmf
```

```
78K0R Object Converter Vx.xx [xx xxx xxxx]
  Copyright(C) NEC Electronics Corporation xxxx

RA78K0R error F4006 : File not found 'sample.lmf'
程序终止。
```

在上述例子中，不存在的加载模块文件设定。错误产生同时目标转换器退出目标转换操作。

2. 设定不存在目标转换选项。

```
C>oc78k0r k0r.lmf -a
```

```
78K0R Object Converter Vx.xx [xx xxx xxxx]
  Copyright(C) NEC Electronics Corporation xxxx

RA78K0R error F4018 : Option is not recognized '-a'
Please enter 'OC78K0R--' , if you want help messages.
程序终止。
```

在上述例子中，不存在的目标转换选项设定。错误产生同时目标转换器退出目标转换操作。

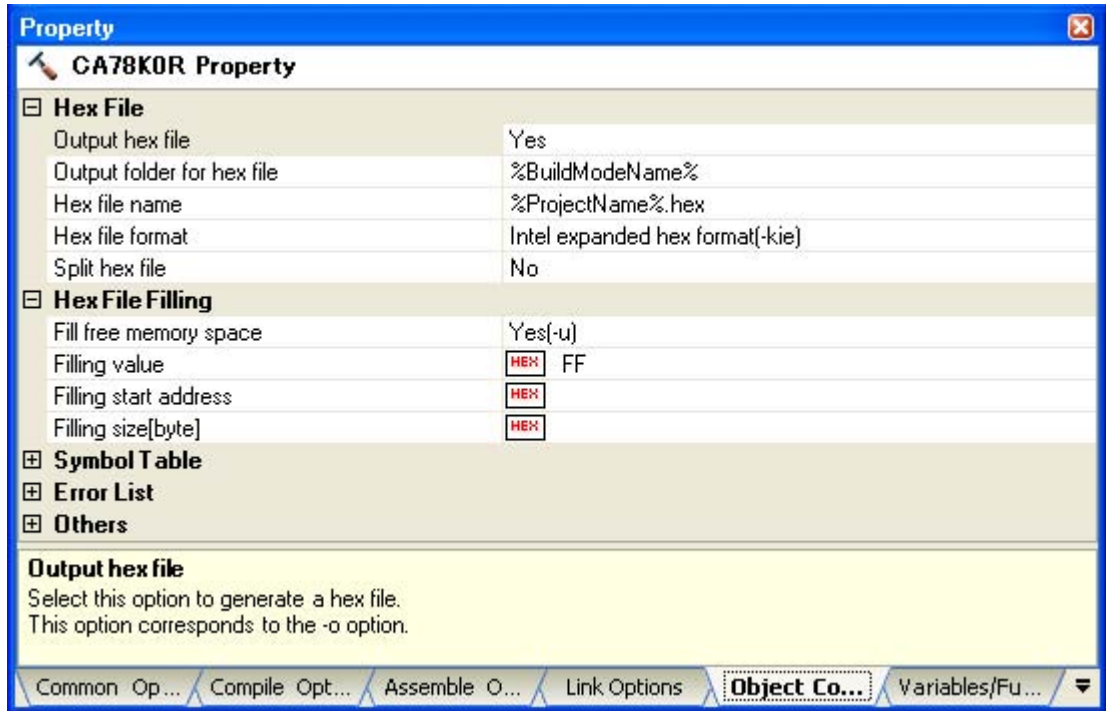
(3) 在 CubeSuite 中设置选项

本章节描述了通过 CubeSuite 怎样设置目标转换选项。

通过 CubeSuite 的[工程树面板](#)，选择 Build Tool 节点。然后通过 [View] 菜单选择 [Property]。打开[属性面板](#)。再选择 [\[Object Convert Options\]](#) 标签。

可以通过在该列表中设定需要的属性设置不同的目标转换选项。

图 B-32. 属性面板: [Object Convert Option] 标签



B.4.4 选项

(1) 类型

编译选项是目标转换器操作的详细指令。
目标转换选项的类型和注解如下。

表 B-23. 目标转换选项

| 分类 | 选项 | 说明 |
|------------|------|----------------------------------|
| Hex 文件输出设定 | -o | 设定 hex 文件的格式。 |
| | -no | |
| 符号表格文件输出设定 | -s | 设定符号表格文件的输出。 |
| | -ns | |
| 目标地址顺序分类设定 | -r | 依照地址顺序将十六进制格式目标进行分类。 |
| | -nr | |
| 目标存储值设定 | -u | 输出设定的存储值作为之前没有十六进制目标输出的地址区的目标代码。 |
| | -nu | |
| 错误表文件输出指定 | -e | 输出错误列表文件 |
| | -ne | |
| 参数文件说明 | -f | 通过指定文件输入选项和输入文件名。 |
| Hex 格式设定 | -ki | Intel 标准 hex 格式 |
| | -kie | Intel 扩展 hex 格式 |
| | -kt | 扩展 tektronix 格式 |
| | -km | Motorola S 类型格式 (标准地址) |
| | -kme | Motorola S 类型格式 (32 位地址) |

| 分类 | 选项 | 说明 |
|--------------------------|-----|-----------------------------|
| 设备文件查找路径说明 | -y | 从指定的路径中读取器件文件。 |
| 用于内嵌 Flash 存储产品的文件分离输出设定 | -zf | 将文件划分成两个部分：一个用于引导区，一个用于其他区。 |
| 帮助说明 | -- | 在屏幕上输出帮助信息。 |

Hex 文件输出设定

Hex 文件输出设定选项如下。

- `-o/-no`

-o/-no**[Description format]**

```
-o[output-file-name]
-no
```

- 省略时解释
- `-oinput-file-name.hex`

[Function]

- `-o` 选项设定 hex 文件的输出。
这也同时指定输出文件名和路径。
- `-no` 选项设定没有 hex 文件输出。

[Application]

- 使用 `-o` 选项设定 hex 文件的输出位置或改变其文件名。
- 当仅为输出符号表格文件执行目标转换时，设定 `-no` 选项。该操作将缩短目标转换时间。

[说明]

- 如果 “*output-file-name*” 当 `-o` 选项设定时省略，hex 文件 “*input-file-name.hex*” 将输出到当前文件夹。
- 如果仅在 “*output-file-name*” 中设定了路径名，“*input-file-name.hex*” 将输出到指定路径。
- 如果同时指定了 `-o` 和 `-no` 选项，则最后指定的选项无效。
- 如果 `-zf` 选项设定，文件类型如下。

| 文件 | 文件类型 |
|----------------------|------|
| 在引导区 ROM 程序分区输出文件 | .hxb |
| 在程序分区而不是引导区 ROM 输出文件 | .hxf |

[Example of use]

- 输出 hex 文件（`sample.hex`），描述如下：

```
C>oc78k0r k0r.lmf -osample.hex
```

符号表格文件输出设定

符号表格文件输出设定选项如下。

- `-s/-ns`

`-s/-ns`

[Description format]

```
-s[output-file-name]  
-ns
```

- 省略时解释
- `-input-file-name.sym`

[Function]

- `-s` 选项设定符号表格文件的输出。该选项同时设定输出位置和文件名。
- `-ns` 选项设定没有符号表格文件输出。

[Application]

- 使用 `-s` 选项设定符号表格文件的输出位置或改变其文件名。
- 当仅为输出符号表格文件执行目标转换时，设定 `-ns` 选项。
该操作将缩短目标转换时间。

[说明]

- 如果 “*output-file-name*” 当 `-s` 选项设定时省略，符号表格文件 “*input-file-name.sym*” 将输出到当前文件夹。
- 如果仅在 “*output-file-name*” 中设定了路径名，“*input-file-name.sym*” 将输出到指定路径。
- 如果 `-s` 选项和 `-ns` 选项同时设定，最后设定的选项有效。

[Example of use]

- 输出符号表格文件（`sample.sym`），描述如下：

```
C>oc78k0r k0r.lmf -ssample.sym
```

目标地址顺序分类设定

目标地址顺序分类设定选项如下。

-r/-nr

-r/-nr

[Description format]

```
-r  
-nr
```

- 省略时解释
-r

[功能]

- r 选项依照地址顺序输出十六进制格式目标的分类。
- nr 选项依照存储在加载模块文件中的顺序输出十六进制格式目标。

[Application]

- 当 hex 格式目标不需要依照地址顺序分类时，使用 -nr 选项设定。

[说明]

- 如果 -r 选项和 -nr 选项同时设定，最后设定的选项有效。
- 如果 -no 选项设定，-r 和 -nr 选项无效。

[Example of use]

- 依照地址顺序将十六进制格式目标进行分类，描述如下：

```
C>oc78k0r k0r.hex -r
```

目标存储值设定

目标存储值设定选项如下。

- `-u/-nu`

`-u/-nu`

[Description format]

```
-ufilling-value[, [start-address], size]
-nu
```

- 省略时解释
- `-u0FFH` (填入 `0FFH`)

[功能]

- `-u` 选项输出设定的存储值作为之前没有十六进制目标输出的地址区的目标代码。
- `-nu` 选项使 `-u` 选项失效。

[Application]

- 没有十六进制目标输出的地址区将可能写入不需要的代码。当这样的地址程序因为任意理由访问时，其所对应的操作是无法预知的。通过设定 `-u` 选项，在没有输出 `hex` 文件的地址区预先写入代码。

[说明]

- 可以用于填充值的范围是 `0H ~ 0FFH`。
该值可以以二进制、八进制、十进制或者十六进制的数字来设定。如果数值超过此范围或指定的不是数字，会产生终止错误。
- 设定地址区的起始地址作为 *起始地址* 执行存储。
可以用于设定值的范围是 `0H` 到 `0FFEFFH`。
该值可以以二进制、八进制、十进制或者十六进制的数字来设定。如果数值超出此范围，或者不是数值的值设定则产生异常中断错误。如果起始地址省略，假设 `0` 地址设定。
- 设定地址区的大小作为 *大小* 执行存储。
允许设定数值的范围为 `1H` 到 `0FFF00H`。
该值可以以二进制、八进制、十进制或者十六进制的数字来设定。如果数值超过此范围或指定的不是数字，会产生终止错误。当 *起始地址* 设定时，*大小* 设定不能省略。
- 如果 *起始地址* 设定，而 *大小* 设定省略，目标转换器将假设内部 `ROM` 的范围已设定来执行数据处理。
- 如果 `-u` 选项和 `-nu` 选项同时设定，最后设定的选项有效。
- 如果 `-no` 选项设定，`-u` 和 `-nu` 选项无效。
- 两个或两个以上地址范围不能通过 `-u` 选项设定。
- *起始地址* 和 *大小* 通过 `-u` 选项的设定格式和设定的解释如下：

(1) `-u` 填充值

如果目标设备包含内部 `ROM`，内部 `ROM` 范围

(2) `-ufilling-value, size`

从地址 `0` 到 “`size - 1`”

(3) `-ufilling-value, start-address, size`

从 *start-address* 到 “*start-address + size - 1*”

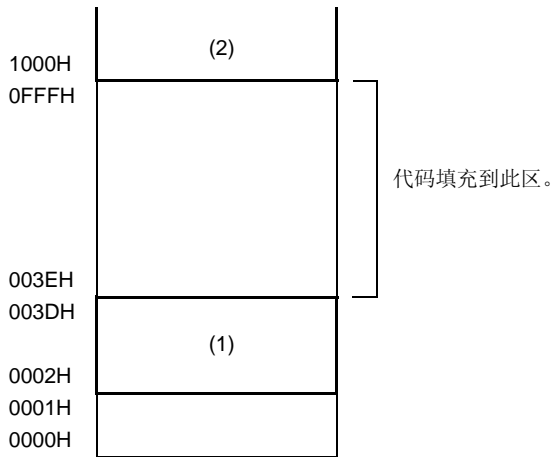
[Example of use]

- 使用代码填写 hex 文件目标没有输出的地址区。
假定下述 hex 文件存在。在这种情况下，代码不能写入到地址区的 003EH ~ 0FFFH。

```

: 020000000200FC
: 10002002B41000BFC80FE2B40000944F7083A20EC ; (1)
: 100012001A6720FE2822006521FED350D25014FE1A ; (1)
: 10002200B900059F2835002431B900059F28350005 ; (1)
: 0C003200242156AF0A8302A807A830560C
: 01000003B5D0d0026A3... ; (2)
: 1010100024A5F622B667... ; (2)
:
: 00000001FF

```



将 00H 填充到地址区 003EH ~ 0FFFH，描述如下：

```
C>oc78k0r k0r.lmf -u00h,003eh,0fc2h
```

错误表文件输出指定

错误表文件输出指定如下：

- `-e/-ne`

`-e/-ne`

[Description format]

```
-e[output-file-name]  
-ne
```

- 省略时解释
-ne

[功能]

- `-e` 选项指定错误表文件的输出。
这也同时指定输出文件名和路径。
- `-ne` 选项使 `-e` 选项失效。

[Application]

- 使用 `-e` 选项设定错误列表文件的输出位置或改变其文件名。

[说明]

- 当 `-e` 选项设定时输出文件名省略，输出文件名将为 “*input-file-name.eoc*”。
- 如果当 `-e` 选项设定时驱动器名省略，错误列表文件将输出到当前驱动器。
- 如果 `-e` 选项和 `-ne` 选项同时设定，最后设定的选项有效。

[Example of use]

- 创建错误列表文件 `k0r.eoc`，描述如下：

```
C>oc78k0r k0r.lmf -ek0r.eoc
```

参数文件说明

参数文件选项说明如下。

-f

-f

[Description format]

```
-f file-name
```

- 省略时解释
选项或输出文件名仅能从命令行输入。

[功能]

- f 选项从指定文件输入选项和输入文件名。

[Application]

- 当启动目标转换器的信息不能适用于命令行时，使用 -f 选项
- 当每次执行目标转换重复设定选项时，在参数文件中描述选项和设定 -f 选项。

[说明]

- 如果文件名省略，产生异常中断错误。
- 参数文件嵌套是禁止的。如果 -f 选项在参数文件范围内设定，则产生异常中断错误。
- 能在参数文件中描述的字符数是没有限制的。
- 使用空格键、tab 键或者换行码（LF）分离选项或输入文件名。
- 参数文件中的选项和输入文件名可以在命令行设定的参数文件位置扩展。
- 最后设定的扩展选项有效。
- 下述字符“;”或“#”都假设为注释，相当于换行码（LF）或 EOF。
- 如果两个或两个以上 -f 选项设定时产生异常中断错误。

[Example of use]

- 使用参数文件（78k0.poc）执行目标转换。
参数文件（78k0.poc）的内容如下。

```
; parameter file  
k0r.lmf -osample.hex  
-ssample.sym -r
```

从命令行输入下述内容。

```
C>oc78k0r k0r.lmf -f78k0r.poc
```

Hex 格式设定

hex 格式设定选项如下。

- [-ki/-kie/-kt/-km/-kme](#)

-ki/-kie/-kt/-km/-kme**[Description format]**

```
-ki
-kie
-kt
-km
-kme
```

- 省略时解释
- kie

[功能]

- 这些选项设定了要输出 hex 文件的格式。

[Application]

- 使用这些选项从“Intel 标准”，“Intel 扩展”，“扩展 tektronix”，“Motorola S 类型（标准地址）”和“Motorola S 类型（32 位地址）”中设置输出的 hex 文件格式。

[说明]

- 本章节描述各个选项。

| 选项 | Hex 格式 | 范围 |
|------|-----------------------|---------------------------|
| -ki | Intel 标准 | 0H ~ FFFFH（上限为 64 KB） |
| -kie | Intel 扩展 | 0H ~ FFFFFH（上限为 1 MKB） |
| -kt | 扩展 tektronix | 0H ~ FFFFFFFFH（上限为 4 GB） |
| -km | Motorola S 类型（标准地址） | 0H ~ FFFFFFFFH（上限为 16 MB） |
| -kme | Motorola S 类型（32 位地址） | 0H ~ FFFFFFFFH（上限为 4 GB） |

[Example of use]

- 设定 hex 文件作为 Motorola S 格式（标准地址）输出，描述如下：

```
C>oc78k0r k0r.lmf -km
```

设备文件查找路径说明

设备文件查找路径选项说明如下。

-y

-y

[Description format]

`-ypath-name`

- 省略时解释

依照下述次序确定读取设备文件的路径。

- (1) 在设备文件的安装文件中的注册路径
- (2) 该路径用于启动 **oc78k0.exe**
- (3) 当前文件夹
- (4) 环境变量 **PATH**

[功能]

-y 选项从指定路径中读取设备文件。

[Application]

- 使用 -y 选项设定设备文件存在的路径。

[说明]

- 如果不是路径名的内容在 -y 选项后设定，则会产生异常中断错误。
- 如果路径名在 -y 选项后省略，则会产生异常中断错误。
- 依照下述次序确定读取设备文件的路径。

- (1) 由 -y 选项设定的路径
- (2) 在设备文件的安装文件中的注册路径
- (3) 该路径用于启动 **OC78K0**
- (4) 当前文件夹
- (5) 环境变量 **PATH**

[Example of use]

- 设定设备文件为文件夹 **C:\78k0\dev**，描述如下：

```
C>oc78k0r k0r.lmf -yC:\78k0r\dev
```

- 设定设备文件的路径为 C:\Program Files\NEC Electronics Tools\device files，描述如下：

```
C>oc78k0r k0r.lmf -y" C:\Program Files\NEC Electronics Tools\device files"
```

用于内嵌 Flash 存储产品的文件分离输出设定

用于内嵌 Flash 存储产品的文件分离输出设定如下。

- `-zf`

-zf

[Description format]

`-zf`

- 省略时解释
没有分离输出

[功能]

- 使用 `-zf` 选项将文件划分成两个部分：一个用于引导区，一个用于其他区。

[说明]

- 当设定链接到内嵌 Flash 存储产品的引导区 ROM 程序时，将该选项添加到将文件划分成几个分离十六进制格式文件（其中文件用于引导区，另用于其他区。）的操作。
- 如果 `-zf` 选项设定，在引导区 ROM 程序分区输出的文件类型是“`hxb`”，在其他程序分区的输出文件类型是“`hxf`”。

注意事项 不要为不具有 Flash 存储区自编程函数的设备设定该选项。

[Example of use]

- 划分 hex 文件为分离文件：用于引导区的 `k0r.hxb` 和用于其他区的 `k0r`，描述如下：

```
C>oc78k0r k0r.lmf -zf
```

帮助说明

帮助选项如下。

--

--

[Description format]

--

- 省略时解释
无显示

[功能]

- 选项在显示器上输出帮助消息。

[Application]

- 帮助消息是目标转换选项说明的列表。
当执行目标转换时参阅这些内容。

[说明]

- 当 -- 选项设定时，所有其他选项无效。

注意事项 该选项不能通过 **CubeSuite** 设定。

[Example of use]

- 在显示器上输出帮助消息，描述如下：

```
C>oc78k0r --
```

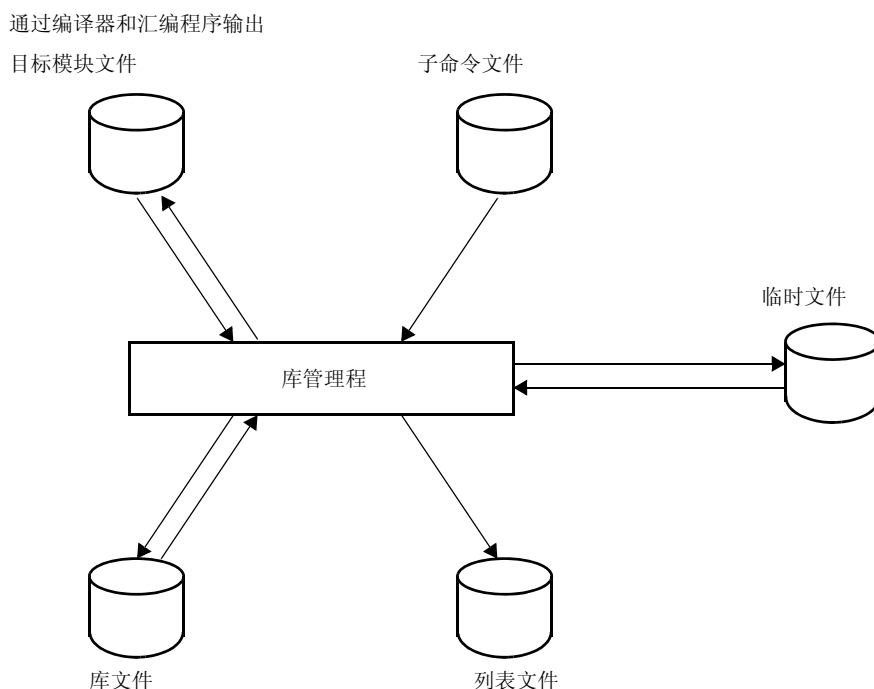
```
78K0R Object Converter Vx.xx [xx xxx xx]
  Copyright(C) NEC Electronics Corporation xxxx

usage : oc78k0r [option[...]] input-file [option[...]]
The option is as follows ([ ] means omissible).
-ffile           :Input option or input-file name from specified file.
-o[file]/-no     :Create HEX module file [with specified name] / Not.
-s[file]/-ns     :Create symbol table file [with specified name] / Not.
-e[file]/-ne     :Create the error list file [with the specified name] / Not.
-r/-nr          :Sort HEX object by address / Not.
-uvalue[, [start],size]/-nu :Fill up HEX object with specified value / Not.
-kkind          :Select hex format. I;intel format IE;intel extend format
                T;tex format M;s format ME;s-32bit format
-ydirectory     :Set device file search path.
-zf             :Create boot hex module file (HXB), and flash hex module file(HXF).
--             :Show this message.
DEFAULT ASSIGNMENT: -o -s -r -u0ffh
```

B.5 库管理程序

库管理程序编辑 CA78K0 目标模块文件和在模块单元中的库文件。它也输出列表文件。如果出现目标转换器错误，将显示错误消息用以说明导致错误的原因。

图 B-33. 库管理程序的 I/O 文件



B.5.1 I/O 文件

库管理程序的 I/O 文件显示如下。
详细的输出文件见 "3.5 库管理程序"。

表 B-24. 库管理程序的 I/O 文件

| 类型 | 文件名 | 说明 | 默认文件类型 |
|--------|--------|--|----------------------------|
| 输入文件 | 子命令文件 | - 包含执行命令参数和子命令的文件（用户创建文件） | None |
| 输出文件 | 列表文件 | - 文件包含库文件信息输出的结果 | .lst |
| I/O 文件 | 目标模块文件 | - 通过编译器和汇编程序输出目标模块文件 | .rel |
| | 库文件 | - 文件用来输入库文件以及通过库管理程序输出，并且更新内容 | .lib |
| | 临时文件 | - 在形成库时文件通过库管理程序自动建立在库管理程序的执行结束时，临时文件删除。 | Lbxxxxx.\$y (y = 1 ~ 6) |

B.5.2 功能

(1) 库模块的构成

汇编程序和链接器在它们输出每个模块时间里文件。

这就意味着如果大量模块建立，那么大量的文件也会生成。因此汇编程序具备收集多个目标模块在单个文件中的功能。此功能叫做模块库构成，并且文件由库组织而成这叫做库文件。

库文件能输入到链接器中。通过建立库文件，其包含的模块公用于许多程序，在运行模块编程时，用户可以方便文件管理和操作有效和容易。

(2) 库文件编辑

库管理程序包括以下关于库文件的编辑功能。

- 库文件的附加模块
- 从库文件中删除模块
- 在库文件中替换模块
- 从库文件中取回模块

关于这些功能的详细信息，请参阅“[B.5.5 子命令](#)”。

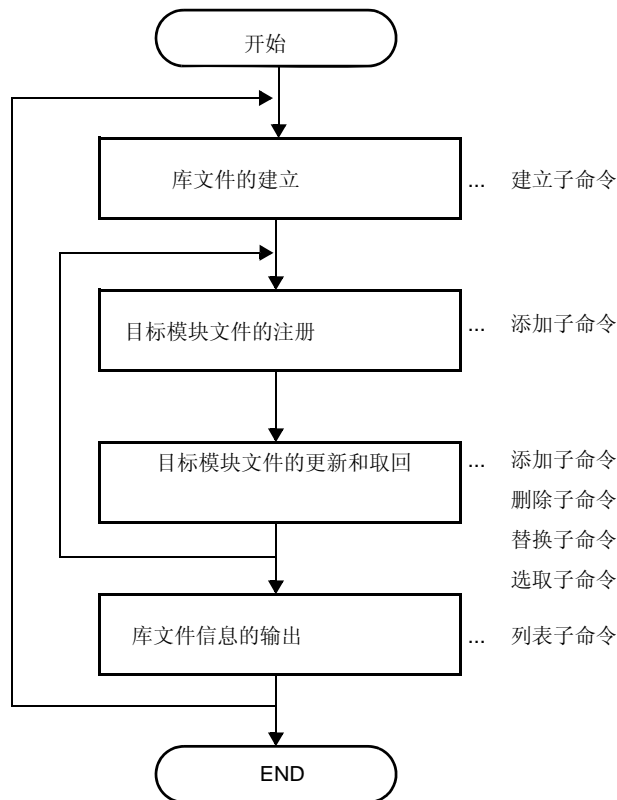
(3) 库文件信息的输出

库管理程序包含的功能有编辑和输出以下存储在库文件中的信息条目。

- 模块名称
- 已建程序
- 注册日期
- 更改日期
- PUBLIC 符号信息

注意事项 库管理程序执行功能 (2) 和 (3) 在应用子命令中做说明。在顺序的解释每条子命令时，库管理程序执行进行。子命令操作请参阅“[B.5.5 子命令](#)”。

建立库文件的一般步骤如下。



B.5.3 操作方法

(1) 库管理程序启动

以下两种方法可用于启动库管理程序。

(a) 从命令行启动

```
X:[path-name]>lb78k0[Δoption] ...
```

| | |
|--------|--|
| X | 当前驱动器名称 |
| 路径名称 | 当前文件夹名 |
| lb78k0 | 库管理程序的命令名 |
| 选项 | 键入库管理程序操作的详细指令。 当设定两个或更多创建库选项时，使用空格将选项分类。库创建选项不区分大写和小写字符。详细的库创建选项见 "B.5.4 选项"。 带有空格的路径附在一对双引号 (") 中。 |

示例 限定每页行数 20 行且每行字符数 80 个，说明如下：

```
C>lb78k0 -l120 -lw80
```

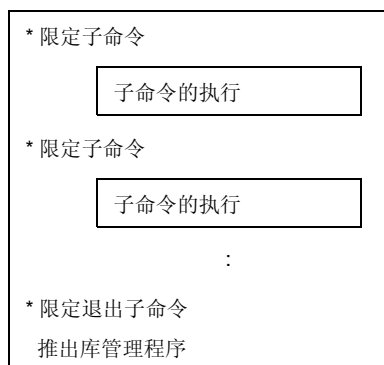
当库管理程序启动时，执行启动消息出现在显示器上。

```
78K0 Librarian Vx.xx [xx xxx xxxxx]
    Copyright(C) NEC Electronics Corporation xxxxx
*
```

在星号标记 (*) 之后，指定库管理程序子命令。

```
*create k0.lib
*add k0.lib k0main.rel k0sub.rel
*exit
```

当完成输入子命令时，每条子命令进程。当子命令执行完成时，"*" 再次在屏幕上出现并且库管理程序等待下条子命令的进入。库管理程序重复此操作直到退出子命令的进入。



最多可以设定每行 128 个字符。

如果所有必须的操作数数据没有放置在一行上，在下一行使用 "&" 继续设定。设定可以继续不超过 15 行。

(b) 从子命令文件启动

子命令文件就是库管理程序子命令存储文件。

在库管理程序启动时如果子命令文件没有指定，则复合子命令必须在 "*" 出现后指定。通过建立子命令文件，这些复合子命令文件可以立即执行。

当相同子命令在每次库信息显示重复指定的话，子命令文件可以使用。

在使用子命令文件时，说明 "<" 在此文件名称之前。

通过子命令文件启动库管理程序如下：

```
x>lb78k0Δ<subcommand-file-name[Δoption] ...
```

| | |
|----------------------|----------------|
| < | 在指定子命令文件时确认已添加 |
| subcommand-file-name | 子命令存储文件 |

备注 使用编辑器生成子命令文件。

子命令文件内容格式如下：

```
Subcommand-name operand-data
Subcommand-name operand-data
:
exit
```

- 当重复子命令，标识 "&" 将在每行行尾持续显示。
 - 从每行行尾用分号符号来标识将作为注释，且将不通过库管理程序命令来解释。
 - 如果在子命令文件中的最后一条子命令不是退出子命令，则库管理程序将自动解释指定的退出子命令。
 - 库管理程序从子命令文件中读取子命令并执行它们。
- 库管理程序在完成子命令文件中的所有子命令后退出。

示例 使用编辑器创建子命令文件 k0.plk，然后启动库管理程序。

```
; library creation command
create k0r.lib
add k0r.lib k0rmain.rel &
k0rsub.rel
;
exit
```

```
C>lb78k0r <k0r.slb
```

(2) 执行开始和结束消息**(a) 执行开始消息**

当库管理程序启动时，执行启动消息出现在显示器上。

```
78K0 Librarian Vx.xx [xx xxx xxxx]
Copyright(C) NEC Electronics Corporation xxxx
*
```

(b) 执行结束消息

库管理程序不会输出执行结束消息。在所有处理完成后用户键入退出子命令时，库管理程序返回控制到主操作系统中。

```
*create k0.lib
*add k0.lib k0main.rel k0sub.rel
*exit
```

如果库管理程序检测到在库管理过程中有致命错误，导致库管理不能进行，则库管理程序输出在显示器上输出消息，同时将控制权返回给主操作系统。

示例 指定的建库选项不存在。

```
C>lb78k0 -a
```

```
78K0 Librarian Vx.xx [xx xxx xxxx]
    Copyright(C) NEC Electronics Corporation xxxx
RA78K0 error F5018 : Option is not recognized '-z'
Usage : LB78K0 [options]
```

在上述例子中，不存在建库选项设定。出错产生并且库管理程序中止库管理执行。

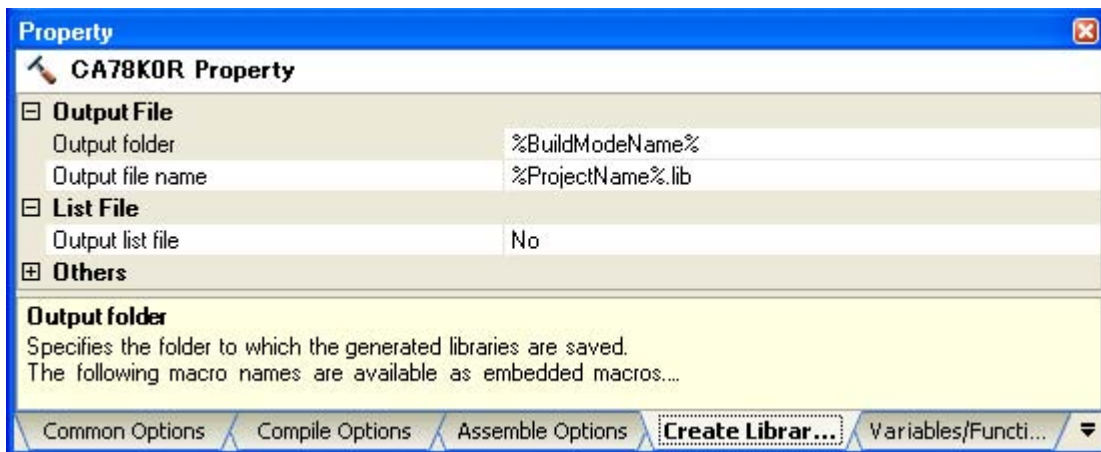
(3) 在 CubeSuite 中设置选项

本章节描述了通过 CubeSuite 怎样设置建库选项。

通过 CubeSuite 的 [工程树面板](#)，选择 Build Tool 节点。然后通过 [View] 菜单选择 [Property]。打开 [属性面板](#)。再选择 [\[Create Library Options\]](#) 标签。

可以通过此属性页给各种建库选项设置必要的值。

图 B-34. 属性面板: [Create Library Options] 标签



B.5.4 选项**(1) 类型**

建库选项是库管理程序的执行的详细指令。

建库选项的类型和注解如下。

表 B-25. 创建库选项

| 分类 | 选项 | 说明 |
|------------|------|------------------|
| 列表文件格式设定 | -lw | 改变列表文件中每行字符数。 |
| | -ll | 在列表文件的每页打印变更的行数。 |
| | -lf | 在链接列表文件的最后插入换页码。 |
| | -nlf | |
| 临时文件创建路径设定 | -t | 在指定的路径建立临时文件。 |
| 帮助说明 | -- | 在屏幕上输出帮助信息。 |

列表文件格式设定

列表文件格式指定如下：

- `-lw`
- `-ll`
- `-lf/-nlf`

`-lw`

[Description format]

```
-lw[number-of-characters]
```

- 省略时解释
- `-lw132`（显示器输出为 80 个字符）

[功能]

- `-lw` 选项指定表文件中每行的字符数。

[Application]

- 使用 `-lw` 选项改变列表文件每行的字符数。

[说明]

- `-lw` 参数可指定的字符数范围为 72~260，（显示器输出为 80 个字符）。
- 如果数值超过此范围或指定的不是数字，会产生终止错误。
- 如果字符数省略，那么会设定为 132。然而当列表文件显示输出是，则会设定为 80。
- 指定的字符数不包含终止符（CR，LF）。
- 如果列表子命令没有设定，`-lw` 选项无效。
- 如果两个或以上 `-lw` 选项设定，后设定的选项有效。

[Example of use]

- 设定 80 作为列表文件每行的字符数，描述如下：

```
C>lb78k0 -lw80
```

-ll

[Description format]

```
-ll[number-of-lines]
```

- 省略时解释
- ll0 (No page breaks)

[Function]

- ll 选项指定列表文件中每页行数。

[Application]

- 使用 -ll 选项改变列表文件中每页行数。

[说明]

- ll 选项可指定的行数范围为 0 和 20~32767。
- 如果数值超过此范围或指定的不是数字，会产生终止错误。
- 如果没有指定，默认值为 0。
- 如果行数指定为 0，将不会生成分页符。
- 如果列表子命令没有设定，-ll 选项无效。
- 如果两个或以上 -ll 选项设定，后设定的选项有效。

[Example of use]

- 设定 20 作为列表文件每页的行数，描述如下：

```
C>lb78k0 -ll20
```

-lf/-nlf

[Description format]

```
-lf  
-nlf
```

- 省略时解释
-nlf

[功能]

- lf 选项在列表文件的最后插入换页码 (FF)。
- nlf 选项禁止 -lf 选项。

[Application]

- 如果在打印的列表文件内容后添加分页符，使用 -lf 选项插入换页码。

[说明]

- 如果列表子命令没有设定，-lf 选项无效。
- 如果同时指定了 -lf 和 -nlf 选项，则最后指定的选项有效。

[Example of use]

- 在列表文件的最后插入换页码。

```
C>lb78k0r -lf
```

临时文件创建路径设定

临时文件创建路径设定选项如下。

-t

-t

[Description format]

`-t path-name`

- 省略时解释
环境变量 **TMP** 指定的路径
则为当前路径，如果环境变量 **TMP** 不设定。

[功能]

- t 选项设定创建临时文件的路径。

[Application]

- 使用 -t 选项设定临时文件的创建位置。

[说明]

- 仅路径可以设定作为路径名。
- 路径名不能省略。
- 即使之前创建的临时文件存在，如果文件没有保护，它将重写。
- 只要需要的存储大小可用，临时文件将在存储器中扩展。
如果没有足够的内存分配，则临时文件的内容将为写入磁盘内。这样临时文件可以稍后通过已保存的磁盘文件来访问。
- 当库队列结束时，临时文件删除。当通过 [CTRL] + [C] 按键退出库队列时，临时文件也删除。
- 依照下述次序确定创建临时文件的路径。

(1) 由 -t 选项设定的路径

(2) 通过环境变量 **TMP** 设定的路径（当 -t 选项省略时）

(3) 当前路径（当 **TMP** 没有设定时）

注意事项 当（1）或（2）设定时，如果临时文件不能在指定的路径创建，则发生异常中断错误。

[Example of use]

- 输出临时文件到文件夹 **C:\tmp**，描述如下：

```
C>lb78k0 -tC:\tmp
```

- 从 **C:\Program Files\NEC Electronics Tools\include** 输出临时文件，描述如下：

```
C>lb78k0r -t" C:\Program Files\NEC Electronics Tools\temporary files"
```

帮助说明

帮助选项如下。

--

--

[Description format]

--

- 省略时解释
无显示

[功能]

- 选项在显示器上输出帮助消息。

[Application]

- 帮助消息是子命令说明的列表。当执行库管理程序时参阅这些内容。

[说明]

- 当 -- 选项设定时，所有其他选项无效。

注意事项 该选项不能通过 **CubeSuite** 设定。

[Example of use]

- 在显示器上输出帮助消息，描述如下：

```
C>1b78k0r --
```

```
78K0R Librarian Vx.xx [xx xxx xx]
  Copyright(C) NEC Electronics Corporation xxxx
+-----+
| Subcommands : create,add,delete,replace,pick,list,help,exit |
| | |
| Usage : subcommand[ option] masterLBF[ option] transaction[ option] |
| | |
|           transaction ::= OMFname |
|                           LBFname[ (modulename[,...])] |
| | |
| <create > : create masterLBF[ transaction] |
| <add    > : add masterLBF transaction |
| <delete > : delete masterLBF(modulename[,...]) |
| <replace> : replace masterLBF transaction |
| <pick  > : pick masterLBF(modulename[,...]) |
| <list  > : list[ option] masterLBF[(modulename[,...])] |
|           option : -p = output public symbol |
|                   -np = no output public symbol |
|                   -o filename = specify output file name |
| <help  > : help |
| <exit  > : exit |
| | |
+-----+
```

B.5.5 子命令

(1) 类型

库管理程序操作中子命令的详细指令。

子命令的类型和注解如下。

表 B-26. 子命令

| 子命令名称 | 缩略。 | 说明 |
|---------|-----|-----------------|
| 创建 | c | 建立新库文件。 |
| add | a | 添加模块到库文件中。 |
| 删除 | d | 从库文件中删除模块。 |
| replace | r | 替换库文件中的模块为另一模块。 |
| 拾取 | p | 从库文件中恢复模块。 |
| 列表 | l | 在库文件中输出模块信息。 |
| 帮助 | h | 在屏幕上输出帮助信息。 |
| 退出 | e | 退出库管理程序。 |

(2) 子命令文件的一般格式

```
*subcommand[Δoption]Δlibrary-file-name[Δoption]transaction[Δoption]
```

| | |
|--------------------------|--|
| <i>library-file-name</i> | 在库文件名称替换为 '!' 之前库文件名称立刻要制定。 |
| <i>transaction</i> | <i>transaction</i> = Δ <i>object-module-file-name</i> Δ <i>Dlibrary-file-name</i> [Δ(Δ <i>module-name</i> [Δ, ?c])] |

备注 子命令和选项不区分大写和小写字符。

创建

[Description format]

```
createΔlibrary-file-name[Δtransaction]
cΔlibrary-file-name[Δtransaction] ; abbreviated form
```

[功能]

- 用建立子命令建立新的库文件。

[说明]

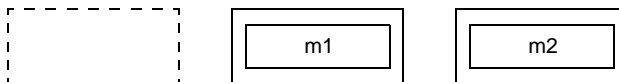
- 已建立库文件的大小为 0。
- 当处理设定，在库文件建立时，模块注册。
- *library-file-name*:
如果指定文件已存在，它将覆盖。
- *处理*:
目标模块文件出现的公用符号与库文件中的公用符号相同时不能注册。
模块名称与库文件中的模块同名的话不能注册。
- 如果产生错误，则处理中断且库文件不能建立。

[Example of use]

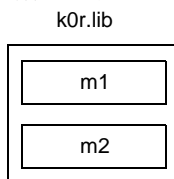
- 在库文件 (k0r.lib) 建立时注册模块 m1 和 m2, 说明如下:

```
*create k0r.lib m1.rel m2.rel
```

< 在文件建立之前 >



< 在文件建立之后 >



add**[Description format]**

```
addΔlibrary-file-nameΔtransaction
aΔlibrary-file-nameΔtransaction ; Abbreviated form
```

[功能]

- 添加子命令是添加模块到库文件中。

[说明]

- 模块可以添加到库文件中即使没有模块储存在库中。
- 如果模块以相同名称添加到已存在的库文件中，则会产生中止性错误。
- 如果该模块以相同公用符号添加到库文件中，则会产生中止性错误。

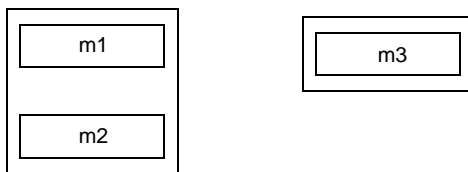
[Example of use]

- To add module m3 to the library file (k0r.lib), describe as:

```
*add k0r.lib m3.rel
```

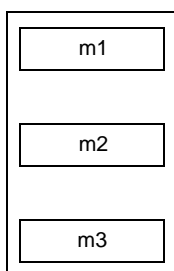
< 在模块添加之前 >

k0r.lib



< 在模块添加之后 >

k0r.lib



删除

[Description format]

```
delete library-file-name( module-name[ , ... ] )
d library-file-name( module-name[ , ... ] ) ; Abbreviated form
```

[功能]

- 删除子命令是从库文件中删除模块。

[说明]

- 如果在库文件中指定的模块不存在，则会产生错误。
- 如果错误产生，处理中断且库文件的环境不会改变。

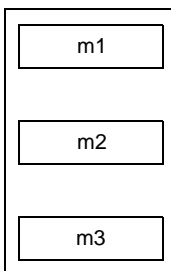
[Example of use]

- 删除模块 m1 到 m3 从库文件 (k0r.lib) 中，说明如下：

```
*delete k0r.lib ( m1.rel , m3.rel )
```

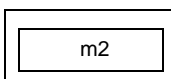
< 在模块删除之前 >

k0r.lib



< 在模块删除之后 >

k0r.lib



replace**[Description format]**

```
replace $\Delta$ library-file-name $\Delta$ transaction
r $\Delta$ library-file-name $\Delta$ transaction ; Abbreviated form
```

[功能]

- 替换子命令替换在库文件中的已存在的模块以及在另外目标模块文件中的模块。

[说明]

- 如果没有模块以相同名称替换到已存在的库文件中，则会产生中止性错误。
- 如果该模块以相同公用符号替换到库文件中，则会产生中止性错误。
- 已替换的目标模块文件名必须以相同的文件名注册到库文件中。
- 如果错误产生，处理中断且库文件的环境不会改变。

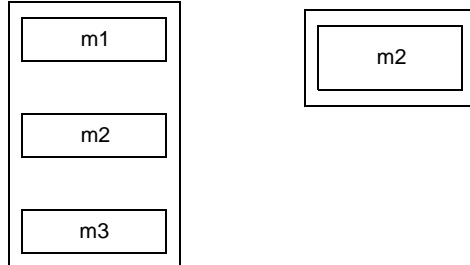
[Example of use]

- 替换模块 m2 到库文件 (k0r.lib), 说明如下:

```
*replace k0r.lib m2.rel
```

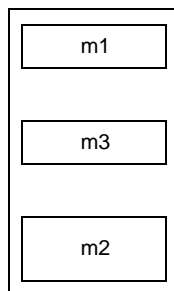
< 在模块替换之前 >

k0r.lib



< 在模块替换之后 >

k0r.lib



因为新模块 (m2) 的注册在库文件中的模块 (m2) 删除后，所以在库文件中的 m2 排在队列中的最后。

拾取

[Description format]

```
pickΔlibrary-file-nameΔ(Δmodule-name[Δ, ... ]Δ)
pΔlibrary-file-nameΔ(Δmodule-name[Δ, ... ]Δ) ; Abbreviated form
```

[功能]

- 拾取子命令是从存在的库文件中恢复指定的模块。

[说明]

- 恢复模块在其注册在库文件中后将成为带有文件名称的目标模块文件。
- 如果在库文件中指定的模块不存在，则会产生错误。
- 如果错误产生，则处理中断。然而，在两个或以上的模块指定时如果错误产生，则恢复的模块在模块导致错误之前恢复有效并保存在磁盘上。

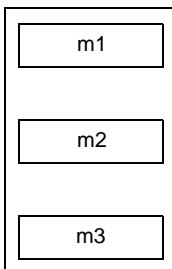
[Example of use]

- 恢复模块 m2 从库文件 (k0r.lib) 中，说明如下：

```
*pick k0r.lib ( m2.rel )
```

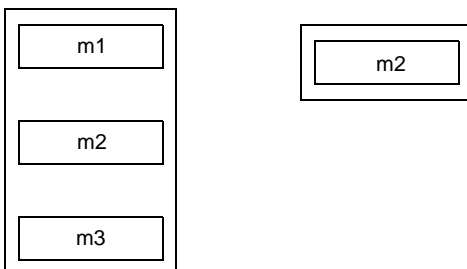
< 在模块恢复之前 >

k0r.lib



< 在模块恢复之后 >

k0r.lib



列表

[Description format]

```
list[Δoption]Δlibrary-file-name[Δ(Δmodule-name[Δ, ... ]Δ)]
l[Δoption]Δlibrary-file-name[Δ(Δmodule-name[Δ, ... ]Δ)]      ; Abbreviated form
  option : -public/-nopublic
          : -oΔfile-name
```

[功能]

- 列表子命令是在库文件中输出模块信息。

[说明]

- 两个或以上的选项能设定。选项不区分大写和小写字符。
- -o:
 - 如果输出文件名称省略，则产生错误。
 - 如果文件类型省略，那么"输入文件名称.lst"设定。
- -public/-nopublic:
 - 它也能设定为 -p/-np。
 - 公用设置公用符号信息的输出。
 - nopublic 选项使 -public 选项禁止。
 - 如果 -public 选项和 -nopublic 选项同时设定，最后设定的选项具有优先级。

[Example of use]

- 在 (k0r.lib) 中输出模块信息到列表文件 (k0r.lst)。此时，设定 -p 选项为了输出公用符号信息。

```
*list -p -ok0r.lst k0r.lib
```

列表文件 (k0r.plv) 的内容如下。

```
78K0R librarian Vx.xx                                DATE : xx xxx xx  PAGE  1

LIB-FILE NAME : k0r.lib                            (xx xxx xxxxx)

0001  k0rmain.rel      (xx xxx xxxxx)

      MAIN      START

NUMBER OF PUBLIC SYMBOLS :                2

0002  k0rsub.rel      (xx xxx xxxxx)

      CONVAH

NUMBER OF PUBLIC SYMBOLS :                1
```


退出

[Description format]

```
exit  
e      ; Abbreviated form
```

[功能]

- 退出子命令退出库管理程序。

[说明]

- 使用子命令退出库管理程序。

[Example of use]

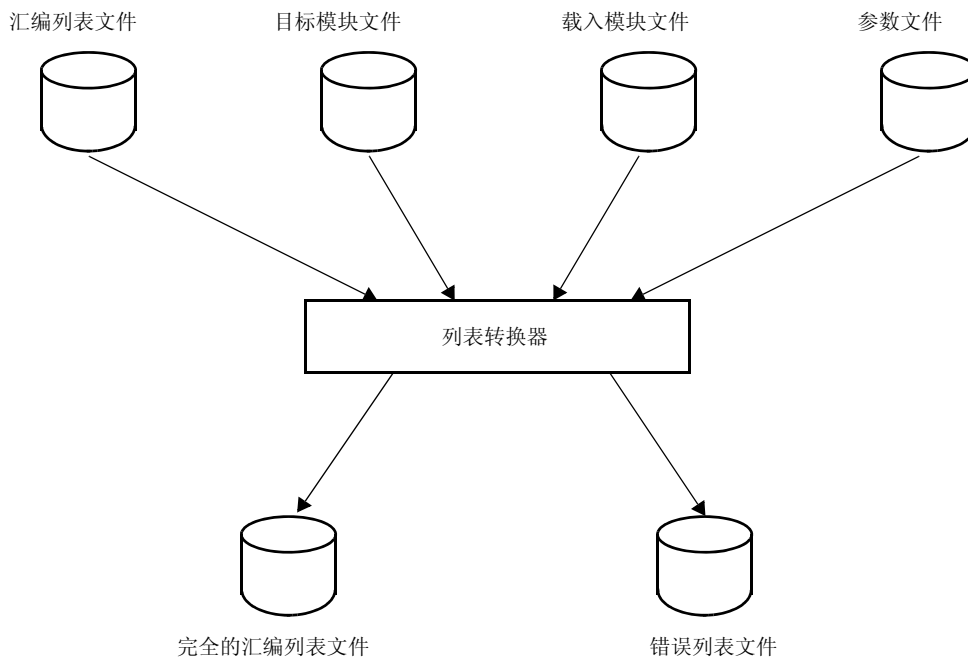
- 退出库管理程序，说明如下：

```
*exit
```

B.6 列表转换器

列表转换器输入汇编列表文件以及通过汇编程序输出目标模块文件并且通过链接器输出载入模块文件。它嵌入实地址在浮动地址里并在输入文件中做标识以及输出绝对汇编列表。

图 B-35. 列表转换器的 I/O 文件



B.6.1 I/O 文件

列表转换器的 I/O 文件如下所列：

表 B-27. 列表转换器的 I/O 文件

| 类型 | 文件名 | 说明 | 默认文件类型 |
|------|-----------|---------------------------------------|--------|
| 输入文件 | 目标模块文件 | - 包含与机器语言和机器语言地址定位相关的重定位信息与符号信息的二进制文件 | .rel |
| | 汇编列表文件 | - 包含如汇编表和交叉参考表等汇编信息的文件 | .prn |
| | 载入模块文件 | - 目标代码的二进制映像文件作为链接操作的结果输出。 | .lmf |
| | 参数文件 | - 包含执行命令参数的文件（用户创建文件） | .plv |
| 输出文件 | 完全的汇编列表文件 | - 列表文件嵌入实地址在浮动地址里并在输入文件中做标识 | .p |
| | 错误列表文件 | - 文件包含在转换列表过程中生成的错误信息。 | .elv |

B.6.2 功能

(1) 解决汇编程序的缺陷（浮动汇编程序）

列表转换器提通过在汇编列表文件中嵌入位置和目标代码，提供了解决浮动汇编程序的缺陷的方法。

- 绝对汇编列表通过列表转换器完全支持地址应用到实际程序运行中。
- 外部标识的实际值嵌入在列表中。
- 浮动值作为实际值嵌入在列表中。
- 在符号表和交叉引用列表中符号值的实际值嵌入在列表中。

例如：绝对汇编列表文件可以通过显示在下面的列表转换器来获取。

示例 1. 重定位数据以如下显示嵌入。

- 汇编列表

| | | | |
|----|----|-------|---|
| 22 | 22 | ----- | CSEG |
| 23 | 23 | 00000 | START : |
| 24 | 24 | | |
| 25 | 25 | | ; chip initialize |
| 26 | 26 | 00000 | RCBF80000 MOVW SP , #_@STBEG |
| 27 | 27 | | |
| 28 | 28 | 00004 | CD201A MOV HDTSA , #1AH |
| 29 | 29 | 00007 | 3620FE MOVW HL , #LOWW (HDTSA) ; set hex 2-code data in HL register |
| 30 | 30 | | |
| 31 | 31 | 0000A | RFD0000 CALL !CONVAH ; convert ASCII <- HEX |
| 32 | 32 | | ; output BC-register <- ASCII code |
| 33 | 33 | 0000D | 3421FE MOVW DE , #LOWW (STASC) ; set DE <- store ASCII code table |
| 34 | 34 | 00010 | 63 MOV A , B |
| 35 | 35 | 00011 | 99 MOV [DE] , A |
| 36 | 36 | 00012 | A5 INCW DE |
| 37 | 37 | 00013 | 62 MOV A , C |
| 38 | 38 | 00014 | 99 MOV [DE] , A |
| 39 | 39 | | |
| 40 | 40 | 00015 | EF FE BR \$\$ |
| 41 | 41 | | |
| 42 | 42 | | END |

- 完全的汇编列表

| | | | |
|----|----|-------|---|
| 22 | 22 | ----- | CSEG |
| 23 | 23 | 000D2 | START : |
| 24 | 24 | | |
| 25 | 25 | | ; chip initialize |
| 26 | 26 | 000D2 | RCBF820FE MOVW SP , #_@STBEG |
| 27 | 27 | | |
| 28 | 28 | 000D6 | CD201A MOV HDTSA , #1AH |
| 29 | 29 | 000D9 | 3620FE MOVW HL , #LOWW (HDTSA) ; set hex 2-code data in HL register |
| 30 | 30 | | |
| 31 | 31 | 000DC | RFDE900 CALL !CONVAH ; convert ASCII <- HEX |
| 32 | 32 | | ; output BC-register <- ASCII code |
| 33 | 33 | 000DF | 3421FE MOVW DE , #LOWW (STASC) ; set DE <- store ASCII code table |
| 34 | 34 | 000E2 | 63 MOV A , B |
| 35 | 35 | 000E3 | 99 MOV [DE] , A |
| 36 | 36 | 000E4 | A5 INCW DE |
| 37 | 37 | 000E5 | 62 MOV A , C |
| 38 | 38 | 00014 | 99 MOV [DE] , A |
| 39 | 39 | | |
| 40 | 40 | 000E7 | EF FE BR \$\$ |
| 41 | 41 | | |
| 42 | 42 | | END |

2. 目标代码以如下显示嵌入。

- 汇编列表

```

22 22 ----- CSEG
23 23 00000 START :
24 24
25 25 ; chip initialize
26 26 00000 RCBF80000 MOVW SP , #_@STBEG
27 27
28 28 00004 CD201A MOV HDTSA , #1AH
29 29 00007 3620FE MOVW HL , #LOWW ( HDTSA ) ; set hex 2-code data in HL register
30 30
31 31 0000A RFD0000 CALL !CONVAH ; convert ASCII <- HEX
32 32 ; output BC-register <- ASCII code
33 33 0000D 3421FE MOVW DE , #LOWW ( STASC ) ; set DE <- store ASCII code table
34 34 00010 63 MOV A , B
35 35 00011 99 MOV [ DE ] , A
36 36 00012 A5 INCW DE
37 37 00013 62 MOV A , C
38 38 00014 99 MOV [ DE ] , A
39 39
40 40 00015 EFFE BR $$
41 41
42 42 END

```

- 完全的汇编列表

```

22 22 ----- CSEG
23 23 000D2 START :
24 24
25 25 ; chip initialize
26 26 000D2 RCBF820FE MOVW SP , #_@STBEG
27 27
28 28 000D6 CD201A MOV HDTSA , #1AH
29 29 000D9 3620FE MOVW HL , #LOWW ( HDTSA ) ; set hex 2-code data in HL register
30 30
31 31 000DC RFDE900 CALL !CONVAH ; convert ASCII <- HEX
32 32 ; output BC-register <- ASCII code
33 33 000DF 3421FE MOVW DE , #LOWW ( STASC ) ; set DE <- store ASCII code table
34 34 000E2 63 MOV A , B
35 35 000E3 99 MOV [ DE ] , A
36 36 000E4 A5 INCW DE
37 37 000E5 62 MOV A , C
38 38 00014 99 MOV [ DE ] , A
39 39
40 40 000E7 EFFE BR $$
41 41
42 42 END

```

B.6.3 操作方法

(1) 列表转换器的启动

以下两种方法可以用于启动列表转换器。

(a) 从命令行启动

```
X:[path-name]>lc78k0r[Δoption] ... input-file-name[Δoption] ... [Δ]
```

| | |
|-----------------|--|
| X | 当前驱动器名称 |
| 路径名 | 当前文件夹名 |
| lc78k0 | 列表转换器的命令名 |
| 选项 | 输入列表转换器操作的详细指令。 当设定两个或更多列表转换选项时，使用空格将选项分类。列表转换选项不区分大写和小写字符。 有关列表转换选项的详细内容，参阅“B.6.4 选项”。 带有空格的路径附在一对双引号（" "）中。 |
| input-file-name | 汇编列表原名 带空格的文件名路径附在一对双引号（" "）中。 使用 ".prn" 扩展名。 |

注意事项 如果只有汇编列表原名指定在命令行中，则目标模块文件和载入模块文件原名必须与汇编列表文件一致。

文件类型也必须如下列所示。

| 文件名 | 类型 |
|--------|------|
| 目标模块类型 | .rel |
| 载入模块文件 | .lmf |

示例 如果原名与汇编列表文件 (k0rmain.prn) 和载入模块文件 (sample.lmf) 不同，描述如下以便限定载入模块文件的输入 (sample.lmf)。

```
C>lc78k0r k0rmain.prn -lsample.lmf
```

(b) 从参数文件启动

当启动列表转换器时需要的数据无法适用于命令行时，或者当相同的列表转换选项重复指定，每次列表转换被执行时，使用参数文件。

通过参数文件启动汇编程序，在命令行上设定参数文件选项（-f）。

通过参数文件启动目标转换器如下：

```
X>lc78k0r[Δinput-file-name]Δ-fparameter-file-name
```

| | |
|-------|-------------------|
| -f | 参数文件说明选项 |
| 参数文件名 | 包含启动列表转换器所需要数据的文件 |

备注 使用编辑器创建参数文件。

参数文件内容的书写规则如下：

```
[[[Δ]option[Δoption] ... [Δ]Δ]] ...
```

- 如果命令行省略输入文件名，参数文件中仅可以指定 1 个输入文件名。
- 输入文件名也可以在加在选项后。
- 在参数文件中写入所有的列表转换选项以及在命令行中定义的输出文件名。

示例 使用编辑器创建参数文件 k0r.plv，然后启动列表转换器。

```
; parameter file
k0rmain -lk0r.lmf
-ek0r.elv
```

```
C>ra78k0r -fk0rmain.pra
```

(2) 执行开始和结束消息

(a) 执行开始消息

当列表转换器启动时，执行启动消息出现在显示器上。

```
List Conversion Program for RA78K0R Vx.xx [xx xxx xxxx]
      Copyright(C) NEC Electronics Corporation xxxx

Pass1 : start ...
Pass2 : start ...
```

(b) 执行结束消息

如果没有检测到列表转换器列表转换错误，则列表转换器输出下述消息到显示器并返回控制权到主操作系统。

```
完成转换。
```

如果在列表转换执行期间列表转换器检测到致命错误致使列表转换无法继续，则列表转换器输出消息到显示器，取消列表转换并返回控制到主操作系统。

示例 设定不存在列表转换选项。

```
List Conversion Program for RA78K0R Vx.xx [xx xxx xxxx]
      Copyright(C) NEC Electronics Corporation xxxx

RA78K0R error F6018 : Option is not recognized '-a'
Please enter 'LC78K0R --', if you want help messages.
程序终止。
```

(3) 在 CubeSuite 中设置选项

CubeSuite 包括在汇编选项中的列表转换选项。

了解设置汇编选项的详细信息参阅 [\[Assemble Options\]](#) 标签 在属性面板中。

B.6.4 选项

(1) 类型

列表转换选项是列表转换器操作的详细指令。
列表转换选项的类型和注解如下。

表 B-28. 列表转换选项

| 分类 | 选项 | 说明 |
|--------------|-----|-------------------|
| 目标模块文件输入指定 | -r | 输入目标模块文件。 |
| 载入模块文件输入规格 | -l | 输入载入模块文件。 |
| 绝对汇编列表文件输出指定 | -o | 输出绝对汇编列表文件。 |
| 错误表文件输出指定 | -e | 输出错误列表文件 |
| | -ne | |
| 参数文件说明 | -f | 通过指定文件输入选项和输入文件名。 |
| 帮助说明 | -- | 在屏幕上输出帮助信息。 |

目标模块文件输入指定

目标模块文件输入设定选项如下。

-r

-r

[Description format]

```
-r[input-file-name]
```

- 省略时解释
- r[*assemble-list-file-name.rel*]

[功能]

- r 选项指定输入目标模块文件。

[Application]

- 在目标模块文件的原名与汇编列表文件的原名不一致时，即如果它的文件类型不是 ".rel"，使用 -r 选项。

[说明]

- 当致命错误产生时，绝对汇编列表文件不能输出。
- 如果只有输入文件名的原名指定，那么列表转换器将 ".rel" 作为文件类型添加到文件名中并且输入此文件。

[Example of use]

- 如果原名与汇编列表文件 (*k0rmain.prn*) 和目标模块文件 (*sample.rel*) 不同，描述如下以便限定载入模块文件的输入 (*sample.rel*)。

```
C>lc78k0r k0rmain.prn -lsample.rel
```

载入模块文件输入规格

载入模块文件输入设定选项如下。

-l

-l

[Description format]

```
-l[input-file-name]
```

- 省略时解释
- l[*assemble-list-file-name.lmf*]

[功能]

- l 选项设定载入模块文件的输入。

[Application]

- 在载入模块文件的原名与汇编列表文件的原名不一致时，即如果它的文件类型不是 ".lmf"，使用 -l 选项。

[说明]

- 当致命错误产生时，绝对汇编列表文件不能输出。
- 如果只有输入文件名的原名指定，那么列表转换器将 ".lmf" 作为文件类型添加到文件名中并且输入此文件。

[Example of use]

- 如果原名与汇编列表文件 (*k0rmain.prn*) 和载入模块文件 (*sample.lmf*) 不同，描述如下以便限定载入模块文件的输入 (*sample.lmf*)。

```
C>l c78k0r k0rmain.prn -l sample.lmf
```

绝对汇编列表文件输出指定

绝对汇编列表文件输出指定选项如下：

-o

-o

[Description format]

```
-o[output-file-name]
```

- 省略时解释
- o[assemble-list-file-name.p]

[功能]

- o 选项指定绝对汇编列表文件的输出。这也同时指定输出文件名和路径。

[Application]

- 使用 -o 选项指定输出绝对汇编列表文件的路径或改变它的文件名。

[说明]

- 如果相同的设备指定作为出错文件的文件名，则会产生中止性错误。
- 如果指定 -o 选项时省略了文件名，则输出文件名为 "assemble-list-file-name.p"。
- 如果只有输出文件名的原名指定，那么列表转换器将 ".p" 作为文件类型添加到文件名中并且输出此文件。
- 如果指定 -o 选项时省略了驱动器名，则会将输出绝对汇编列表文件到当前驱动器上。

[Example of use]

- 输出绝对汇编列表文件 (sample.p)，说明如下：

```
C>lc78k0r k0rmain.prn -osample.p -lk0r.lmf
```

错误表文件输出指定

错误表文件输出指定如下：

- `-e/-ne`

-e/-ne**[Description format]**

```
-e[output-file-name]
-ne
```

- 省略时解释
-ne

[功能]

- `-e` 选项指定错误表文件的输出。
这也同时指定输出文件名和路径。
- `-ne` 选项使 `-e` 选项失效。

[Application]

- 使用 `-e` 选项将错误消息保存为文件。

[说明]

- 如果相同的设备指定作为绝对汇编列表文件的文件名，则会产生中止性错误。
- 如果指定 `-e` 选项时省略了文件名，则输出文件名为 "`assemble-list-file-name.elv`"。
- 如果只有输出文件名的原名指定，那么列表转换器将 ".`elv`" 作为文件类型添加到文件名中并且输出此文件。
- 如果当 `-e` 选项设定时驱动器名省略，错误列表文件将输出到当前驱动器。
- 如果 `-e` 选项和 `-ne` 选项同时设定，最后设定的选项有效。

[Example of use]

- 建立出错列表文件 (`sample.elv`)，说明如下：

```
C>lc78k0r k0rmain.prn -esample.elv
```

出错列表文件 (`sample.elv`) 的内容如下。

```
RA78K0R 警告 W6701: 载入模块文件要比目标模块文件还旧 'k0main.lmf, k0main.rel'
Pass1 : start

RA78K0R 警告 W6702: 载入模块文件要比汇编模块文件还旧 'k0rmain.lmf, k0rmain.prn'

Pass2 : start
```

参数文件说明

参数文件选项说明如下。

-f

-f

[Description format]

```
-f file-name
```

- 省略时解释
选项或输出文件名仅能从命令行输入。

[功能]

- f 选项从指定文件输入选项和输入文件名。

[Application]

- 当启动列表转换器的信息不能适用于命令行时，使用 -f 选项
- 当每次执行列表转换重复设定选项时，在参数文件中描述选项和设定 -f 选项。

[说明]

- 如果文件名省略，产生异常中断错误。
- 如果只有输出文件名的原名指定，那么列表转换器将 ".plv" 作为文件类型添加到文件名中并且打开此文件。
- 参数文件嵌套是禁止的。如果 -f 选项在参数文件范围内设定，则产生异常中断错误。
- 能在参数文件中描述的字符数是没有限制的。
- 使用空格键、tab 键或者换行码 (LF) 分离选项或输入文件名。
- 参数文件中的选项和输入文件名可以在命令行设定的参数文件位置扩展。
- 最后设定的扩展选项有效。
- 如果两个或两个以上 -f 选项设定时产生异常中断错误。
- 下述字符 ";" 或 "#" 都假设为注释，相当于换行码 (LF) 或 EOF。

[Example of use]

- 使用参数文件 (k0r.plv) 执行列表转换。
参数文件 (k0r.plv) 的内容如下。

```
: parameter file
k0rmain -lk0r.lmf
-ek0r.elv
```

从命令行输入下述内容。

```
C>lc78k0r -fk0r.plv
```

帮助说明

帮助选项如下。

--

--

[Description format]

--

- 省略时解释
- 无显示

[功能]

- 选项在显示器上输出帮助消息。

[Application]

- 帮助消息是列表转换选项说明的列表。当执行列表转换时参阅这些内容。

[说明]

- 当 -- 选项设定时，所有其他选项无效。

注意事项 该选项不能通过 **CubeSuite** 设定。

[Example of use]

- 在显示器上输出帮助消息，描述如下：

```
C>lc78k0r --
```

```
List Conversion Program for RA78K0R Vx.xx [xx xxx xx]
  Copyright(C) NEC Electronics Corporation xxxx

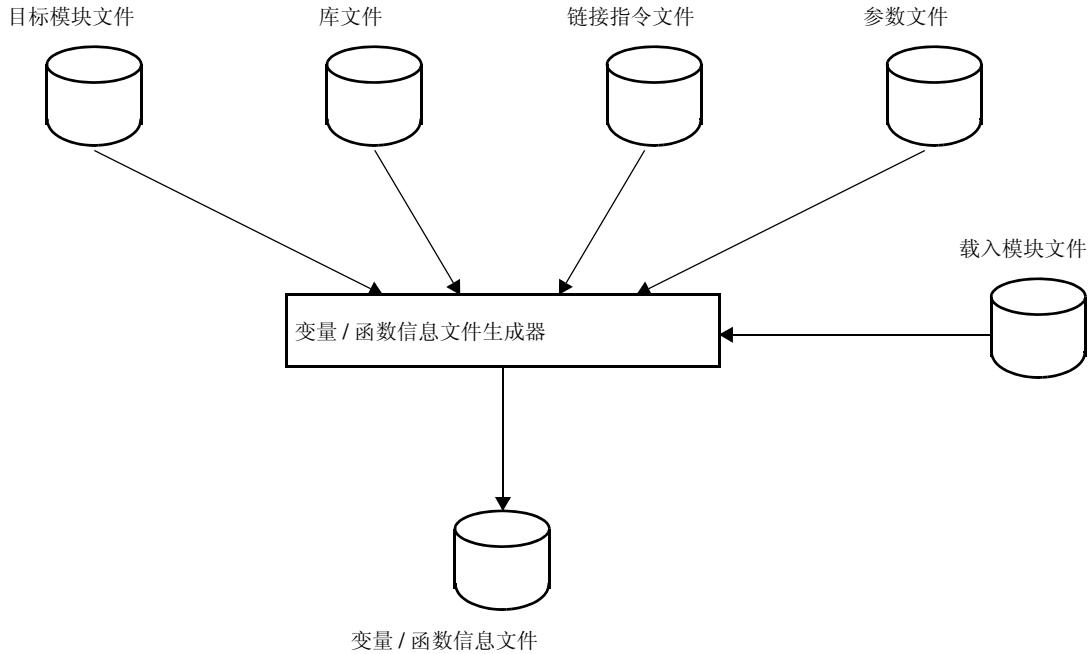
usage : LC78K0R [option[...]] input-file [option[...]]
The option is as follows ([ ] means omissible).
-r[file]: 设定目标模块文件。
-l[file]: 设定载入模块文件。
-o[file]: 设定输出列表文件 (完全汇编列表文件)。
-ffile      :Input option or input-file name from specified file.
-e[file]: 创建错误列表文件。
--          :Show this message.
```

B.7 变量 / 函数信息文件生成器

变量 / 函数信息文件生成器通过使用 C 编译程序或汇编程序输出大量的目标模块文件，并输出变量 / 函数信息文件，该文件包含有效分配的变量和函数的信息。

如果出现错误，将输出显示错误消息，用以说明导致错误的原因。当产生错误，将不能输出变量 / 函数信息文件。

图 B-36. 变量 / 函数信息文件生成器的 I/O 文件



B.7.1 I/O 文件

变量 / 函数信息文件生成器的 I/O 文件显示如下。

关于输出文件的详细信息，请参阅“3.7 变量 / 函数信息文件生成器”。

表 B-29. 变量 / 函数信息文件生成器的 I/O 文件

| 类型 | 文件名 | 说明 | 默认文件类型 |
|------|-------------|--|--------|
| 输入文件 | 目标模块文件 | - 二进制文件包含机器语言信息，与机器语言分配地址和符号信息有关的重定向信息。 - 由编译器或汇编程序输出文件 | .rel |
| | 库文件 | - 包含两个或两个以上的对象模块文件的文件 - 通过库管理程序输出文件 | .lib |
| | 链接指令文件 | - 包含用于链接器的链接指令的文件（用户创建文件） | .dr |
| | 参数文件 | - 包含执行程序所需参数的文件（用户创建的文件） | .plk |
| | 载入模块文件 | - 在自动编程期间，载入模块文件并重新输入 | .lmf |
| 输出文件 | 变量 / 函数信息文件 | - 指定文件分配到 saddr 区域和调用表区域；它是可以引用的变量和函数列表。 | .vfi |

B.7.2 功能

(1) 生成变量 / 函数信息文件

当决定变量和函数的重定向时，变量 / 函数信息文件生成器对引用数量计数，并输出带有效分配它们信息的文件。

通过设定优化由 C 编译器分配的 `saddr` 区域和调用表区域，这个信息文件能用于减少代码。

(2) ROM/RAM 用法显示

在链接到标准输出之后，变量 / 函数信息文件生成器显示 ROM/RAM 用法。

B.7.3 变量 / 函数信息

(1) 区域

(a) `saddr` 区域

78K0R 有个区域能够用 8 位类型寻址调用 `saddr` 寻址来寻址（短距离寻址）。

`saddr` 寻址目标在 FFE20H 的起始 256 个字节内。注意：在此分配用户变量，然而，因为这个区域也包含通用寄存器和端口。`saddr` 区域指向由变量 / 函数信息文件生成器排列的 192 个字节内 (FFE20H to FFEDFH)。

(b) `CALLT` 表区域

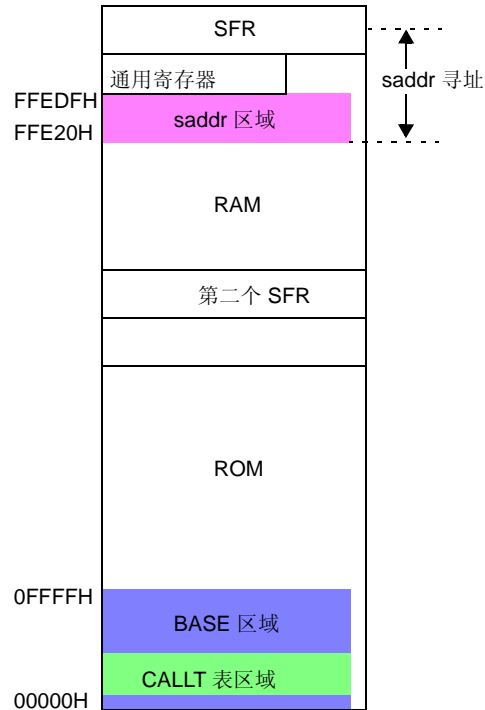
从 00080H 到 000BFH 的区域能注册为 32BASE 区域地址的分支目标。

(c) `BASE` 区域

这个区域从 00000H 到 0FFFFH。它是 2 字节调用指令的分支目标（`CALLT`）。

它能够通过编译程序定义的 `BASE` 重定向属性来设定。

图 B-37. 78K0R 的存储映像



(2) 变量信息

(a) 引用计数

变量 / 函数信息文件生成器对在重定向期间引用符号的次数进行计数。

(b) 检测空闲区域

在正常分配之后，变量 / 函数信息文件生成器检测在 **saddr** 区域中的空闲区域大小和起始地址。

(c) 决定优先级

通过表达式计算的值如下，对代码增加速度（每字节）进行统计，决定优先级别（较高值意味着较高优先级）。

$$\text{number-of-references} / \text{symbol-size} * \text{reference-type} \text{ 注意}$$

注 引用类型

near: 1 (改变从 near 区域到 saddr 区域增加 1 个字节代码)

far: 2 (改变从 far 区域到 saddr 区域增加 2 个字节代码)

sreg: 0 (通过 sreg 设定没有分配的目标，变量已经分配到 saddr 区域)

示例

| 变量 ; variable | 引用数量 | 符号大小 | 引用类型 | 优先 |
|------------------|------|-------|------|------------------|
| sym1 | 10 次 | 2 字节 | near | $10 / 2 * 1 = 5$ |
| sym2 | 6 次 | 1 个字节 | far | $6 / 1 * 2 = 12$ |
| sym3 | 6 次 | 1 个字节 | sreg | $6 / 1 * 0 = 0$ |

改变 far 变量 sym2，它在 saddr 区域引用了 6 次，对代码效率要比引用 10 次并需要 2 个字节的 near 变量 sym1 高。变量 sym3 不符合分配的要求，由于它已经分配到 saddr 区域了。

备注 以下变量不能优化。

| | |
|-----------------------|---|
| 常量 | 常量不符合分配在 saddr 区域，因为它们已分配到镜像源代码区域中。然而，对引用它们的次数进行计数，并作为注释输出文件。 |
| sreg 变量 | 已经设定的 sreg 变量不符合分配的要求。然而，对引用它们的次数进行计数，并作为注释输出文件。 |
| 静态变量 | 静态变量不符合分配要求，无论它们是在文件中还是在函数里。然而，对引用它们的次数进行计数，并作为注释输出文件。 |
| 在 C 源代码中不能定义的变量 | 在 C 源代码中不能定义的变量不符合分配的要求（例如：在汇编程序或运行时间库中的定义）。它们也不输出到输出文件中。 |
| 在启动区定义和在 flash 区引用的变量 | 在启动区定义和在 flash 区引用的变量不符合分配的要求。然而，对引用它们的次数进行计数，并作为注释输出文件。 |
| 未引用的变量 | 它们也不输出到输出文件中。 |

(d) 校准考虑

以下变量能分配到奇数地址。

- 1 个字节大小的变量（字符型，无符号字符型，枚举型，结构体，和联合体）
- 1 个字节大小变量的数组（字符，无符号字符）
- 1 个字节大小的枚举型，结构体，联合体变量数组，并有一个元素。

(3) 函数信息

(a) 引用计数

变量 / 函数信息文件生成器对在重定向期间引用符号的次数进行计数。

(b) 检测空闲区域

在正常分配之后，变量 / 函数信息文件生成器检测在调用区域和 BASE 区域中的空闲区域大小和起始地址。

(c) 决定优先级

通过以下表达式计算数值来决定优先级（较高数值意味着较高优先级）。

| |
|---|
| $number-of-references * reference-type$ <small>注意</small> |
|---|

注 引用类型

- near: 1 (改变从 near 区域到 callt 区域增加 1 个字节代码)
- far: 2 (改变从 far 区域到 callt 区域增加 2 个字节代码)
- callt: 0 (函数已经分配到没有分配目标的 callt 区域)

示例

| 功能 | 引用数量 | 引用类型 | 优先 |
|-------|------|-------|---------------|
| func1 | 10 次 | near | $10 * 1 = 10$ |
| func2 | 10 次 | far | $10 * 2 = 20$ |
| func3 | 10 次 | callt | $10 * 0 = 0$ |

改变 far 函数 func2，该函数在 callt 区域引用 10 次，提高代码效率比引用 10 次的 near 函数 func1 要高。函数 func3 不符合分配的要求，由于它已经分配到 callt 区域了。

备注 以下函数不能优化。

| | |
|-----------------|--|
| 在 flash 区域的函数 | 在 flash 区域的函数不符合分配的要求，因为它们不能在 callt 区域寄存。然而，对引用它们的次数进行计数，并作为注释输出文件。 如果在启动区的函数并从 flash 区域引用，也不可能分配；这些将作为注释输出到文件。 |
| 调用函数 | 这些不符合分配是因为它们已经注册在调用表中。然而，对引用它们的次数进行计数，并作为注释输出文件。 |
| 静态函数 | 静态函数不符合分配要求，无论它们是在文件中还是在函数里。然而，对引用它们的次数进行计数，并作为注释输出文件。 |
| 在 C 源代码中不能定义的函数 | 在 C 源代码中不能定义的函数不符合分配的要求（例如：在汇编程序或运行时间库中的定义）。它们也不输出到输出文件中。 |
| 未定义的函数 | 它们也不输出到输出文件中。 |

(d) 对函数分配状态的考虑

在 near 引用的情况下，相关函数已经分配到 BASE 区域。因此，不需要对 BASE 区域的可分配空间进行统计。在 far 引用的情况下，不可能提示何处相关函数已经分配。

- near 引用

相关函数已经注册在调用区域，因为它已经分配到 BASE 区域。

- far 引用区域的相关函数不在 BASE 区域里。

如果在 BASE 区域有足够的空闲区域，则相关函数注册到调用区域。

如果在 BASE 区域没有足够的空闲区域，则相关函数不能注册到调用区域。

(4) 不输出到变量 / 函数信息文件中的符号

以下符号不输出到变量 / 函数信息文件中。

- 未引用符号

- 在库中定义的符号
- 除在载入模块以外的 EXTERN 符号
- 在汇编程序源代码中定义的符号
- 设置段的重定位属性用 AT
- RTOS 任务或 RTOS 的中断处理
- 固化的 ROM 函数
- 向量中断函数
- 符号类型为 T_NULL

B.7.4 操作方法

(1) 启动变量 / 函数信息文件生成器

以下两种方法能用于启动变量 / 函数信息文件生成器。

(a) 从命令行启动

```
X:[path-name]>vf78k0r[Δoption] ... object-module-file-name[Δobject-module-file-name]
... [Δoption] ... [Δ]
```

| | |
|-------------------------|---|
| X | 当前驱动器名称 |
| 路径名 | 当前文件夹名 |
| vf78k0r | 变量 / 函数信息文件生成器的命令名称 |
| 选项 | 为变量 / 函数信息文件生成器操作，输入详细的指令。 当设定两个或更多变量 / 函数重定向选项时，使用空格将选项分类。变量 / 函数重定向选项不区分大写和小写字符。有关变量 / 函数重定向选项的详细内容，参阅“ B.7.5 选项 ”。 带有空格的路径附在一对双引号 (") 中。 |
| object-module-file-name | 用于生成变量 / 函数信息文件的目标模块文件的名称 作为输入模块可输入多达 1024 个项目。 带空格的文件名路径附在一对双引号 (") 中。 |

注意事项 在设定相同选项及作为那些在链接程序中设定目标模块文件之后，添加选项指定在变量 / 函数信息文件生成器中。

示例 输出变量 / 函数信息文件 (k0r.vfi)，描述如下：

```
C>vf78k0r k0rmain.rel k0rsub.rel -vok0r.vfi
```

(b) 从参数文件启动

当启动 C 编译器所需数据没有放置在命令行上时，或者每次生成变量 / 函数信息文件时重复指定了相同的编译选项，则可使用参数文件。

通过参数文件启动汇编程序，在命令行上设定参数文件选项 (-f)。

通过参数文件启动目标转换器如下：

```
X>vf78k0r[Δobject-module-file]Δ-fparameter-file-name
```

| | |
|-------|----------------------------|
| -f | 参数文件说明选项 |
| 参数文件名 | 文件包含启动变量 / 函数信息文件生成器所需的数据。 |

备注 使用编辑器创建参数文件。

参数文件内容的书写规则如下：

```
[[[Δ]option[Δoption] ... [Δ]Δ]] ...
```

- 如果命令行省略源文件名，参数文件中仅可以指定 1 个源文件名。
- 源文件名也可以在加在选项后。
- 在参数文件中写入所有变量 / 函数重定位选项以及在命令行中设定的输出文件名。

示例 使用编辑器创建参数文件 k0r.plk，然后启动变量 / 函数信息文件生成器。

```
; parameter file
k0rmain.rel k0rsub.rel -ok0r.lmf -pk0r.map -e
-tC:\tmp
```

```
C>vf78k0r -fk0r.plk -vok0r.vfi
```

(2) 执行开始和结束消息

(a) 执行开始消息

当变量 / 函数信息文件生成器启动时，在显示器上会显示执行启动消息。

```
78K0R Var-Func-Inf Vx.xx [xx xxx xxxx]
Copyright(C) NEC Electronics Corporation xxxx
```

(b) 执行结束消息

如果从变量 / 函数信息文件生成器中没有检测到出错结果，则变量 / 函数信息文件生成器在屏幕上输出以下信息并返回控制给主操作系统。

```
Target chip : uPD78xxx
Device file : Vx.xx

VF check complete, 0 error(s) and 0 warning(s) found.
```

如果在变量 / 函数信息文件生成期间，变量 / 函数信息文件生成器检测到致命错误，不能继续变量 / 函数信息文件的生成进程时，变量 / 函数信息文件生成器在屏幕上输出消息，取消变量 / 函数信息文件生成并返回控制给主操作系统。

输出示例如下。

- 设定不存在的目标模块文件。

```
C>vf78k0r samp1.rel samp2.rel -vosamp.vfi
```

```
78K0R Var-Func-Inf Vx.xx [xx xxx xxxx]
    Copyright(C) NEC Electronics Corporation xxxx

VF78K0R error F0006 : File not found 'samp1.rel'
VF78K0R error F0006 : File not found 'samp2.rel'
程序终止。
```

在上述例子中，不存在的目标模块文件设定。错误产生且变量 / 函数信息文件生成器终止执行。

- 设定了不存在的变量 / 函数重定位选项。

```
C>vf78k0r k0rmain.rel k0rsub.rel -z
```

```
78K0R Var-Func-Inf Vx.xx [xx xxx xxxx]
    Copyright(C) NEC Electronics Corporation xxxx

VF78K0R error F0018 : Option is not recognized '-z'
程序终止。
```

在以上例子中，设定了不存在的变量 / 函数重定位选项。错误产生且变量 / 函数信息文件生成器终止执行。

(3) 在 CubeSuite 中设置选项

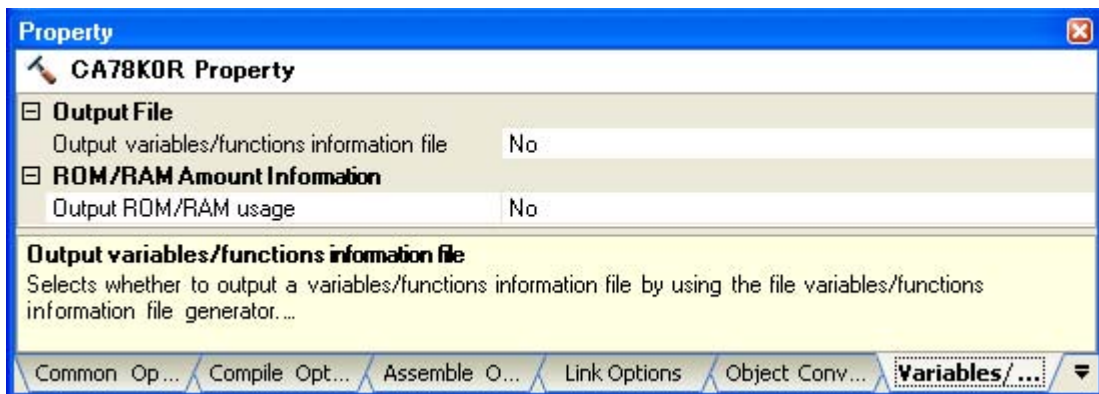
本章节描述了通过 CubeSuite，如何设置变量 / 函数重定位选项。

通过 CubeSuite 的 [工程树面板](#)，选择 Build Tool 节点。然后通过 [View] 菜单选择 [Property]。打开 [属性面板](#)。

再选择 [\[Variables/Functions Relocation Options\]](#) 标签。

可以通过该标签给各种选项设置必要属性。

图 B-38. 属性面板: [Memory Bank Relocation Options] 标签



B.7.5 选项**(1) 类型**

变量 / 函数重定位选项是有关变量 / 函数信息文件生成器操作的详细指令。

变量 / 函数重定位选项的类型和说明显示如下。

表 B-30. 变量 / 功能重定位选项

| 分类 | 选项 | 说明 |
|-----------------------------|-----|--------------------------------|
| 变量 / 函数信息文件输出说明 | -vo | 设定变量 / 函数信息文件的输出。 |
| 空闲的 <code>saddr</code> 区域说明 | -vs | 设定 <code>saddr</code> 区域的边界大小。 |
| ROM/RAM 用法显示 | -vx | 在链接到标准输出之后，输出 ROM/RAM 用法。 |

变量 / 函数信息文件输出说明

变量 / 函数信息文件输出选项说明如下。

- **-vo**

-vo

[Description format]

```
-vooutput-file-name
```

- 省略时解释
这个选项不能省略，（除非设定了 **-vx** 选项时）

[功能]

- **-vo** 选项设定变量 / 函数信息文件的输出。
这也同时指定输出文件名和路径。

[Application]

- 使用 **-vo** 选项设定变量 / 函数信息文件的输出。

[说明]

- 默认文件类型为 ".vfi"。
- *i* 输出文件名称 “*i*”，其中包含设定的路径名称。
- 即使设定了 **-vo** 选项，在链接完成前产生错误，则不能输出变量 / 函数信息文件。
- **-vo** 和 **-vx** 两个选项不能在同时设定。

[Example of use]

- 输出变量 / 函数信息文件 (**kOr.vfi**), 描述如下:

```
C>vf78k0r k0rmain.rel k0rsub.rel -vok0r.vfi
```

空闲的 `saddr` 区域说明

空闲 `saddr` 区域选项说明如下。

- `-vs`

-vs

[Description format]

```
-vs[size]
```

- 省略时解释

-vs0

[功能]

- 在通过这个工具分配变量给 `saddr` 区域之后，由于处理顺序和排列的关系，在编译或链接期间可能会产生排列错误。在这种状况下，在 `saddr` 区域进行边界分配能够避免这种错误。

-vs 选项设定 `saddr` 区域的边界大小。

[Application]

- 通过这个工具在分配变量到 `saddr` 区域后，使用 `-vs` 选项在编译或链接期间，避免分配错误。

[说明]

- 根据“大小”设定 `saddr` 区域的边界大小（字节数）。

- 该值可以设定为十进制、十六进制、或二进制数字。

最多可达 192 (用十进制数字)。如果设定了 193 或更多，则产生错误。

- 如果设定的空闲区域数量大于实际空闲数量，则产生错误。

- 如果设定了 `-vo` 选项，则 `-vs` 选项则有效。

[Example of use]

- 设定 `saddr` 区域的边界大小为 10 个字节（用十进制数字），描述如下：

```
C>vf78k0r k0rmain.rel k0rsub.rel -vok0r.vfi -vs10
```

- 设定 `saddr` 区域的边界大小为 0AH 个字节（用十六进制数字），描述如下：

```
C>vf78k0r k0rmain.rel k0rsub.rel -vok0r.vfi -vs0AH
```

- 设定 `saddr` 区域的边界大小为 1010B 个字节（用二进制数字），描述如下：

```
C>vf78k0r k0rmain.rel k0rsub.rel -vok0r.vfi -vs1010B
```

ROM/RAM 用法显示

ROM/RAM 用法输出选项如下。

- **-vx**

-vx

[Description format]

-vx

- 省略时解释
ROM/RAM 用法不输出到标注输出。

[功能]

- 在链接到标准输出之后，-vx 选项输出 ROM/RAM 用法。

[Application]

- 在链接之后，使用 -vx 选项输出 ROM/RAM 用法。

[说明]

- -vo 和 -vx 两个选项不能在同时设定。
- ROM/RAM 用法输出例子如下所示。
- 当使用默认存储区域名称时

```
*** Memory Area Information ***
ROM : xxxxxH byte(s) real data
RAM : xxxxxH byte(s) real data

*** Memory Area Information in ROM ***
ROM : xxxxxH byte(s)

*** Memory Area Information in RAM ***
RAM : xxxxxH byte(s)
```

- 当使用存储指令定义存储区域名称时

```
*** Memory Area Information ***
ROM : xxxxxH byte(s) real data
RAM : xxxxxH byte(s) real data

*** Memory Area Information in ROM ***
ROM : xxxxxH byte(s)
ROM1 : xxxxxH byte(s)

*** Memory Area Information in RAM ***
RAM : xxxxxH byte(s)
RAM1 : xxxxxH byte(s)
```

首先输出全部使用的数量，接下去为每个已定义存储区域的用法。

[Example of use]

- 在链接到标准输出之后，输出 ROM/RAM 用法，描述如下：

```
C>vf78k0r k0rmain.rel k0rsub.rel -vx
```

附录 C 索引

| | | | |
|-------------------------|-------------------------|----------------------------------|-------------------------|
| 符号 | | 帮助 | 540 |
| --/?/-h (CC78K0R) | 384 | C 编译器 | 332 |
| -- (LB78K0R) | 531 | 编辑器面板 | 230 |
| -- (LC78K0R) | 554 | 标签跳跃 | 234 |
| -- (LK78K0R) | 474 | 标准库 | 336 |
| -- (OC78K0R) | 518 | Boot-flash relink function | 476 |
| -- (RA78K0R) | 428 | Build tool 版本 | 24 |
| .asm | 385 | C | |
| .cer | 333 | -c (CC78K0R) | 343 |
| .dr | 430, 555 | -c (RA78K0R) | 392 |
| .ecc | 333 | 参数文件 | 385, 430, 489, 502, 542 |
| .elk | 430 | -ccza (LK78K0R) | 472 |
| .elv | 542 | 查找和替换对话框 | 252 |
| .eoc | 489 | [Quick Search] 标签 | 253 |
| .er | 333 | [Whole Replace] 标签 | 262 |
| .era | 385 | [Whole Search] 标签 | 256 |
| .her | 333 | [Quick Replace] 标签 | 259 |
| .hex | 489 | 重命名工具栏对话框 | 297 |
| .lib | 430, 520, 555 | 重排命令对话框 | 300 |
| .lmf | 430, 489, 542, 555 | 重新构建 | 78, 81 |
| .lst | 520 | 创建 | 534 |
| .map | 430 | [Commands] 标签 | 294 |
| .p | 542 | -common (CC78K0R) | 382 |
| .plk | 430, 555 | -common (RA78K0R) | 426 |
| .plv | 542 | -compati (RA78K0R) | 427 |
| .poc | 489 | 错误列表 | 107, 112, 114 |
| .pra | 385 | 错误列表文件 | 96 |
| .prn | 385, 542 | D | |
| .rel | 385, 430, 520, 542, 555 | -d (CC78K0R) | 355 |
| .sym | 489 | -d (LK78K0R) | 442 |
| .vfi | 555 | -d (RA78K0R) | 425 |
| A | | 打开程序对话框 | 313 |
| -a (CC78K0R) | 358 | E | |
| add | 535 | -e (CC78K0R) | 362 |
| Assemble list | 103 | -e (LC78K0R) | 552 |
| B | | -e (LK78K0R) | 455 |
| -b (LK78K0R) | 456 | | |

| | | | |
|---------------------------------------|--------|----------------------|--------|
| -e (OC78K0R) | 512 | -i (RA78K0R) | 398 |
| -e (RA78K0R) | 417 | INC78K0R | 357 |
| exit | 541 | | |
| F | | J | |
| -f (CC78K0R) | 375 | -j (LK78K0R) | 438 |
| -f (LC78K0R) | 553 | -j (RA78K0R) | 394 |
| -f (LK78K0R) | 459 | 交叉引用列表 | 105 |
| -f (OC78K0R) | 513 | 交叉引用列表文件 | 100 |
| -f (RA78K0R) | 419 | 进度状态对话框 | 266 |
| 分项 | 31 | 局部符号列表 | 111 |
| 符号列表 | 104 | | |
| G | | K | |
| -g (CC78K0R) | 351 | -k (CC78K0R) | 353 |
| -g (LK78K0R) | 439 | -ka (RA78K0R) | 401 |
| -g (RA78K0R) | 395 | -kd (LK78K0R) | 446 |
| -ga (RA78K0R) | 397 | -ki (OC78K0R) | 514 |
| 改变构建模式 | 74 | -kie (OC78K0R) | 514 |
| 改变输出文件名称 | 36 | -kl (LK78K0R) | 450 |
| -gb (LK78K0R) | 470 | -km (LK78K0R) | 444 |
| [General - Build/Debug] 分项 | 284 | -km (OC78K0R) | 514 |
| [General - Display] 分项 | 271 | -kme (OC78K0R) | 514 |
| [General - External Tools] 分项 | 280 | -kp (LK78K0R) | 448 |
| [General - Font and Color] 分项 | 275 | -ks (RA78K0R) | 403 |
| [General - Startup and Exit] 分项 | 269 | -kt (OC78K0R) | 514 |
| [General - Text Editor] 分项 | 273 | 快速构建 | 78, 81 |
| [General - Update] 分项 | 286 | 库管理程序 | 520 |
| -gi (LK78K0R) | 468 | -kx (RA78K0R) | 404 |
| -go (LK78K0R) | 466 | | |
| 工程树面板 | 134 | L | |
| 公用符号列表 | 110 | -l (LC78K0R) | 550 |
| 构建 | 78, 80 | -lf (CC78K0R) | 371 |
| 构建模式 | 73, 74 | -lf (LB78K0) | 528 |
| 构建模式设置对话框 | 248 | -lf (LK78K0R) | 454 |
| | | -lf (RA78K0R) | 416 |
| H | | -lh (RA78K0R) | 410 |
| 汇编程序 | 385 | -li (CC78K0R) | 372 |
| 汇编源文件 | 93 | 链接列表文件 | 107 |
| 活动工程 | 72 | 链接器 | 430 |
| | | 链接顺序对话框 | 246 |
| I | | Map 列表 | 109 |
| -i (CC78K0R) | 357 | 列表 | 539 |
| -i (LK78K0R) | 457 | 列表转换器 | 542 |
| | | 另存为对话框 | 311 |
| | | -ll (CC78K0R) | 369 |

| | | | |
|---------------------------|-----|-------------------------------------|---------------|
| -ll (LB78K0) | 527 | -np (LK78K0R) | 443 |
| -ll (LK78K0R) | 452 | -np (RA78K0R) | 400 |
| -ll (RA78K0R) | 408 | -nq (CC78K0R) | 349 |
| -lt (CC78K0R) | 370 | -nr (CC78K0R) | 345, 347, 348 |
| -lt (RA78K0R) | 413 | -nr (OC78K0R) | 509 |
| 路径编辑对话框 | 244 | -ns (LK78K0R) | 440 |
| -lw (CC78K0R) | 368 | -ns (OC78K0R) | 508 |
| -lw (LB78K0) | 526 | -nu (OC78K0R) | 510 |
| -lw (RA78K0R) | 406 | -nv (CC78K0R) | 374 |
| M | | -nz (CC78K0R) | 377 |
| -m (CC78K0R) | 380 | O | |
| -ma (CC78K0R) | 383 | -o (CC78K0R) | 344 |
| -mi (CC78K0R) | 381 | -o (LC78K0R) | 551 |
| -mi (LK78K0R) | 471 | -o (LK78K0R) | 437 |
| 目标转换器 | 489 | -o (OC78K0R) | 507 |
| N | | -o (RA78K0R) | 393 |
| -nccza (LK78K0R) | 472 | [Other - User Information] 分项 | 288 |
| -ncompati (RA78K0R) | 427 | P | |
| -ne (LC78K0R) | 552 | -p (CC78K0R) | 352 |
| -ne (LK78K0R) | 455 | -p (LK78K0R) | 443 |
| -ne (OC78K0R) | 512 | -p (RA78K0R) | 400 |
| -ne (RA78K0R) | 417 | 批量构建 | 78, 83 |
| -ng (CC78K0R) | 351 | 批量构建对话框 | 250 |
| -ng (LK78K0R) | 439 | Q | |
| -ng (RA78K0R) | 395 | -q (CC78K0R) | 349 |
| -nga (RA78K0R) | 397 | 清除 | 86 |
| -nj (LK78K0R) | 438 | [Quick Replace] 标签 | 259 |
| -nj (RA78K0R) | 394 | [Quick Search] 标签 | 253 |
| -nka (RA78K0R) | 401 | R | |
| -nkd (LK78K0R) | 446 | -r (CC78K0R) | 345 |
| -nkl (LK78K0R) | 450 | -r (LC78K0R) | 549 |
| -nkm (LK78K0R) | 444 | -r (OC78K0R) | 509 |
| -nkp (LK78K0R) | 448 | -rd (CC78K0R) | 347 |
| -nks (RA78K0R) | 403 | -rs (CC78K0R) | 348 |
| -nkx (RA78K0R) | 404 | S | |
| -nlf (LB78K0) | 528 | -s (LK78K0R) | 440 |
| -nlf (LK78K0R) | 454 | -s (OC78K0R) | 508 |
| -nlf (RA78K0R) | 416 | -sa (CC78K0R) | 359 |
| -no (CC78K0R) | 344 | -se (CC78K0R) | 364 |
| -no (LK78K0R) | 437 | | |
| -no (OC78K0) | 507 | | |
| -no (RA78K0R) | 393 | | |

- self (LK78K0R) 473
- selfw (LK78K0R) 473
- 删除 536
- 删除构建模式 75
- 设定启动区变量 / 函数信息文件对话框 307
- 设定引导区载入模块文件对话框 309
- 设置编译选项 40
- 设置变量 / 函数重新配置选项 62
- 设置创建库选项 61
- 设置汇编选项 45
- 设置连接选项 48
- 设置目标转换选项 59
- 拾取 538
- 输出符号信息 38
- 输出汇编列表 37
- 输出面板 233
- 输出映射信息 38
- 属性面板 145
 - [Assemble Options] 标签 178
 - [Build Settings] 标签 203
 - [Category Information] 标签 229
 - [Common Options] 标签 148
 - [Create Library Options] 标签 198
 - [File Information] 标签 227
 - [Individual Assemble Options] 标签 220
 - [Individual Compile Options] 标签 206
 - [Link Options] 标签 184
 - [Object Convert Options] 标签 193
 - [Variables/Functions Relocation Options] 标签
201
 - [Compile Options] 标签 158
- T**
- t (CC78K0R) 376
- t (LB78K0R) 529
- t (LK78K0R) 460
- t (RA78K0R) 420
- 添加构建模式 73
- 添加文件到工程中 27
- 添加文件对话框 236
- 添加文件夹和文件对话框 238
- 添加现有的文件对话框 303
- 跳转定位对话框 265
- 替换 537
- [Toolbars] 标签 292
- U**
- u (CC78K0R) 356
- u (OC78K0R) 510
- V**
- v (CC78K0R) 374
- vo (VF78K0R) 564
- vs (VF78K0R) 565
- vx (VF78K0R) 566
- W**
- w (CC78K0R) 373
- w (LK78K0R) 464
- 完全的汇编列表 114
- 文本编辑对话框 242
- Include 文件 123
- 文件夹浏览对话框 305
- 文件显示顺序 32
- 文件依赖关系 33
- [Whole Replace] 标签 262
- [Whole Search] 标签 256
- X**
- x (CC78K0R) 366
- 新建工具栏对话框 296
- 选项对话框 267
 - [General - Build/Debug] 分项 284
 - [General - Display] 分项 271
 - [General - External Tools] 分项 280
 - [General - Font and Color] 分项 275
 - [General - Startup and Exit] 分项 269
 - [General - Text Editor] 分项 273
 - [General - Update] 分项 286
 - [Other - User Information] 分项 288
- 选择程序对话框 315
- 选择外部文本编辑器对话框 317
- Y**
- y (CC78K0R) 379
- y (LK78K0R) 462
- y (OC78K0R) 515

| | |
|---------------------|----------|
| -y (RA78K0R) | 423 |
| 用户设置对话框 | 290 |
| [Commands] 标签 | 294 |
| [Toolbars] 标签 | 292 |
| 预处理列表文件 | 98 |
| 运行时间库 | 336 |
| Z | |
| -z (CC78K0R) | 377 |
| -zb (LK78K0R) | 465 |
| -ze (RA78K0R) | 422 |
| -zf (OC78K0R) | 517 |
| 载入模块文件 | 489, 542 |
| 执行创建 | 78 |
| 主窗口 | 130 |
| 自定义键盘对话框 | 298 |
| 字符串输入对话框 | 240 |
| 子命令 | 533 |
| -zn (RA78K0R) | 422 |
| -zs (RA78K0R) | 422 |
| | |
| 载入模块文件 | 430 |
| 重链接功能 | 476 |

详细信息请联系:

中国区

MCU 技术支持热线:

电话: +86-400-700-0606 (普通话)

服务时间: 9:00-12:00, 13:00-17:00 (不含法定节假日)

网址:

<http://www.cn.necel.com/> (中文)

<http://www.necel.com/> (英文)

[北京]

日电电子(中国)有限公司

中国北京市海淀区知春路 27 号

量子芯座 7, 8, 9, 15 层

电话: (+86) 10-8235-1155

传真: (+86) 10-8235-7679

[深圳]

日电电子(中国)有限公司深圳分公司

深圳市福田区益田路卓越时代广场大厦 39 楼

3901, 3902, 3909 室

电话: (+86) 755-8282-9800

传真: (+86) 755-8282-9899

[上海]

日电电子(中国)有限公司上海分公司

中国上海市浦东新区银城中路 200 号

中银大厦 2409-2412 和 2509-2510 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[香港]

香港日电电子有限公司

香港九龙旺角太子道西 193 号新世纪广场

第 2 座 16 楼 1601-1613 室

电话: (+852) 2886-9318

传真: (+852) 2886-9022

2886-9044

上海恩益禧电子国际贸易有限公司

中国上海市浦东新区银城中路 200 号

中银大厦 2511-2512 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[成都]

日电电子(中国)有限公司成都分公司

成都市二环路南三段 15 号天华大厦 7 楼 703 室

电话: (+86)28-8512-5224

传真: (+86)28-8512-5334

[长春]

日电电子(中国)有限公司长春分公司

吉林省长春市朝阳区

西安大路 727 号中银大厦 A 座 1609 室

电话: (+86)431-8859-7533 / 8859-8533

传真: (+86)431-8680-2944

[大连]

日电电子(中国)有限公司长春分公司

大连市中山路 88 号天安国际大厦 2701 室

电话: (+86)411-8230-8815 / 8230-8825

传真: (+86)411-8230-8835