

CubeSuite+ V2.01.00

統合開発環境

ユーザーズマニュアル RH850 ビルド編

対象デバイス

RH850 ファミリ

お知らせ：

本資料の以下のページに訂正、追加がございます。

- Page 30 2.4.3 マップ情報を出力する
- Page 31 2.4.4 ライブラリ情報を出力する
- Page 96 3.2 リンク・マップ・ファイル
- Page 107 3.4 ライブラリ・リスト・ファイル

本資料に記載の全ての情報は発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、
 家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、
 防災・防犯装置、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

このマニュアルの使い方

このマニュアルは、RH850 ファミリ用アプリケーション・システムを開発する際の統合開発環境である CubeSuite+ について説明します。

CubeSuite+ は、RH850 ファミリの統合開発環境（ソフトウェア開発における、設計、実装、デバッグなどの各開発フェーズに必要なツールをプラットフォームである IDE に統合）です。統合することで、さまざまなツールを使い分ける必要がなく、本製品のみを使用して開発のすべてを行うことができます。

対 象 者 このマニュアルは、CubeSuite+ を使用してアプリケーション・システムを開発するユーザを対象としています。

目 的 このマニュアルは、CubeSuite+ の持つソフトウェア機能をユーザに理解していただき、これらのデバイスを使用するシステムのハードウェア、ソフトウェア開発の参照用資料として役立つことを目的としています。

構 成 このマニュアルは、大きく分けて次の内容で構成しています。

[第 1 章 概 説](#)

[第 2 章 機 能](#)

[第 3 章 ビルドの出カリスト](#)

[付録 A ウィンドウ・リファレンス](#)

[付録 B コマンド・リファレンス](#)

[付録 C 索 引](#)

読 み 方 このマニュアルを読むにあたっては、電気、論理回路、マイクロコンピュータに関する一般知識が必要となります。

凡 例	データ表記の重み	: 左が上位桁、右が下位桁
	アクティブ・ロウの表記	: <u>XXX</u> （端子、信号名称に上線）
	注	: 本文中につけた注の説明
	注意	: 気をつけて読んでいただきたい内容
	備考	: 本文中の補足説明
	数の表記	: 10 進数 ... XXXX
		16 進数 ... 0xFFFF

関連資料 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

資 料 名		資料番号	
		和文	英文
CubeSuite+ 統合開発環境 ユーザーズ・マニュアル	起動編	R20UT2444J	R20UT2444E
	V850 設計編	R20UT2134J	R20UT2134E
	RL78 設計編	R20UT2136J	R20UT2136E
	78K0R 設計編	R20UT2137J	R20UT2137E
	78K0 設計編	R20UT2138J	R20UT2138E
	RH850 コーディング編	R20UT2584J	R20UT2584E
	RX コーディング編	R20UT2470J	R20UT2470E
	V850 コーディング編	R20UT0553J	R20UT0553E
	コーディング編（CX コンパイラ）	R20UT2139J	R20UT2139E
	RL78, 78K0R コーディング編	R20UT2140J	R20UT2140E
	78K0 コーディング編	R20UT2141J	R20UT2141E
	RH850 ビルド編	このマニュアル	R20UT2585E
	RX ビルド編	R20UT2472J	R20UT2472E
	V850 ビルド編	R20UT0557J	R20UT0557E
	ビルド編（CX コンパイラ）	R20UT2142J	R20UT2142E
	RL78, 78K0R ビルド編	R20UT2143J	R20UT2143E
	78K0 ビルド編	R20UT0783J	R20UT0783E
	RX デバッグ編	R20UT2350J	R20UT2350E
	V850 デバッグ編	R20UT2446J	R20UT2446E
	RL78 デバッグ編	R20UT2445J	R20UT2445E
	78K0R デバッグ編	R20UT0732J	R20UT0732E
	78K0 デバッグ編	R20UT0731J	R20UT0731E
	解析編	R20UT2447J	R20UT2447E
	メッセージ編	R20UT2448J	R20UT2448E

注意 上記関連資料は、予告なしに内容を変更することがあります。設計などには、必ず最新の資料を使用してください。

この資料に記載されている会社名、製品名などは、各社の商標または登録商標です。

目 次

第1章 概 説 … 8

- 1.1 概 要 … 8
- 1.2 著作権について … 9
- 1.3 特 長 … 10

第2章 機 能 … 11

- 2.1 概 要 … 11
 - 2.1.1 ロード・モジュールを作成する … 11
 - 2.1.2 ユーザ・ライブラリを作成する … 12
- 2.2 ビルド・ツールのバージョンを変更する … 14
- 2.3 ビルド対象ファイルを設定する … 15
 - 2.3.1 プロジェクトにファイルを追加する … 15
 - 2.3.2 プロジェクトからファイルを外す … 19
 - 2.3.3 ファイルをビルド対象から外す … 20
 - 2.3.4 ファイルをカテゴリに分類する … 20
 - 2.3.5 ファイルの表示順を変更する … 21
 - 2.3.6 ファイルの依存関係を更新する … 22
- 2.4 出力ファイルの種類を設定する … 25
 - 2.4.1 出力ファイル名を変更する … 25
 - 2.4.2 アセンブル・リストを出力する … 28
 - 2.4.3 マップ情報を出力する … 29
 - 2.4.4 ライブラリ情報を出力する … 30
- 2.5 コンパイル・オプションを設定する … 32
 - 2.5.1 コード・サイズを優先した最適化を行う … 33
 - 2.5.2 実行速度を優先した最適化を行う … 33
 - 2.5.3 インクルード・パスを追加する … 33
 - 2.5.4 定義マクロを設定する … 35
 - 2.5.5 レジスタ・モードを変更する … 37
- 2.6 アセンブル・オプションを設定する … 38
 - 2.6.1 インクルード・パスを追加する … 39
 - 2.6.2 定義マクロを設定する … 40
- 2.7 リンク・オプションを設定する … 42
 - 2.7.1 ユーザ・ライブラリを追加する … 43
 - 2.7.2 オーバーレイ・セクションの選択機能を使用するための準備をする … 45
- 2.8 ヘキサ出力オプションを設定する … 54
 - 2.8.1 ヘキサ・ファイルの出力を設定する … 54
 - 2.8.2 空き領域を充てんする … 56
 - 2.8.3 複数のオブジェクトを結合する … 58
- 2.9 ライブラリ生成オプションを設定する … 60
 - 2.9.1 ライブラリ・ファイルの出力を設定する … 60
- 2.10 個別にビルド・オプションを設定する … 63

2.10.1	プロジェクト単位でビルド・オプションを設定する	...	63
2.10.2	ファイル単位でコンパイル／アセンブル・オプションを設定する	...	63
2.11	ビルドの設定をする	...	67
2.11.1	他のプロジェクトのビルド・オプションをインポートする	...	67
2.11.2	ファイルのリンク順を設定する	...	69
2.11.3	サブプロジェクトのビルド順を変更する	...	73
2.11.4	ビルド・オプションを一覧表示する	...	73
2.11.5	ビルド対象プロジェクトを変更する	...	73
2.11.6	ビルド・モードを追加する	...	75
2.11.7	ビルド・モードを変更する	...	77
2.11.8	ビルド・モードを削除する	...	78
2.11.9	現在のビルド・オプションをプロジェクトの標準に設定する	...	79
2.12	ビルドを実行する	...	80
2.12.1	更新ファイルのビルドを実行する	...	83
2.12.2	すべてのファイルのビルドを実行する	...	84
2.12.3	他の処理と平行してビルドを実行する	...	85
2.12.4	ビルド・モードを一括してビルドを実行する	...	86
2.12.5	複数のファイルのコンパイル／アセンブル／リンクを同時に実行する	...	87
2.12.6	ファイル単位でコンパイル／アセンブルする	...	88
2.12.7	ビルドの実行を中止する	...	89
2.12.8	ビルド結果をファイルに保存する	...	90
2.12.9	中間ファイル、生成ファイルを削除する	...	90
2.13	スタックを見積もる	...	92

第3章 ビルドの出力リスト ... 93

3.1	アセンブル・リスト・ファイル	...	93
3.1.1	アセンブル・リストの構成	...	94
3.1.2	アセンブル・リスト	...	94
3.1.3	セクション・リスト	...	95
3.1.4	コマンド・ライン情報	...	95
3.2	リンク・マップ・ファイル	...	96
3.2.1	リンク・マップの構成	...	96
3.2.2	オプション情報	...	97
3.2.3	エラー情報	...	97
3.2.4	リンク・マップ情報	...	98
3.2.5	合計セクション・サイズ	...	98
3.2.6	シンボル情報	...	99
3.2.7	クロス・リファレンス情報	...	101
3.3	リンク・マップ・ファイル（オブジェクト結合時）	...	103
3.3.1	リンク・マップの構成	...	103
3.3.2	ヘッダ情報	...	104
3.3.3	オプション情報	...	104
3.3.4	エラー情報	...	105
3.3.5	エントリ情報	...	105
3.3.6	結合アドレス情報	...	105
3.3.7	アドレス重複情報	...	106
3.4	ライブラリ・リスト・ファイル	...	107
3.4.1	ライブラリ・リストの構成	...	107

3.4.2	オプション情報	...	108
3.4.3	エラー情報	...	109
3.4.4	ライブラリ情報	...	109
3.4.5	ライブラリ内モジュール, セクション, シンボル情報	...	110
3.5	インテル拡張ヘキサ・ファイル	...	111
3.5.1	インテル拡張ヘキサ・ファイルの構成	...	111
3.5.2	スタート・リニア・アドレス・レコード	...	113
3.5.3	拡張リニア・アドレス・レコード	...	113
3.5.4	スタート・アドレス・レコード	...	114
3.5.5	拡張アドレス・レコード	...	114
3.5.6	データ・レコード	...	115
3.5.7	エンド・レコード	...	116
3.6	モトローラ・Sタイプ・ファイル	...	117
3.6.1	モトローラ・Sタイプ・ファイルの構成	...	117
3.6.2	S0 レコード	...	119
3.6.3	S1 レコード	...	119
3.6.4	S2 レコード	...	119
3.6.5	S3 レコード	...	120
3.6.6	S7 レコード	...	120
3.6.7	S8 レコード	...	121
3.6.8	S9 レコード	...	121

付録 A ウィンドウ・リファレンス ... 122

A.1	説明	...	122
-----	----	-----	-----

付録 B コマンド・リファレンス ... 356

B.1	概要	...	356
B.2	入出力ファイル	...	358
B.3	環境変数	...	360
B.4	操作方法	...	361
B.4.1	コマンド・ラインでの操作方法	...	361
B.4.2	サブコマンド・ファイルの使用方法	...	363
B.4.3	CubeSuite+ でのオプション設定	...	365
B.5	オプション	...	366
B.5.1	コンパイル・オプション	...	367
B.5.2	アセンブル・オプション	...	438
B.5.3	リンク・オプション	...	460
B.6	オプションの複数指定	...	526
B.6.1	優先順序	...	526
B.6.2	機能矛盾	...	526
B.6.3	依存関係	...	527
B.6.4	#pragma 指定との関係	...	527

付録 C 索引 ... 528

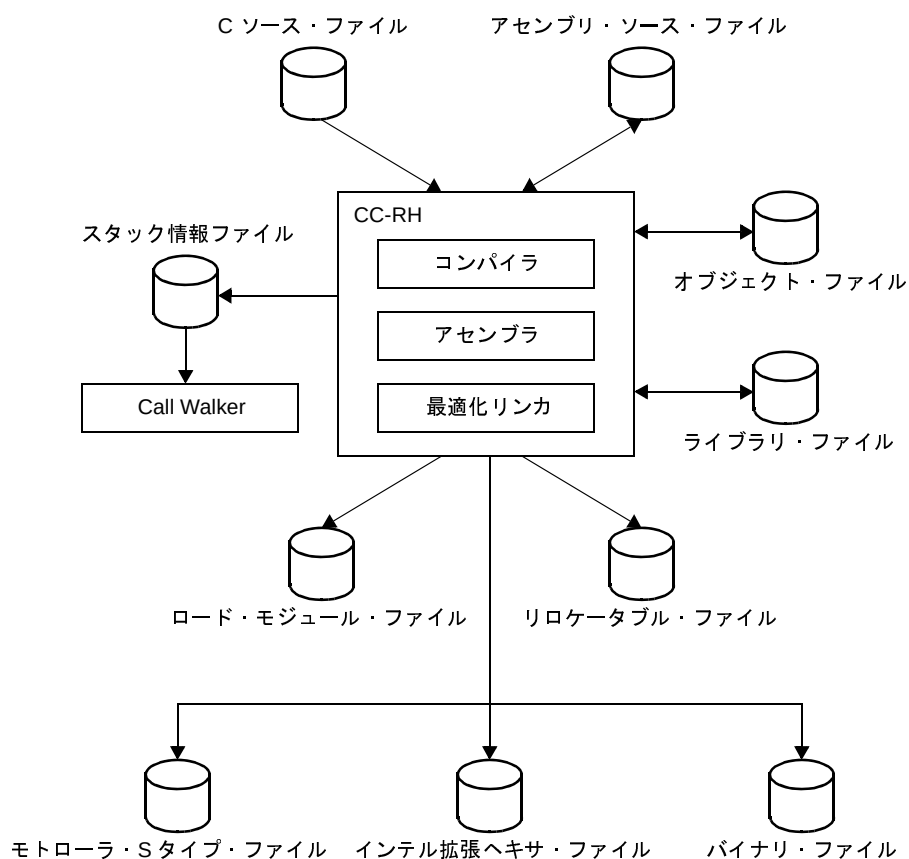
第1章 概 説

この章では、ビルド・ツール（CC-RH）の概要について説明します。

1.1 概 要

CC-RH は、本製品が提供しているコンポーネントの一種であり、GUI ベースで各種情報を設定することにより、ソース・ファイルからロード・モジュール・ファイル、ライブラリ・ファイルなどを目的に応じて生成することができます。

図 1—1 CC-RH の処理の流れ



備考 CC-RH を構成しているコマンドの詳細については、「[付録 B コマンド・リファレンス](#)」、Call Walker については「[2.13 スタックを見積もる](#)」を参照してください。

1.2 著作権について

本ソフトウェアは LLVM Release License に従い、LLVM 技術をベースに開発されました。

その他のソフトウェア構成物は、ルネサス エレクトロニクス株式会社が著作権を有します。

```
=====
LLVM Release License
=====

University of Illinois/NCSA
Open Source License

Copyright (c) 2003-2012 University of Illinois at Urbana-Champaign.
All rights reserved.

Developed by:

    LLVM Team

    University of Illinois at Urbana-Champaign

    http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software
and associated documentation files (the "Software"), to deal with the Software without
restriction, including without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

    * Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimers.

    * Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimers in the documentation and/or other materials provided
with the distribution.

    * Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the
names of its contributors may be used to endorse or promote products derived from this
Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL THE
CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE
SOFTWARE.
```

1.3 特 長

以下に、CC-RH の特長を示します。

- 高性能

高度な最適化を適用し、従来の最適化に加えて大域最適化を実現します。

これにより、コード・サイズ、および実行速度に優れたコードを生成し、かつコンパイル時間も短縮します。

- 高移植性

C プログラムのチェック機能を備えることにより、既存コンパイラ（SuperH RISC engine C/C++ コンパイラ（以降、SHC））向けプログラムからの移行をサポートします。

- 多機能

他のコンポーネントとの連携による静的解析機能などを提供します。

第2章 機能

この章では、CubeSuite+ を使用したビルドの手順、およびビルドに関する主な機能について説明します。

2.1 概要

ここでは、ロード・モジュール、およびユーザ・ライブラリの作成手順について説明します。

2.1.1 ロード・モジュールを作成する

ロード・モジュールの作成手順を以下に示します。

(1) プロジェクトの作成／読み込み

プロジェクトの新規作成、または既存のプロジェクトの読み込みを行います。

備考 プロジェクトの新規作成、および既存のプロジェクトの読み込みについての詳細は、「CubeSuite+ 統合開発環境 ユーザーズマニュアル 起動編」を参照してください。

(2) ビルド対象プロジェクトの設定

ビルド対象とするプロジェクトを設定します（「[2.11 ビルドの設定をする](#)」参照）。

備考 1. プロジェクトにサブプロジェクトがない場合、そのプロジェクトは常にアクティブになります。

2. ビルド・モードを設定する場合は、ビルド・モードを追加してください（「[2.11.6 ビルド・モードを追加する](#)」参照）。

(3) ビルド対象ファイルの設定

ビルド対象ファイルの追加／削除、依存関係の更新などを行います（「[2.3 ビルド対象ファイルを設定する](#)」参照）。

備考 1. ユーザ・ライブラリのプロジェクトへの追加方法については、「[2.7.1 ユーザ・ライブラリを追加する](#)」を参照してください。

2. オブジェクト・ファイル、およびライブラリ・ファイルのリンク順は、ユーザが設定することできます（「[2.11.2 ファイルのリンク順を設定する](#)」参照）。

(4) ロード・モジュールの出力指定

生成するロード・モジュールの種類を設定します（「[2.4 出力ファイルの種類を設定する](#)」参照）。

(5) ビルド・オプションの設定

コンパイラ、アセンブラ、リンカなどに対するオプションを設定します（「[2.5 コンパイル・オプションを設定する](#)」, 「[2.6 アセンブル・オプションを設定する](#)」, 「[2.7 リンク・オプションを設定する](#)」参照）。

(6) ビルドの実行

ビルドを実行します（「[2.12 ビルドを実行する](#)」参照）。

ビルドには、次の種類があります。

- ビルド（「[2.12.1 更新ファイルのビルドを実行する](#)」参照）
- リビルド（「[2.12.2 すべてのファイルのビルドを実行する](#)」参照）
- ラピッド・ビルド（「[2.12.3 他の処理と平行してビルドを実行する](#)」参照）
- バッチ・ビルド（「[2.12.4 ビルド・モードを一括してビルドを実行する](#)」参照）

備考 ビルド処理前、およびビルド処理後に実行したいコマンドがある場合は、[プロパティパネルの\[共通オプション\]タブ](#)の[その他]カテゴリにおいて、[ビルド前に実行するコマンド] プロパティ、および[ビルド後に実行するコマンド] プロパティを設定してください。

ファイル単位でビルド処理前、およびビルド処理後に実行したいコマンドがある場合は、[\[個別コンパイル・オプション\]タブ](#)（Cソース・ファイルの場合）、および[\[個別アセンブル・オプション\]タブ](#)（アセンブリ・ソース・ファイルの場合）において設定することができます。

(7) プロジェクトの保存

プロジェクトの設定内容をプロジェクト・ファイルに保存します。

備考 プロジェクトの保存についての詳細は、「CubeSuite+ 統合開発環境 ユーザーズマニュアル 起動編」を参照してください。

2.1.2 ユーザ・ライブラリを作成する

ユーザ・ライブラリの作成手順を以下に示します。

(1) プロジェクトの作成／読み込み

プロジェクトの新規作成、または既存のプロジェクトの読み込みを行います。

プロジェクトを新規作成する際は、ライブラリ用のプロジェクトを設定します。

備考 プロジェクトの新規作成、および既存のプロジェクトの読み込みについての詳細は、「CubeSuite+ 統合開発環境 ユーザーズマニュアル 起動編」を参照してください。

(2) ビルド対象プロジェクトの設定

ビルド対象とするプロジェクトを設定します（「[2.11 ビルドの設定をする](#)」参照）。

備考 1. プロジェクトにサブプロジェクトがない場合、そのプロジェクトは常にアクティブになります。

2. ビルド・モードを設定する場合は、ビルド・モードを追加してください（「[2.11.6 ビルド・モードを追加する](#)」参照）。

(3) ビルド対象ファイルの設定

ビルド対象ファイルの追加／削除、依存関係の更新などを行います（「[2.3 ビルド対象ファイルを設定する](#)」参照）。

(4) ビルド・オプションの設定

コンパイラ、アセンブラ、ライブラリアンに対するオプションを設定します（「[2.5 コンパイル・オプションを設定する](#)」, 「[2.6 アセンブル・オプションを設定する](#)」, 「[2.9 ライブラリ生成オプションを設定する](#)」参照）。

備考 デバイス共通のライブラリを作成する場合は、[プロパティ パネル](#)の[\[共通オプション\]](#)タブの[\[出力ファイルの種類と場所\]](#)カテゴリにおいて、[\[デバイス共通オブジェクト・ファイルを出力する\]](#)プロパティを設定してください。

(5) ビルドの実行

ビルドを実行します（「[2.12 ビルドを実行する](#)」参照）。

ビルドには、次の種類があります。

- ビルド（「[2.12.1 更新ファイルのビルドを実行する](#)」参照）
- リビルド（「[2.12.2 すべてのファイルのビルドを実行する](#)」参照）
- ラピッド・ビルド（「[2.12.3 他の処理と平行してビルドを実行する](#)」参照）
- バッチ・ビルド（「[2.12.4 ビルド・モードを一括してビルドを実行する](#)」参照）

備考 ビルド処理前、およびビルド処理後に実行したいコマンドがある場合は、[プロパティ パネル](#)の[\[共通オプション\]](#)タブの[\[その他\]](#)カテゴリにおいて、[\[ビルド前に実行するコマンド\]](#)プロパティ、および[\[ビルド後に実行するコマンド\]](#)プロパティを設定してください。

ファイル単位でビルド処理前、およびビルド処理後に実行したいコマンドがある場合は、[\[個別コンパイル・オプション\]](#)タブ（Cソース・ファイルの場合）、および[\[個別アセンブル・オプション\]](#)タブ（アセンブリ・ソース・ファイルの場合）において設定することができます。

(6) プロジェクトの保存

プロジェクトの設定内容をプロジェクト・ファイルに保存します。

備考 プロジェクトの保存についての詳細は、「CubeSuite+ 統合開発環境 ユーザーズマニュアル 起動編」を参照してください。

2.2 ビルド・ツールのバージョンを変更する

プロジェクト（メイン・プロジェクト、またはサブプロジェクト）で使用するビルド・ツール（コンパイラ・パッケージ）のバージョンを変更することができます。

プロジェクト・ツリーでビルド・ツール・ノードを選択したのち、[プロパティパネル](#)の[\[共通オプション\]](#)タブを選択します。[\[バージョン選択\]](#)カテゴリの[\[使用するコンパイラ・パッケージのバージョン\]](#)プロパティで[\[常にインストール済みの最新版\]](#)、または該当バージョンを選択してください。

図 2—1 [\[使用するコンパイラ・パッケージのバージョン\]](#) プロパティ



備考 1. メイン・プロジェクト、およびサブプロジェクトで使用するビルド・ツールが同じ場合、それらのビルド・ツール・ノードをすべて選択し、プロパティを設定することで、ビルド・ツールのバージョンをまとめて変更することができます。

2. ほかの実行環境で作成したプロジェクトを開いた場合など、インストールしていないコンパイラ・パッケージを選択している場合は、そのバージョンも表示します。
3. コンパイラ・パッケージによってオプションに変更がある場合は、選択したバージョンにあわせて、ビルド・ツールの各プロパティの表示を切り替えます。

バージョンの変更により非表示になるプロパティについては、プロジェクト・ファイル中に設定値を残しておき、再表示の際に値を再現します。

なお、オプションの変更は、以下の規則に基づいて行い、変更情報は[出力パネル](#)に表示します。

- 旧バージョンから新バージョンへ変更した場合は、オプションの設定の引き継ぎ、および変換（必要な場合のみ）を行います。
 - 新バージョンから旧バージョンへ変更した場合は、同一オプションの設定の引き継ぎのみを行います。
- 旧バージョンのみに存在するオプションについては、デフォルト値を設定します。

2.3 ビルド対象ファイルを設定する

ビルドを実行する前に、ビルド対象となるファイル（C ソース・ファイル、アセンブリ・ソース・ファイルなど）をプロジェクトに追加しておく必要があります。

ここでは、プロジェクトにおけるファイルの設定に関する操作を説明します。

2.3.1 プロジェクトにファイルを追加する

プロジェクトにファイルを追加するには、次の方法があります。

- 既存のファイルを追加する場合
- 空のファイルを作成して追加する場合

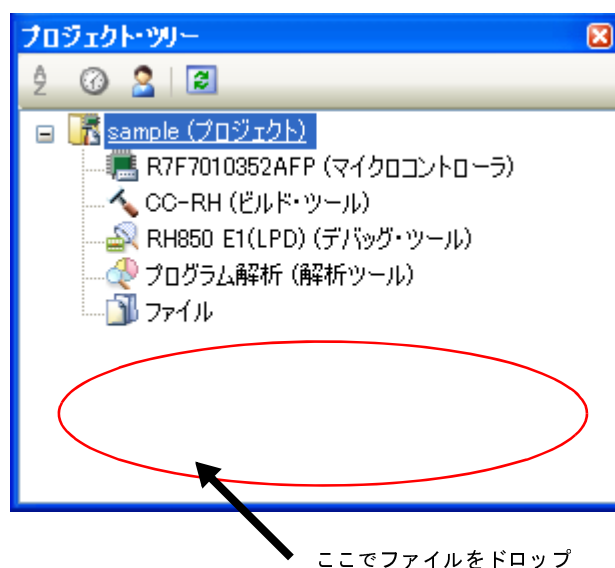
(1) 既存のファイルを追加する場合

(a) ファイル単位で追加する

エクスプローラなどからファイルをドラッグし、プロジェクト・ツリー下部の空白部分にドロップしてください。

ファイルの追加先はファイル・ノード以下となります。

図 2—2 プロジェクト・ツリー パネル（ファイルのドロップ位置）



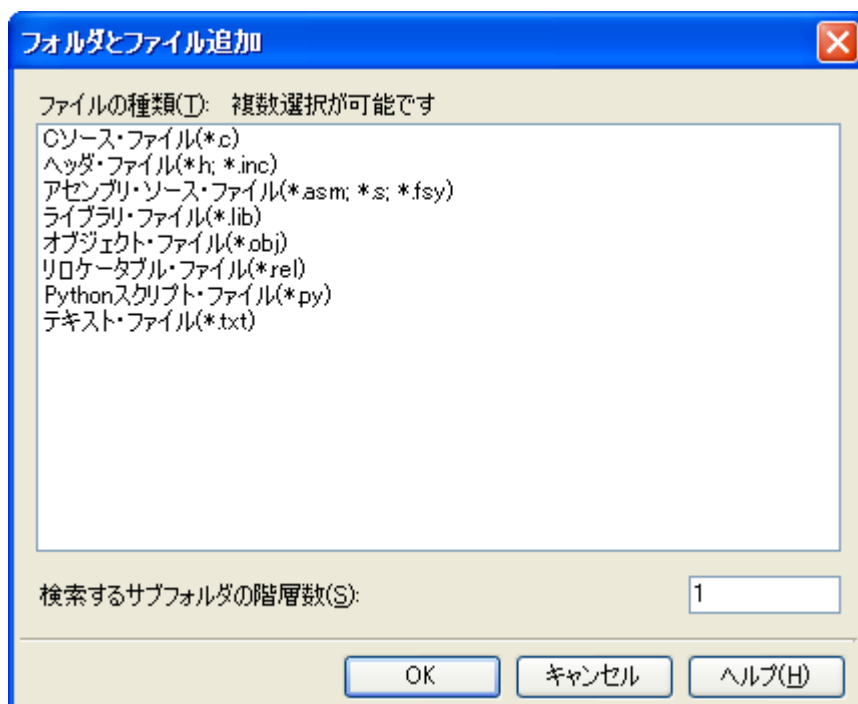
(b) フォルダ単位で追加する

エクスプローラなどからフォルダをドラッグし、プロジェクト・ツリー下部の空白部分にドロップすると、[フォルダとファイル追加 ダイアログ](#)がオープンします。

備考 複数のフォルダを同時にドラッグし、プロジェクト・ツリーにドロップすることにより、複数のフォルダを同時にプロジェクトに追加することもできます。

注意 フォルダ名が 200 文字を越えるフォルダをドロップした場合、201 文字目以降は切り捨てたカテゴリ名で、プロジェクト・ツリーに追加します。

図 2—3 フォルダとファイル追加 ダイアログ



ダイアログ上で、プロジェクトに追加するファイルの種類を選択し、プロジェクトに追加するサブフォルダの階層数を指定したのち、[OK] ボタンをクリックしてください。

備考 ファイルの種類は、[Ctrl] キー＋左クリック、または [Shift] キー＋左クリックにより、複数選択することができます。

何も選択しない場合は、すべての種類を選択したものとみなします。

フォルダの追加先はファイル・ノード以下となります。

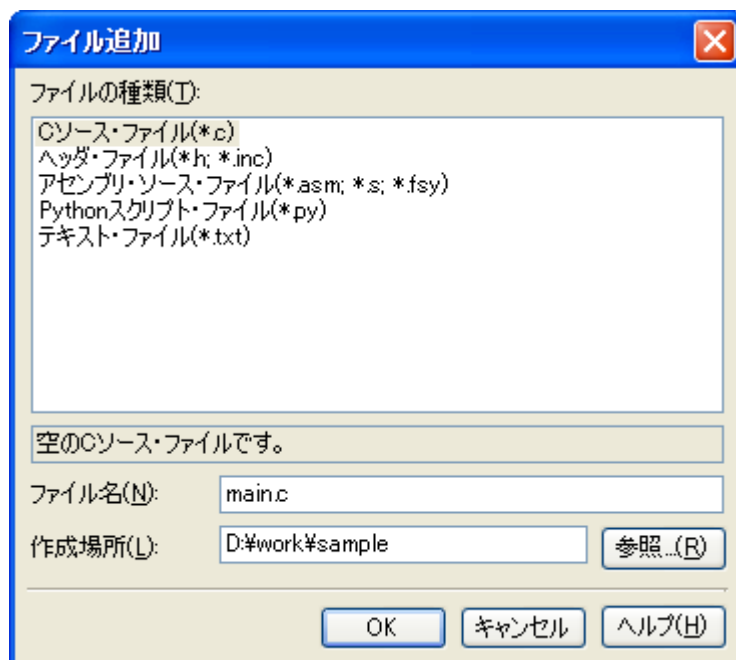
なお、フォルダはプロジェクト・ツリーではカテゴリとなります。

備考 ユーザが作成したカテゴリ・ノードが存在する場合、カテゴリ・ノード上でファイルをドロップすると、カテゴリ・ノード以下に追加することができます（カテゴリ・ノードについては、「[2.3.4 ファイルをカテゴリに分類する](#)」を参照してください）。

(2) 空のファイルを作成して追加する場合

プロジェクト・ツリー でプロジェクト・ノード、サブプロジェクト・ノード、ファイル・ノードのいずれかを選択し、コンテキスト・メニューの [追加] → [新しいファイルを追加 ...] を選択すると、[ファイル追加ダイアログ](#) がオープンします。

図 2—4 ファイル追加 ダイアログ



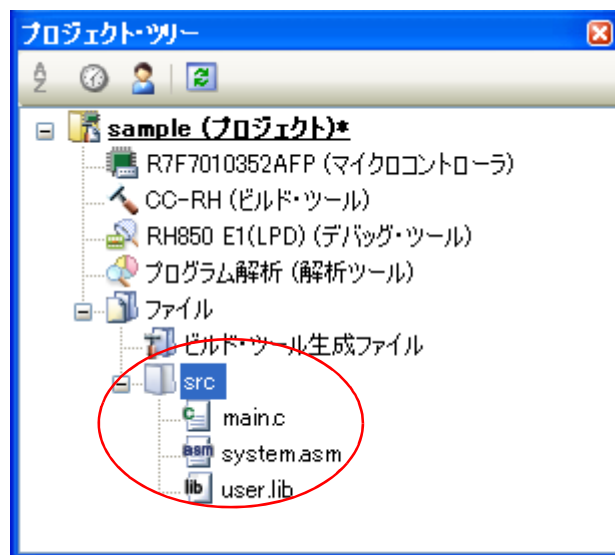
ダイアログ上で、新しく作成するファイルを指定し、[OK] ボタンをクリックしてください。
ファイルの追加先はファイル・ノード以下となります。

ファイル、およびフォルダ追加後のプロジェクト・ツリーは、以下のようになります。

図 2—5 プロジェクト・ツリー パネル (ファイル main.c 追加後)



図 2—6 プロジェクト・ツリー パネル (フォルダ src 追加後)



備考 ファイル・ノード以下におけるファイルの追加位置は、現在のファイルの表示順の設定に依存します。
 ファイルの表示順の変更方法については、「[2.3.5 ファイルの表示順を変更する](#)」を参照してください。

注意 1. パスが異なれば、同名のソース・ファイルを追加することができます。

ただし、それらの出力ファイル名の設定がデフォルトのままの場合、出力ファイル名が同名になるため、ビルドを正しく実行することができません(例えば、D: ¥ sample1 ¥ func.c, D: ¥ sample2 ¥ func.c を追加した場合、これらの出力ファイル名は、デフォルトではどちらも func.obj となります)。

この問題を回避するために、ソース・ファイルの個別ビルド・オプションで、出力ファイル名をそれぞれ異なる名前に設定してください。

C ソース・ファイルの出力ファイル名の変更は、[\[個別コンパイル・オプション\]](#) タブの [\[出力ファイル\]](#) カテゴリの [\[オブジェクト・ファイル名\]](#) プロパティで行います。

アセンブリ・ソース・ファイルの出力ファイル名の変更は、[\[個別アセンブル・オプション\]](#) タブの [\[出力ファイル\]](#) カテゴリの [\[オブジェクト・ファイル名\]](#) プロパティで行います。

個別ビルド・オプションの設定方法については、「[2.10.2 ファイル単位でコンパイル/アセンブル・オプションを設定する](#)」を参照してください。

2. 同名のソース・ファイルを追加した場合、デバッグ時に対象のソースをオープンすることができません。
3. プロジェクトに追加可能なファイル数は、メイン・プロジェクト、およびサブプロジェクトごとに 5000 個までです。

新しいファイルを追加した場合、[ファイル追加 ダイアログ](#)で指定した場所に、空のファイルを作成します。

プロジェクト・ツリー でファイル名をダブルクリックすることにより、[エディタ パネル](#)をオープンし、ファイルを編集することができます。

以下に、[エディタ パネル](#)でオープン可能なファイルを示します。

- C ソース・ファイル (*.c)
- アセンブリ・ソース・ファイル (*.asm, *.s)

- シンボル・アドレス・ファイル (*.fsy)
- ヘッダ・ファイル (*.h, *.inc)
- アセンブリ・プログラム用リスト・ファイル (*.lst, *.lis)
- アセンブリ・プログラム用プリプロセッサ展開ファイル (*.exp)
- アセンブル・リスト・ファイル (*.prn)
- リンク順指定ファイル (*.mtls)
- リンク・マップ・ファイル (*.map)
- ライブラリ・リスト・ファイル (*.lbp)
- インテル拡張ヘキサ・ファイル (*.hex)
- モトローラ・Sタイプ・ファイル (*.mot)
- テキスト・ファイル (*.txt)

備考 1. 以下のいずれかの方法により、上記以外のファイルもエディタパネルでオープンすることができます。

- ファイルをドラッグし、エディタパネルにドロップする。
- ファイルを選択し、コンテキスト・メニューの「内部エディタで開く...」を選択する。

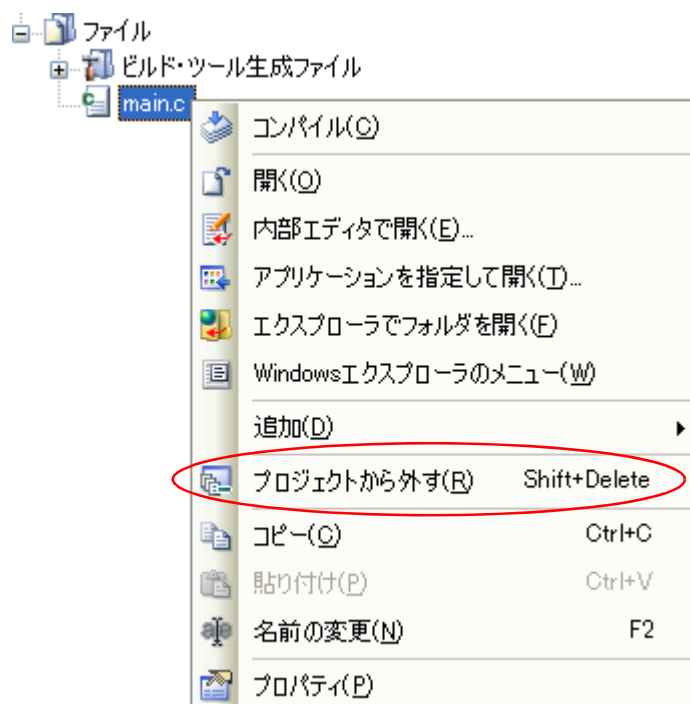
2. オプションダイアログで、外部テキスト・エディタを使用する設定になっている場合は、設定している外部テキスト・エディタでオープンします。

それ以外のファイルは、ホスト OS で関連付けられているアプリケーションで起動します。

2.3.2 プロジェクトからファイルを外す

プロジェクトに追加しているファイルをプロジェクトから外すには、プロジェクト・ツリーでプロジェクトから外すファイルを選択し、コンテキスト・メニューの「プロジェクトから外す」を選択してください。

図 2—7 「プロジェクトから外す」項目



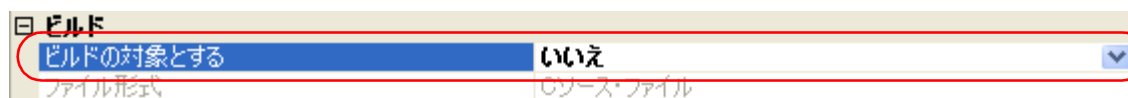
2.3.3 ファイルをビルド対象から外す

プロジェクトに追加しているファイルのうち、特定のファイルをビルド対象から外すことができます。

プロジェクト・ツリーでビルド対象から外すファイルを選択したのち、[プロパティ パネル](#)の[\[ビルド設定\] タブ](#)を選択します。

[ビルド] カテゴリの[ビルドの対象とする] プロパティで[いいえ]を選択してください。

図 2—8 [ビルドの対象とする] プロパティ



備考 この機能を適用できるファイルは、C ソース・ファイル、アセンブリ・ソース・ファイル、オブジェクト・ファイル、ライブラリ・ファイルです。

2.3.4 ファイルをカテゴリに分類する

プロジェクトに追加しているファイルをプロジェクト・ツリー上で見やすくしたり、機能ごとに管理しやすくするために、ファイル・ノード以下にカテゴリ・ノードを作成して、ファイルを分類することができます。

カテゴリ・ノードを作成するには、プロジェクト・ツリーでプロジェクト・ノード、サブプロジェクト・ノード、ファイル・ノードのいずれかを選択し、コンテキスト・メニューの[追加] → [新しいカテゴリを追加]を選択してください。

図 2—9 [新しいカテゴリを追加] 項目 (ファイル・ノードの場合)

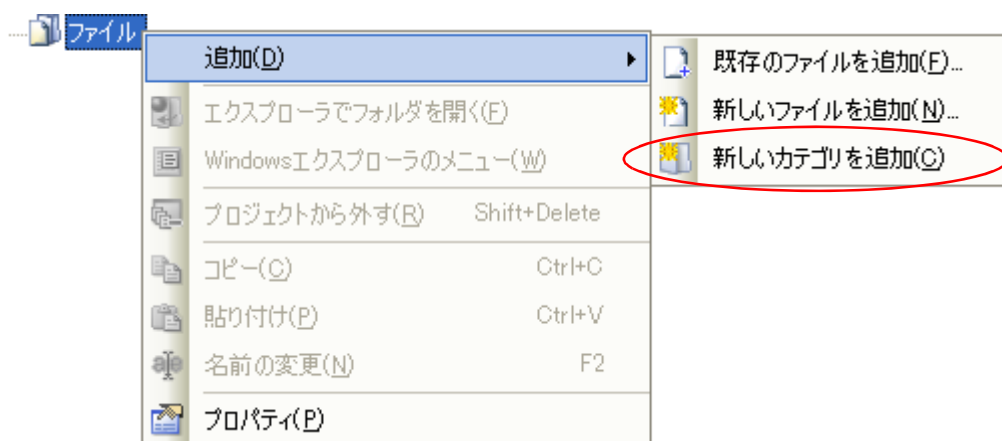
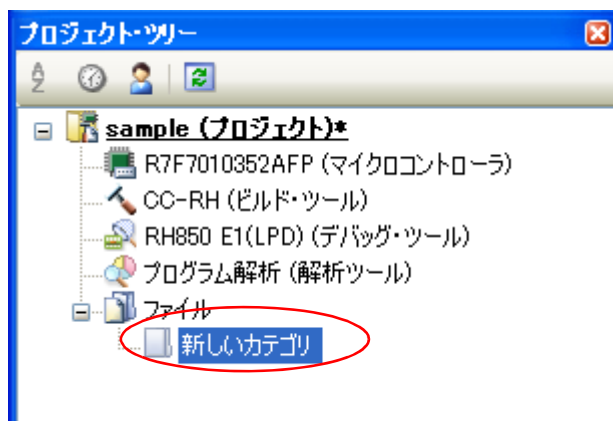


図 2—10 プロジェクト・ツリー パネル (カテゴリ・ノード追加後)



備考 1. カテゴリ名は、デフォルトで“新しいカテゴリ”となります。

カテゴリ名の変更は、カテゴリ・ノードのコンテキスト・メニューの「名前の変更」から行うことができます。

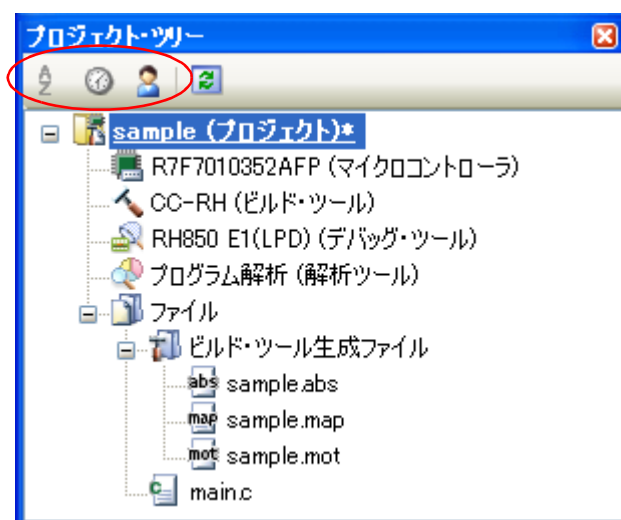
2. すでに存在するカテゴリ・ノードと同名のカテゴリ・ノードを追加することもできます。
3. カテゴリのネスト数の上限は20です。

作成したカテゴリ・ノードにファイルを分類するには、ファイルのドラッグ・アンド・ドロップにより行うことができます。

2.3.5 ファイルの表示順を変更する

プロジェクト・ツリー上のボタンで、ファイル、およびカテゴリ・ノードの表示順を変更することができます。

図 2—11 ツールバー (プロジェクト・ツリー パネル)



プロジェクト・ツリーパネルのツールバーで、以下のいずれかのボタンを選択してください。

ボタン	説明
  	カテゴリ・ノード、およびファイルを名前順でソートします。  : 昇順  : 降順  : 昇順
  	カテゴリ・ノード、およびファイルをタイムスタンプ順でソートします。  : 降順  : 昇順  : 降順
	カテゴリ・ノードとファイルをユーザが指定した順で表示します（デフォルト）。 カテゴリ・ノード、およびファイルをドラッグ・アンド・ドロップすることにより、表示順を任意に変更することができます。

2.3.6 ファイルの依存関係を更新する

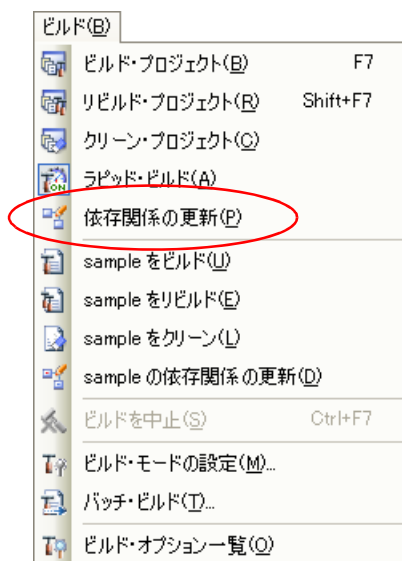
コンパイル・オプションの設定、アセンブル・オプションの設定で、ファイルの依存関係に影響する変更（インクルード・ファイルのパスの変更、C ソース・ファイル、およびアセンブリ・ソース・ファイル中にヘッダ・ファイルのインクルード文を追加など）を行った場合は、該当ファイルの依存関係を更新する必要があります。

ファイルの依存関係の更新は、プロジェクト全体（メイン・プロジェクト、およびサブプロジェクト）、またはアクティブ・プロジェクトに対して行います。

(1) プロジェクト全体の場合

[ビルド] メニュー→ [依存関係の更新] を選択してください。

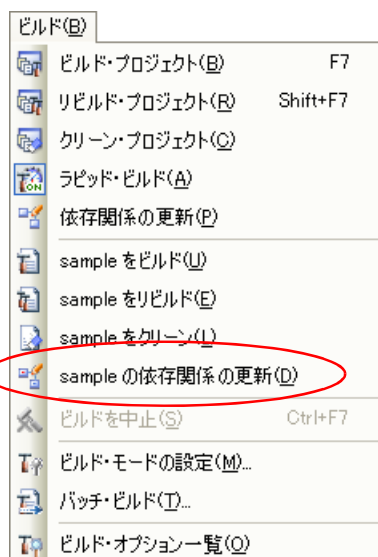
図 2—12 「依存関係の更新」項目



(2) アクティブ・プロジェクトの場合

[ビルド] メニュー→ [アクティブ・プロジェクトの依存関係の更新] を選択してください。

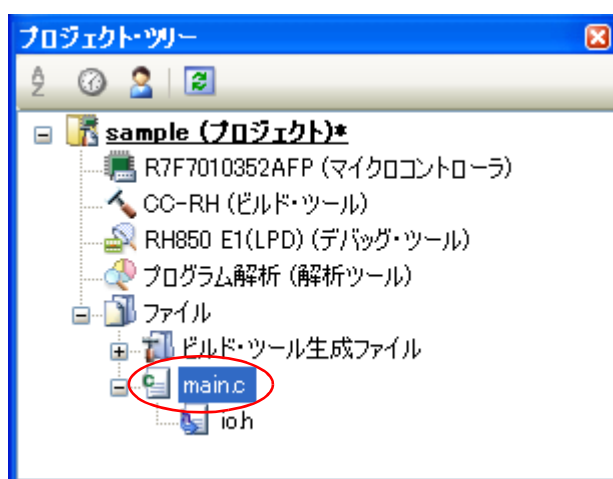
図 2—13 「アクティブ・プロジェクトの依存関係の更新」項目



備考 ファイルの依存関係を更新する際、[エディタ パネル](#)で編集集中のファイルがある場合は、該当ファイルを一括して保存します。

依存関係ファイル（インクルード・ファイル）は、プロジェクト・ツリー上のソース・ファイルにぶら下げて表示することができます。

図 2—14 プロジェクト・ツリー パネル（依存関係ファイル表示後）



なお、依存関係ファイルの表示は、以下のタイミングで更新します。

- プロジェクトを読み込んだのち、初めてビルドを実行したとき
- ツールバーの  をクリックしたとき
- [ビルド] メニュー→[依存関係の更新] を選択したとき
- [ビルド] メニュー→[アクティブ・プロジェクトの依存関係の更新] を選択したとき

備考 1. 依存関係ファイルの表示は、[オプション ダイアログ](#)の[全般 - ビルド／デバッグ] カテゴリの[プロジェクト・ツリーに依存関係ファイルを表示する] がチェック状態の場合のみ有効となります。

2. プロジェクト・ツリーに表示している依存関係ファイルの情報は、プロジェクト・ファイルには保存しません。

注意 1. CubeSuite+ は、インクルード・ファイルの依存関係のチェックにおいて、`#if` などの条件文やコメントをサポートしません。

そのため、ビルドに不要なインクルード・ファイルを、必要なファイルであると認識する場合があります（以下の例において、`header1.h`、`header5.h` は、ビルドに必要であると判断します）。

```
#if      0

#include  "header1.h"      /* 依存関係ありと判断する */
#else
/* ! zero */
#include  "header2.h"      /* 依存関係あり */
#endif

#define   AAA
#ifdef   AAA
#include  "header3.h"      /* 依存関係あり */
#else
#include  "header4.h"      /* 依存関係あり */
#endif

/*
#include  "header5.h"      /* 依存関係ありと判断する */
*/
```

2. CubeSuite+ は、インクルード・ファイルの依存関係のチェックにおいて、コメント文のあとに記述したインクルード文をサポートしません。

そのため、ビルドに必要なインクルード・ファイルを、不要なファイルであると認識する場合があります（以下の例において、`header6.h`、`header7.h` は、ビルドに不要であると判断します）。

```
/* comment */ #include  "header6.h"      /* 依存関係なしと判断する */

/*
comment
*/ #include  "header7.h"      /* 依存関係なしと判断する */
```

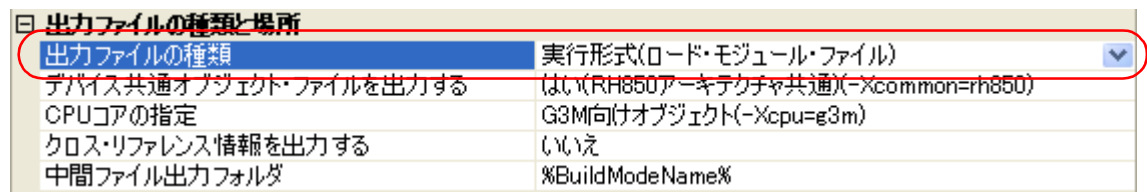

2.4 出力ファイルの種類を設定する

ビルドの生成物として出力するファイルの種類を設定します。

プロジェクト・ツリーでビルド・ツール・ノードを選択し、プロパティパネルの[共通オプション]タブを選択します。

[出力ファイルの種類と場所] カテゴリの [出力ファイルの種類] プロパティにおいて、ファイルの種類を選択してください。

図 2—15 [出力ファイルの種類] プロパティ



(1) [実行形式 (ロード・モジュール・ファイル)] を選択した場合 (デフォルト)

- ロード・モジュール・ファイルを生成します。
- ロード・モジュール・ファイルがデバッグ対象となります。

(2) [実行形式 (ヘキサ・ファイル)] を選択した場合

- ヘキサ・ファイルを生成します。
- ヘキサ・ファイルがデバッグ対象となります。

注意 ライブラリ用のプロジェクトの場合、本プロパティは常に [ライブラリ形式] となり、変更することはできません。

2.4.1 出力ファイル名を変更する

ビルド・ツールが出力するロード・モジュール・ファイル、ヘキサ・ファイル、ライブラリ・ファイルは、デフォルトで次の名前を設定しています。

- ロード・モジュール・ファイル名 : %ProjectName%.abs
- ヘキサ・ファイル名 : %ProjectName%.mot
- ライブラリ・ファイル名 : lib%ProjectName%.lib

備考 “%ProjectName%” はプレースホルダで、プロジェクト名に置換します。

これらのファイル名の変更方法を、以下に示します。

(1) ロード・モジュール・ファイル名を変更する場合

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの \[リンク・オプション\] タブ](#)を選択します。

〔出力ファイル〕カテゴリの〔出力ファイル名〕プロパティにおいて、変更するロード・モジュール・ファイル名を入力してください。

図 2—16 〔出力ファイル名〕プロパティ



本プロパティは、次のプレースホルダに対応しています。

%ActiveProjectName% : アクティブ・プロジェクト名に置換します。

%MainProjectName% : メイン・プロジェクト名に置換します。

%ProjectName% : プロジェクト名に置換します。

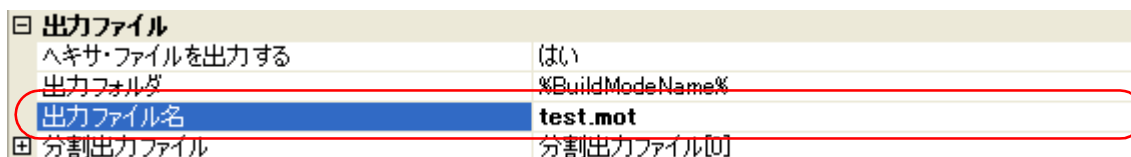
備考 [\[共通オプション\] タブ](#)の〔よく使うオプション (リンク)〕カテゴリの〔出力ファイル名〕プロパティでも、同様に変更することができます。

(2) ヘキサ・ファイル名を変更する場合

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの \[ヘキサ出力オプション\] タブ](#)を選択します。

〔出力ファイル〕カテゴリの〔出力ファイル名〕プロパティにおいて、変更するヘキサ・ファイル名を入力してください。

図 2—17 〔出力ファイル名〕プロパティ



本プロパティは、次のプレースホルダに対応しています。

%ActiveProjectName% : アクティブ・プロジェクト名に置換します。

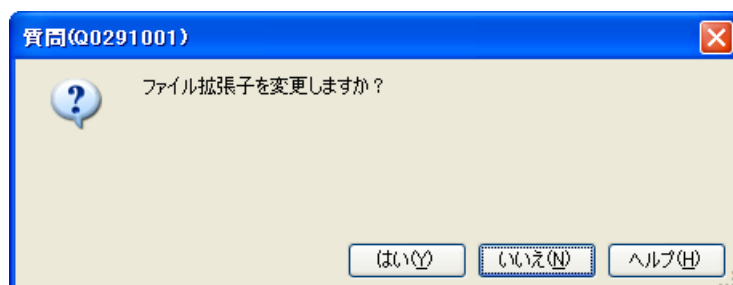
%MainProjectName% : メイン・プロジェクト名に置換します。

%ProjectName% : プロジェクト名に置換します。

備考 [\[共通オプション\] タブ](#)の〔よく使うオプション (ヘキサ出力)〕カテゴリの〔出力ファイル名〕プロパティでも、同様に変更することができます。

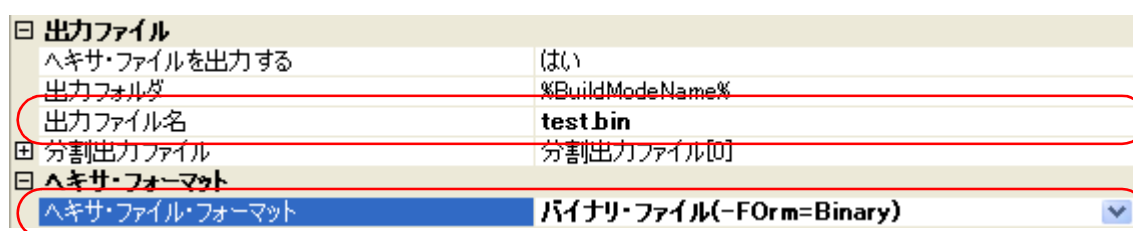
なお、[ヘキサ・フォーマット] カテゴリの [ヘキサ・ファイル・フォーマット] プロパティを変更すると、以下のメッセージ ダイアログがオープンします。

図 2—18 メッセージ ダイアログ



ダイアログ上で [はい] を選択すると、[ヘキサ・ファイル・フォーマット] プロパティで選択したフォーマットに応じて、出力ファイル名の拡張子を変更することができます。

図 2—19 [出力ファイル名]、および [ヘキサ・ファイル・フォーマット] プロパティ

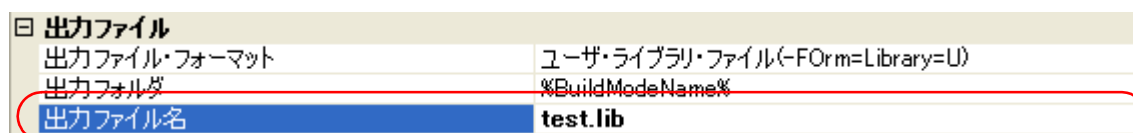


(3) ライブラリ・ファイル名を変更する場合

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの \[ライブラリ生成オプション\] タブ](#) を選択します。

[出力ファイル] カテゴリの [出力ファイル名] プロパティにおいて、変更するライブラリ・ファイル名を入力してください。

図 2—20 [出力ファイル名] プロパティ



本プロパティは、次のプレースホルダに対応しています。

%ActiveProjectName% : アクティブ・プロジェクト名に置換します。

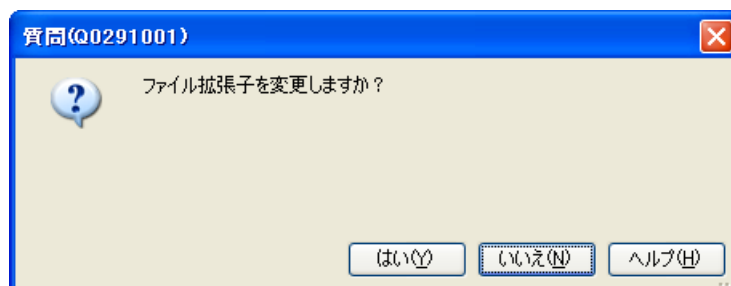
%MainProjectName% : メイン・プロジェクト名に置換します。

%ProjectName% : プロジェクト名に置換します。

備考 [共通オプション] タブの [よく使うオプション (ライブラリ生成)] カテゴリの [出力ファイル名] プロパティでも、同様に変更することができます。

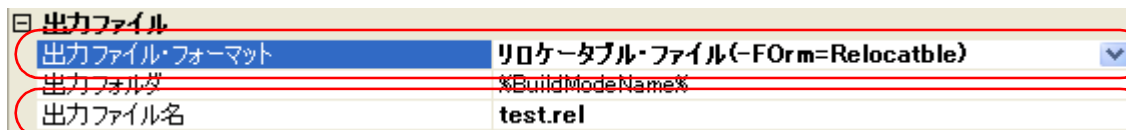
なお、[出力ファイル・フォーマット] プロパティを変更すると、以下のメッセージ ダイアログがオープンします。

図 2—21 メッセージ ダイアログ



ダイアログ上で [はい] を選択すると、[出力ファイル・フォーマット] プロパティで選択したフォーマットに応じて、出力ファイル名の拡張子を変更することができます。

図 2—22 [出力ファイル・フォーマット], および [出力ファイル名] プロパティ



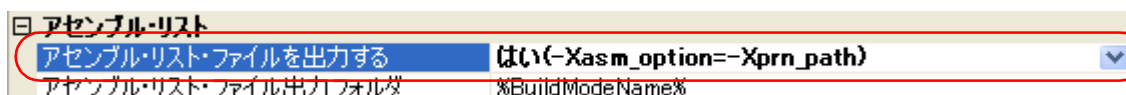
2.4.2 アセンブル・リストを出力する

アセンブル・リスト (アセンブル結果のコード) は、アセンブル・リスト・ファイルに出力します。

プロジェクト・ツリーでビルド・ツール・ノードを選択し、プロパティ パネルの [コンパイル・オプション] タブを選択します。

アセンブル・リスト・ファイルを出力するには、[アセンブル・リスト] カテゴリの [アセンブル・リスト・ファイルを出力する] プロパティで [はい (-Xasm_option=-Xprn_path)] を選択してください。

図 2—23 [アセンブル・リスト・ファイルを出力する] プロパティ



アセンブル・リスト・ファイルを出力する場合、出力フォルダを設定することができます。

(1) 出力フォルダの設定

「アセンブル・リスト・ファイル出力フォルダ」プロパティにおいて、テキスト・ボックスへの直接入力、または [...] ボタンにより行います。

本プロパティは、次のプレースホルダに対応しています。

%BuildModeName% : ビルド・モード名に置換します。

デフォルトでは、“%BuildModeName%”を設定しています。

なお、ファイル名は、ソース・ファイルの拡張子を“.prn”で置き換えた名前となります。

備考 アセンブル・リスト・ファイルについての詳細は、「[3.1 アセンブル・リスト・ファイル](#)」を参照してください。

2.4.3 マップ情報を出力する

マップ情報（リンク結果の情報）は、リンク・マップ・ファイルに出力します。

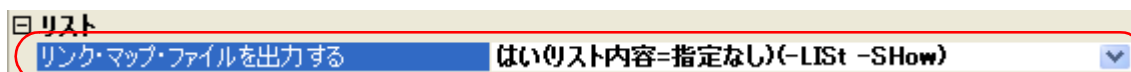
プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネル](#)の「[リンク・オプション](#)」タブを選択します。

リンク・マップ・ファイルを出力するには、[リスト] カテゴリの「リンク・マップ・ファイルを出力する」プロパティを設定します。

(1) 出力フォーマットに従った情報を出力する場合

「リンク・マップ・ファイルを出力する」プロパティで「はい (リスト内容 = 指定なし) (-LIST -SHow)」または「はい (リスト内容 = すべて) (-LIST -SHow=ALL)」を選択してください。

図 2—24 「リンク・マップ・ファイルを出力する」プロパティ（出力フォーマットに従った情報を出力する場合）



備考 -SHow オプションと -SHow=ALL オプションの違いについては、「[-SHow](#)」の「備考」を参照してください。

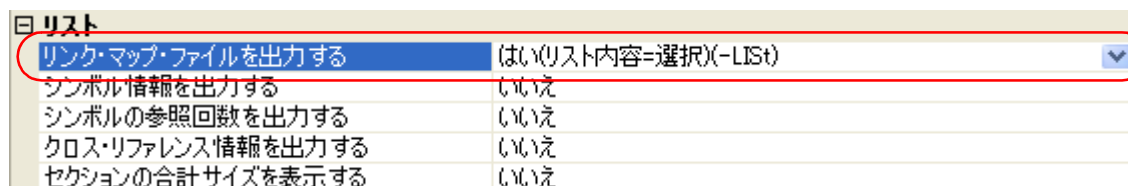
(2) 出力する情報を指定する場合

「リンク・マップ・ファイルを出力する」プロパティで「はい (リスト内容 = 選択) (-LIST)」を選択すると、以下のプロパティを表示します。

- 「シンボル情報を出力する」プロパティ
- 「シンボルの参照回数を出力する」プロパティ
- 「クロス・リファレンス情報を出力する」プロパティ
- 「セクションの合計サイズを表示する」プロパティ

出力する情報の各プロパティで「はい」を選択してください。

図 2—25 「リンク・マップ・ファイルを出力する」プロパティ（出力する情報を指定する場合）



リンク・マップ・ファイルの出力先は、プロジェクト・フォルダです。

また、プロジェクト・ツリーのビルド・ツール生成ファイル・ノードにも表示します。

~~ファイル名は、出力ファイル名の拡張子を“.map”で置き換えた名前となります。~~

ファイル名は、プロジェクト・ファイル名の拡張子を“.map”で置き換えた名前となります。

備考 リンク・マップ・ファイルについての詳細は、「[3.2 リンク・マップ・ファイル](#)」を参照してください。

2.4.4 ライブラリ情報を出力する

ライブラリ情報（ライブラリ作成結果の情報）は、ライブラリ・リスト・ファイルに出力します。

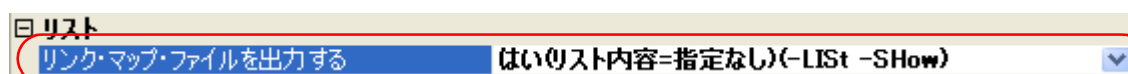
プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティパネルの「ライブラリ生成オプション」タブ](#)を選択します。

ライブラリ・リスト・ファイルを出力するには、[リスト] カテゴリの [リンク・マップ・ファイルを出力する] プロパティを設定します。

(1) 出力フォーマットに従った情報を出力する場合

[リンク・マップ・ファイルを出力する] プロパティで [はい (リスト内容 = 指定なし) (-LIST -SHow)]、または [はい (リスト内容 = すべて) (-LIST -SHow=ALL)] を選択してください。

図 2—26 「リンク・マップ・ファイルを出力する」プロパティ（出力フォーマットに従った情報を出力する場合）



備考 -SHow オプションと -SHow=ALL オプションの違いについては、「[-SHow](#)」の [備考] を参照してください。

(2) 出力する情報を指定する場合

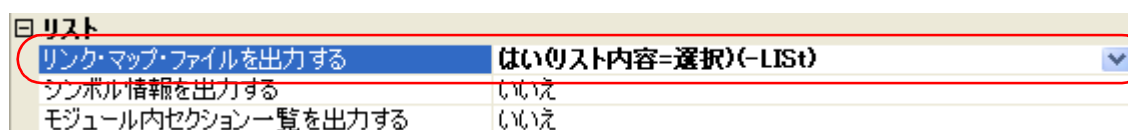
[リンク・マップ・ファイルを出力する] プロパティで [はい (リスト内容 = 選択) (-LIST)] を選択すると、以下のプロパティを表示します。

- [シンボル情報を出力する] プロパティ
- [モジュール内セクション一覧を出力する] プロパティ^{注1}
- [クロス・リファレンス情報を出力する] プロパティ^{注2}
- [セクションの合計サイズを表示する] プロパティ^{注2}

- 注 1. [出力ファイル] カテゴリの [出力ファイル・フォーマット] プロパティで [ユーザ・ライブラリ・ファイル (-FOrm=Library=U)], または [システム・ライブラリ・ファイル (-FOrm=Library=S)] を選択した場合のみ表示します。
2. [出力ファイル] カテゴリの [出力ファイル・フォーマット] プロパティで [リロケートブル・ファイル (-FOrm=Relocate)] を選択した場合のみ表示します。

出力する情報の各プロパティで [はい] を選択してください。

図 2—27 [リンク・マップ・ファイルを出力する] プロパティ (出力する情報を指定する場合)



ライブラリ・リスト・ファイルの出力先は、プロジェクト・フォルダです。

また、プロジェクト・ツリーのビルド・ツール生成ファイル・ノードにも表示します。

~~ファイル名は、出力ファイル名の拡張子を“.lbp”で置き換えた名前となります。~~

ファイル名は、プロジェクト・ファイル名の拡張子を“.lbp”で置き換えた名前となります。

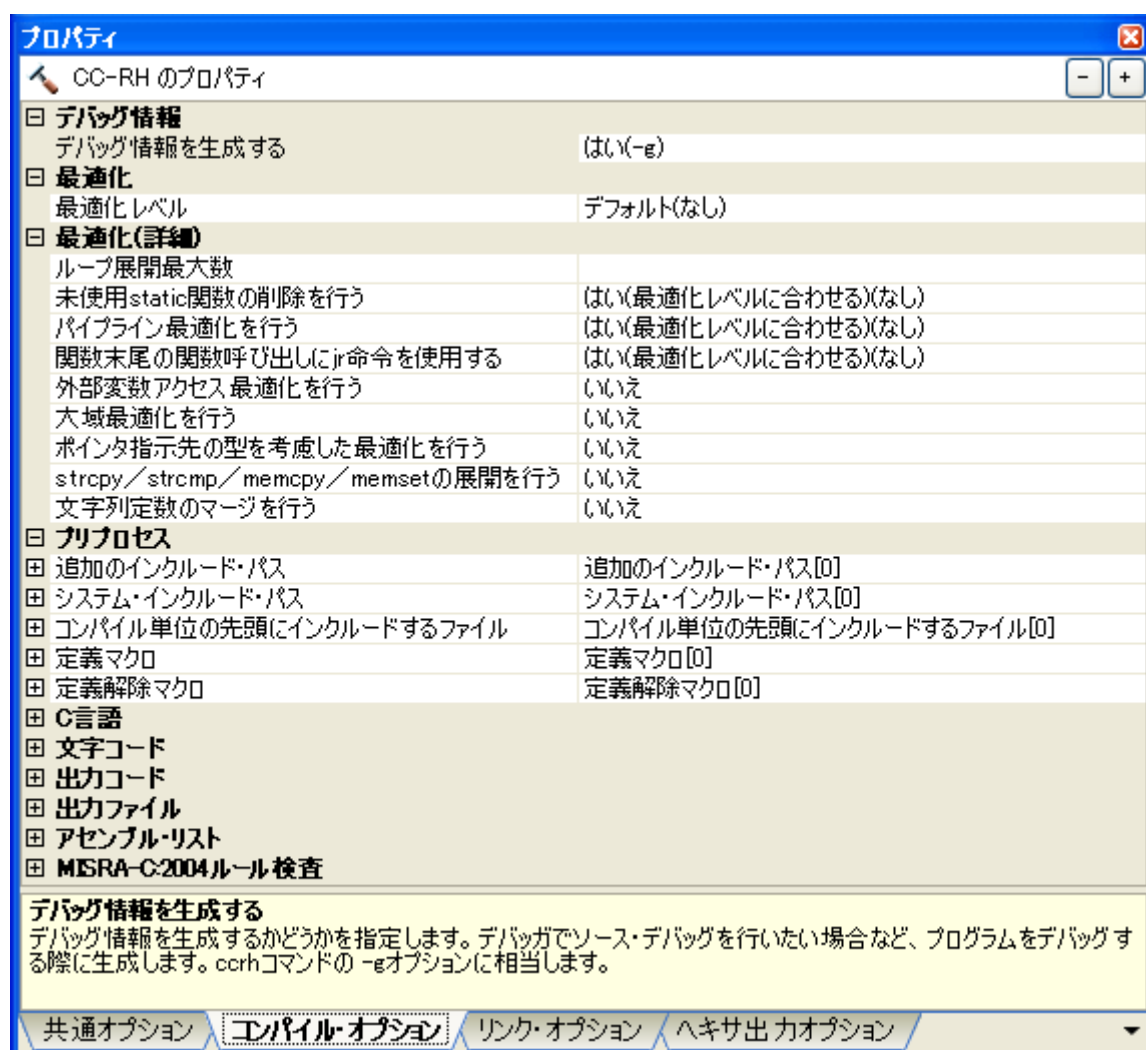
備考 ライブラリ・リスト・ファイルについての詳細は、「[3.4 ライブラリ・リスト・ファイル](#)」を参照してください。

2.5 コンパイル・オプションを設定する

コンパイル・フェーズに対するオプションを設定するには、プロジェクト・ツリーでビルド・ツール・ノードを選択し、**プロパティ パネル**の**［コンパイル・オプション］**タブを選択してください。

タブ上で各プロパティを設定することにより、対応するコンパイル・オプションを設定することができます。

図 2—28 プロパティ パネル：［コンパイル・オプション］タブ



備考 よく使うオプションについては、**［共通オプション］**タブの**［よく使うオプション（コンパイル）］**カテゴリにまとめられています。

2.5.1 コード・サイズを優先した最適化を行う

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの \[コンパイル・オプション\] タブ](#)を選択します。

コード・サイズを優先した最適化を行うには、[最適化] カテゴリの [最適化レベル] プロパティで [サイズ優先 (-Osize)] を選択してください。

図 2—29 [最適化レベル] プロパティ (コード・サイズ優先の場合)



備考 [\[共通オプション\] タブ](#)の [よく使うオプション (コンパイル)] カテゴリの [最適化レベル] プロパティでも、同様に設定することができます。

2.5.2 実行速度を優先した最適化を行う

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの \[コンパイル・オプション\] タブ](#)を選択します。

実行速度を優先した最適化を行うには、[最適化] カテゴリの [最適化レベル] プロパティで [実行速度優先 (-Ospeed)] を選択してください。

図 2—30 [最適化レベル] プロパティ (実行速度優先の場合)



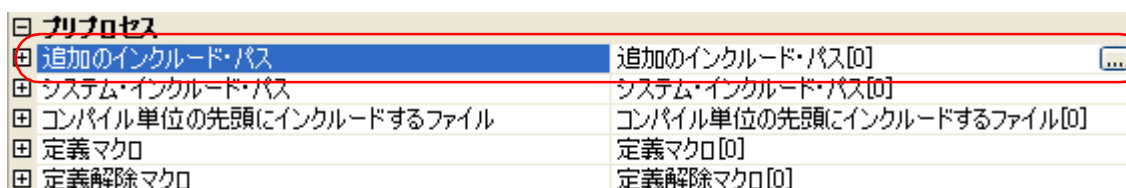
備考 [\[共通オプション\] タブ](#)の [よく使うオプション (コンパイル)] カテゴリの [最適化レベル] プロパティでも、同様に設定することができます。

2.5.3 インクルード・パスを追加する

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの \[コンパイル・オプション\] タブ](#)を選択します。

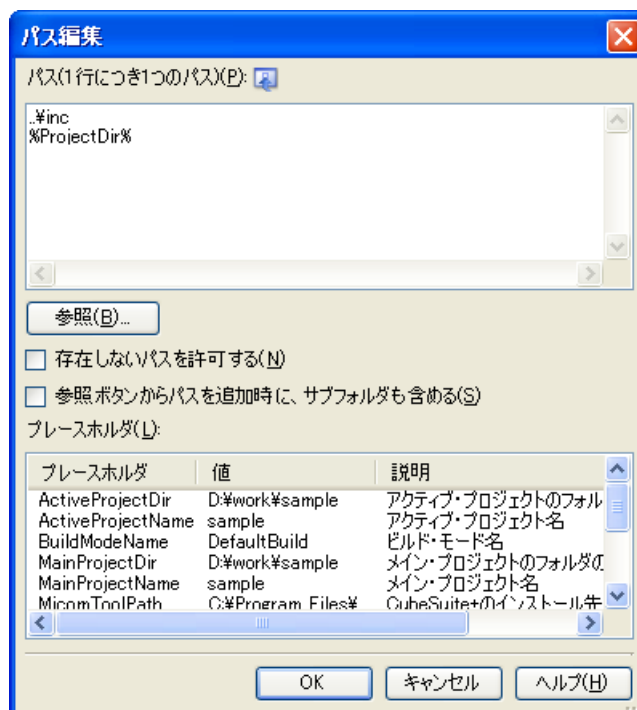
インクルード・パスの設定は、[プリプロセス] カテゴリの [追加のインクルード・パス] プロパティで行います。

図 2—31 [追加のインクルード・パス] プロパティ



[...] ボタンをクリックすると、[パス編集 ダイアログ](#)がオープンします。

図 2—32 パス編集 ダイアログ



[パス (1 行につき 1 つのパス)] にインクルード・パスを 1 行に 1 つずつ入力します。

1 行に 259 文字まで、256 行まで指定可能です。

備考 1. 本プロパティは、プレースホルダに対応しています。

[プレースホルダ] において行をダブルクリックすると、プレースホルダが [パス (1 行につき 1 つのパス)] に反映されます。

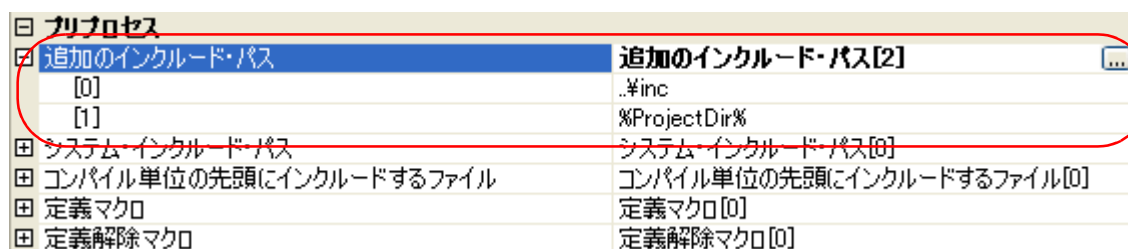
2. インクルード・パスは、以下のいずれかの方法で指定することも可能です。

- エクスプローラなどからフォルダをドラッグ・アンド・ドロップ
- [参照 ...] ボタンをクリックし、[フォルダの参照 ダイアログ](#)によるフォルダの選択
- [プレースホルダ] において行をダブルクリック

3. [参照ボタンからパスを追加時に、サブフォルダも含める] をチェックしたのち、[参照 ...] ボタンからパスの指定を行うと、指定したパスとそのサブフォルダ 5 階層分までのパスを [パス (1 行につき 1 つのパス)] に追加します。

[OK] ボタンをクリックすると、入力したインクルード・パスがサブプロパティとして表示されます。

図 2—33 「追加のインクルード・パス」 プロパティ（インクルード・パス追加後）



インクルード・パスの変更は、[...] ボタン、またはサブプロパティのテキスト・ボックスへの直接入力により行うことができます。

また、プロジェクト・ツリーにインクルード・ファイルを追加すると、そのインクルード・パスをサブプロパティの一番最初に自動で追加します。

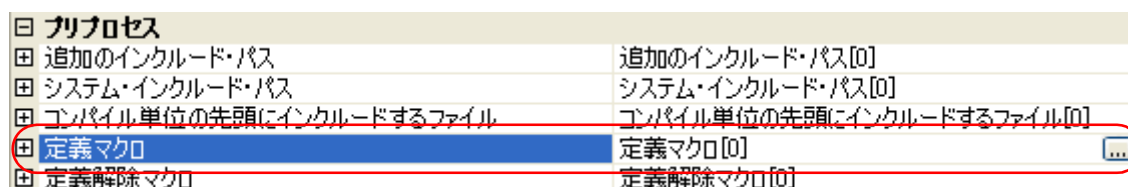
備考 [\[共通オプション\] タブ](#)の「よく使うオプション（コンパイル）」カテゴリの「追加のインクルード・パス」プロパティでも、同様に設定することができます。

2.5.4 定義マクロを設定する

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの「コンパイル・オプション」タブ](#)を選択します。

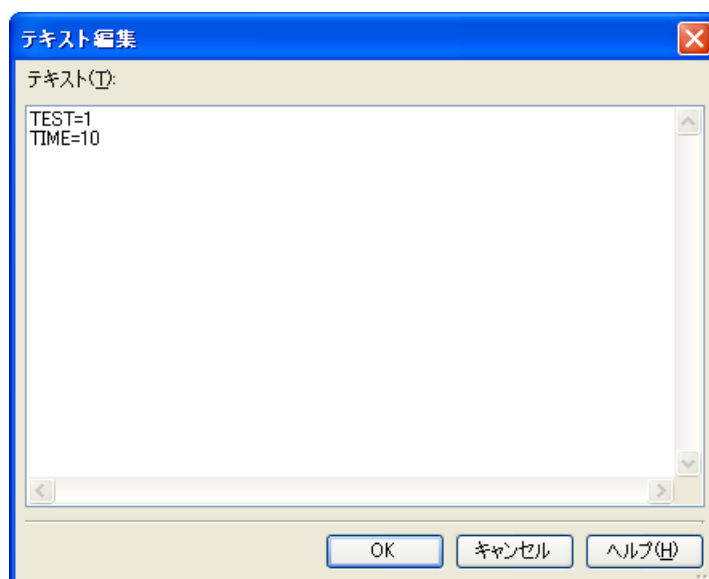
定義マクロの設定は、「プリプロセス」カテゴリの「定義マクロ」プロパティで行います。

図 2—34 「定義マクロ」 プロパティ



[...] ボタンをクリックすると、[テキスト編集 ダイアログ](#)がオープンします。

図 2—35 テキスト編集 ダイアログ



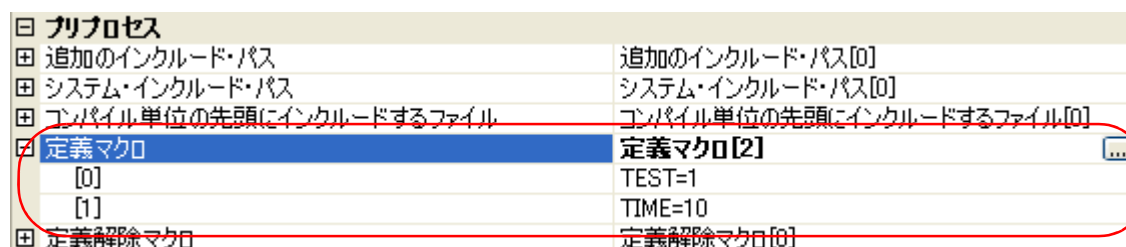
「テキスト」に定義マクロを「マクロ名= 定義値」の形式で 1 行に 1 つずつ入力します。

1 行に 256 文字まで、256 行まで指定可能です。

「= 定義値」の部分は省略可能で、省略した場合、定義値を 1 とします。

「OK」ボタンをクリックすると、入力した定義マクロがサブプロパティとして表示されます。

図 2—36 「定義マクロ」プロパティ（定義マクロ設定後）



定義マクロの変更は、[...] ボタン、またはサブプロパティのテキスト・ボックスへの直接入力により行うことができます。

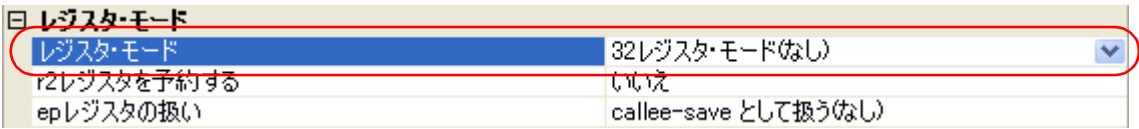
備考 [\[共通オプション\]](#) タブの [\[よく使うオプション \(コンパイル\)\]](#) カテゴリの [\[定義マクロ\]](#) プロパティでも、同様に設定することができます。

2.5.5 レジスタ・モードを変更する

プロジェクト・ツリーでビルド・ツール・ノードを選択し、プロパティパネルの [共通オプション] タブを選択します。

[レジスタ・モード] カテゴリの [レジスタ・モード] プロパティで、レジスタ・モードを変更してください。

図 2—37 [レジスタ・モード] プロパティ



以下のレジスタ・モードを選択することができます。

レジスタ・モード	作業用レジスタ	レジスタ変数用レジスタ
32 レジスタ・モード (なし)	r10 ~ r19	r20 ~ r29
22 レジスタ・モード (-Xreg_mode=22)	r10 ~ r14	r25 ~ r29
汎用レジスタ・モード (-Xreg_mode=common)	r10 ~ r14	r25 ~ r29

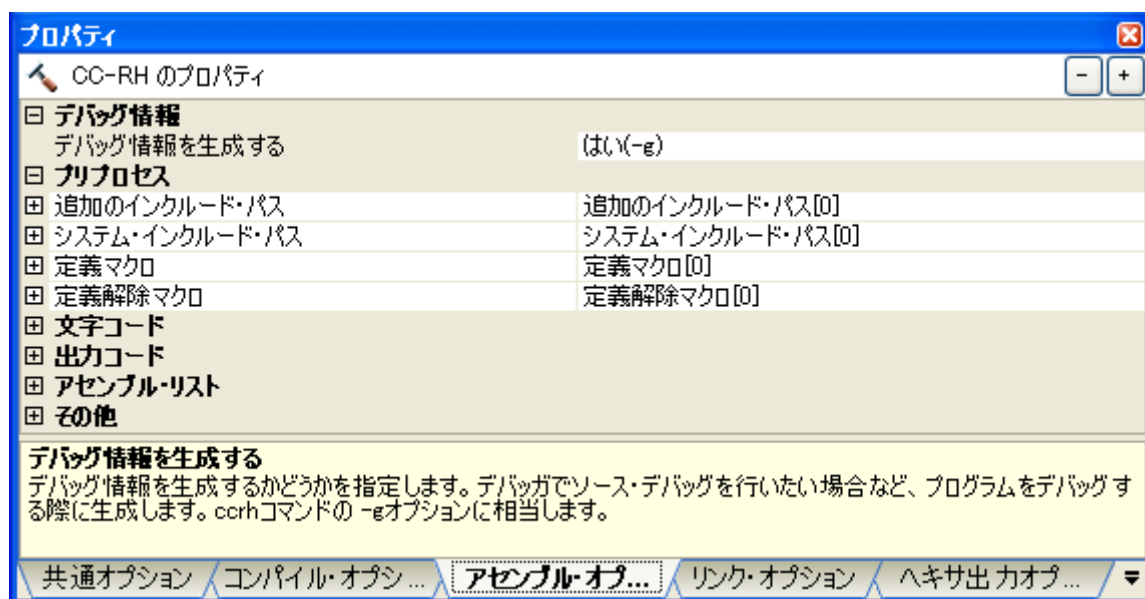
備考 レジスタ・モードについての詳細は、「CubeSuite+ 統合開発環境 ユーザーズマニュアル RH850 コーディング編」を参照してください。

2.6 アセンブル・オプションを設定する

アセンブル・フェーズに対するオプションを設定するには、プロジェクト・ツリーでビルド・ツール・ノードを選択し、**プロパティ パネル**の**「アセンブル・オプション」**タブを選択してください。

タブ上で各プロパティを設定することにより、対応するアセンブル・オプションを設定することができます。

図 2—38 プロパティ パネル：「アセンブル・オプション」タブ



備考 よく使うオプションについては、**「共通オプション」**タブの**「よく使うオプション（アセンブル）」**カテゴリにまとめられています。

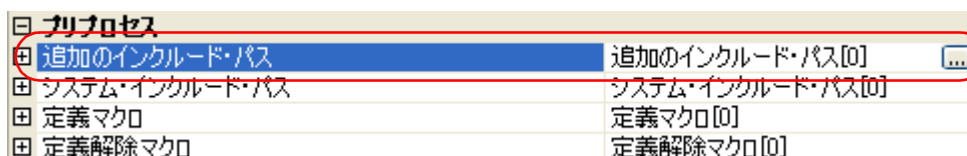
注意 上記のタブは、**「共通オプション」**タブの**「ビルド方法」**カテゴリの**「一括ビルドを行う」**プロパティで**「いいえ」**を選択した場合のみ表示します。

2.6.1 インクルード・パスを追加する

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの \[アセンブル・オプション\] タブ](#)を選択します。

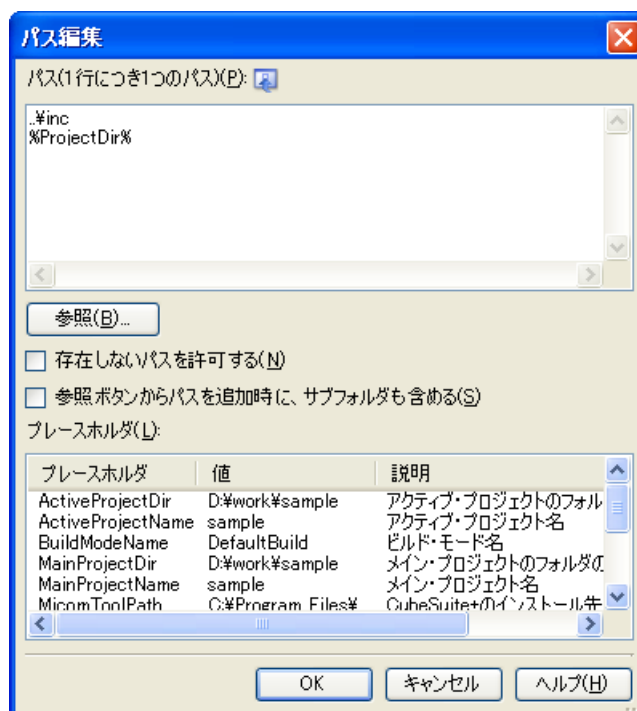
インクルード・パスの設定は、[プリプロセス] カテゴリの [追加のインクルード・パス] プロパティで行います。

図 2—39 [追加のインクルード・パス] プロパティ



[...] ボタンをクリックすると、[パス編集 ダイアログ](#)がオープンします。

図 2—40 パス編集 ダイアログ



[パス (1 行につき 1 つのパス)] にインクルード・パスを 1 行に 1 つずつ入力します。

1 行に 259 文字まで、256 行まで指定可能です。

備考 1. 本プロパティは、プレースホルダに対応しています。

[プレースホルダ] において行をダブルクリックすると、プレースホルダが [パス (1 行につき 1 つのパス)] に反映されます。

2. インクルード・パスは、以下のいずれかの方法で指定することも可能です。

- エクスプローラなどからフォルダをドラッグ・アンド・ドロップ
 - [参照...] ボタンをクリックし、[フォルダの参照 ダイアログ](#)によるフォルダの選択
 - [プレースホルダ] において行をダブルクリック
3. [参照ボタンからパスを追加時に、サブフォルダも含める] をチェックしたのち、[参照...] ボタンからパスの指定を行うと、指定したパスとそのサブフォルダ 5 階層分までのパスを [パス (1 行につき 1 つのパス)] に追加します。

[OK] ボタンをクリックすると、入力したインクルード・パスがサブプロパティとして表示されます。

図 2—41 「追加のインクルード・パス」プロパティ（インクルード・パス追加後）



インクルード・パスの変更は、[...] ボタン、またはサブプロパティのテキスト・ボックスへの直接入力により行うことができます。

また、プロジェクト・ツリーにインクルード・ファイルを追加すると、そのインクルード・パスをサブプロパティの一番最初に自動で追加します。

備考 [\[共通オプション\] タブ](#)の[\[よく使うオプション \(アセンブル\)\]](#)カテゴリの[\[追加のインクルード・パス\]](#)プロパティでも、同様に設定することができます。

2.6.2 定義マクロを設定する

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネル](#)の[\[アセンブル・オプション\] タブ](#)を選択します。

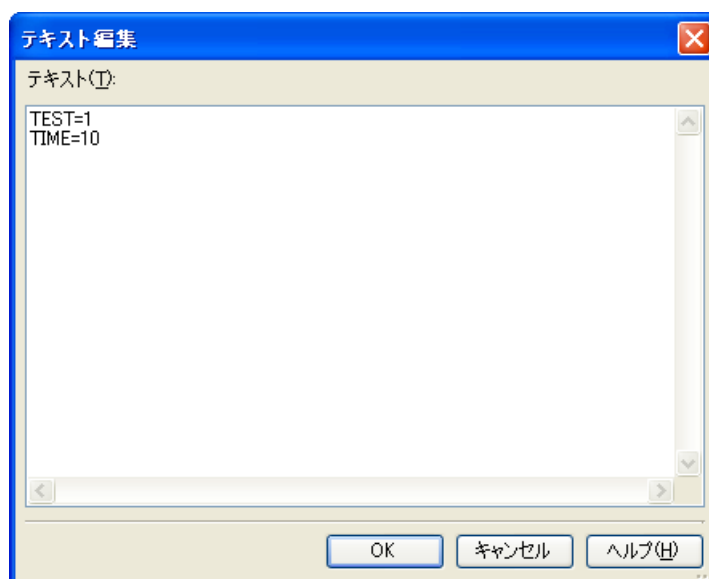
定義マクロの設定は、[プリプロセス] カテゴリの[\[定義マクロ\]](#)プロパティで行います。

図 2—42 「定義マクロ」プロパティ



[...] ボタンをクリックすると、[テキスト編集 ダイアログ](#)がオープンします。

図 2—43 テキスト編集 ダイアログ



【テキスト】に定義マクロを「マクロ名= 定義値」の形式で 1 行に 1 つずつ入力します。

1 行に 256 文字まで、256 行まで指定可能です。

「= 定義値」の部分は省略可能で、省略した場合、定義値を 1 とします。

【OK】ボタンをクリックすると、入力した定義マクロがサブプロパティとして表示されます。

図 2—44 【定義マクロ】プロパティ（定義マクロ設定後）



定義マクロの変更は、[...] ボタン、またはサブプロパティのテキスト・ボックスへの直接入力により行うことができます。

備考 **【共通オプション】** タブの **【よく使うオプション（アセンブル）】** カテゴリの **【定義マクロ】** プロパティでも、同様に設定することができます。

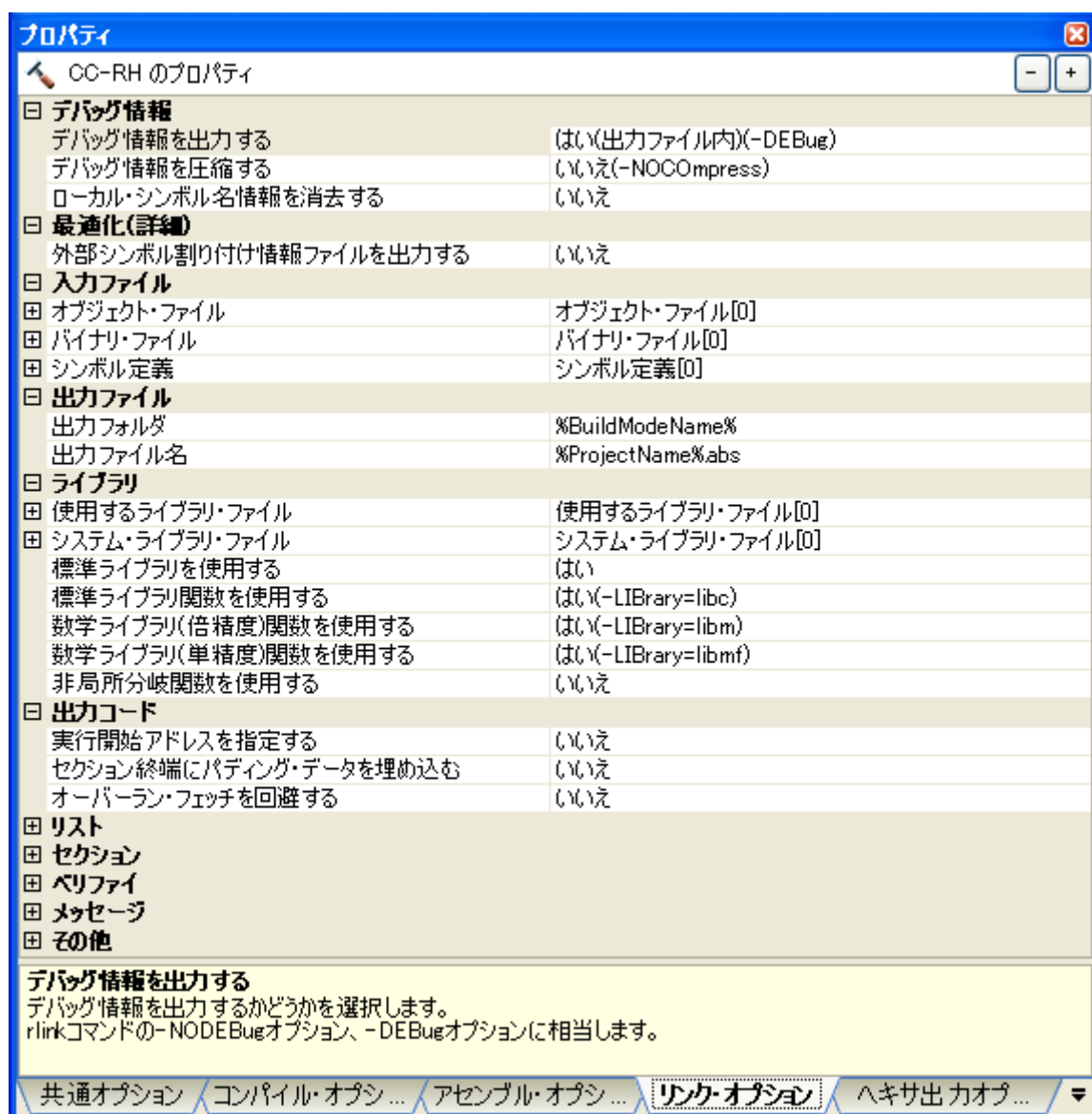
2.7 リンク・オプションを設定する

リンク・フェーズに対するオプションを設定するには、プロジェクト・ツリーでビルド・ツール・ノードを選択し、プロパティパネルの[リンク・オプション]タブを選択してください。

タブ上で各プロパティを設定することにより、対応するリンク・オプションを設定することができます。

注意 本タブは、ライブラリ用のプロジェクトの場合は表示しません。

図 2—45 プロパティパネル:[リンク・オプション]タブ



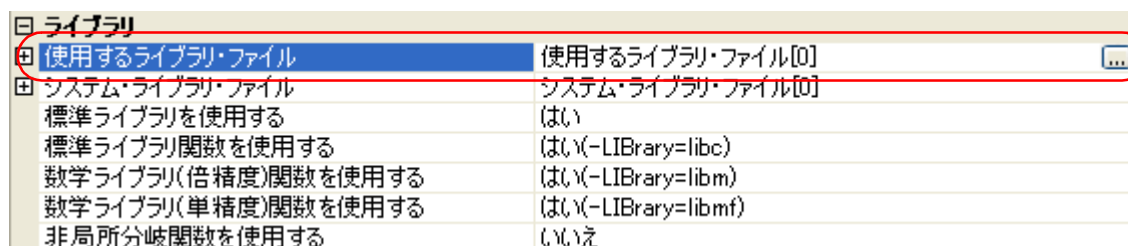
備考 よく使うオプションについては、[共通オプション]タブの[よく使うオプション(リンク)]カテゴリにまとめられています。

2.7.1 ユーザ・ライブラリを追加する

プロジェクト・ツリーでビルド・ツール・ノードを選択し、**プロパティ パネル**の **［リンク・オプション］** タブを選択します。

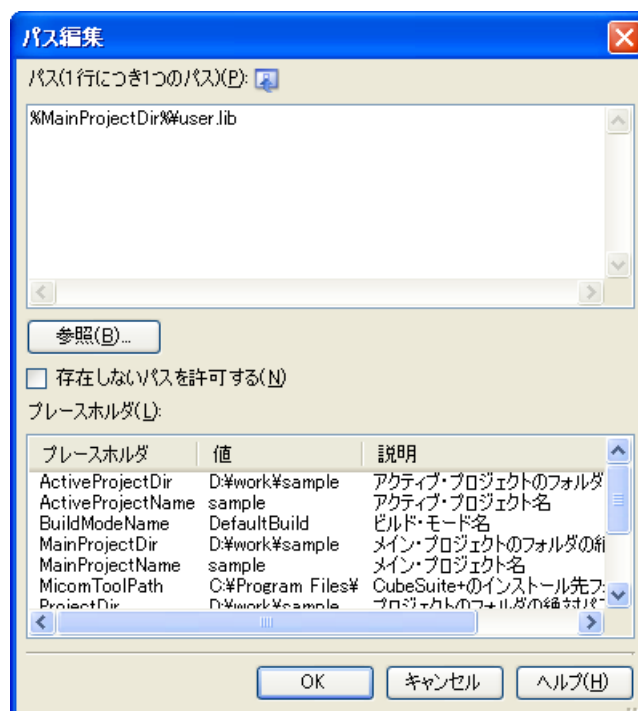
ユーザ・ライブラリの追加は、**［ライブラリ］** カテゴリの **［使用するライブラリ・ファイル］** プロパティで行います。

図 2—46 「使用するライブラリ・ファイル」 プロパティ



[...] ボタンをクリックすると、**パス編集 ダイアログ**がオープンします。

図 2—47 パス編集 ダイアログ



[パス (1 行につき 1 つのパス)] にライブラリ・ファイル (パスを含む) を 1 行に 1 つずつ入力します。

1 行に 259 文字まで、65536 行まで指定可能です。

備考 1. 本プロパティは、プレースホルダに対応しています。

[プレースホルダ] において行をダブルクリックすると、プレースホルダが [パス (1 行につき 1 つのパス)] に反映されます。

2. ライブラリ・ファイルは、以下のいずれかの方法で指定することも可能です。

- エクスプローラなどからフォルダをドラッグ・アンド・ドロップ
- [参照 ...] ボタンをクリックし、使用するライブラリ・ファイルを指定ダイアログによるフォルダの選択
- [プレースホルダ] において行をダブルクリック

[OK] ボタンをクリックすると、入力したライブラリ・ファイルがサブプロパティとして表示されます。

図 2—48 「使用するライブラリ・ファイル」 プロパティ (ライブラリ・ファイル設定後)



ライブラリ・ファイルの変更は、[...] ボタン、またはサブプロパティのテキスト・ボックスへの直接入力により行うことができます。

備考 [共通オプション] タブの [よく使うオプション (リンク)] カテゴリの [使用するライブラリ・ファイル] プロパティでも、同様に設定することができます。

2.7.2 オーバーレイ・セクションの選択機能を使用するための準備をする

CC-RH が使用する最適化リンカ（rlink）では、プログラム中に定義した複数のセクションを同じアドレスに割り付けることができます。このように割り付けられたセクションを“オーバーレイ・セクション”と呼びます。

同じアドレスに割り付けられたセクションのうち、デバッグ対象とするオーバーレイ・セクション（優先セクション）を選択する機能はデバッグ・ツールが提供しています。この機能を“オーバーレイ・セクションの選択機能”と呼びます。

オーバーレイ・セクションを設定したロード・モジュールは、プログラム実行前に優先セクションを切り替えてデバッグすることが可能になります。

オーバーレイ・セクションの選択機能を実現するための、ロード・モジュールの作成方法を以下に示します。

(1) ROM 領域の内容を RAM にコピー

ROM 領域の内容を RAM 領域にコピーすることにより、コードやデータを RAM 上へ展開します。

(2) ビルド・オプションの設定

オーバーレイ・セクションの選択機能に対応するため、ROM から RAM ヘマップするセクション、およびオーバーレイ・セクションを設定します。

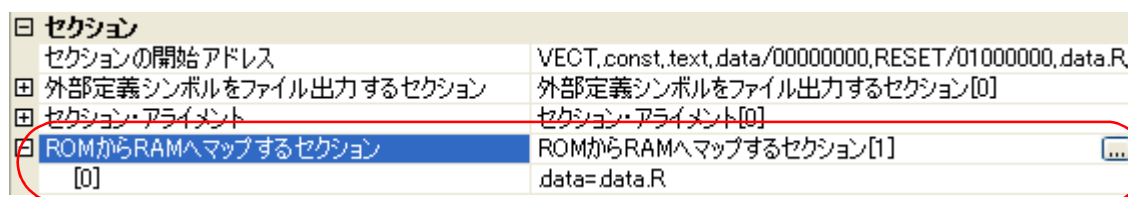
プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの \[リンク・オプション\] タブ](#)を選択します。

(a) ROM から RAM ヘマップするセクションの設定

ROM から RAM ヘマップするセクションの設定は、[セクション] カテゴリの [ROM から RAM ヘマップするセクション] プロパティで行います。

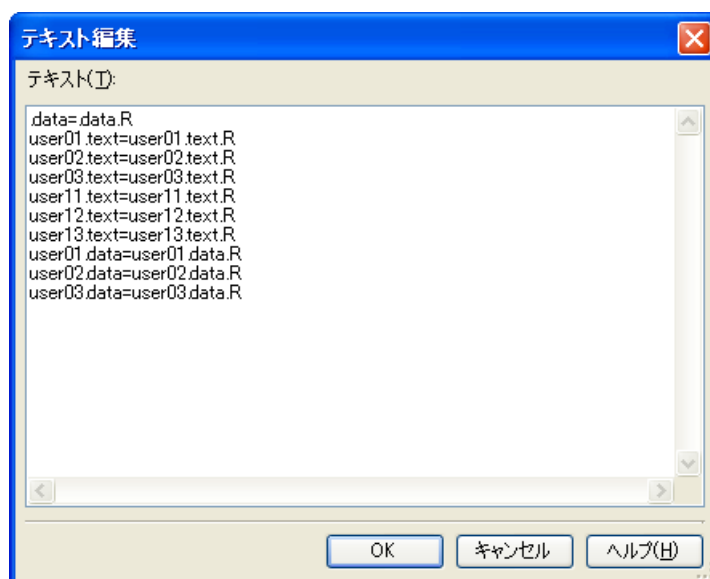
これにより、ROM セクションと同サイズの RAM セクションを確保し、ROM セクション内定義シンボルを RAM セクション上のアドレスでリロケーションします。

図 2—49 [ROM から RAM ヘマップするセクション] プロパティ



[...] ボタンをクリックすると、[テキスト編集 ダイアログ](#)がオープンします。

図 2—50 テキスト編集 ダイアログ

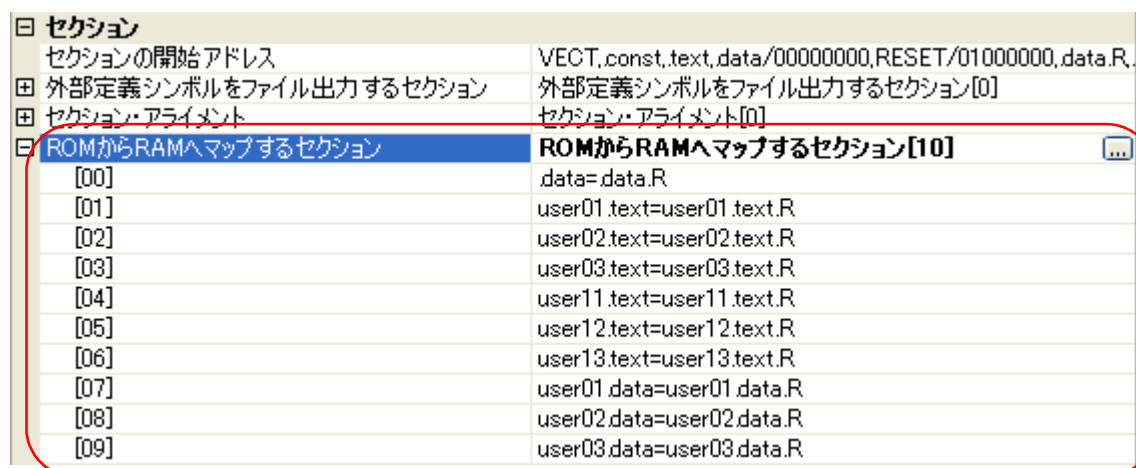


【テキスト】にセクション名を「ROM セクション名=RAM セクション名」の形式で1行に1つずつ入力します。

1行に 32767 文字まで、65535 行まで指定可能です。

【OK】ボタンをクリックすると、入力したセクション名がサブプロパティとして表示されます。

図 2—51 「ROM から RAM へマップするセクション」プロパティ（セクション設定後）



セクション名の変更は、[...] ボタン、またはサブプロパティのテキスト・ボックスへの直接入力により行うことができます。

(b) ROM セクション, RAM セクション (オーバーレイ・セクション) の設定

セクションの設定は, [セクション] カテゴリの [セクションの開始アドレス] プロパティで行います。

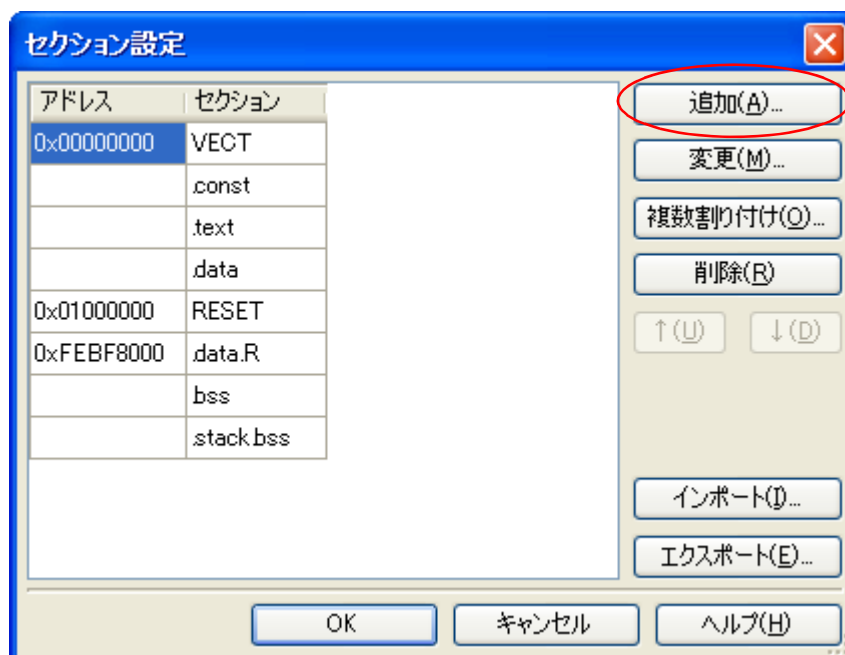
図 2—52 [セクションの開始アドレス] プロパティ

セクション	
セクションの開始アドレス	VECT,const,text,data/00000000,RESET/01000000,data...
外部定義シンボルをファイル出力するセクション	外部定義シンボルをファイル出力するセクション[0]
セクション・アライメント	セクション・アライメント[0]
ROMからRAMへマップするセクション	ROMからRAMへマップするセクション[10]

- ROM セクションの設定

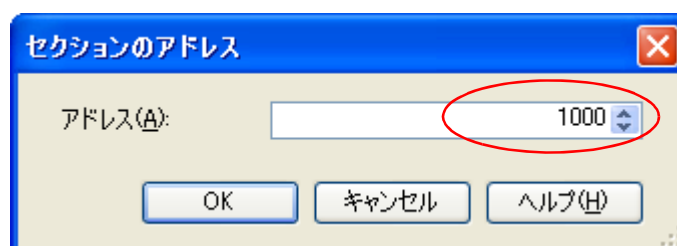
[...] ボタンをクリックすると, セクション設定 ダイアログがオープンします。

図 2—53 セクション設定 ダイアログ



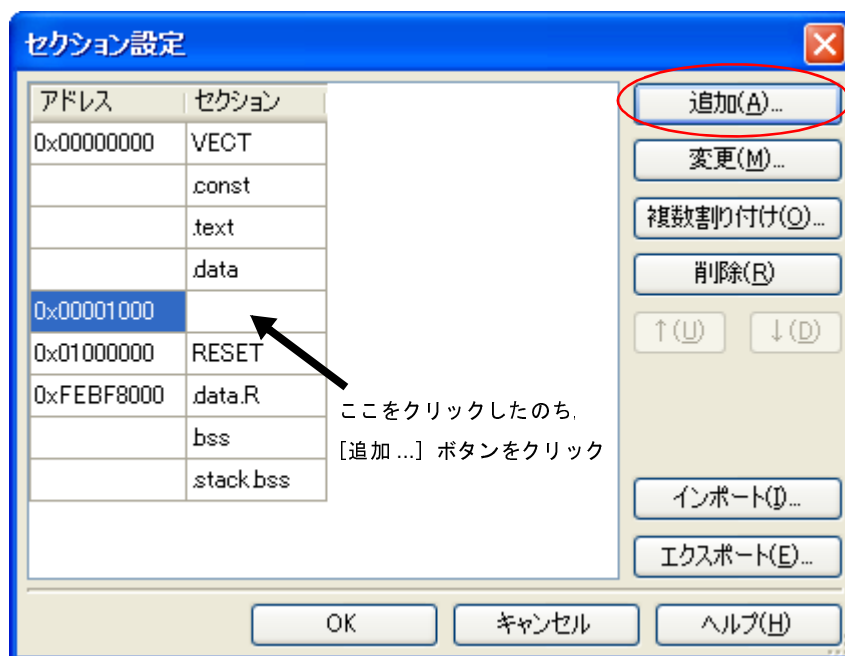
[追加 ...] ボタンをクリックすると, セクションのアドレス ダイアログがオープンします。

図 2—54 セクションのアドレス ダイアログ



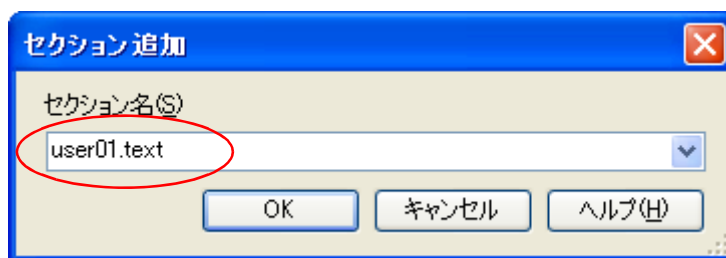
「[アドレス]」に追加したい ROM セクションのアドレスを入力し、[OK] ボタンをクリックすると、入力したアドレスが[セクション設定 ダイアログ](#)の「[アドレス]」に追加されます。

図 2—55 セクション設定 ダイアログ (ROM セクションのアドレス追加後)



追加されたアドレス行のセクション欄をクリックしたのち、[追加 ...] ボタンをクリックすると、[セクション追加 ダイアログ](#)がオープンします。

図 2—56 セクション追加 ダイアログ



「[セクション名]」に追加したい ROM セクション名を入力し、[OK] ボタンをクリックすると、入力したセクションが[セクション設定 ダイアログ](#)の「[セクション]」に追加されます。

図 2—57 セクション設定 ダイアログ (ROM セクション追加後)



他の ROM セクションについても同様に、アドレスとセクション名の設定を行います。

備考 アドレス欄をクリックしたのち、[追加 ...] ボタンをクリックすると、[セクションのアドレス ダイアログ](#)がオープンし、新しいアドレスを追加することができます。

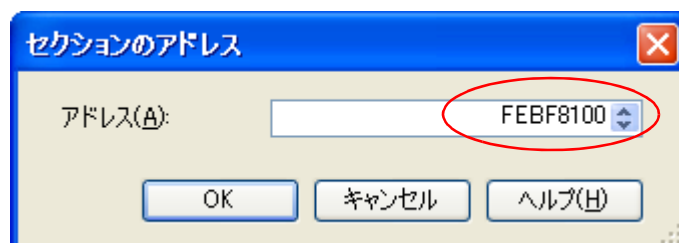
図 2—58 セクション設定 ダイアログ (複数 ROM セクション追加後)



- RAM セクション（オーバーレイ・セクション）の設定

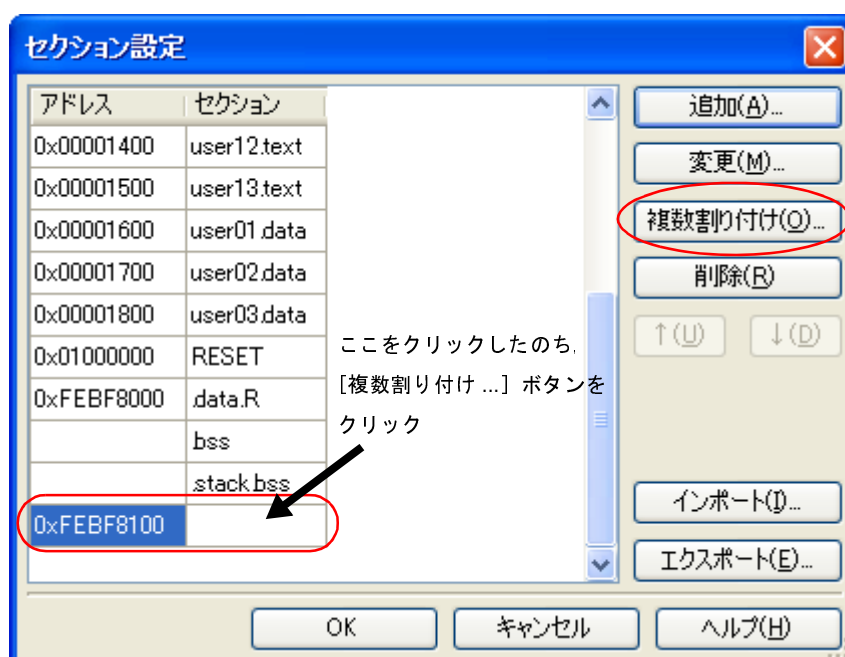
追加済みのアドレスをクリックしたのち、[追加] ボタンをクリックすると、[セクションのアドレス ダイアログ](#)がオープンします。

図 2—59 セクションのアドレス ダイアログ



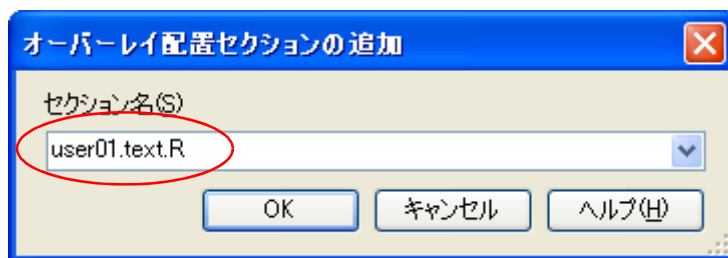
[アドレス] に追加したい RAM セクションのアドレスを入力し、[OK] ボタンをクリックすると、入力したアドレスが[セクション設定 ダイアログ](#)の [アドレス] に追加されます。

図 2—60 セクション設定 ダイアログ (RAM セクションのアドレス追加後)



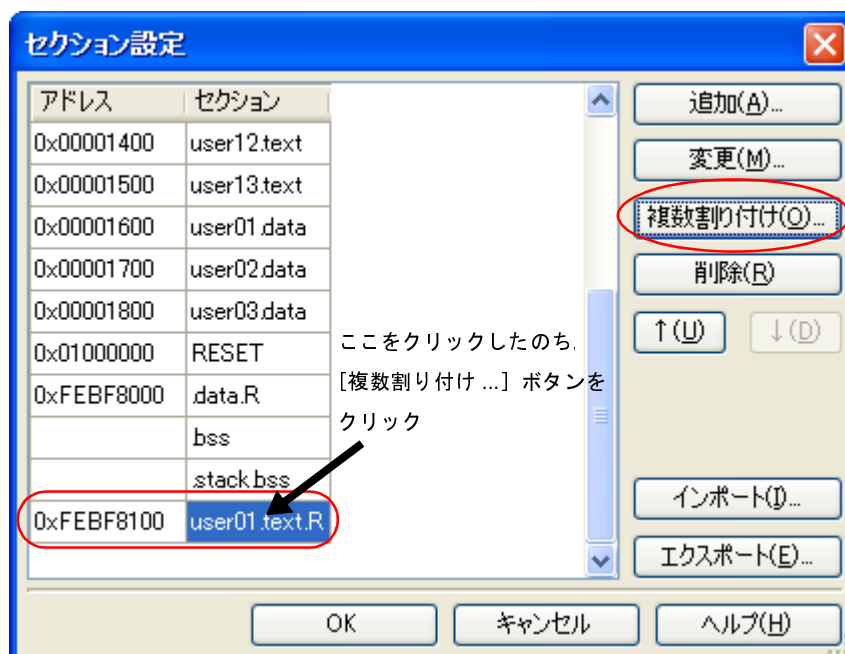
追加されたアドレス行（アドレス欄、またはセクション欄）をクリックしたのち、[複数割り付け ...] ボタンをクリックすると、[オーバーレイ配置セクションの追加 ダイアログ](#)がオープンします。

図 2—61 オーバーレイ配置セクションの追加 ダイアログ



「セクション名」に追加したいRAM セクション名を入力し、「OK」ボタンをクリックすると、入力したセクションがセクション設定 ダイアログの「セクション」に追加されます。

図 2—62 セクション設定 ダイアログ (RAM セクション追加後)



同一アドレスに配置するセクションについても同様に、「複数割り付け...」ボタンで追加します。追加したセクションは、「オーバーレイ n 」(n : 1 で始まる数字)に表示されます。

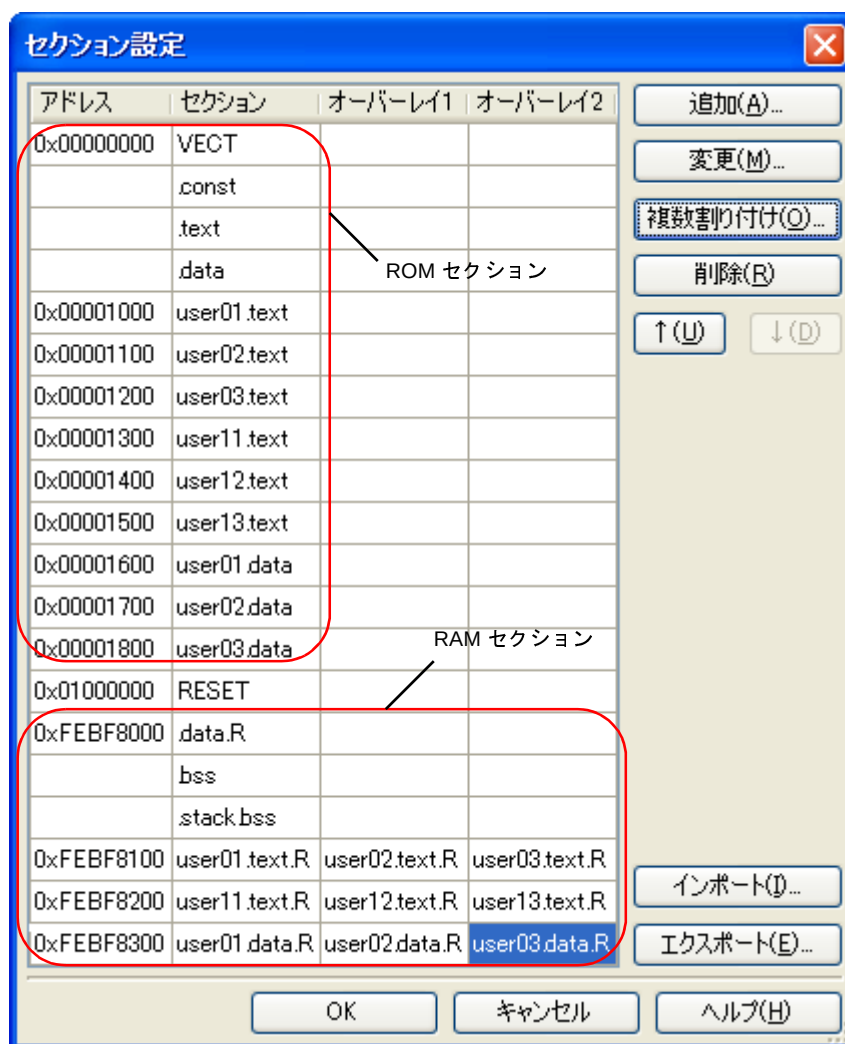
図 2—63 セクション設定 ダイアログ (オーバーレイ・セクション追加後)



他の RAM セクションについても同様に、アドレスとセクション名の設定を行います。

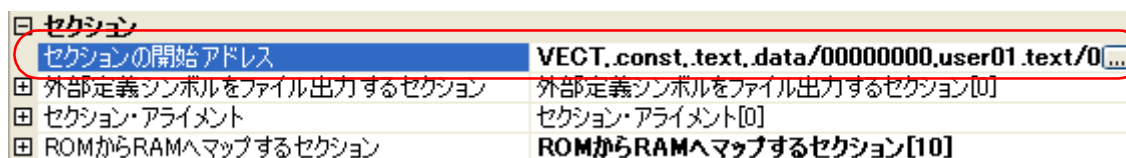
備考 アドレス欄をクリックしたのち、[追加 ...] ボタンをクリックすると、[セクションのアドレス ダイアログ](#)がオープンし、新しいアドレスを追加することができます。

図 2—64 セクション設定 ダイアログ (複数 RAM セクション追加後)



[OK] ボタンをクリックすると、設定した ROM セクション、RAM セクション（オーバーレイ・セクション）がテキスト・ボックスに表示されます。

図 2—65 [セクションの開始アドレス] プロパティ (セクション設定後)



(3) プロジェクトのビルドの実行

プロジェクトのビルドを実行します。

オーバーレイ・セクションの選択機能を実現するためのロード・モジュール・ファイルが生成されます。

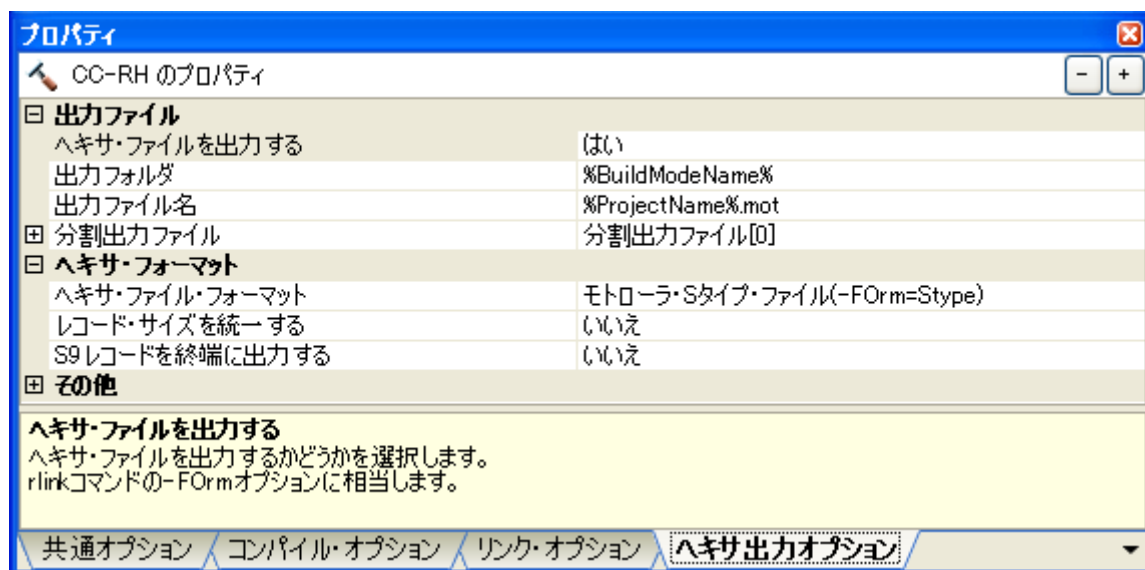
2.8 ヘキサ出力オプションを設定する

ヘキサ出力フェーズに対するオプションを設定するには、プロジェクト・ツリーでビルド・ツール・ノードを選択し、**プロパティ パネル**の**「ヘキサ出力オプション」**タブを選択してください。

タブ上で各プロパティを設定することにより、対応するヘキサ出力オプションを設定することができます。

注意 本タブは、ライブラリ用のプロジェクトの場合は表示しません。

図 2—66 プロパティ パネル : 「ヘキサ出力オプション」 タブ



備考 よく使うオプションについては、**「共通オプション」**タブの**「よく使うオプション（ヘキサ出力）」**カテゴリにまとめられています。

2.8.1 ヘキサ・ファイルの出力を設定する

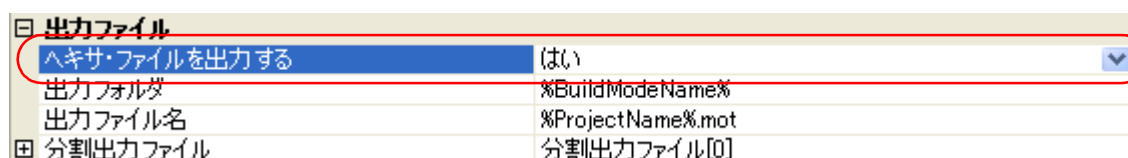
プロジェクト・ツリーでビルド・ツール・ノードを選択し、**プロパティ パネル**の**「ヘキサ出力オプション」**タブを選択します。

(1) ヘキサ・ファイルの出力設定

ヘキサ・ファイルの出力設定は、**「出力ファイル」**カテゴリの**「ヘキサ・ファイルを出力する」**プロパティで行います。

ヘキサ・ファイルを出力する場合は**「はい」**、出力しない場合は**「いいえ」**を選択してください。

図 2—67 「ヘキサ・ファイルを出力する」プロパティ



ヘキサ・ファイルを出力する場合、出力フォルダ、および出力ファイル名を設定することができます。

(a) 出力フォルダの設定

〔出力フォルダ〕プロパティにおいて、テキスト・ボックスへの直接入力、または [...] ボタンにより行います。

テキスト・ボックスには 247 文字まで指定可能です。

本プロパティは、次のプレースホルダに対応しています。

%ActiveProjectDir%	: アクティブ・プロジェクト・フォルダの絶対パスに置換します。
%ActiveProjectName%	: アクティブ・プロジェクト名に置換します。
%BuildModeName%	: ビルド・モード名に置換します。
%MainProjectDir%	: メイン・プロジェクト・フォルダの絶対パスに置換します。
%MainProjectName%	: メイン・プロジェクト名に置換します。
%MicomToolPath%	: 本製品のインストール・フォルダの絶対パスに置換します。
%ProjectDir%	: プロジェクト・フォルダの絶対パスに置換します。
%ProjectName%	: プロジェクト名に置換します。
%TempDir%	: テンポラリ・フォルダの絶対パスに置換します。
%WinDir%	: Windows システム・フォルダの絶対パスに置換します。

デフォルトでは、“%BuildModeName%” を設定しています。

(b) 出力ファイル名の設定

〔出力ファイル名〕プロパティにおいて、テキスト・ボックスへの直接入力により行います。

テキスト・ボックスには 259 文字まで指定可能です。

本プロパティは、次のプレースホルダに対応しています。

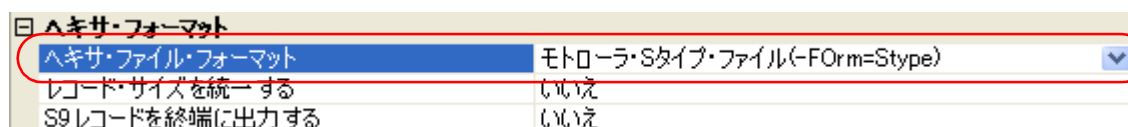
%ActiveProjectName%	: アクティブ・プロジェクト名に置換します。
%MainProjectName%	: メイン・プロジェクト名に置換します。
%ProjectName%	: プロジェクト名に置換します。

デフォルトでは、“%ProjectName%.mot” を設定しています。

(2) ヘキサ・ファイル・フォーマットの設定

〔ヘキサ・フォーマット〕カテゴリの〔ヘキサ・ファイル・フォーマット〕プロパティで、フォーマットを選択してください。

図 2—68 〔ヘキサ・ファイル・フォーマット〕プロパティ



以下のフォーマットを選択することができます。

フォーマット	説明
インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)	インテル拡張ヘキサ・ファイルを出力します。
モトローラ・Sタイプ・ファイル (-FOrm=Stype)	モトローラ・Sタイプ・ファイルを出力します。
バイナリ・ファイル (-FOrm=Binary)	バイナリ・ファイルを出力します。

備考 インテル拡張ヘキサ・ファイル、およびモトローラ・Sタイプ・ファイルについての詳細は、「[3.5 インテル拡張ヘキサ・ファイル](#)」、「[3.6 モトローラ・Sタイプ・ファイル](#)」を参照してください。

2.8.2 空き領域を充てんする

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの \[ヘキサ出力オプション\] タブ](#) を選択します。

(1) ヘキサ・ファイルの出力範囲設定

ヘキサ・ファイルの出力範囲の設定は、[出力ファイル] カテゴリの [分割出力ファイル] プロパティで行います。

図 2—69 [分割出力ファイル] プロパティ

ヘキサ・ファイルを出力する	はい
出力フォルダ	%BuildModeName%
出力ファイル名	%ProjectName%.mot
分割出力ファイル	分割出力ファイル[0]

[...] ボタンをクリックすると、[テキスト編集 ダイアログ](#)がオープンします。

図 2—70 テキスト編集 ダイアログ

テキスト編集

テキスト(T):

```
file1.abs=sec1:sec2
file2.abs=10000-1ffff
```

プレースホルダ(L):

プレースホルダ	値	説明
ActiveProjectDir	D:\work\sample	アクティブ・プロジェクトのフォルダの絶対パス
ActiveProjectName	sample	アクティブ・プロジェクト名
BuildModeName	DefaultBuild	ビルド・モード名
MainProjectDir	D:\work\sample	メイン・プロジェクトのフォルダの絶対パス
MainProjectName	sample	メイン・プロジェクト名

OK キャンセル ヘルプ(H)

〔テキスト〕に分割出力ファイル名を「ファイル名=先頭アドレス-終了アドレス」(先頭アドレス, 終了アドレス: 出力範囲の先頭アドレス, 終了アドレス), または「ファイル名=セクション名」(セクション名: 出力するセクション名)の形式で1行に1つずつ入力します。

セクション名を複数指定する場合は, 「ファイル名=セクション名:セクション名」のように, コロンで区切って入力します。

先頭アドレス, 終了アドレスは16進数で指定します。

1行に259文字まで, 65535行まで指定可能です。

〔OK〕ボタンをクリックすると, 入力した分割出力ファイル名がサブプロパティとして表示されます。

図 2—71 〔分割出力ファイル〕プロパティ (分割出力ファイル名設定後)

■ 出力ファイル	
ヘキサ・ファイルを出力する	はい
出力フォルダ	%BuildModeName%
出力ファイル名	%ProjectName%.mot
■ 分割出力ファイル	分割出力ファイル[2]
[0]	file1.abs=sec1:sec2
[1]	file2.abs=10000-1ffff

分割出力ファイル名の変更は, [...] ボタン, またはサブプロパティのテキスト・ボックスへの直接入力により行うことができます。

(2) 空き領域の充てん方法設定

出力範囲の空き領域の充てん方法を設定します。

(a) 空き領域を乱数で充てんする場合

〔ヘキサ・フォーマット〕カテゴリの〔出力範囲のメモリの空き領域をデータで充てんする〕プロパティで〔はい (乱数) (-SPace=Random)〕を選択してください。

図 2—72 〔出力範囲のメモリの空き領域をデータで充てんする〕プロパティ

■ ヘキサ・フォーマット	
ヘキサ・ファイル・フォーマット	モトローラ・Sタイプ・ファイル(-FOrm=Stype)
レコード・サイズを統一する	いいえ
出力範囲のメモリの空き領域をデータで充てんする	はい(乱数)(-SPace=Random)
S9レコードを終端に出力する	いいえ

(b) 空き領域を充てんするデータを指定する場合

〔ヘキサ・フォーマット〕カテゴリの〔出力範囲のメモリの空き領域をデータで充てんする〕プロパティで〔はい (データ指定) (-SPace=<数値>)〕を選択すると, 〔空き領域出力データ〕プロパティを表示します。

図 2—73 「出力範囲のメモリの空き領域をデータで充てんする」, および「空き領域出力データ」プロパティ

日 ヘキサ・フォーマット	
ヘキサ・ファイル・フォーマット	モトローラ・Sタイプ・ファイル(-FOrm=Stype)
レコード・サイズを統一する	いいえ
出力範囲のメモリの空き領域をデータで充てんする	はい(データ指定)(-SPace=<数値>)
空き領域出力データ	HEX 00
S9レコードを終端に出力する	いいえ

テキスト・ボックスに空き領域の充てん値を直接入力してください。

指定可能な値の範囲は 00 ~ FFFFFFFF (16 進数) です。

デフォルトでは, “FF” を設定しています。

2. 8. 3 複数のオブジェクトを結合する

マルチコア用プロジェクトにおいて, 構成アプリケーションとして設定した各プロジェクトのインテル拡張ヘキサ・ファイル, またはモトローラ・Sタイプ・ファイルを1つのヘキサ・ファイルに結合することができます (この機能を“オブジェクト結合機能”と呼びます)。

なお, インテル拡張ヘキサ・ファイルとモトローラ・Sタイプ・ファイルを混在させることはできません。

(1) ヘキサ・ファイルの出力設定

ブート・ローダ・プロジェクト, および各アプリケーション・プロジェクトにおいて, ヘキサ・ファイルの出力, およびヘキサ・フォーマットの設定を行います。

プロジェクト・ツリーでビルド・ツール・ノードを選択し, プロパティパネルの「ヘキサ出力オプション」タブを選択します。

「出力ファイル」カテゴリの「ヘキサ・ファイルを出力する」プロパティで「はい」を選択します。

図 2—74 「ヘキサ・ファイルを出力する」プロパティ

日 出力ファイル	
ヘキサ・ファイルを出力する	はい
出力フォルダ	%BuildModeName%
出力ファイル名	%ProjectName%.mot
分割出力ファイル	分割出力ファイル[0]
オブジェクト結合機能を使用する	いいえ

「ヘキサ・フォーマット」カテゴリの「ヘキサ・ファイル・フォーマット」プロパティで「インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)」, または「モトローラ・Sタイプ・ファイル (-FOrm=Stype)」を選択します。

図 2—75 「ヘキサ・ファイル・フォーマット」プロパティ

日 ヘキサ・フォーマット	
ヘキサ・ファイル・フォーマット	モトローラ・Sタイプ・ファイル(-FOrm=Stype)
レコード・サイズを統一する	いいえ
S9レコードを終端に出力する	いいえ

注意 ブート・ローダ・プロジェクト、およびすべてのアプリケーション・プロジェクトで同一の項目を選択する必要があります。

(2) ヘキサ・ファイルの結合指定

ブート・ローダ・プロジェクトにおいて、1つのヘキサ・ファイルに結合することを指定します。

〔出力ファイル〕カテゴリの〔オブジェクト結合機能を使用する〕プロパティで〔はい〕を選択します。

図 2—76 〔オブジェクト結合機能を使用する〕プロパティ

出力ファイル	
ヘキサ・ファイルを出力する	はい
出力フォルダ	%BuildModeName%
出力ファイル名	%ProjectName%.mot
分割出力ファイル	分割出力ファイル[0]
オブジェクト結合機能を使用する	はい
結合ヘキサ・ファイルの出力フォルダ	%BuildModeName%_merged

結合したヘキサ・ファイルの出力先は、〔結合ヘキサ・ファイルの出力フォルダ〕プロパティで指定することができます（デフォルト：%BuildModeName%_merged）。

なお、ファイル名は、〔出力ファイル名〕プロパティと同名となります。

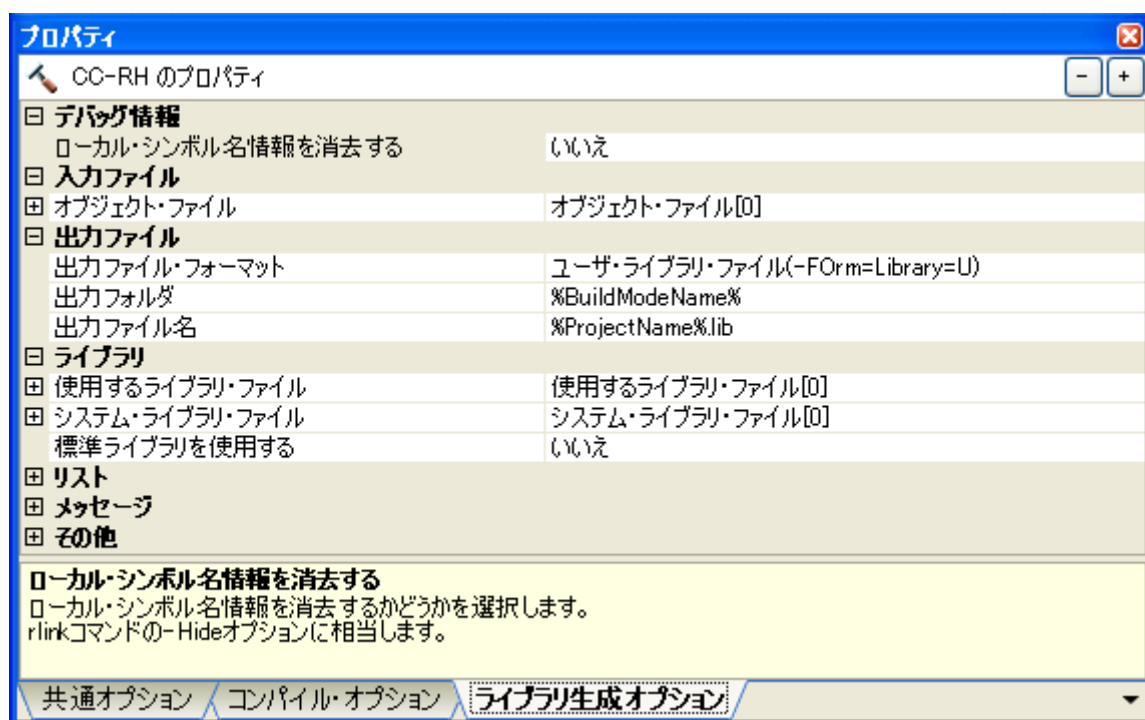
2.9 ライブラリ生成オプションを設定する

ライブラリ生成フェーズに対するオプションを設定するには、プロジェクト・ツリーでビルド・ツール・ノードを選択し、**プロパティ パネル**の**［ライブラリ生成オプション］**タブを選択してください。

タブ上で各プロパティを設定することにより、対応するライブラリ生成オプションを設定することができます。

注意 本タブは、ライブラリ用のプロジェクトの場合のみ表示します。

図 2—77 プロパティ パネル：［ライブラリ生成オプション］タブ



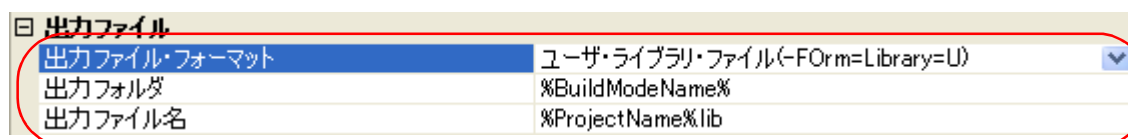
備考 よく使うオプションについては、**［共通オプション］**タブの**［よく使うオプション（ライブラリ生成）］**カテゴリにまとめられています。

2.9.1 ライブラリ・ファイルの出力を設定する

プロジェクト・ツリーでビルド・ツール・ノードを選択し、**プロパティ パネル**の**［ライブラリ生成オプション］**タブを選択します。

ライブラリ・ファイルの出力設定は、**［出力ファイル］**カテゴリで行います。

図 2—78 〔出力ファイル〕カテゴリ



(1) 出力フォーマットの設定

〔出力ファイル・フォーマット〕プロパティで、フォーマットを選択してください。

以下のフォーマットを選択することができます。

フォーマット	説明
ユーザ・ライブラリ・ファイル (-FOrm=Library=U)	ユーザ・ライブラリ・ファイルを出力します。
システム・ライブラリ・ファイル (-FOrm=Library=S)	システム・ライブラリ・ファイルを出力します。 システム・ライブラリ・ファイルは、ユーザ・ライブラリ・ファイルの後にリンクされます。 ユーザ・ライブラリ・ファイルの後にリンクするライブラリを作成する場合は、本項目を選択してください。
リロケートブル・ファイル (-FOrm=Relocate)	リロケートブル・ファイルを出力します。

(2) 出力フォルダの設定

〔出力フォルダ〕プロパティにおいて、テキスト・ボックスへの直接入力、または [...] ボタンにより行います。

テキスト・ボックスには 247 文字まで指定可能です。

本プロパティは、次のプレースホルダに対応しています。

- %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。
- %ActiveProjectName% : アクティブ・プロジェクト名に置換します。
- %BuildModeName% : ビルド・モード名に置換します。
- %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。
- %MainProjectName% : メイン・プロジェクト名に置換します。
- %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。
- %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。
- %ProjectName% : プロジェクト名に置換します。
- %TempDir% : テンポラリ・フォルダの絶対パスに置換します。
- %WinDir% : Windows システム・フォルダの絶対パスに置換します。

デフォルトでは、“%BuildModeName%”を設定しています。

(3) 出力ファイル名の設定

〔出力ファイル名〕プロパティにおいて、テキスト・ボックスへの直接入力により行います。

拡張子を省略した場合は、〔出力ファイル・フォーマット〕プロパティの選択に依存して、自動的に付加します。

- 〔ユーザ・ライブラリ・ファイル (-FOrm=Library=U)〕を選択している場合 : .lib
- 〔システム・ライブラリ・ファイル (-FOrm=Library=S)〕を選択している場合 : .lib
- 〔リロケートブル・ファイル (-FOrm=Relocate)〕を選択している場合 : .rel

テキスト・ボックスには 259 文字まで指定可能です。

本プロパティは、次のプレースホルダに対応しています。

- %ActiveProjectName% : アクティブ・プロジェクト名に置換します。
- %MainProjectName% : メイン・プロジェクト名に置換します。
- %ProjectName% : プロジェクト名に置換します。

デフォルトでは, “%ProjectName%.lib” を設定しています。

2.10 個別にビルド・オプションを設定する

ビルド・オプションの設定は、プロジェクト単位、またはファイル単位で行います。

プロジェクト単位 → 「2.10.1 プロジェクト単位でビルド・オプションを設定する」参照

ファイル単位 → 「2.10.2 ファイル単位でコンパイル／アセンブル・オプションを設定する」参照

2.10.1 プロジェクト単位でビルド・オプションを設定する

プロジェクト（メイン・プロジェクト、またはサブプロジェクト）に対するビルド・オプションを設定するには、プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネル](#)を表示します。

フェーズに対する各タブを選択し、必要なプロパティを設定することにより、ビルド・オプションを設定することができます。

コンパイル・フェーズ → [\[コンパイル・オプション\] タブ](#)

アセンブル・フェーズ → [\[アセンブル・オプション\] タブ](#)

リンク・フェーズ → [\[リンク・オプション\] タブ](#)

ヘキサ出力フェーズ → [\[ヘキサ出力オプション\] タブ](#)

ライブラリ生成フェーズ → [\[ライブラリ生成オプション\] タブ](#)

2.10.2 ファイル単位でコンパイル／アセンブル・オプションを設定する

プロジェクトに追加している各ソース・ファイルに対して、コンパイル・オプション、またはアセンブル・オプションを個別に設定することができます。

(1) C ソース・ファイルにコンパイル・オプションを設定する場合

プロジェクト・ツリーでCソース・ファイルを選択し、[プロパティ パネル](#)の[\[ビルド設定\] タブ](#)を選択します。

[ビルド] カテゴリの[\[個別コンパイル・オプションを設定する\]](#) プロパティで[\[はい\]](#)を選択すると、[メッセージ ダイアログ](#)がオープンします。

図 2—79 [個別コンパイル・オプションを設定する] プロパティ

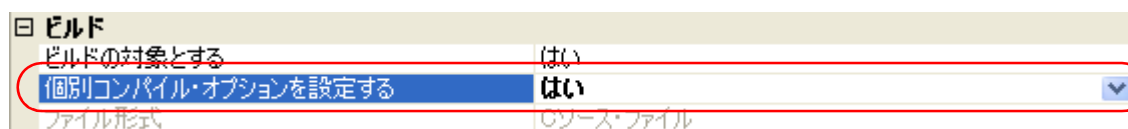
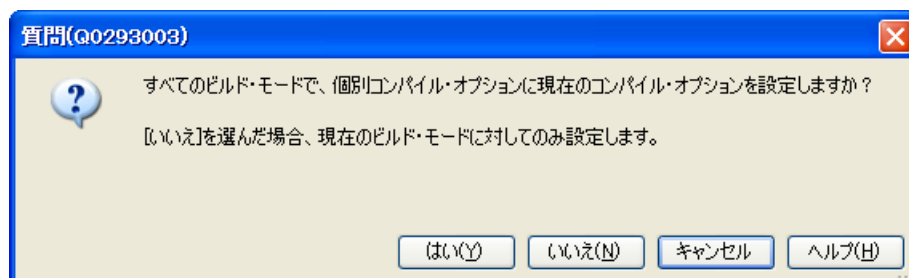
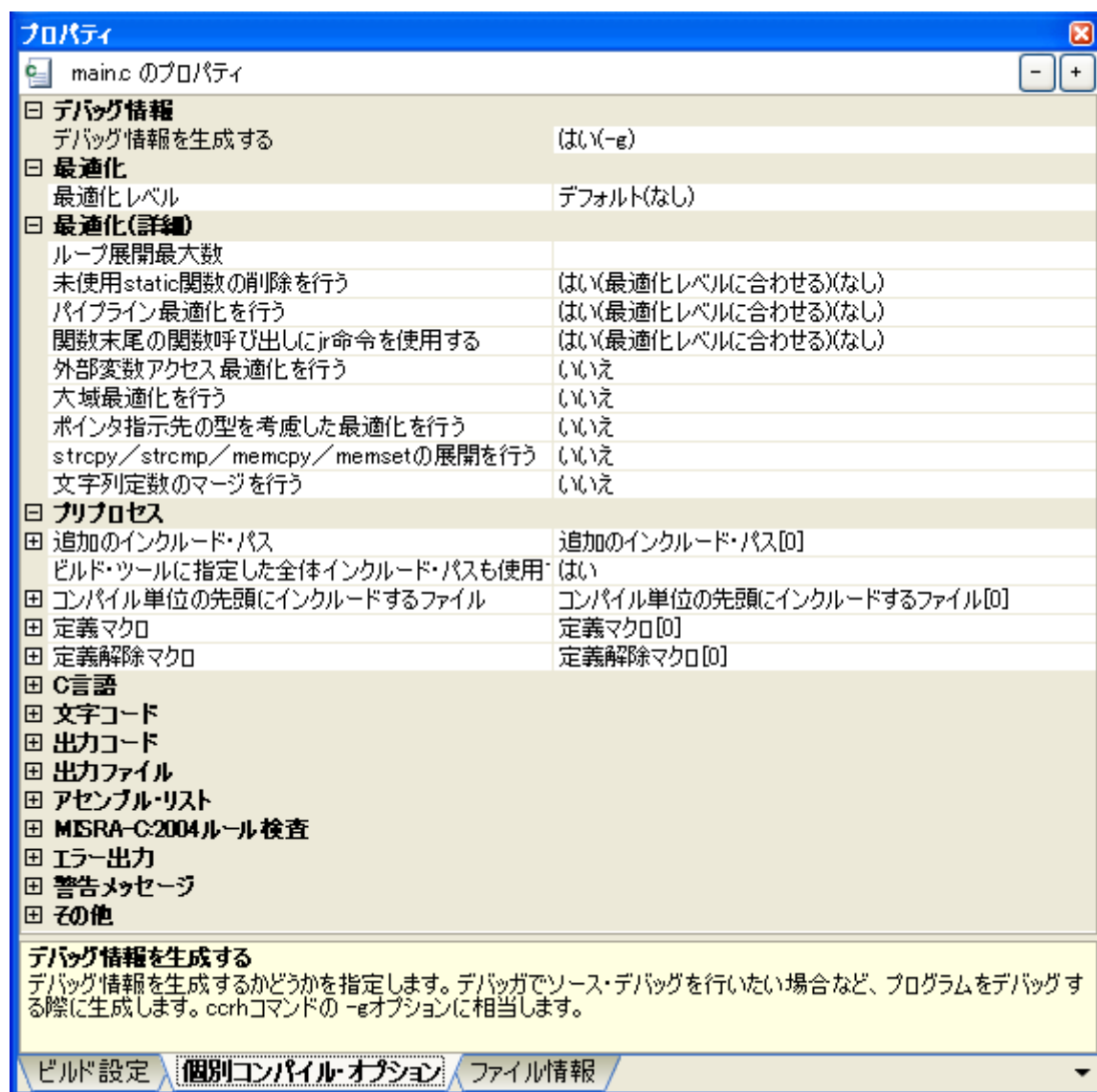


図 2—80 メッセージ ダイアログ



ダイアログ上で [はい] をクリックすると、[個別コンパイル・オプション] タブを表示します。

図 2—81 プロパティ パネル：[個別コンパイル・オプション] タブ



タブ上で必要なプロパティを設定することにより、C ソース・ファイルに対するコンパイル・オプションを設定することができます。

なお、本タブは、以下のプロパティを除いて、デフォルトでは [共通オプション] タブ、および [コンパイル・オプション] タブの設定内容を継承します。

- [プリプロセス] カテゴリの [追加のインクルード・パス]、[ビルド・ツールに指定した全体インクルード・パスも使用する]
- [出力ファイル] カテゴリの [オブジェクト・ファイル名]

(2) アセンブリ・ソース・ファイルにアセンブル・オプションを設定する場合

プロジェクト・ツリーでアセンブリ・ソース・ファイルを選択し、プロパティパネルの[ビルド設定]タブを選択します。

[ビルド] カテゴリの[個別アセンブル・オプションを設定する] プロパティで [はい] を選択すると、メッセージダイアログ（一括ビルドを行う場合）、またはメッセージダイアログ（一括ビルドを行わない場合）がオープンします。

図 2—82 [個別アセンブル・オプションを設定する] プロパティ

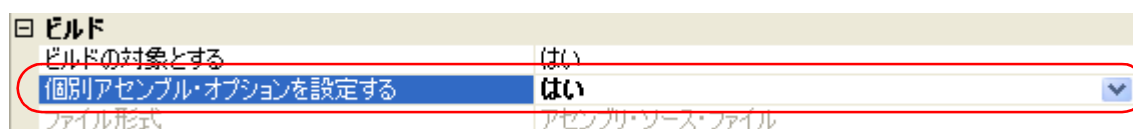


図 2—83 メッセージ ダイアログ（一括ビルドを行う場合）

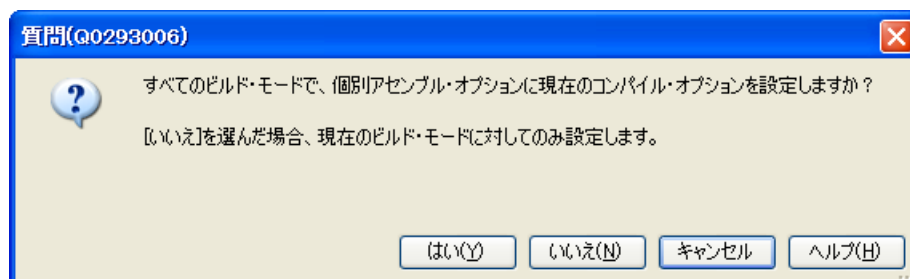
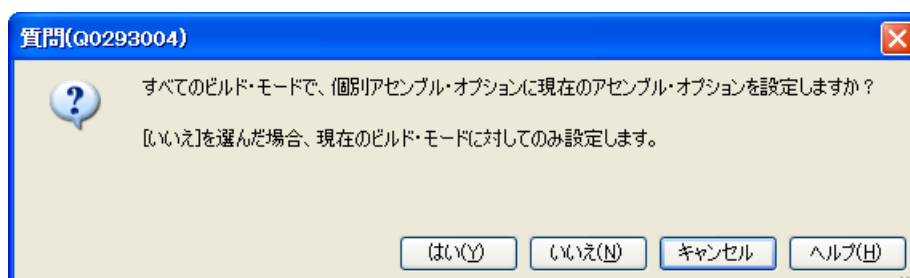
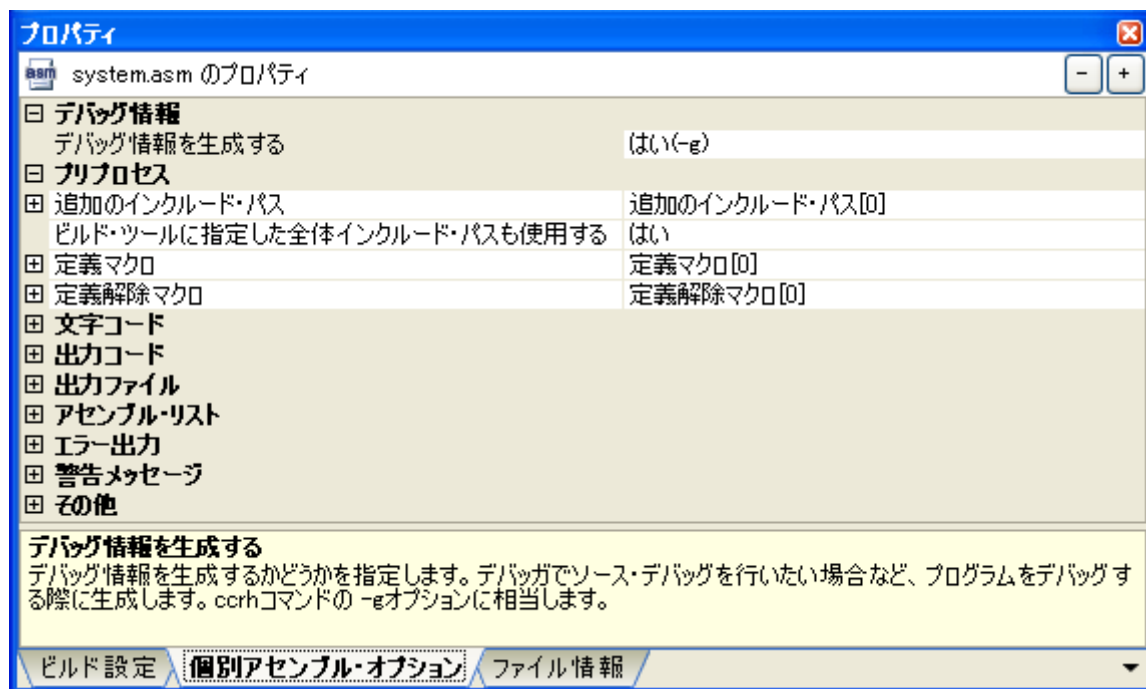


図 2—84 メッセージ ダイアログ（一括ビルドを行わない場合）



ダイアログ上で [はい] をクリックすると、[個別アセンブル・オプション] タブを表示します。

図 2—85 プロパティ パネル：[個別アセンブル・オプション] タブ



タブ上で必要なプロパティを設定することにより、アセンブリ・ソース・ファイルに対するアセンブル・オプションを設定することができます。

なお、本タブは、以下のプロパティを除いて、デフォルトでは [\[共通オプション\] タブ](#)、および [\[コンパイル・オプション\] タブ](#) / [\[アセンブル・オプション\] タブ](#) の設定内容を継承します。

- [プリプロセス] カテゴリの [追加のインクルード・パス]、[ビルド・ツールに指定した全体インクルード・パスも使用する]
- [出力ファイル] カテゴリの [オブジェクト・ファイル名]

2.11 ビルドの設定をする

ここでは、ビルドに関する以下の操作を説明します。

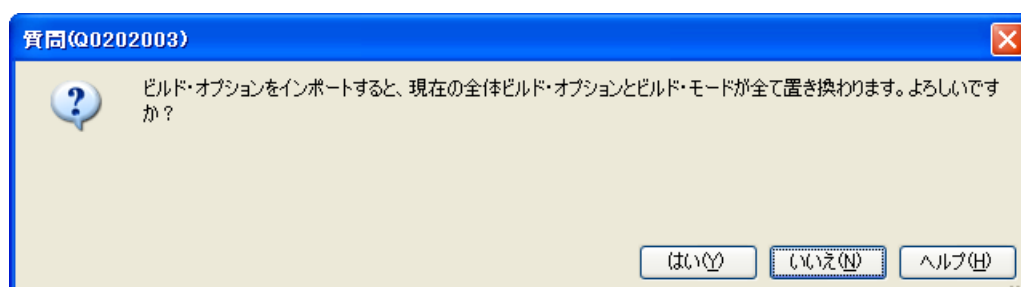
- 他のプロジェクトのビルド・オプションをインポートする
- ファイルのリンク順を設定する
- サブプロジェクトのビルド順を変更する
- ビルド・オプションを一覧表示する
- ビルド対象プロジェクトを変更する
- ビルド・モードを追加する
- ビルド・モードを変更する
- ビルド・モードを削除する
- 現在のビルド・オプションをプロジェクトの標準に設定する

2.11.1 他のプロジェクトのビルド・オプションをインポートする

他のプロジェクトのビルド・オプションを現在のプロジェクトにインポートすることができます。

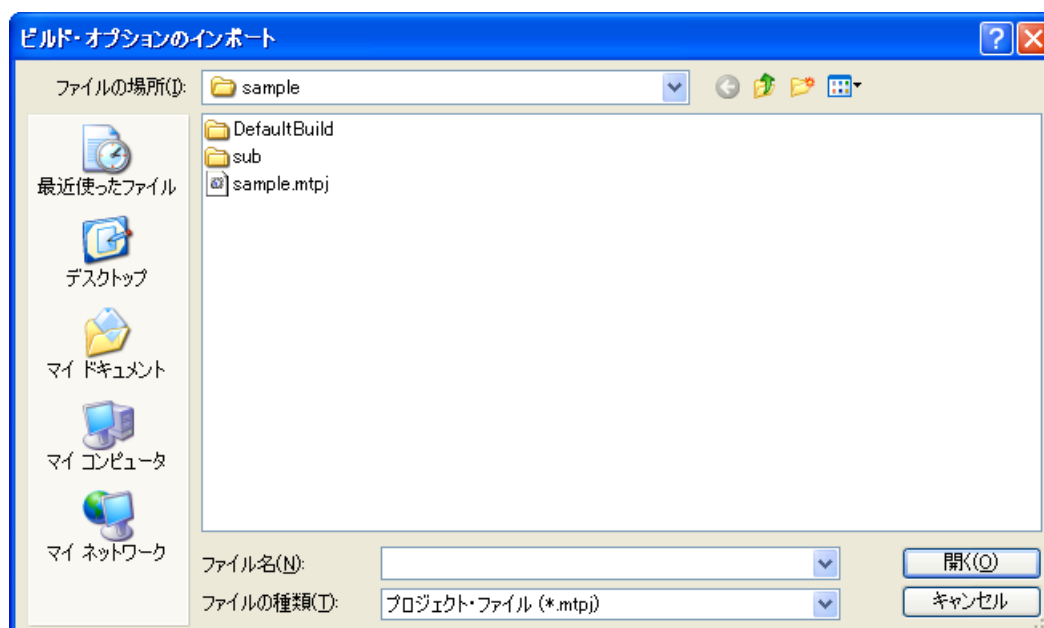
プロジェクト・ツリーでビルド・ツール・ノードを選択し、コンテキスト・メニューの「ビルド・オプションのインポート ...」を選択すると、以下のメッセージ ダイアログがオープンします。

図 2—86 メッセージ ダイアログ



ダイアログ上で「はい」をクリックすると、[ビルド・オプションのインポート ダイアログ](#)がオープンします。

図 2—87 ビルド・オプションのインポート ダイアログ



ダイアログ上でビルド・オプションのインポート対象となるプロジェクト・ファイルを選択し、[開く] ボタンをクリックしてください。

選択したプロジェクト・ファイルのビルド・オプションを現在のプロジェクトにインポートします。

備考 1. インポート可能なプロジェクトの条件を以下に示します。

- ビルド・ツールが同じである。
- プロジェクトの種類（アプリケーション、ライブラリなど）が同じである。
- 同じメジャー・バージョンの CubeSuite+ で作成したプロジェクトである。

2. インポート対象となるビルド・オプションは、ビルド・ツールのプロパティで設定した全体オプションのみです。

標準ビルド・オプションの設定（[「2. 11. 9 現在のビルド・オプションをプロジェクトの標準に設定する」](#) 参照）や個別オプションのインポートは行いません。

3. インポート対象のすべてのビルド・モードのインポートも行います。

なお、現在のプロジェクトのビルド・モードは“DefaultBuild”以外は削除します。

4. 使用するビルド・ツールのバージョンのインポートも行います。

2.11.2 ファイルのリンク順を設定する

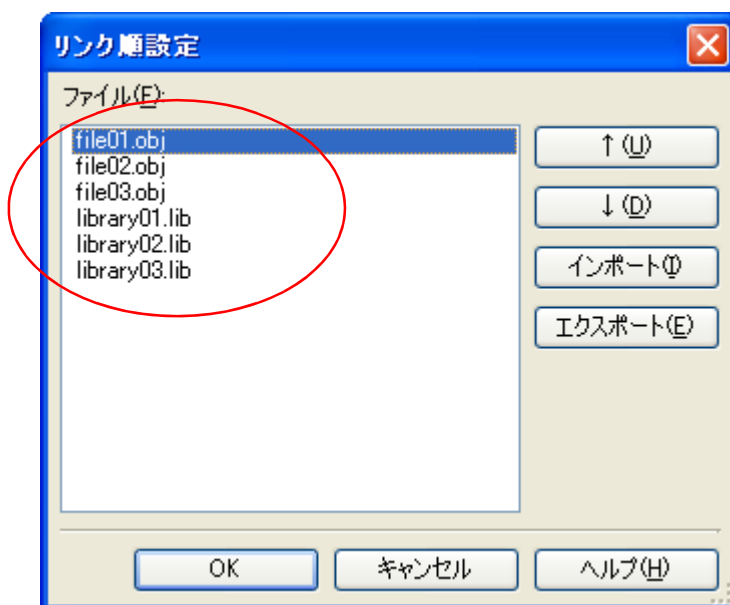
オブジェクト・ファイル、およびライブラリ・ファイルのリンク順は、自動で決定しますが、ユーザが設定することもできます。

以下に、操作手順を示します。

(1) リンク順設定 ダイアログのオープン

プロジェクト・ツリーでビルド・ツール・ノードを選択し、コンテキスト・メニューの [リンク順を設定する ...] を選択すると、[リンク順設定 ダイアログ](#) がオープンします。

図 2—88 リンク順設定 ダイアログ



[ファイル] には、以下のファイルのファイル名一覧を、リンカへの入力順で表示します。

- 選択しているメイン・プロジェクト、またはサブプロジェクトに追加しているソース・ファイルから生成するオブジェクト・ファイル
- 選択しているメイン・プロジェクト、またはサブプロジェクトのプロジェクト・ツリーに直接追加したオブジェクト・ファイル
- 選択しているメイン・プロジェクト、またはサブプロジェクトのプロジェクト・ツリーに直接追加したライブラリ・ファイル

備考 デフォルトでは、プロジェクトに追加している順番となります。

新規に追加したソース・ファイルから生成するオブジェクト・ファイル、および新規に追加したオブジェクト・ファイルは、一覧の最後のオブジェクト・ファイルの次に追加します。

新規に追加したライブラリ・ファイルは、一覧の最後に追加します。

(2) ファイルの表示順の変更

ファイルの表示順を変更することにより、リンカへのファイルの入力順を設定することができます。

以下のいずれかの方法により、ファイルの表示順を変更します。

- [↑], および [↓] ボタンによるファイル名の移動
- ファイル名のドラッグ・アンド・ドロップ
- リンク順指定ファイルの利用

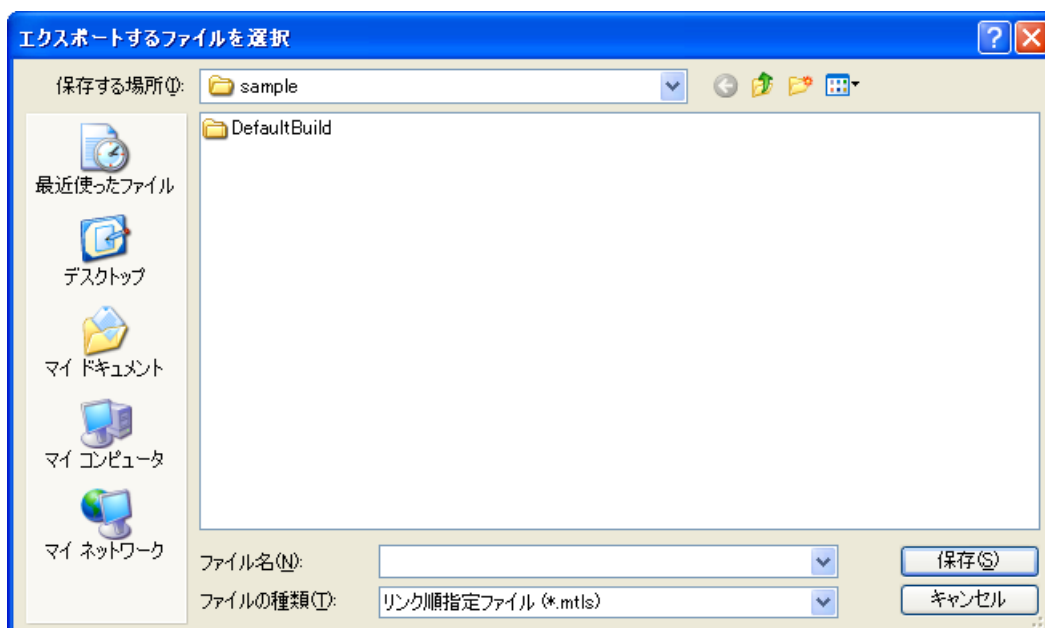
備考 リンク順指定ファイルを利用することにより、ファイル・ベースで表示順を変更することができます。

以下に、操作手順を示します。

(a) リンク順指定ファイルの生成

リンク順設定 ダイアログの [エクスポート] ボタンをクリックすると、エクスポートするファイルを選択 ダイアログがオープンします。

図 2—89 エクスポートするファイルを選択 ダイアログ



ダイアログ上で、リンク順設定 ダイアログの [ファイル] に表示しているファイル名一覧を出力するファイル（リンク順指定ファイル）を指定します。

[保存] ボタンをクリックすると、リンク順指定ファイルが生成されます。

注意 リンク順指定ファイルには、ファイル名のみを出力します。

同名のファイルが存在する場合は、リンク順指定ファイルのインポート後にポップアップ表示でファイルの存在場所を確認してください。

(b) リンク順指定ファイルの編集

エディタでリンク順指定ファイルをオープンし、ファイル名の記述順を変更します。

リンク順指定ファイルの記述例を以下に示します。

```
# CubeSuite+ Vx.xx.xx Link order specification file
# SampleProject: xxxx 年 xx 月 xx 日

file01.obj
file03.obj
library02.lib
file02.obj
library01.lib
library03.lib
:
```

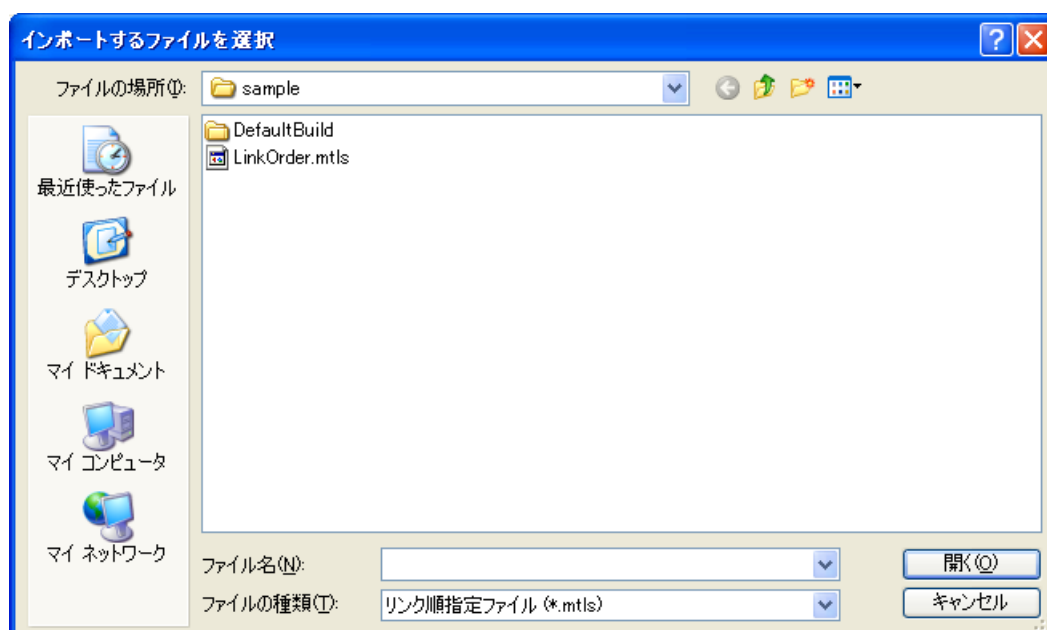
リンク順指定ファイルを編集する際の注意事項を以下に示します。

- ファイル名は1行に1つずつ指定してください。
- ファイル名の大文字／小文字は区別しません。
- “#” で始まる行は、コメント行とみなします。
- 空白文字（半角スペース、タブ）は無視します。

(c) リンク順指定ファイルのインポート

リンク順設定ダイアログの「インポート」ボタンをクリックすると、インポートするファイルを選択ダイアログがオープンします。

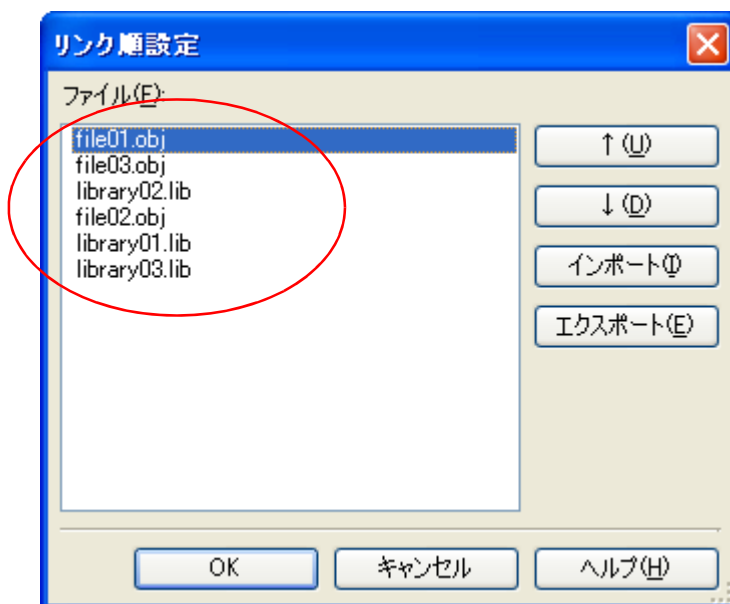
図 2—90 インポートするファイルを選択 ダイアログ



ダイアログ上でリンク順指定ファイルを選択し、[開く] ボタンをクリックしてください。

選択したリンク順指定ファイルからファイル名の記述順を取得し、[リンク順設定 ダイアログ](#)の[ファイル] に反映します。

図 2—91 リンク順設定 ダイアログ (リンク順設定後)



注意 1. リンク順指定ファイルに記述しているが、プロジェクトには追加していないファイルは、表示されません。

該当ファイルが存在する場合は、[出力 パネル](#)にファイル名一覧が表示されます。

2. プロジェクトに追加しているが、リンク順指定ファイルには記述していないファイルは、[ファイル] の最後に表示されます。

3. 同名のファイルが存在する場合は、ポップアップ表示（ファイル名にマウス・カーソルをあわせると表示されます）でファイルの存在場所を確認してください。

リンク順の変更が必要な場合は、[↑]、および[↓] ボタン、またはファイル名のドラッグ・アンド・ドロップにより行ってください。

(3) ファイルのリンク順の設定

[リンク順設定 ダイアログ](#)で [OK] ボタンをクリックすることにより、リンクへのファイルの入力順を設定することができます。

2.11.3 サブプロジェクトのビルド順を変更する

ビルドの実行は、サブプロジェクト、メイン・プロジェクトの順で行いますが、複数のサブプロジェクトを追加している場合、サブプロジェクトのビルド順はプロジェクト・ツリーでの表示順となります。

プロジェクト・ツリーでのサブプロジェクトの表示順を変更するには、移動するサブプロジェクトをドラッグし、移動先でドロップしてください。

2.11.4 ビルド・オプションを一覧表示する

プロジェクト（メイン・プロジェクト、およびサブプロジェクト）に対して、[プロパティ パネル](#)で現在設定しているビルド・オプションを一覧表示することができます。

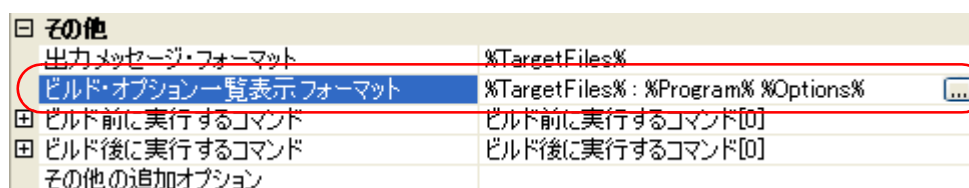
[ビルド] メニュー→[ビルド・オプション一覧]を選択すると、プロジェクトに対する現在のオプションの設定内容を、[出力 パネル](#)の[ビルド・ツール] タブにビルド順に表示します。

備考 ビルド・オプション一覧の表示フォーマットは、変更することができます。

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネル](#)の[共通オプション] タブを選択します。

[その他] カテゴリの[ビルド・オプション一覧表示フォーマット] プロパティを設定してください。

図 2—92 [ビルド・オプション一覧表示フォーマット] プロパティ



次のプレースホルダに対応しています。

%Program% : 実行中のプログラム名に置換します。

%Options% : ビルド時のコマンド・ライン・オプションに置換します。

%TargetFiles% : コンパイル／アセンブル中のファイル名、またはリンク後の出力ファイル名に置換します。

デフォルトでは、“%TargetFiles% : %Program% %Options%”を設定しています。

2.11.5 ビルド対象プロジェクトを変更する

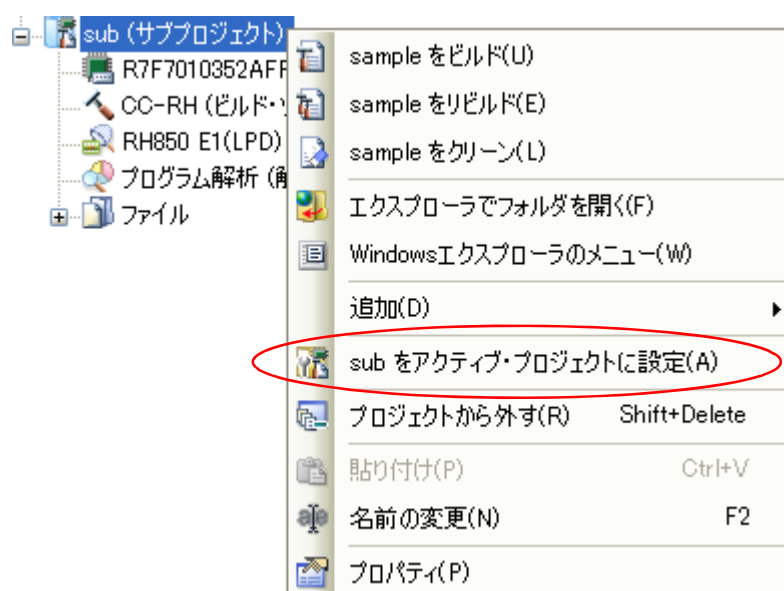
特定のプロジェクト（メイン・プロジェクト、またはサブプロジェクト）を対象にビルドを行う場合、そのプロジェクトを“アクティブ・プロジェクト”として設定する必要があります。

注意 アクティブ・プロジェクトがデバッグ対象プロジェクトとなります。

なお、デバッグ・ツールと接続中はアクティブ・プロジェクトを設定することができないため、アクティブ・プロジェクトを設定する場合はデバッグ・ツールと切断してから行ってください。

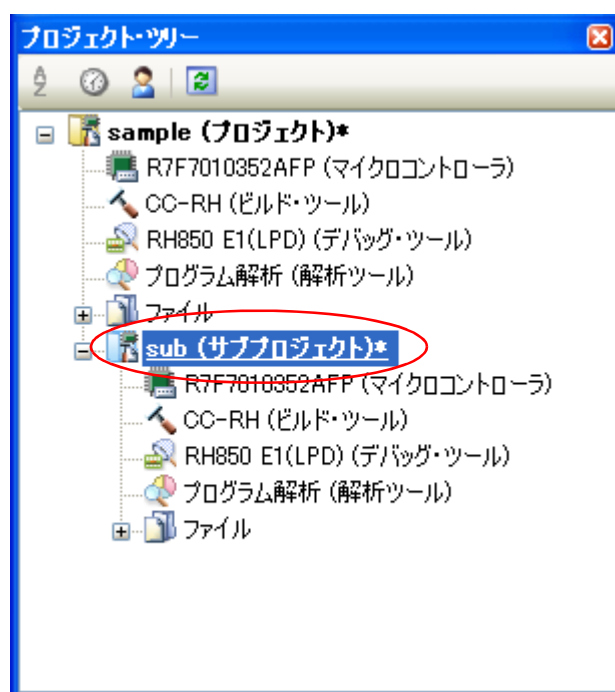
アクティブ・プロジェクトを設定するには、プロジェクト・ツリーでアクティブ・プロジェクトに設定するメイン・プロジェクト・ノード、またはサブプロジェクト・ノード選択し、コンテキスト・メニューの「選択しているプロジェクトをアクティブ・プロジェクトに設定」を選択してください。

図 2—93 「選択しているプロジェクトをアクティブ・プロジェクトに設定」項目



アクティブ・プロジェクトを設定すると、そのプロジェクトには下線を付加します。

図 2—94 アクティブ・プロジェクト



備考 1. プロジェクトの作成直後は、メイン・プロジェクトがアクティブ・プロジェクトとなります。

2. アクティブ・プロジェクトに設定しているサブプロジェクトをプロジェクトから外した場合は、メイン・プロジェクトがアクティブ・プロジェクトとなります。

注意 ツールバーのボタンからビルドを実行すると、プロジェクト全体がビルド対象となります。

アクティブ・プロジェクトのみをビルド対象とするためには、アクティブ・プロジェクトのコンテキスト・メニューからビルドを実行してください。

2.11.6 ビルド・モードを追加する

ビルドの目的に応じてビルド・オプションや定義マクロを変更したい場合、それらの設定を一括して変更することができます。

ビルド・オプションや定義マクロの設定をまとめたものをビルド・モードと呼び、ビルド・モードを変更することにより、ビルド・オプションや定義マクロの設定を毎回変更する必要がなくなります。

ビルド・モードは、デフォルトでは“DefaultBuild”のみ用意していますので、ビルドの目的に応じてユーザが追加してください。

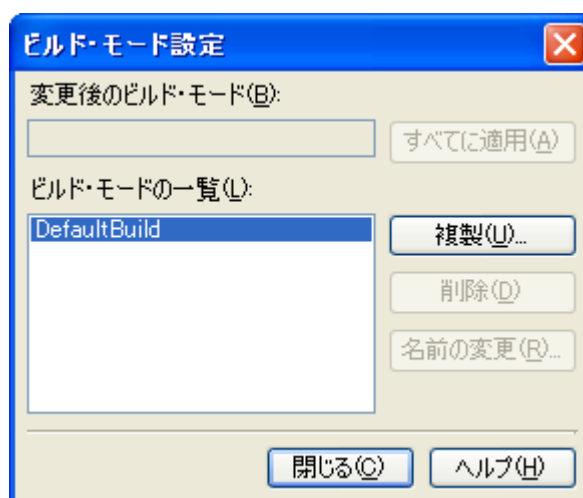
以下に、ビルド・モードの追加方法を示します。

(1) ビルド・モードの新規作成

新規のビルド・モードの作成は、既存のビルド・モードの複製により行います。

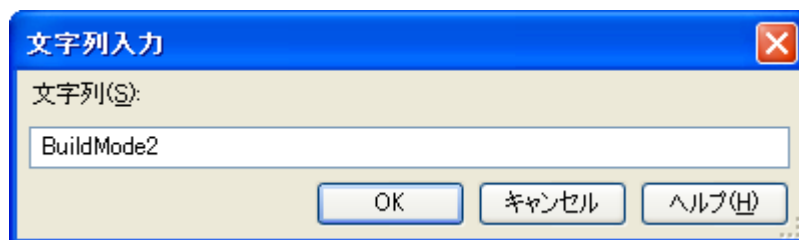
[ビルド] メニュー→ [ビルド・モードの設定 ...] を選択すると、[ビルド・モード設定 ダイアログ](#)がオープンします。

図 2—95 ビルド・モード設定 ダイアログ



[ビルド・モードの一覧] から複製元のビルド・モードを選択したのち、[複製 ...] ボタンをクリックすると、[文字列入力 ダイアログ](#)がオープンします。

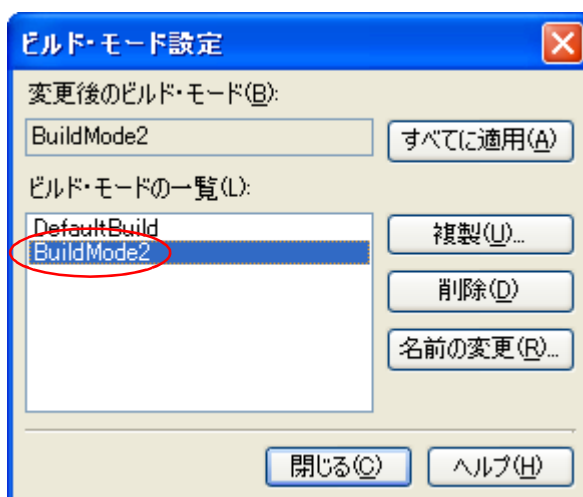
図 2—96 文字列入力 ダイアログ



ダイアログ上で新規作成するビルド・モードの名前を入力し、[OK] ボタンをクリックすると、その名前でビルド・モードを複製します。

プロジェクトに属するメイン・プロジェクト、およびすべてのサブプロジェクトのビルド・モードに、作成したビルド・モードを追加します。

図 2—97 ビルド・モード設定 ダイアログ（ビルド・モード追加後）



(2) ビルド・モードの変更

ビルド・モードを新規に作成したビルド・モードに変更します（「[2.11.7 ビルド・モードを変更する](#)」参照）。

(3) ビルド・モードの設定内容の変更

プロジェクト・ツリー でビルド・ツール・ノードを選択したのち、[プロパティ パネル](#)でビルド・オプションや定義マクロの設定を変更します。

備考 ビルド・モードの作成は、プロジェクトの変更とみなします。

プロジェクトを閉じる際に、ビルド・モードを保存するかどうかの確認を行います。

2.11.7 ビルド・モードを変更する

ビルドの目的に応じてビルド・オプションや定義マクロを変更したい場合、それらの設定を一括して変更することができます。

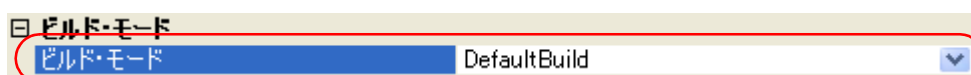
ビルド・オプションや定義マクロの設定をまとめたものをビルド・モードと呼び、ビルド・モードを変更することにより、ビルド・オプションや定義マクロの設定を毎回変更する必要がなくなります。

(1) メイン・プロジェクト、またはサブプロジェクトのビルド・モードを変更する場合

プロジェクト・ツリーで対象プロジェクトのビルド・ツール・ノードを選択したのち、[プロパティパネルの\[共通オプション\] タブ](#)を選択します。

[ビルド・モード] カテゴリの [ビルド・モード] プロパティで変更するビルド・モードを選択してください。

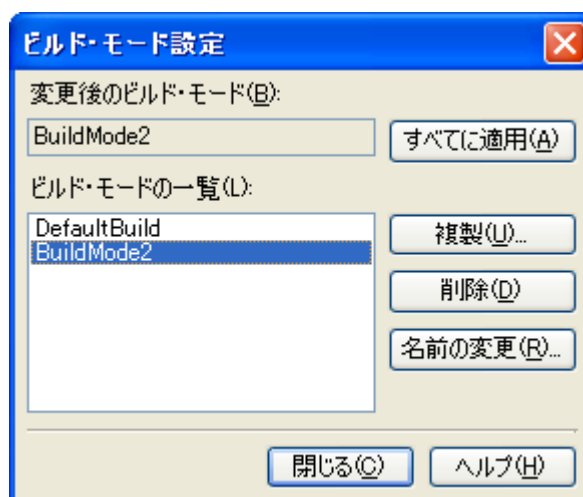
図 2—98 [ビルド・モード] プロパティ



(2) プロジェクト全体のビルド・モードを変更する場合

[ビルド] メニュー→ [ビルド・モードの設定 ...] を選択すると、[ビルド・モード設定 ダイアログ](#)がオープンします。

図 2—99 ビルド・モード設定 ダイアログ



[ビルド・モードの一覧] から変更するビルド・モードを選択すると、[変更後のビルド・モード] に選択したビルド・モードを表示します。

[すべてに適用] ボタンをクリックすると、プロジェクトに属するメイン・プロジェクト、およびすべてのサブプロジェクトのビルド・モードを、ダイアログ上で選択したビルド・モードに変更します。

注意 選択したビルド・モードが存在しないプロジェクトについては、“DefaultBuild”を選択したビルド・モード名で複製し、複製したビルド・モードに変更します。

備考 1. ビルド・モードは、デフォルトでは“DefaultBuild”のみ用意しています。

ビルド・モードの追加方法については、「[2.11.6 ビルド・モードを追加する](#)」を参照してください。

2. ビルド・モードの一覧でビルド・モードを選択したのち、[名前の変更] ボタンをクリックすることにより、ビルド・モードの名前を変更することができます。

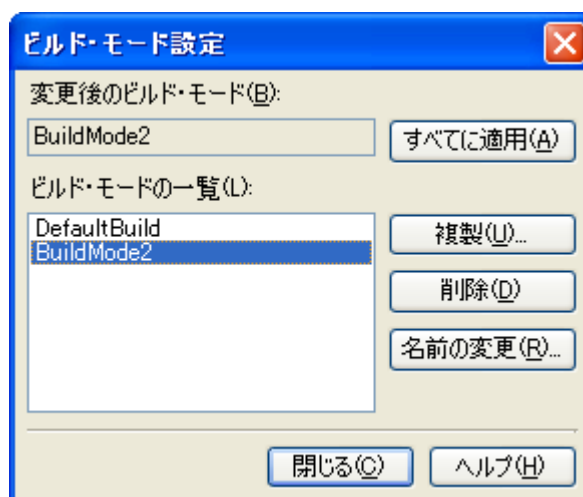
ただし、“DefaultBuild”は名前を変更することができません。

2.11.8 ビルド・モードを削除する

ビルド・モードの削除は、[ビルド・モード設定 ダイアログ](#)で行います。

[ビルド] メニュー→[ビルド・モードの設定 ...] を選択すると、ダイアログがオープンします。

図 2—100 ビルド・モード設定 ダイアログ



[ビルド・モードの一覧] で削除するビルド・モードを選択したのち、[削除] ボタンをクリックすると、以下のメッセージ ダイアログがオープンします。

図 2—101 メッセージ ダイアログ



処理を継続するには、ダイアログ上で [OK] をクリックしてください。

選択したビルド・モードをプロジェクトから削除します。

注意 “DefaultBuild” を削除することはできません。

2. 11. 9 現在のビルド・オプションをプロジェクトの標準に設定する

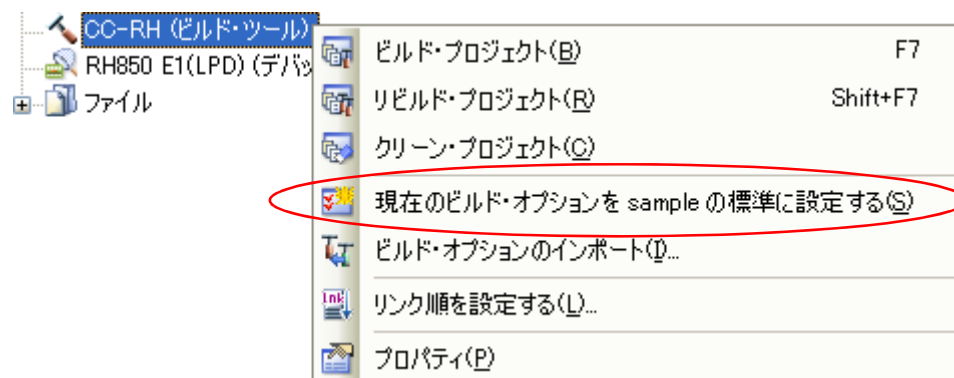
プロパティ パネルにおいて、標準ビルド・オプションの設定に変更を加えると、プロパティの値を太字表示します。

図 2—102 プロパティ パネル（標準ビルド・オプション変更後）

出力ファイルの種類と場所	
出力ファイルの種類	実行形式(ヘキサ・ファイル)
デバイス共通オブジェクト・ファイルを出力する	はい(RH850アーキテクチャ共通)(-Xcommon=rh850)
CPUコアの指定	G3M向けオブジェクト(-Xcpu=g3m)
クロス・リファレンス情報を出力する	いいえ
中間ファイル出力フォルダ	%BuildModeName%

現在選択しているプロジェクト（メイン・プロジェクト、またはサブプロジェクト）のビルド・オプションを標準ビルド・オプションとする（太字表示を解除する）には、プロジェクト・ツリーでビルド・ツール・ノードを選択し、コンテキスト・メニューの「現在のビルド・オプションを選択しているプロジェクトの標準に設定する」を選択してください。

図 2—103 「現在のビルド・オプションを選択しているプロジェクトの標準に設定する」項目



標準ビルド・オプションに設定後のプロパティの値は、以下ようになります。

図 2—104 プロパティ パネル（標準ビルド・オプション設定後）

出力ファイルの種類と場所	
出力ファイルの種類	実行形式(ヘキサ・ファイル)
デバイス共通オブジェクト・ファイルを出力する	はい(RH850アーキテクチャ共通)(-Xcommon=rh850)
CPUコアの指定	G3M向けオブジェクト(-Xcpu=g3m)
クロス・リファレンス情報を出力する	いいえ
中間ファイル出力フォルダ	%BuildModeName%

注意 メイン・プロジェクトを選択している場合、メイン・プロジェクトの設定のみ行います。サブプロジェクトを追加していても、サブプロジェクトの設定は行いません。

2.12 ビルドを実行する

ここでは、ビルドの実行に関する操作を説明します。

(1) ビルドの種類

ビルドには、次の種類があります。

表 2—1 ビルドの種類

種類	説明
ビルド	ビルド対象ファイルのうち、更新したファイルのみビルドを実行します。 →「 2.12.1 更新ファイルのビルドを実行する 」参照
リビルド	ビルド対象のすべてのファイルのビルドを実行します。 →「 2.12.2 すべてのファイルのビルドを実行する 」参照
ラピッド・ビルド	ビルド設定の変更と平行してビルドを実行します。 →「 2.12.3 他の処理と平行してビルドを実行する 」参照
バッチ・ビルド	プロジェクトが持つビルド・モードを一括してビルドを実行します。 →「 2.12.4 ビルド・モードを一括してビルドを実行する 」参照

備考 1. ビルドの実行は、サブプロジェクト、メイン・プロジェクトの順で行います。

サブプロジェクトは、プロジェクト・ツリーでの表示順にビルドを行います（「[2.11.3 サブプロジェクトのビルド順を変更する](#)」参照）。

ただし、プロジェクト間に依存関係がある場合は、ビルド対象プロジェクトよりも先に、そのプロジェクトが依存しているプロジェクトのビルドを行います。

2. ビルド、リビルド、バッチ・ビルドを実行する際、[エディタ パネル](#)で編集集中のファイルがある場合は、該当ファイルを一括して保存します。

(2) 実行結果の表示

ビルドの実行結果（ビルド・ツールの出力メッセージ）は、[出力 パネル](#)の各タブに表示します。

- ビルド、リビルド、バッチ・ビルドの場合

→ [すべてのメッセージ] タブ、および [ビルド・ツール] タブ

- ラピッド・ビルドの場合

→ [ラピッド・ビルド] タブ

図 2—105 ビルドの実行結果（ビルド、リビルド、バッチ・ビルドの場合）

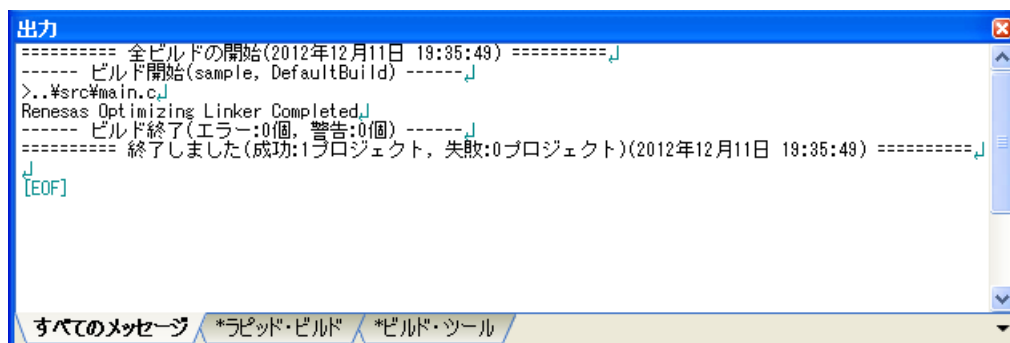
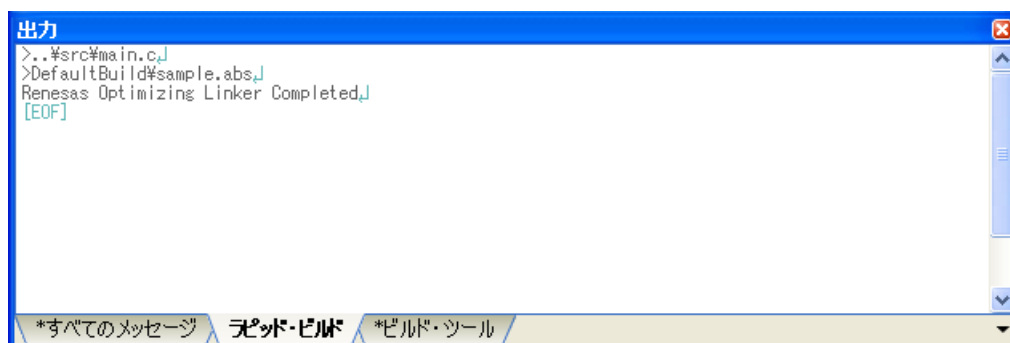


図 2—106 ビルドの実行結果（ラピッド・ビルドの場合）

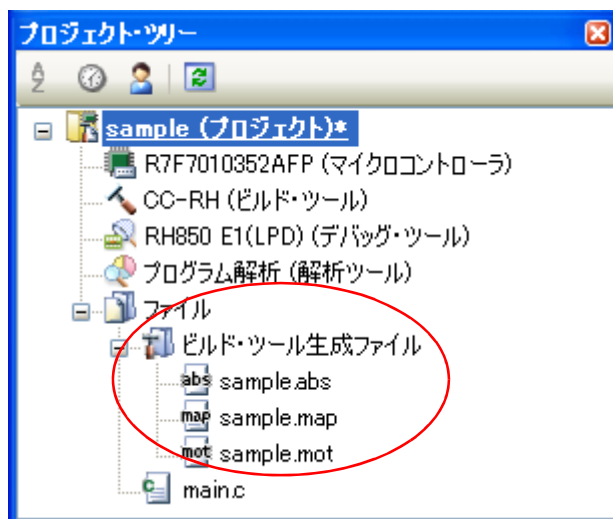


備考 1. [ラピッド・ビルド] タブの表示文字列は、淡色表示になります。

2. 出力したメッセージからファイル名／行番号を獲得できる場合、メッセージ上でダブルクリックすると、ファイルの該当する行へジャンプすることができます。
3. 警告メッセージ、またはエラー・メッセージを表示している行にキャレットがある状態で、[F1] キーを押下すると、その行のメッセージに関するヘルプを表示することができます。

ビルド・ツールの生成ファイルは、プロジェクト・ツリーパネルのビルド・ツール生成ファイル・ノードに表示します。

図 2—107 ビルド・ツールの生成ファイル



備考 ビルド・ツール生成ファイル・ノードに表示するのは、以下のファイルです。

- ライブラリ用のプロジェクト以外の場合
 - ロード・モジュール・ファイル (*.abs)
 - インテル拡張ヘキサ・ファイル (*.hex)
 - モトローラ・S タイプ・ファイル (*.mot)
 - バイナリ・ファイル (*.bin)
 - リンク・マップ・ファイル (*.map)
 - ライブラリ・ファイル (*.lib)
 - ライブラリ・リスト・ファイル (*.lbp)
 - エラー・メッセージ・ファイル (*.err)
- ライブラリ用のプロジェクトの場合
 - ライブラリ・ファイル (*.lib)
 - エラー・メッセージ・ファイル (*.err)

注意 ビルド・ツール生成ファイル・ノードは、ビルド時に作成するノードです。

ビルド後にプロジェクトの再読み込みを行った場合、本ノードは表示されなくなります。

2.12.1 更新ファイルのビルドを実行する

ビルド対象ファイルのうち、更新したファイルのみビルドを実行します（以降，“ビルド”と呼びます）。

ビルドの実行は、プロジェクト全体（メイン・プロジェクト、およびサブプロジェクト）、またはアクティブ・プロジェクト（「2.11.5 ビルド対象プロジェクトを変更する」参照）に対して行います。

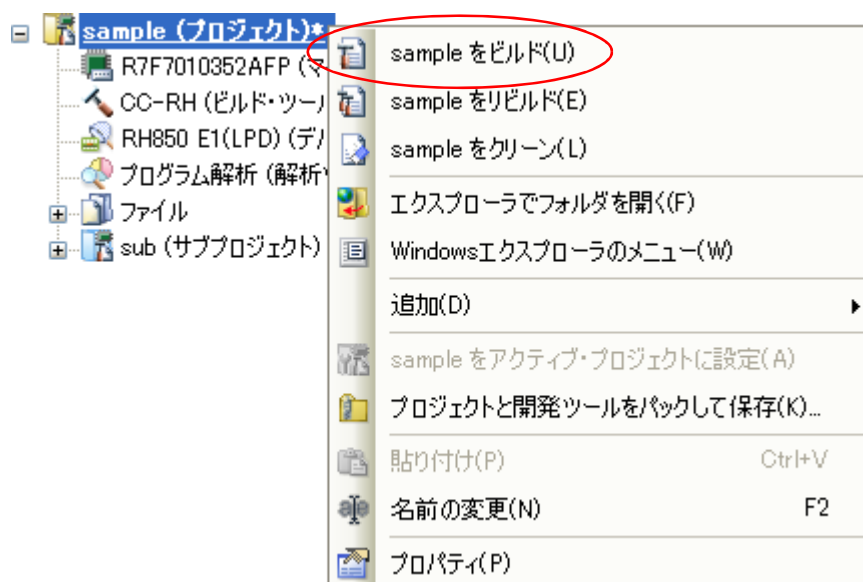
(1) プロジェクト全体のビルドを実行する場合

ツールバーの  ボタンをクリックしてください。

(2) アクティブ・プロジェクトのビルドを実行する場合

プロジェクトを選択し、コンテキスト・メニューの「アクティブ・プロジェクトをビルド」を選択してください。

図 2—108 「アクティブ・プロジェクトをビルド」項目



備考 ヘッダ・ファイルを編集後にビルドを実行してもインクルードしているソース・ファイルがビルドされない場合は、ファイルの依存関係を更新してください（「2.3.6 ファイルの依存関係を更新する」参照）。

2.12.2 すべてのファイルのビルドを実行する

ビルド対象のすべてのファイルのビルドを実行します（以降，“リビルド”と呼びます）。

また、クロス・リファレンス・ファイルの削除も行います。

リビルドの実行は、プロジェクト全体（メイン・プロジェクト、およびサブプロジェクト）、またはアクティブ・プロジェクト（「2.11.5 ビルド対象プロジェクトを変更する」参照）に対して行います。

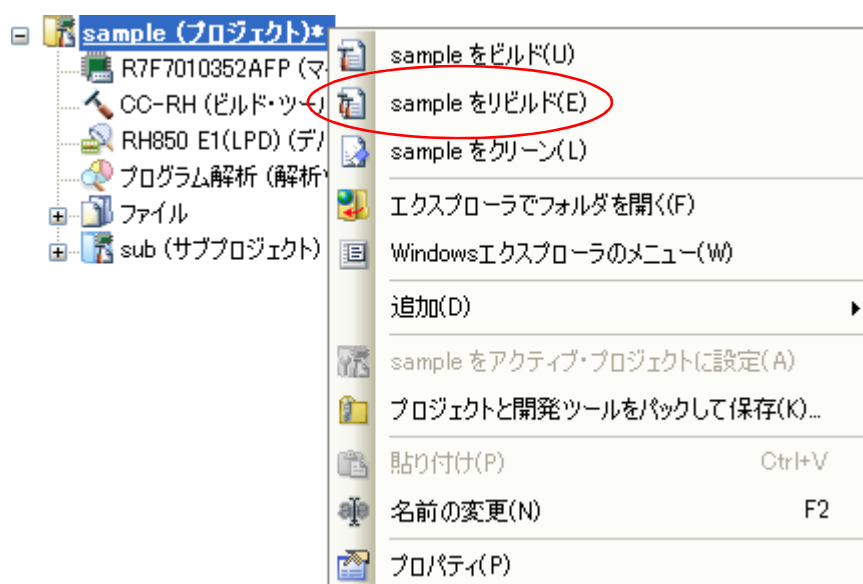
(1) プロジェクト全体のリビルドを実行する場合

ツールバーの  ボタンをクリックしてください。

(2) アクティブ・プロジェクトのリビルドを実行する場合

プロジェクトを選択し、コンテキスト・メニューの「アクティブ・プロジェクトをリビルド」を選択してください。

図 2—109 「アクティブ・プロジェクトをリビルド」項目



2.12.3 他の処理と平行してビルドを実行する

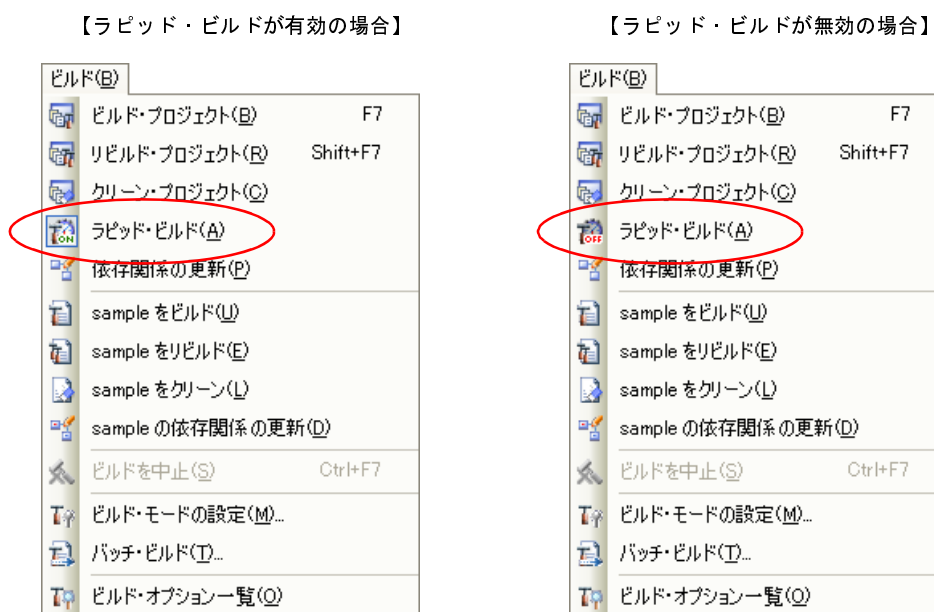
以下のタイミングでビルドを自動で開始する機能があります（以降，“ラピッド・ビルド”と呼びます）。

- プロジェクトに追加している C ソース・ファイル，アセンブリ・ソース・ファイル，シンボル・アドレス・ファイル，ヘッダ・ファイル，オブジェクト・ファイル，リロケータブル・ファイル，およびライブラリ・ファイルを更新したとき
- プロジェクトにビルド対象ファイルを追加，または削除したとき
- オブジェクト・ファイル，およびライブラリ・ファイルのリンク順を変更したとき
- ビルド・ツール，およびビルド対象ファイルのプロパティを変更したとき

ラピッド・ビルドを有効にすることにより，上記の操作と平行してビルドを行うことができます。

ラピッド・ビルドの有効／無効は，[ビルド] メニュー→[ラピッド・ビルド] の選択により，切り替えます。デフォルトでは，有効となっています。

図 2—110 [ラピッド・ビルド] 項目



- 備考 1.** ソース・ファイル編集後，[Ctrl] + [S] キーの押下により，こまめに上書き保存することを推奨します。
- 2.** ラピッド・ビルドの有効／無効は，プロジェクト全体（メイン・プロジェクト，およびサブプロジェクト）に対して設定します。
- 3.** ラピッド・ビルドの実行中に，ラピッド・ビルドを無効に切り替えた場合は，その場でラピッド・ビルドの実行を中止します。

注意 この機能は，**オプションダイアログ**の[ビルド／デバッグ]カテゴリの「登録されたファイルの変更を監視する」をチェックした場合，外部テキスト・エディタでも有効となります。

2.12.4 ビルド・モードを一括してビルドを実行する

プロジェクト（メイン・プロジェクト、およびサブプロジェクト）が持つビルド・モードを一括して、ビルド／リビルド／クリーンを行うことができます（以降、“バッチ・ビルド”と呼びます）。

備考 ビルド、リビルド、クリーンについては、それぞれ以下を参照してください。

- ビルド → 「2.12.1 更新ファイルのビルドを実行する」参照
- リビルド → 「2.12.2 すべてのファイルのビルドを実行する」参照
- クリーン → 「2.12.9 中間ファイル、生成ファイルを削除する」参照

[ビルド] メニュー→ [バッチ・ビルド ...] を選択すると、**バッチ・ビルド ダイアログ**がオープンします。

図 2—111 バッチ・ビルド ダイアログ



ダイアログ上には、現在開いているプロジェクトが持つメイン・プロジェクト、およびサブプロジェクトの名前と、それらが持つビルド・モード、定義マクロの組み合わせの一覧を表示します。

バッチ・ビルドを行いたいメイン・プロジェクト、およびサブプロジェクトとビルド・モードの組み合わせをチェック・ボックスにより選択し、[ビルド] / [リビルド] / [クリーン] ボタンをクリックしてください。

備考 バッチ・ビルド順は、プロジェクトのビルド順に従い、サブプロジェクト、メイン・プロジェクトの順となります。

1つのメイン・プロジェクト、またはサブプロジェクトについて複数のビルド・モードを選択した場合は、そのサブプロジェクトで選択しているすべてのビルド・モードでビルドを行ったのち、次のサブプロジェクト、またはメイン・プロジェクトのビルドを行います。

2.12.5 複数のファイルのコンパイル／アセンブル／リンクを同時に実行する

ビルド対象ファイルが複数存在する場合、ccrh コマンド 1 回の呼び出しでファイルを一括してコンパイル／アセンブル／リンクを行うことができます（以降，“一括ビルド”と呼びます）。

ccrh コマンドの呼び出しイメージを以下に示します。

例 ビルド対象ファイルが aaa.c, bbb.c, ccc.c の場合

- 一括ビルドを行う場合

```
>ccrh -Xcommon=v850e3v5 aaa.c bbb.c ccc.c ← aaa.abs を生成
```

- 一括ビルドを行わない場合

```
>ccrh -Xcommon=v850e3v5 aaa.c ← aaa.obj を生成
>ccrh -Xcommon=v850e3v5 bbb.c ← bbb.obj を生成
>ccrh -Xcommon=v850e3v5 ccc.c ← ccc.obj を生成
>ccrh -Xcommon=v850e3v5 aaa.obj bbb.obj ccc.obj ← aaa.abs を生成
```

一括ビルドを行うかどうかは、プロパティで設定します。

プロジェクト・ツリーでビルド・ツール・ノードを選択し、[プロパティ パネルの \[共通オプション\] タブ](#)を選択します。

[ビルド方法] カテゴリの [一括ビルドを行う] プロパティで、[はい] を選択してください。

図 2—112 [一括ビルドを行う] プロパティ



このとき、[\[アセンブル・オプション\] タブ](#)は非表示となります。

アセンブリ・ソース・ファイル（個別オプションを設定しているファイルを除く）のアセンブルには、[\[コンパイル・オプション\] タブ](#)の設定を使用します。

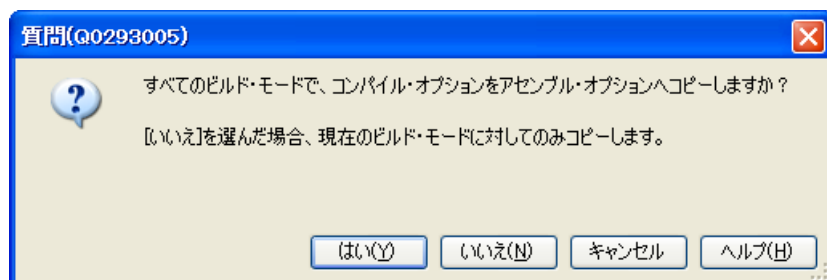
備考 1. 個別オプションを設定しているファイル、およびビルド前実行の対象となっているファイルは、一括ビルドの対象となります。

一括ビルドの対象外となったファイルについては、個別にビルドを行います。

2. ソース・ファイルが、生成するオブジェクト・ファイル、および関連するプロパティやプロジェクトなどより古い場合は、ソース・ファイルではなく、オブジェクト・ファイルがビルド対象となります。

3. [一括ビルドを行う] プロパティを [はい] から [いいえ] に変更すると、以下のメッセージダイアログがオープンします。

図 2—113 メッセージ ダイアログ



ダイアログ上で [はい] をクリックすると、すべてのビルド・モードの同一の [コンパイル・オプション] タブの設定を [アセンブル・オプション] タブにコピーします。

[いいえ] をクリックすると、現在のビルド・モードの同一の [コンパイル・オプション] タブの設定を [アセンブル・オプション] タブにコピーします。

[キャンセル] をクリックすると、[アセンブル・オプション] タブは非表示となる前の状態で表示します。

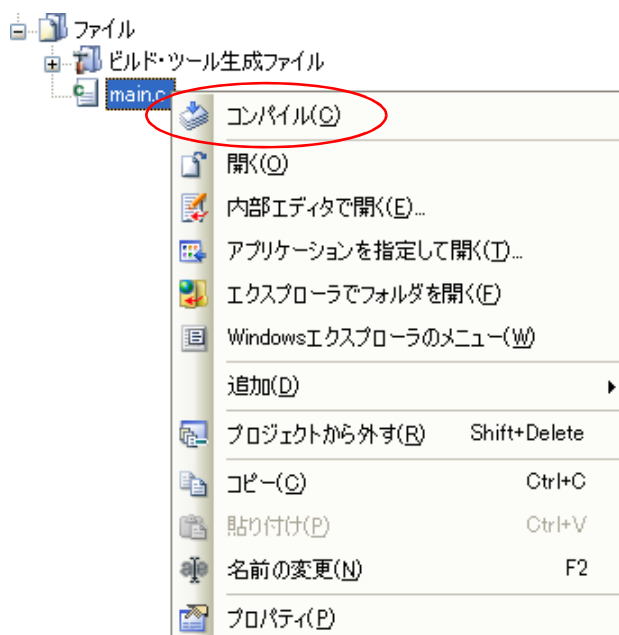
2. 12. 6 ファイル単位でコンパイル／アセンブルする

プロジェクトに追加している各ソース・ファイルに対して、コンパイル、またはアセンブルのみを行うことができます。

(1) C ソース・ファイルをコンパイルする場合

プロジェクト・ツリーでCソース・ファイルを選択し、コンテキスト・メニューの [コンパイル] を選択してください。

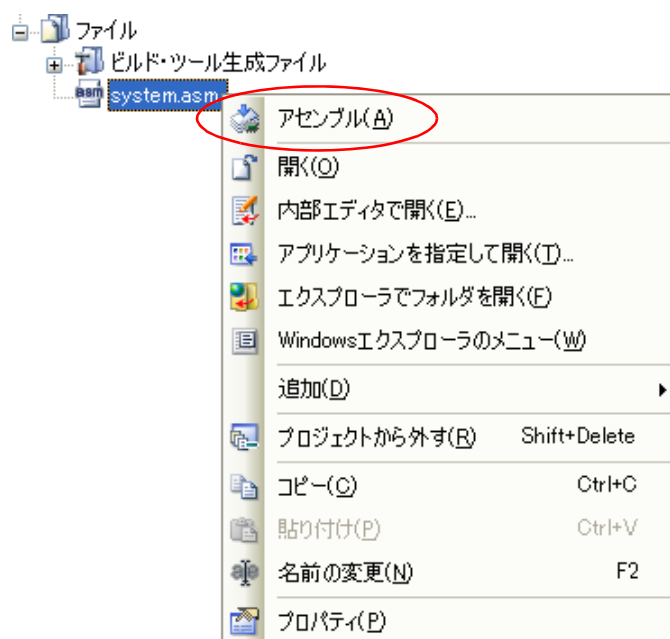
図 2—114 [コンパイル] 項目



(2) アセンブリ・ソース・ファイルをアセンブルする場合

プロジェクト・ツリーでアセンブリ・ソース・ファイルを選択し、コンテキスト・メニューの「アセンブル」を選択してください。

図 2—115 「アセンブル」項目



2.12.7 ビルドの実行を中止する

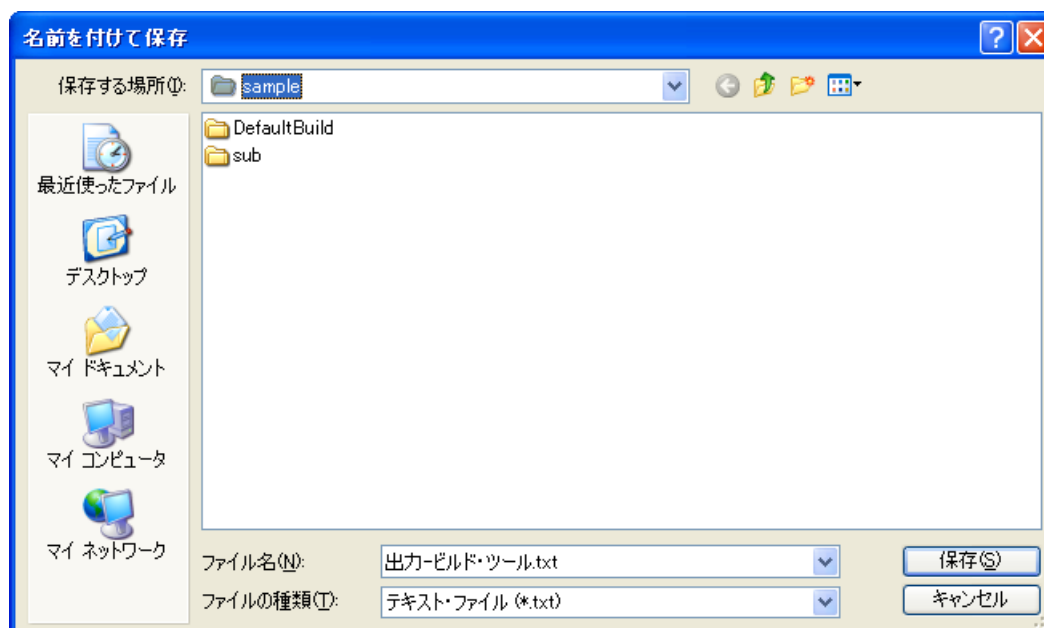
実行中のビルド、リビルド、バッチ・ビルドを中止するには、ツールバーの  ボタンをクリックしてください。

2.12.8 ビルド結果をファイルに保存する

出力パネルに表示するビルドの実行結果（ビルド・ツールの出力メッセージ）をテキスト・ファイルに保存することができます。

パネル上で［ビルド・ツール］タブを選択し、［ファイル］メニュー→［名前を付けて 出力ビルド・ツールを保存 ...］を選択すると、名前を付けて保存ダイアログがオープンします。

図 2—116 名前を付けて保存 ダイアログ



ダイアログ上で、保存するテキスト・ファイル名と保存場所を指定し、［保存］ボタンをクリックしてください。

2.12.9 中間ファイル、生成ファイルを削除する

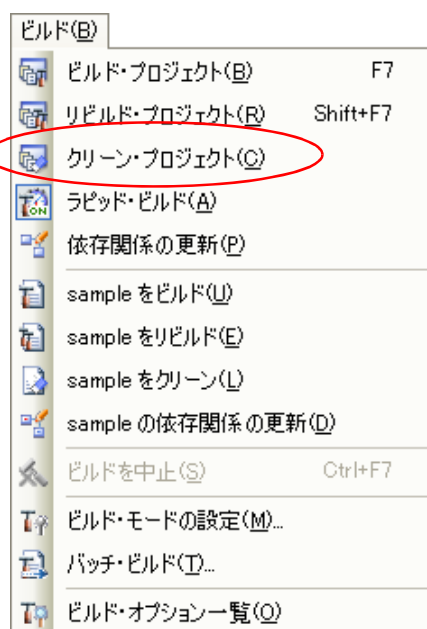
ビルドの実行により出力した中間ファイル、生成ファイルをすべて削除することができます（以降、“クリーン”と呼びます）。

クリーンの実行は、プロジェクト全体（メイン・プロジェクト、およびサブプロジェクト）、またはアクティブ・プロジェクト（「2.11.5 ビルド対象プロジェクトを変更する」参照）に対して行います。

(1) プロジェクト全体のクリーンを実行する場合

[ビルド] メニュー→ [クリーン・プロジェクト] を選択してください。

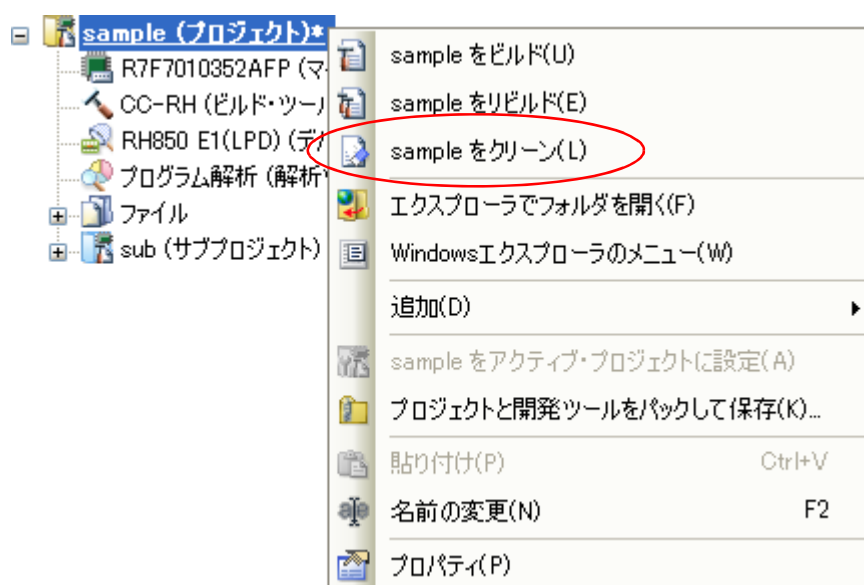
図 2—117 「クリーン・プロジェクト」項目



(2) アクティブ・プロジェクトのクリーンを実行する場合

プロジェクトを選択し、コンテキスト・メニューの [アクティブ・プロジェクトをクリーン] を選択してください。

図 2—118 「アクティブ・プロジェクトをクリーン」項目



2.13 スタックを見積もる

スタックを見積もるには、Call Walker を使用します。

Call Walker では、静的に解析処理を行うことにより、シンボルの呼び出し関係をツリー形式で表示すると共に、シンボル単位のスタック情報（シンボル名、属性、アドレス、サイズ、スタック・サイズ、ファイル名）をリスト形式で表示します。

Call Walker の起動は、Windows の [スタート] → [すべてのプログラム] → [Renesas Electronics CubeSuite+] → [ユーティリティ] → [Call Walker] を選択することにより行います。

備考 Windows 8 の場合は、スタート画面の [Call Walker] をダブルクリックしてください。

また、Call Walker の終了は、Call Walker の [File] メニュー → [Exit] を選択することにより行います。

操作手順については、Call Walker の [Help] メニュー → [Help Topics] を参照してください。

第3章 ビルドの出力リスト

この章では、ビルドにより各コマンドが出力する各種リストのフォーマットなどについて説明します。

3.1 アセンブル・リスト・ファイル

ここでは、アセンブル・リスト・ファイルについて説明します。

アセンブル・リストとは、ソースをコンパイル、アセンブルして出力するコードをリスト形式にしたものです。

これにより、コンパイル、アセンブルした結果が、どのようなコードになっているかを確認することができます。

アセンブル・リスト・ファイルの出力方法を以下に示します。

(1) CubeSuite+ の場合

プロジェクト・ツリー パネルでビルド・ツール・ノードを選択したのち、プロパティ パネルの [コンパイル・オプション] タブを選択します。

アセンブル・リスト・ファイルを出力するには、[アセンブル・リスト] カテゴリの [アセンブル・リスト・ファイルを出力する] プロパティで [はい (-Xasm_option=-Xprn_path)] を選択します。

出力先は、[アセンブル・リスト・ファイル出力フォルダ] プロパティで指定します。

ファイル名は、ソース・ファイル名の拡張子を “.prn” で置き換えた名前となります。

なお、[出力コード] カテゴリの [アセンブリ・ソース・ファイルにコメントを出力する] プロパティで [はい (-Xpass_source)] を選択すると、出力するアセンブル・リスト・ファイル中に、アセンブリ・ソース・プログラムに対応した C ソース・プログラムをコメントとして出力します。

(2) コマンド・ラインの場合

-Xprn_path オプションを指定すると、カレント・フォルダに、ソース・ファイル名の拡張子を “.prn” で置き換えた名前でアセンブル・リスト・ファイルを出力します。

-Xprn_path オプションのパラメータにより、出力フォルダ、ファイル名を指定することもできます。

また、-Xpass_source オプションを同時に指定すると、出力するアセンブル・リスト・ファイル中に、アセンブリ・ソース・プログラムに対応した C ソース・プログラムをコメントとして出力します。

3.1.1 アセンブル・リストの構成

アセンブル・リストの構成と内容を以下に示します。

出力情報	説明
アセンブル・リスト	アセンブル時のロケーション・カウンタ値、コード、行番号、ソース・プログラム
セクション・リスト	セクションの種類、サイズ、名称
コマンド・ライン情報	アセンブラのコマンド・ライン文字列

3.1.2 アセンブル・リスト

アセンブル時のロケーション・カウンタ値、コード、行番号、ソース・プログラムを出力します。

アセンブル・リストの出力例を以下に示します。

(1) OFFSET	(2) CODE	(3) (4) NO SOURCE STATEMENT
00000000		1 #CC-RH Compiler RH850 Assembler Source
00000000		2 #@ CC-RH Version : VX.XX.XXx [XX Xxx XXXX]
00000000		3 #@ Command : main.c -Xcommon=v850e3v5 -S
00000000		4 #@ compiled at Xxx Xxx XX XX:XX:XX XXXX
00000000		5 .cseg text
00000000		6 ld.w \$_data,r12
00000000	440E0000	-- movhi 0x0, gp, r1
00000004	21670100	-- ld.w 0x0[r1], r12
00000008		7
00000000		8 .dseg data
00000000	00000000	9 _data: .dw 0
00000004		10

項番	説明
(1)	ロケーション・カウンタ値 その行のソース・プログラムに対して生成したコードの先頭に対するロケーション・カウンタ値を出力します。
(2)	コード その行のソース・プログラムに対して生成したコード（機械語命令、またはデータ）を出力します。 1バイトごとに2桁の16進数で表記します。
(3)	行番号 その行の行番号を出力します。 10進数で表記します。
(4)	ソース・プログラム その行のソース・プログラムを出力します。 その行の命令に対して命令展開が生じた場合、その命令展開によって生成した機械語命令の命令列を逆アセンブルした結果を、“--”以降に示します。 コンパイラ情報（1～4行目）は、コンパイラが出力したアセンブリ・ソース・ファイルをアセンブルした場合のみ出力します。

3.1.3 セクション・リスト

セクションの種類、サイズ、名称を出力します。

セクション・リストの出力例を以下に示します。

Section List		
(1)	(2)	(3)
Attr	Size	Name
TEXT	8 (00000008)	.text
DATA	4 (00000004)	.data

項番	説明
(1)	セクションの種類 セクションの種類を再配置属性として出力します。
(2)	セクション・サイズ セクションのサイズを出力します（単位：バイト）。 10 進数、およびかっこ内に 16 進数で表記します。
(3)	セクション名 セクション名を出力します。

3.1.4 コマンド・ライン情報

アセンブラのコマンド・ライン文字列を出力します。

コマンド・ライン情報の出力例を以下に示します。

Command Line Parameter	
a.asm -Xcommon=v850e3v5 -Xprn_path	(1)

項番	説明
(1)	コマンド・ライン文字列 アセンブラに指定したコマンド・ライン文字列を出力します。

3.2 リンク・マップ・ファイル

ここでは、リンク・マップ・ファイルについて説明します。

リンク・マップとは、リンク結果の情報が書かれたもので、セクションの配置アドレスなどの情報を知ることができます。

リンク・マップ・ファイルの出力方法を以下に示します。

(1) CubeSuite+ の場合

プロジェクト・ツリー パネルでビルド・ツール・ノードを選択したのち、プロパティ パネルの [リンク・オプション] タブを選択します。

リンク・マップ・ファイルを出力するには、[リスト] カテゴリの [リンク・マップ・ファイルを出力する] プロパティで [はい] を選択します。

出力先は、[出力ファイル] カテゴリの [出力フォルダ] プロパティで指定します。

また、プロジェクト・ツリーのビルド・ツール生成ファイル・ノードにも表示します。

~~ファイル名は、出力ファイル名の拡張子を“.map”で置き換えた名前となります。~~
ファイル名は、プロジェクト・ファイル名の拡張子を“.map”で置き換えた名前となります。

(2) コマンド・ラインの場合

-form=library, または -extract オプションの未指定時に、-list オプションを指定すると、カレント・フォルダに、出力ファイル名の拡張子を“.map”で置き換えた名前で行リンク・マップ・ファイルを出力します。

また、-show オプションを同時に指定することにより、リンク・マップ・ファイルへの出力情報を指定することができます。

詳細については、「[3.2.1 リンク・マップの構成](#)」を参照してください。

3.2.1 リンク・マップの構成

リンク・マップの構成と内容を以下に示します。

出力情報	説明	-show オプション 指定	-show オプション 省略時
オプション情報	コマンド・ライン、サブコマンド・ファイルで指定したオプション列	—	出力する
エラー情報	エラー・メッセージ	—	出力する
リンク・マップ情報	セクション名、先頭/最終アドレス、サイズ、種別	—	出力する
合計セクション・サイズ	RAM, ROM, およびプログラム・セクションの合計サイズ	-show=total_size	出力しない
シンボル情報	静的定義シンボル名、アドレス、サイズ、種別（アドレス順）、最適化実行の有無 -show=reference を指定した場合は、各シンボルの参照回数も出力します。	-show=symbol -show=reference	出力しない
クロス・リファレンス情報	シンボルの参照情報	-show=xreference	出力しない

注意 -show オプションは、-list オプションを指定した場合に有効となります。

-show オプションについての詳細は、「[-Show](#)」を参照してください。

3.2.2 オプション情報

コマンド・ライン、サブコマンド・ファイルで指定したオプション列を出力します。

コマンド・ライン、サブコマンド・ファイルで次のように指定した場合のオプション情報の出力例を以下に示します。

<コマンド・ライン>

```
>rlink -subcommand=test.sub -list -show
```

<サブコマンド・ファイル test.sub >

```
input sample.obj
```

```
*** Options ***

-subcommand=test.sub      (1)
input sample.obj          (2)
-list                     (1)
-show                     (1)
```

項番	説明
(1)	コマンド・ラインで指定したオプション コマンド・ラインで指定したオプションを出力します（指定順）。
(2)	サブコマンド・ファイル内で指定したオプション サブコマンド・ファイル test.sub 内で指定したオプションを出力します。

3.2.3 エラー情報

エラー・メッセージを出力します。

エラー情報の出力例を以下に示します。

```
*** Error information ***

** E0562310 Undefined external symbol "_func_02" referenced in "sample.obj" (1)
```

項番	説明
(1)	エラー・メッセージ エラー・メッセージを出力します。

3.2.4 リンク・マップ情報

各セクションの先頭／最終アドレス、サイズ、種別をアドレス順に出力します。

リンク・マップ情報の出力例を以下に示します。

*** Mapping List ***				
(1)	(2)	(3)	(4)	(5)
SECTION	START	END	SIZE	ALIGN
.text	00000000	0000003b	3c	2
.data	fe600006	fe600003	4	4
.bss	fe600004	fe60000b	8	4

項番	説明
(1)	セクション名 セクション名を出力します。
(2)	先頭アドレス 先頭アドレスを出力します。 16 進数で表記します。
(3)	最終アドレス 最終アドレスを出力します。 16 進数で表記します。
(4)	セクション・サイズ セクション・サイズを出力します（単位：バイト）。 16 進数で表記します。
(5)	セクションのアライメント数 セクションのアライメント数を出力します。

3.2.5 合計セクション・サイズ

-show=total_size オプションを指定した場合、ROM セクション、RAM セクション、およびプログラム・セクションの合計サイズを出力します。

合計セクション・サイズの出力例を以下に示します。

*** Total Section Size ***	
RAMDATA SECTION:	00000660 Byte(s) (1)
ROMDATA SECTION:	00000174 Byte(s) (2)
PROGRAM SECTION:	000016d6 Byte(s) (3)

項番	説明
(1)	RAM データ・セクションの合計サイズ RAM データ・セクションの合計サイズを出力します。 16 進数で表記します。
(2)	ROM データ・セクションの合計サイズ ROM データ・セクションの合計サイズを出力します。 16 進数で表記します。
(3)	プログラム・セクションの合計サイズ プログラム・セクションの合計サイズを出力します。 16 進数で表記します。

3.2.6 シンボル情報

-show=symbol オプションを指定した場合は、外部定義シンボル、または静的内部定義シンボルのアドレス、サイズ、種別、最適化実行の有無をアドレス順で出力します。

また、-show=reference オプションを指定した場合は、各シンボルの参照回数も出力します。

シンボル情報の出力例を以下に示します。

```

*** Symbol List ***

SECTION=(1)
FILE=(2)

          (3)          (4)          (5)
          START        END          SIZE
(6)      (7)      (8)      (9)      (10)  (11)
SYMBOL   ADDR      SIZE      INFO      COUNTS  OPT

SECTION=.text
FILE=sample.obj
          00000000      00000023      24
_main
          00000000          0      func ,g      0
_func_01
          00000018          0      func ,g      0
SECTION=.bss
FILE=sample.obj
          fe600004      fe60000b      8
_gvall
          fe600004          4      data ,g      0

```

項番	説明
(1)	セクション名 セクション名を出力します。

項番	説明
(2)	ファイル名 ファイル名を出力します。
(3)	先頭アドレス (2) のファイルに含まれる該当セクションの先頭アドレスを出力します。 16 進数で表記します。
(4)	最終アドレス (2) のファイルに含まれる該当セクションの最終アドレスを出力します。 16 進数で表記します。
(5)	セクション・サイズ (2) のファイルに含まれる該当セクションのセクション・サイズを出力します（単位：バイト）。 16 進数で表記します。
(6)	シンボル名 シンボル名を出力します。
(7)	シンボル・アドレス シンボル・アドレスを出力します。 16 進数で表記します。
(8)	シンボル・サイズ シンボル・サイズを出力します（単位：バイト）。 16 進数で表記します。
(9)	シンボル種別 データ種別、および宣言種別を出力します。 - データ種別 func : 関数名 data : 変数名 entry : エントリ関数名 none : 未設定（ラベル、アセンブラ・シンボル） - 宣言種別 g : 外部定義 l : 内部定義
(10)	シンボル参照回数 シンボル参照回数を出力します。 16 進数で表記します。 -show=reference オプションを指定した場合のみ出力します。 参照回数を出力しないときは、“*”を出力します。
(11)	最適化実行の有無 最適化実行の有無を出力します。 ch : 最適化によって変更されたシンボル cr : 最適化によって生成されたシンボル mv : 最適化によって移動されたシンボル

3.2.7 クロス・リファレンス情報

-show=xreference オプションを指定した場合、シンボルの参照情報（クロス・リファレンス情報）を出力します。

クロス・リファレンス情報の出力例を以下に示します。

```

*** Cross Reference List ***

(1)  (2)      (3)      (4)      (5)
No   Unit Name Global.Symbol Location External Information
0001 sample1
      SECTION=.text
            _main
                        00000000
            _func_01
                        00000018
      SECTION=.data
            _gval3
                        fe600000 0003(00000032:.text)
                        0003(00000038:.text)
      SECTION=.bss
            _gval1
                        fe600004 0001(0000001a:.text)
                        0001(00000020:.text)
            _gval2
                        fe600008 0002(00000026:.text)
                        0002(0000002c:.text)
0002 sample2
      SECTION=.text
            _func02
                        00000024 0001(0000000a:.text)
0003 sample3
      SECTION=.text
            _func03
                        00000030 0001(00000010:.text)

```

項番	説明
(1)	Unit 番号 オブジェクト単位の識別番号を出力します。
(2)	オブジェクト名 オブジェクト名をリンク時の入力指定順で出力します。
(3)	シンボル名 シンボル名をセクションごとに配置アドレスの昇順で出力します。
(4)	シンボルの配置アドレス シンボルの配置アドレスを出力します。 -form=relocate オプションを指定した場合は、セクションの先頭からの相対値となります。

項番	説明
(5)	参照している外部シンボルのアドレス 参照している外部シンボルのアドレスを以下の形式で出力します。 Unit 番号 (アドレス, または セクション内オフセット: セクション名)

3.3 リンク・マップ・ファイル（オブジェクト結合時）

ここでは、マルチコア用プロジェクトにおいて、オブジェクト結合機能（「[2.8.3 複数のオブジェクトを結合する](#)」参照）を使用する場合に出力するリンク・マップ・ファイルについて説明します。

リンク・マップ・ファイルを出力することにより、オブジェクト結合の詳細情報を知ることができます。

リンク・マップ・ファイルの出力方法を以下に示します。

(1) CubeSuite+ の場合

ブート・ローダ・プロジェクトにおいて設定を行います。

プロジェクト・ツリーパネルでビルド・ツール・ノードを選択したのち、プロパティパネルの[リンク・オプション]タブを選択します。

リンク・マップ・ファイルを出力するには、[リスト]カテゴリの[リンク・マップ・ファイルを出力する]プロパティで[はい]を選択します。

出力先は、[ヘキサ出力オプション]タブの[出力ファイル]カテゴリの[結合ヘキサ・ファイルの出力フォルダ]プロパティで指定したフォルダとなります。

ファイル名は、[出力ファイル名]プロパティで指定したファイル名の拡張子を“.map”で置き換えた名前となります。

(2) コマンド・ラインの場合

入力ファイルとして、1つ以上のインテル拡張ヘキサ・ファイル、またはモトローラ・Sタイプ・ファイルを指定します。

注意 入力ファイルの種類は同一である必要があります（インテル拡張ヘキサ・ファイルとモトローラ・Sタイプ・ファイルを結合することはできません）。

-list オプションを指定すると、カレント・フォルダに、リンク・マップ・ファイルを出力します。

同時に、入力ファイルにインテル拡張ヘキサ・ファイルを指定した場合は -form=hexadecimal オプション、モトローラ・Sタイプ・ファイルを指定した場合は -form=stype オプションを指定します。

出力ファイル名は“先頭入力ファイル名_combine.map”となります。

3.3.1 リンク・マップの構成

リンク・マップの構成と内容を以下に示します。

出力情報	説明
ヘッダ情報	最適化リンクのバージョン情報、およびリンク時刻
オプション情報	コマンド・ライン、サブコマンド・ファイルで指定したオプション列
エラー情報	エラー・メッセージ
エントリ情報	実行開始アドレス
結合アドレス情報	結合元ファイル、および連続範囲データの開始／終了アドレス、サイズ
アドレス重複情報	重複した結合元ファイル、および重複範囲データの開始／終了アドレス、サイズ

3.3.2 ヘッダ情報

バージョン情報、およびリンク時刻を出力します。

ヘッダ情報の出力例を以下に示します。

Renesas Optimizing Linker (VX.XX.XX) XX-Xxx-XXXX XX:XX:XX (1)	
項番	説明
(1)	最適化リンカのバージョン情報、およびリンク時刻 最適化リンカのバージョン情報、およびリンク時刻を出力します。

3.3.3 オプション情報

コマンド・ライン、サブコマンド・ファイルで指定したオプション列を出力します。

コマンド・ライン、サブコマンド・ファイルで次のように指定した場合のオプション情報の出力例を以下に示します。

<コマンド・ライン>

```
>rlink -subcommand=test.sub -list
```

<サブコマンド・ファイル test.sub >

```
input sample1.mot
input sample2.mot
form stype
output result
```

*** Options ***	
-subcommand=test.sub	(1)
input sample1.mot	(2)
input sample2.mot	(2)
form stype	(2)
output result	(2)
-list	(1)

項番	説明
(1)	コマンド・ラインで指定したオプション コマンド・ラインで指定したオプションを出力します（指定順）。
(2)	サブコマンド・ファイル内で指定したオプション サブコマンド・ファイル test.sub 内で指定したオプションを出力します。

3.3.4 エラー情報

エラー・メッセージを出力します。

エラー情報の出力例を以下に示します。

```
*** Error information ***
E0562420:"sample1.mot" overlap address "sample2.mot" : "00000100" (1)
```

項番	説明
(1)	エラー・メッセージ エラー・メッセージを出力します。

3.3.5 エントリ情報

実行開始アドレスを出力します。

エントリ情報の出力例を以下に示します。

```
*** Entry address ***
00000100 (1)
```

項番	説明
(1)	実行開始アドレス 実行開始アドレスを出力します。 ただし、実行開始アドレスが“00000000”の場合は出力しません。

3.3.6 結合アドレス情報

結合元ファイル、および連続範囲データの開始／終了アドレス、サイズを出力します。

結合アドレス情報の出力例を以下に示します。

```
*** Combine information ***
(1)          (2)          (3)          (4)
FILE          START      END          SIZE
sample1.mot
                00000100    00000127    28
sample1.mot
                00000200    00000227    28
sample2.mot
                00000250    00000263    14
sample2.mot
                00000300    0000033b    3c
```

項番	説明
(1)	結合元ファイル名 結合元ファイル名を出力します。

項番	説明
(2)	連続範囲データの開始アドレス 連続範囲データの開始アドレスを出力します。 16 進数で表記します。
(3)	連続範囲データの終了アドレス 連続範囲データの終了アドレスを出力します。 16 進数で表記します。
(4)	連続範囲データのサイズ 連続範囲データのサイズを出力します（単位：バイト）。 16 進数で表記します。

3.3.7 アドレス重複情報

重複した結合元ファイル、および重複範囲データの開始／終了アドレス、サイズを出力します。

アドレス重複情報の出力例を以下に示します。

*** Conflict information ***			
(1)	(2)	(3)	(4)
FILE	START	END	SIZE
Conflict 1			
	00000200	00000213	14
sample1.mot			
sample2.mot			

項番	説明
(1)	重複した結合元ファイル名 重複した結合元ファイル名を出力します。
(2)	重複範囲データの開始アドレス 重複範囲データの開始アドレスを出力します。 16 進数で表記します。
(3)	重複範囲データの終了アドレス 重複範囲データの終了アドレスを出力します。 16 進数で表記します。
(4)	重複範囲データのサイズ 重複範囲データのサイズを出力します（単位：バイト）。 16 進数で表記します。

3.4 ライブラリ・リスト・ファイル

ここでは、ライブラリ・リスト・ファイルについて説明します。

ライブラリ・リストとは、ライブラリ作成結果の情報が書かれたものです。

ライブラリ・リスト・ファイルの出力方法を以下に示します。

(1) CubeSuite+ の場合

プロジェクト・ツリー パネルでビルド・ツール・ノードを選択したのち、プロパティ パネルの [ライブラリ生成オプション] タブを選択します。

ライブラリ・リスト・ファイルを出力するには、[リスト] カテゴリの [リンク・マップ・ファイルを出力する] プロパティで [はい] を選択します。

出力先は、[出力ファイル] カテゴリの [出力フォルダ] プロパティで指定します。

また、プロジェクト・ツリーのビルド・ツール生成ファイル・ノードにも表示します。

~~ファイル名は、出力ファイル名の拡張子を“.lbp”で置き換えた名前となります。~~
ファイル名は、プロジェクト・ファイル名の拡張子を“.lbp”で置き換えた名前となります。

(2) コマンド・ラインの場合

-form=library, または -extract オプション指定時に、-list オプションを指定すると、カレント・フォルダに、出力ファイル名の拡張子を“.lbp”で置き換えた名前でライブラリ・リスト・ファイルを出力します。

また、-show オプションを同時に指定することにより、ライブラリ・リスト・ファイルへの出力情報を指定することができます。

詳細については、「[3.4.1 ライブラリ・リストの構成](#)」を参照してください。

3.4.1 ライブラリ・リストの構成

ライブラリ・リストの構成と内容を以下に示します。

出力情報	説明	-show オプション 指定	-show オプション 省略時
オプション情報	コマンド・ライン、サブコマンド・ファイル で指定したオプション列	—	出力する
エラー情報	エラー・メッセージ	—	出力する
ライブラリ情報	ライブラリ情報	—	出力する
ライブラリ内モジュール、 セクション、シンボル情報	ライブラリ内モジュール	—	出力する
	モジュール内シンボル名	-show=symbol	出力しない
	各モジュール内セクション名、シンボル名	-show=section	出力しない

注意 -show オプションは、-list オプションを指定した場合に有効となります。

-show オプションについての詳細は、「[-SHOW](#)」を参照してください。

3.4.2 オプション情報

コマンド・ライン、サブコマンド・ファイルで指定したオプション列を出力します。

コマンド・ライン、サブコマンド・ファイルで次のように指定した場合のオプション情報の出力例を以下に示します。

<コマンド・ライン>

```
>rlink -subcommand=test.sub -list -show
```

<サブコマンド・ファイル test.sub >

```
form library
input extmod1
input extmod2
output usrlib.lib
```

```
*** Options ***

-subcommand=test.sub      (1)
form library              (2)
input extmod1             (2)
input extmod2             (2)
output usrlib.lib         (2)
-list                    (1)
-show                    (1)
```

項番	説明
(1)	コマンド・ラインで指定したオプション コマンド・ラインで指定したオプションを出力します（指定順）。
(2)	サブコマンド・ファイル内で指定したオプション サブコマンド・ファイル test.sub 内で指定したオプションを出力します。

3.4.3 エラー情報

エラー，ワーニングなどのメッセージを出力します。

エラー情報の出力例を以下に示します。

```
*** Error Information ***

** E0561200 Backed up file "sample1.lib" into "usrlib.lbk" (1)
```

項番	説明
(1)	メッセージ メッセージを出力します。

3.4.4 ライブラリ情報

ライブラリの種別を出力します。

ライブラリ情報の出力例を以下に示します。

```
*** Library Information ***

LIBRARY NAME=usrlib.lib (1)
CPU=RH850 (2)
ENDIAN=Little (3)
ATTRIBUTE=user (4)
NUMBER OF MODULE=2 (5)
```

項番	説明
(1)	ライブラリ名 ライブラリ名を出力します。
(2)	マイコン名 マイコン名を出力します。
(3)	エンディアン種別 エンディアン種別を出力します。
(4)	ライブラリ・ファイルの属性 システム・ライブラリであるか，ユーザ・ライブラリであるかを出力します。
(5)	ライブラリ内モジュール数 ライブラリ内モジュール数を出力します。

3.4.5 ライブラリ内モジュール、セクション、シンボル情報

ライブラリ内のモジュールを出力します。

-show=symbol オプションを指定した場合は、モジュール内シンボル名を出力します。

また、-show=section オプションを指定した場合は、モジュール内セクション名も出力します。

ライブラリ内モジュール、セクション、シンボル情報の出力例を以下に示します。

```

*** Library List ***

(1)          (2)
MODULE      LAST UPDATE

(3)
SECTION

(4)
SYMBOL
extmod1
                12-Dec-2011 16:30:00

.text
    _func_01
    _func_02
extmod2
                12-Dec-2011 16:30:10

.text
    _func_03
    _func_04

```

項番	説明
(1)	モジュール名 モジュール名を出力します。
(2)	モジュールを登録した日付 モジュールを登録した日付を出力します。 モジュールが更新された場合は、最新の更新日付を出力します。
(3)	モジュール内セクション名 モジュール内セクション名を出力します。
(4)	セクション内シンボル名 セクション内シンボル名を出力します。

3.5 インテル拡張ヘキサ・ファイル

ここでは、インテル拡張ヘキサ・ファイルについて説明します。

インテル拡張ヘキサ・ファイルの出力方法を以下に示します。

(1) CubeSuite+ の場合

プロジェクト・ツリーパネルでビルド・ツール・ノードを選択したのち、プロパティパネルの[ヘキサ出力オプション]タブを選択します。

インテル拡張ヘキサ・ファイルを出力するには、[ヘキサ・フォーマット]カテゴリの[ヘキサ・ファイル・フォーマット]プロパティで[インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)]を選択します。

出力先は、[出力ファイル]カテゴリの[出力フォルダ]プロパティ、および[出力ファイル名]プロパティで指定します。

また、プロジェクト・ツリーのビルド・ツール生成ファイル・ノードにも表示します。

(2) コマンド・ラインの場合

-form=hexadecimal オプションを指定すると、カレント・フォルダに、出力ファイル名の拡張子を“.hex”で置き換えた名前インテル拡張ヘキサ・ファイルを出力します。

3.5.1 インテル拡張ヘキサ・ファイルの構成

インテル拡張ヘキサ・ファイル（20 ビット）は、スタート・アドレス・レコード、拡張アドレス・レコード、データ・レコード、およびエンド・レコードの4種類のレコード^注により構成されます。

インテル拡張ヘキサ・ファイル（32 ビット）は、スタート・リニア・アドレス・レコード、拡張リニア・スタート・アドレス・レコード、スタート・アドレス・レコード、拡張アドレス・レコード、データ・レコード、およびエンド・レコードの6種類のレコード^注により構成されます。

注 各レコードは、ASCII コードで出力します。

インテル拡張ヘキサ・ファイルの構成と内容を以下に示します。

項番	説明
(5)	コード コードの1バイトごとを2桁の16進数で表したものです。
(6)	チェック・サム :, SS, NLを除くレコード内の各バイト値を16進数で加算した結果の2の補数です(2桁)。
(7)	ニュー・ライン (¥n)

備考 インテル・ヘキサ・フォーマットのロケーション・アドレスは2バイト(16ビット)です。

したがって、64Kの空間しか直接指定はできません。

それを拡張するために、16ビットの拡張アドレスを追加して1M(20ビット)の空間まで扱えるようにしたのがインテル拡張ヘキサ・フォーマットです。

具体的には、16ビットの拡張アドレスを指定するレコード・タイプを追加しています。

この追加した拡張アドレスの4ビットをシフトしてロケーション・アドレスと加算することで、20ビットのアドレスを表現できるようになっています。

3.5.2 スタート・リニア・アドレス・レコード

リニア・アドレスを示します。

:	04	0000	05	XXXXXXXX	SS	NL
(1)	(2)	(3)	(4)	(5)	(6)	(7)

項番	説明
(1)	レコード・マーク
(2)	04 固定
(3)	0000 固定
(4)	レコード・タイプ (05 固定)
(5)	リニア・アドレス値
(6)	チェック・サム
(7)	ニュー・ライン

3.5.3 拡張リニア・アドレス・レコード

ビット32～ビット16の上位16ビット・アドレスを示します。

:	02	0000	04	XXXX	SS	NL
(1)	(2)	(3)	(4)	(5)	(6)	(7)

項番	説明
(1)	レコード・マーク
(2)	02 固定
(3)	0000 固定
(4)	レコード・タイプ (04 固定)

項番	説明
(5)	ビット 32 ～ビット 16 の上位 16 ビット・アドレス値
(6)	チェック・サム
(7)	ニュー・ライン

注 下位 16 ビットは、データ・レコードのロケーション・アドレスを用います。

3.5.4 スタート・アドレス・レコード

エントリ・ポイント・アドレスを示します。

:	04	0000	03	PPPP	XXXX	SS	NL
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

項番	説明
(1)	レコード・マーク
(2)	04 固定
(3)	0000 固定
(4)	レコード・タイプ (03 固定)
(5)	エントリ・ポイント・アドレスのパラグラフ値 ^注
(6)	エントリ・ポイント・アドレスのオフセット値
(7)	チェック・サム
(8)	ニュー・ライン

注 アドレスは (パラグラフ値 <<4) + オフセット値で求められます。

3.5.5 拡張アドレス・レコード

ロード・アドレスのパラグラフ値を示します^注。

注 (データ・レコードを出力する際) セグメントの先頭で、またはデータ・レコードのロード・アドレスのオフセット値が最大値 0xffff を越えてセグメントが新しくなる際、出力します。

:	02	0000	02	PPPP	SS	NL
(1)	(2)	(3)	(4)	(5)	(6)	(7)

項番	説明
(1)	レコード・マーク
(2)	02 固定
(3)	0000 固定
(4)	レコード・タイプ (02 固定)

項番	説明
(5)	セグメントのパラグラフ値
(6)	チェック・サム
(7)	ニュー・ライン

3.5.6 データ・レコード

コードの値を示します。

:	XX	XXXX	00	DD.....DD	SS	NL
(1)	(2)	(3)	(4)	(5)	(6)	(7)

項番	説明
(1)	レコード・マーク
(2)	バイト数 ^注
(3)	ロケーション・アドレス
(4)	レコード・タイプ (00 固定)
(5)	コード コードの1バイトごとを2桁の16進数で表したものです。
(6)	チェック・サム
(7)	ニュー・ライン

注 0x1 ~ 0xff の範囲に限られます (1つのデータ・レコードで示されるコードのバイト数の最小値は1で最大値は255です)。

例

:	04	0100	00	3C58E01B	6C	NL
(1)	(2)	(3)	(4)	(5)	(6)	(7)

項番	説明
(1)	レコード・マーク
(2)	3C58E01B の2桁ずつの16進数で表されるバイトのバイト数
(3)	ロケーション・アドレス
(4)	レコード・タイプ 00
(5)	コードの1バイトごとを2桁の16進数で表したもの
(6)	チェック・サム $04 + 01 + 00 + 00 + 3C + 58 + E0 + 1B = 194$ の2の補数 E6C の下位1バイトを2桁の16進数で表したもの
(7)	ニュー・ライン (¥n)

3.5.7 エンド・レコード

コードの終わりを示します。

:	00	0000	01	FF	NL
(1)	(2)	(3)	(4)	(5)	(6)

項番	説明
(1)	レコード・マーク
(2)	00 固定
(3)	0000 固定
(4)	レコード・タイプ (01 固定)
(5)	FF 固定
(6)	ニュー・ライン

3.6 モトローラ・Sタイプ・ファイル

ここでは、モトローラ・Sタイプ・ファイルについて説明します。

モトローラ・Sタイプ・ファイルの出力方法を以下に示します。

(1) CubeSuite+ の場合

プロジェクト・ツリーパネルでビルド・ツール・ノードを選択したのち、プロパティパネルの[ヘキサ出力オプション]タブを選択します。

モトローラ・Sタイプ・ファイルを出力するには、[ヘキサ・フォーマット]カテゴリの[ヘキサ・ファイル・フォーマット]プロパティで[モトローラ・Sタイプ・ファイル (-FOrm=Stype)]を選択します。

出力先は、[出力ファイル]カテゴリの[出力フォルダ]プロパティ、および[出力ファイル名]プロパティで指定します。

また、プロジェクト・ツリーのビルド・ツール生成ファイル・ノードにも表示します。

(2) コマンド・ラインの場合

-form=stype オプションを指定すると、カレント・フォルダに、出力ファイル名の拡張子を“.mot”で置き換えた名前でもトローラ・Sタイプ・ファイルを出力します。

3.6.1 モトローラ・Sタイプ・ファイルの構成

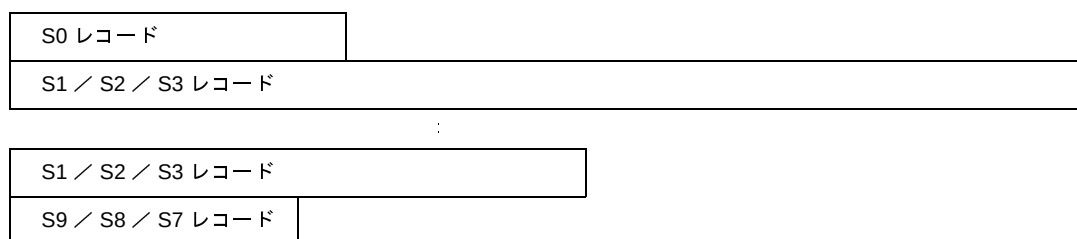
モトローラ・Sタイプ・ファイルは、ヘッダ・レコードであるS0レコード、データ・レコードであるS1／S2／S3レコード、エンド・レコードであるS9／S8／S7レコードの7種類のレコード^{注1}により構成されます^{注2}。

注1. 各レコードは、ASCIIコードで出力します。

2. モトローラ・Sタイプ・ファイルには、16ビット・アドレスのもの、(24ビット)スタンダード・アドレスのもの、32ビット・アドレスのものが存在し、16ビット・アドレスのフォーマットはS0, S1, およびS9レコード、スタンダード・アドレスのフォーマットはS0, S2, およびS8レコード、32ビット・アドレスのフォーマットはS0, S3, およびS7レコードによって構成されます。

モトローラ・Sタイプ・ファイルの構成と内容を以下に示します。

図 3—2 モトローラ・Sタイプ・ファイルの構成



出力情報	説明
S0 レコード	ファイル名
S1 レコード	コードの値
S2 レコード	コードの値
S3 レコード	コードの値
S7 レコード	エントリ・ポイント・アドレス
S8 レコード	エントリ・ポイント・アドレス
S9 レコード	エントリ・ポイント・アドレス

各レコードは、各種フィールドにより以下の形で構成されます。

Sx	XX	YY.....YY	SS	NL
(1)	(2)	(3)	(4)	(5)

項番	説明
(1)	レコード・タイプ S0 : S0 レコード S1 : S1 レコード S2 : S2 レコード S3 : S3 レコード S4 : S4 レコード S5 : S5 レコード S6 : S6 レコード S7 : S7 レコード S8 : S8 レコード S9 : S9 レコード
(2)	レコード長 (3) の 2 桁ずつの 16 進数で表されるバイトのバイト数 + SS で表されるバイト数 ^注
(3)	フィールド
(4)	チェック・サム Sx, SS, NL を除くレコード内の 2 桁ずつの 16 進数で表されるバイトの値を合計したものの 1 の補数を取り、その下位 1 バイトを 2 桁の 16 進数で表したもの
(5)	ニュー・ライン (¥n)

注 1 です。

3.6.2 S0 レコード

ファイル名を示します。

S0	0E	0000	XX.....XX	SS	NL
(1)	(2)	(3)	(4)	(5)	(6)

項番	説明
(1)	S0 固定
(2)	0E 固定
(3)	0000 固定
(4)	主にファイル名 (8 文字) + ファイル形式 (3 文字)
(5)	チェック・サム
(6)	ニュー・ライン

3.6.3 S1 レコード

コードの値を示します。

S1	XX	YYYY	ZZ.....ZZ	SS	NL
(1)	(2)	(3)	(4)	(5)	(6)

項番	説明
(1)	S1 固定
(2)	レコード長
(3)	ロード・アドレス 16 ビット (0x0 ~ 0xFFFF)
(4)	コード コードの 1 バイトごとを 2 桁の 16 進数で表したもの
(5)	チェック・サム
(6)	ニュー・ライン

3.6.4 S2 レコード

コードの値を示します。

S2	XX	YYYYYY	ZZ.....ZZ	SS	NL
(1)	(2)	(3)	(4)	(5)	(6)

項番	説明
(1)	S2 固定
(2)	レコード長
(3)	ロード・アドレス 24 ビット (0x0 ~ 0xFFFFFF)

項番	説明
(4)	コード コードの1バイトごとを2桁の16進数で表したもの
(5)	チェック・サム
(6)	ニュー・ライン

3.6.5 S3 レコード

コードの値を示します。

S3	XX	YYYYYYYY	ZZ.....ZZ	SS	NL
(1)	(2)	(3)	(4)	(5)	(6)

項番	説明
(1)	S3 固定
(2)	レコード長
(3)	ロード・アドレス 32 ビット (0x0 ~ 0xFFFFFFFF)
(4)	コード コードの1バイトごとを2桁の16進数で表したもの
(5)	チェック・サム
(6)	ニュー・ライン

3.6.6 S7 レコード

エントリ・ポイント・アドレスを示します。

S7	XX	YYYYYYYY	SS	NL
(1)	(2)	(3)	(4)	(5)

項番	説明
(1)	S7 固定
(2)	レコード長
(3)	エントリ・ポイント・アドレス 32 ビット (0x0 ~ 0xFFFFFFFF)
(4)	チェック・サム
(5)	ニュー・ライン

3.6.7 S8 レコード

エントリ・ポイント・アドレスを示します。

S8	XX	YYYYYY	SS	NL
(1)	(2)	(3)	(4)	(5)

項番	説明
(1)	S8 固定
(2)	レコード長
(3)	エントリ・ポイント・アドレス 24 ビット (0x0 ~ 0xFFFFFF)
(4)	チェック・サム
(5)	ニュー・ライン

3.6.8 S9 レコード

エントリ・ポイント・アドレスを示します。

S9	XX	YYYY	SS	NL
(1)	(2)	(3)	(4)	(5)

項番	説明
(1)	S9 固定
(2)	レコード長
(3)	エントリ・ポイント・アドレス 16 ビット (0x0 ~ 0xFFFF)
(4)	チェック・サム
(5)	ニュー・ライン

付録 A ウィンドウ・リファレンス

ここでは、ビルドに関するウィンドウ／パネル／ダイアログについての詳細を説明します。

A.1 説 明

以下に、ビルドに関するウィンドウ／パネル／ダイアログの一覧を示します。

表 A—1 ウィンドウ／パネル／ダイアログ一覧

ウィンドウ／パネル／ダイアログ名	機能概要
メイン・ウィンドウ	CubeSuite+ を起動した際、最初にオープンするウィンドウ
プロジェクト・ツリー パネル	プロジェクトの構成要素をツリー表示
プロパティ パネル	プロジェクト・ツリー パネルで選択しているビルド・ツール・ノード、ファイル、カテゴリ・ノードについて、詳細情報を表示、および設定を変更
エディタ パネル	テキスト・ファイル／ソース・ファイルを表示／編集
出力 パネル	ビルド・ツールから出力するメッセージを表示
ファイル追加 ダイアログ	新規ファイルを作成、およびプロジェクトに追加
フォルダとファイル追加 ダイアログ	既存のファイルとフォルダ構成をプロジェクトに追加
文字列入力 ダイアログ	1 行分の文字列を入力、編集
テキスト編集 ダイアログ	複数行のテキストを入力、編集
パス編集 ダイアログ	パス、またはパスを含むファイル名を編集、追加
システム・インクルード・パス順設定 ダイアログ	コンパイラに対して指定するシステム・インクルード・パスを参照、および指定順を設定
ルール番号の指定 ダイアログ	本ダイアログの呼び出し元に設定する MISRA-C:2004 ルール番号を選択
セクション設定 ダイアログ	セクションを追加、変更、削除
セクション追加 ダイアログ セクション編集 ダイアログ オーバーレイ配置セクションの追加 ダイアログ	セクションの追加、編集、複数割り付けを行う際のセクション名を設定
セクションのアドレス ダイアログ	セクションの追加、または変更時のアドレスを設定
削除するセクション ダイアログ	セクションを削除
ファイルの保存設定 ダイアログ	エディタ パネルで編集中のファイルのエンコードと改行コードを設定
リンク順設定 ダイアログ	リンクに入力するファイルを参照、およびリンク順を設定
ビルド・モード設定 ダイアログ	ビルド・モードの追加と削除、および現在のビルド・モードを一括設定

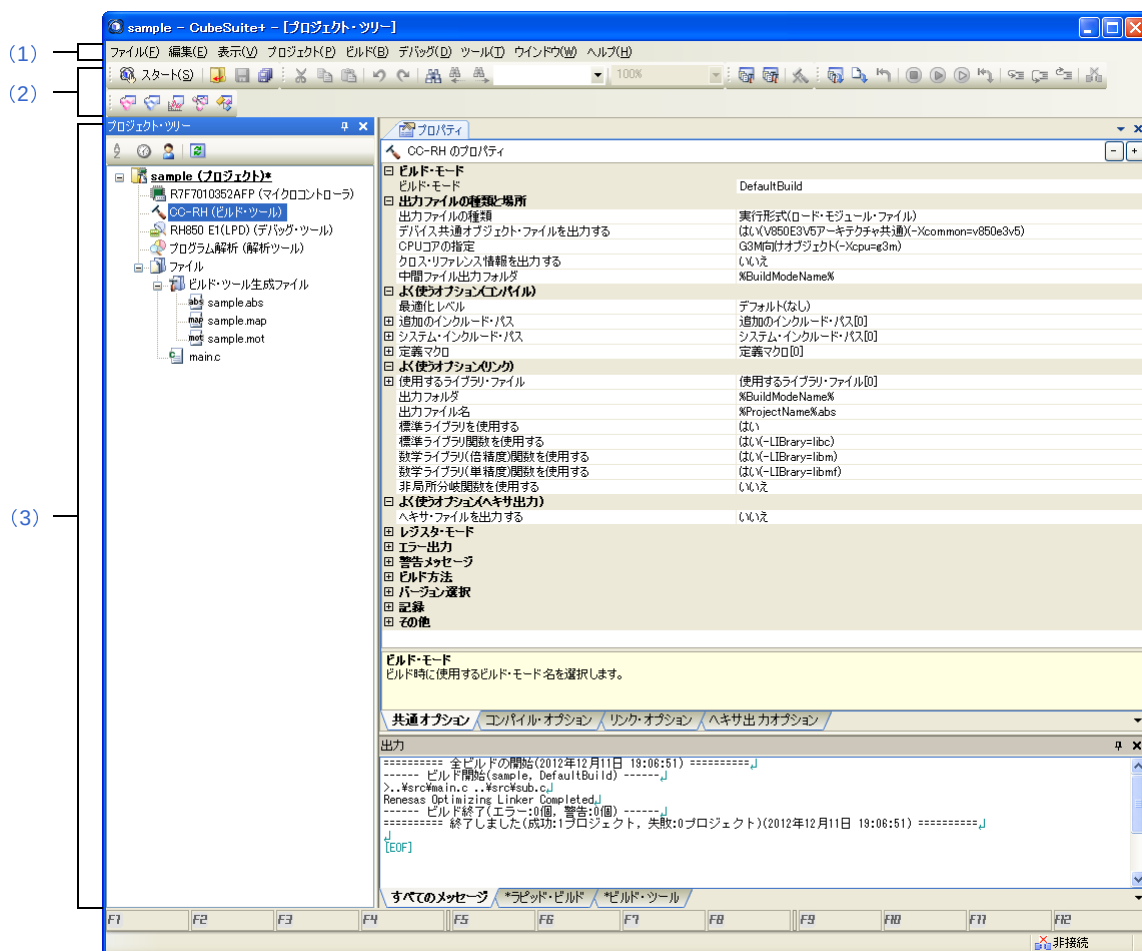
ウィンドウ／パネル／ダイアログ名	機能概要
バッチ・ビルド ダイアログ	プロジェクトが持つビルド・モードを一括して、ビルド、リビルド、クリーンを実行
処理中表示 ダイアログ	処理の進捗状況を表示
オプション ダイアログ	各種環境を設定
既存のファイルを追加 ダイアログ	プロジェクトに既存のファイルを追加
ビルド・オプションのインポート ダイアログ	ビルド・オプションのインポート対象となるプロジェクト・ファイルを選択
フォルダの参照 ダイアログ	本ダイアログの呼び出し元に設定するフォルダを選択
Far Jump ファイルを指定 ダイアログ	本ダイアログの呼び出し元に設定する Far Jump ファイルを選択
MISRA-C:2004 ルール・ファイルの指定 ダイアログ	本ダイアログの呼び出し元に設定する MISRA-C:2004 ルール・ファイルを選択
対象外ファイルの追加 ダイアログ	本ダイアログの呼び出し元に設定する MISRA-C:2004 ルールのチェック対象外のファイルを選択
使用するライブラリ・ファイルを指定 ダイアログ	本ダイアログの呼び出し元に設定するライブラリ・ファイルを選択
名前を付けて保存 ダイアログ	編集中のファイル、または各パネルの内容をファイルに保存
プログラムから開く ダイアログ	ファイルを開くアプリケーションを選択
インポートするファイルを選択 ダイアログ	本ダイアログの呼び出し元にインポートするファイルを選択
エクスポートするファイルを選択 ダイアログ	本ダイアログの呼び出し元の情報をファイルに出力

メイン・ウィンドウ

CubeSuite+ を起動した際、最初にオープンするウィンドウです。

ビルドを行う際は、本ウィンドウからユーザ・プログラムの実行制御、および各パネルのオープンを行います。

図 A—1 メイン・ウィンドウ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]

[オープン方法]

- Windows の [スタート] → [すべてのプログラム] → [Renesas Electronics CubeSuite+] → [CubeSuite+] を選択

備考 Windows 8 の場合は、スタート画面の [CubeSuite+] をダブルクリックしてください。

[各エリアの説明]

(1) メニューバー

ビルド関連のメニューを示します。

(a) [プロジェクト]

[プロジェクト] メニューでは、プロジェクト関連を操作するメニュー項目を表示します。

新しいプロジェクトを作成 ...	現在のプロジェクトを閉じて、新しいプロジェクトを作成するために、プロジェクト作成 ダイアログをオープンします。 開いているプロジェクト、またはファイルを変更し、保存していない場合は、それらを保存するかどうかの確認を行います。
プロジェクトを開く ...	現在のプロジェクトを閉じて、既存のプロジェクトを開くために、プロジェクトを開く ダイアログをオープンします。 開いているプロジェクト、またはファイルを変更し、保存していない場合は、それらを保存するかどうかの確認を行います。
お気に入りのプロジェクト	お気に入りのプロジェクトを開く、または登録するためのカスケード・メニューを表示します。
1 パス	[お気に入りのプロジェクト] → [1 お気に入りのプロジェクトに登録] で登録したプロジェクトを開きます。 プロジェクトを登録していない場合は、“お気に入りのプロジェクト”を表示します。
2 パス	[お気に入りのプロジェクト] → [2 お気に入りのプロジェクトに登録] で登録したプロジェクトを開きます。 プロジェクトを登録していない場合は、“お気に入りのプロジェクト”を表示します。
3 パス	[お気に入りのプロジェクト] → [3 お気に入りのプロジェクトに登録] で登録したプロジェクトを開きます。 プロジェクトを登録していない場合は、“お気に入りのプロジェクト”を表示します。
4 パス	[お気に入りのプロジェクト] → [4 お気に入りのプロジェクトに登録] で登録したプロジェクトを開きます。 プロジェクトを登録していない場合は、“お気に入りのプロジェクト”を表示します。
1 お気に入りのプロジェクトに登録	現在開いているプロジェクトのパスを [お気に入りのプロジェクト] → [1 パス] に登録します。
2 お気に入りのプロジェクトに登録	現在開いているプロジェクトのパスを [お気に入りのプロジェクト] → [2 パス] に登録します。
3 お気に入りのプロジェクトに登録	現在開いているプロジェクトのパスを [お気に入りのプロジェクト] → [3 パス] に登録します。
4 お気に入りのプロジェクトに登録	現在開いているプロジェクトのパスを [お気に入りのプロジェクト] → [4 パス] に登録します。

追加	プロジェクトにサブプロジェクトを追加するためのカスケード・メニューを表示します。
既存のサブプロジェクトを追加 ...	プロジェクトに既存のサブプロジェクトを追加するために、既存のサブプロジェクトを追加 ダイアログをオープンします。
新しいサブプロジェクトを追加 ...	プロジェクトに新しいサブプロジェクトを追加するために、プロジェクト作成 ダイアログをオープンします。
既存のファイルを追加 ...	既存のファイルを追加 ダイアログをオープンし、選択したファイルをプロジェクトに追加します。
新しいファイルを追加 ...	ファイル追加 ダイアログをオープンし、選択した種類でファイルを作成し、プロジェクトに追加します。 追加したファイルはファイルの拡張子に割り当てられたアプリケーションでオープンします。
新しいカテゴリを追加	ファイル・ノードの直下にカテゴリ・ノードを追加し、カテゴリ名が編集可能な状態になります。 カテゴリ名は、デフォルトで“新しいカテゴリ”となります。すでに存在するカテゴリ・ノードと同名のカテゴリ・ノードを追加することもできます。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
選択しているプロジェクト、またはサブプロジェクト名をアクティブ・プロジェクトに設定	選択しているプロジェクト、またはサブプロジェクトをアクティブ・プロジェクトに設定します。
プロジェクトを閉じる	現在開いているプロジェクトを閉じます。 開いているプロジェクト、またはファイルを変更し、保存していない場合は、それらを保存するかどうかの確認を行います。
プロジェクトを保存	現在開いているプロジェクトの設定情報をプロジェクト・ファイルに保存します。
名前を付けてプロジェクトを保存 ...	現在開いているプロジェクトの設定情報を別名のプロジェクト・ファイルに保存するために、名前を付けてプロジェクトを保存 ダイアログをオープンします。
プロジェクトから外す	選択しているサブプロジェクト、またはファイルをプロジェクトから外します。 サブプロジェクト・ファイル、およびファイル自体はファイル・システム上から削除しません。
プロジェクトと開発ツールをバックして保存 ...	本製品一式とプロジェクト一式を指定したフォルダにコピーして、1つのフォルダにまとめて保存します。 また、プロジェクトのみをバックして保存することも可能です。

(b) [ビルド]

[ビルド] メニューでは、ビルド関連を操作するメニュー項目を表示します。

ビルド・プロジェクト	プロジェクトのビルドを行います。サブプロジェクトを追加している場合は、サブプロジェクトのビルドも行います。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
リビルド・プロジェクト	プロジェクトのリビルドを行います。サブプロジェクトを追加している場合は、サブプロジェクトのリビルドも行います。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
クリーン・プロジェクト	プロジェクトのクリーンを行います。サブプロジェクトを追加している場合は、サブプロジェクトのクリーンも行います。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
ラビッド・ビルド	ラビッド・ビルド機能の有効（デフォルト）／無効を選択します（トグル）。
依存関係の更新	プロジェクトのビルド対象ファイルの依存関係を更新します。サブプロジェクトを追加している場合は、サブプロジェクトのビルド対象ファイルの依存関係も更新します。
アクティブ・プロジェクトをビルド	アクティブ・プロジェクトのビルドを行います。 アクティブ・プロジェクトがメイン・プロジェクトの場合、サブプロジェクトのビルドは行いません。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
アクティブ・プロジェクトをリビルド	アクティブ・プロジェクトのリビルドを行います。 アクティブ・プロジェクトがメイン・プロジェクトの場合、サブプロジェクトのリビルドは行いません。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
アクティブ・プロジェクトをクリーン	アクティブ・プロジェクトのクリーンを行います。 アクティブ・プロジェクトがメイン・プロジェクトの場合、サブプロジェクトのクリーンは行いません。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
アクティブ・プロジェクトの依存関係の更新	アクティブ・プロジェクトのビルド対象ファイルの依存関係を更新します。
ビルドを中止	実行中のビルド、リビルド、バッチ・ビルド、クリーンを中止します。
ビルド・モードの設定 ...	ビルド・モードの変更、追加等を行うために、 ビルド・モード設定 ダイアログ をオープンします。
バッチ・ビルド ...	バッチ・ビルドを行うために、 バッチ・ビルド ダイアログ をオープンします。
ビルド・オプション一覧	現在設定しているビルド・オプションを 出力 パネル に一覧表示します。

(2) ツールバー

ビルド関連のボタン群を示します。

(a) ビルド・ツールバー

ビルド・ツールバーでは、ビルド関連を操作するボタン群を表示します。

	プロジェクトのビルドを行います。サブプロジェクトを追加している場合は、サブプロジェクトのビルドも行います。 なお、本ボタンは、ビルド・ツールが実行中の場合は無効となります。
	プロジェクトのリビルドを行います。サブプロジェクトを追加している場合は、サブプロジェクトのリビルドも行います。 なお、本ボタンは、ビルド・ツールが実行中の場合は無効となります。
	実行中のビルド、リビルド、バッチ・ビルド、クリーンを中止します。

(3) パネル表示エリア

以下のパネルを表示するエリアです。

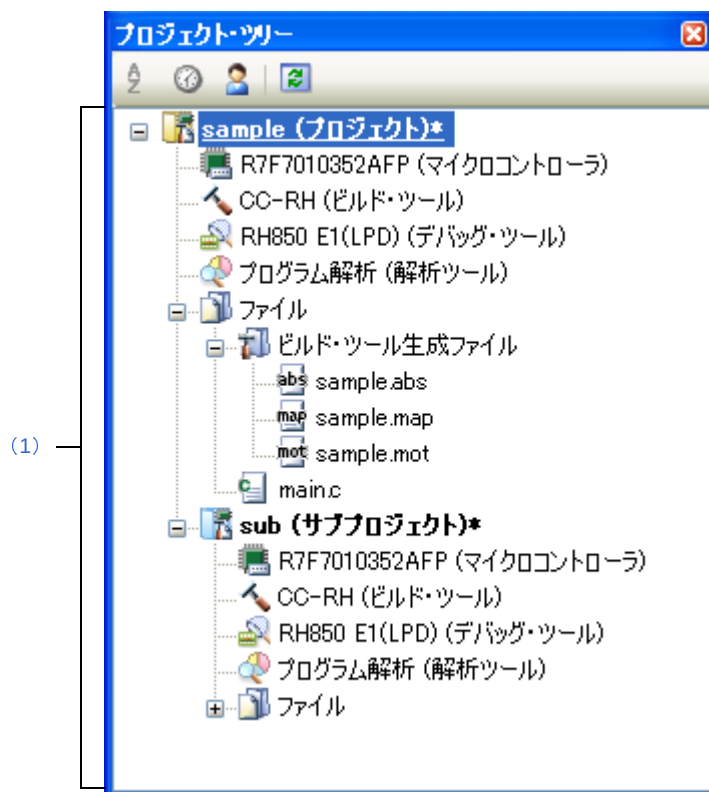
- プロジェクト・ツリー パネル
- プロパティ パネル
- エディタ パネル
- 出力 パネル

表示内容についての詳細は、各パネルの項を参照してください。

プロジェクト・ツリー パネル

プロジェクトを構成するビルド・ツール、ソース・ファイル等の構成要素をツリー表示します。

図 A—2 プロジェクト・ツリー パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [[編集] メニュー (プロジェクト・ツリー パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [プロジェクト・ツリー] を選択

[各エリアの説明]

(1) プロジェクト・ツリー エリア

プロジェクトの構成要素を以下のノードでツリー表示します。

ノード	説明
プロジェクト名 (プロジェクト) (以降, “プロジェクト・ノード” と呼びます。)	プロジェクトの名前です。
ビルド・ツール名 (ビルド・ツール) (以降, “ビルド・ツール・ノード” と呼びます。)	使用するビルド・ツール (CC-RH) です。
ファイル (以降, “ファイル・ノード” と呼びます。)	プロジェクトに追加しているファイルを直下に表示します。
ビルド・ツール生成ファイル (以降, “ビルド・ツール生成ファイル・ノード” と呼びます。)	ビルド時に生成するノードで、ビルド・ツールによって生成したファイルのうち、以下のものを直下に表示します。 - ライブラリ用のプロジェクト以外の場合 - ロード・モジュール・ファイル (*.abs) - インテル拡張ヘキサ・ファイル (*.hex) - モトローラ・S タイプ・ファイル (*.mot) - バイナリ・ファイル (*.bin) - リンク・マップ・ファイル (*.map) - ライブラリ・ファイル (*.lib) - ライブラリ・リスト・ファイル (*.lbp) - スタック情報ファイル (*.sni) - エラー・メッセージ・ファイル (*.err) - ライブラリ用のプロジェクトの場合 - ライブラリ・ファイル (*.lib) - ライブラリ・リスト・ファイル (*.lbp) - エラー・メッセージ・ファイル (*.err) 本ノードに表示しているファイルは、名前の変更、削除、移動を行うことができません。 なお、本ノードは常にファイル・ノード以下に生成します。 ビルド後にプロジェクトの再読み込みを行った場合、本ノードは表示されなくなります。
カテゴリ名 (以降, “カテゴリ・ノード” と呼びます。)	ファイルを分類するためにユーザが作成するカテゴリです (「2.3.4 ファイルをカテゴリに分類する」参照)。 なお、本ノードは常にファイル・ノード以下に作成します。
サブプロジェクト名 (サブプロジェクト) (以降, “サブプロジェクト・ノード” と呼びます。)	プロジェクトに追加しているサブプロジェクトです。

各構成要素（ノード、またはファイル）を選択すると、その詳細情報（プロパティ）を **プロパティ パネル** に表示し、設定の変更を行うことができます。

備考 複数の構成要素を選択している場合は、その構成要素に共通するタブのみ表示します。

なお、複数のファイルを選択し、共通するプロパティの値が異なる場合、その値は空欄となります。

本エリアは、次の機能を備えています。

(a) ファイルの追加

以下のいずれかの方法により、ファイルの追加を行うことができます。

ファイルの追加先はファイル・ノード以下となります。

- 既存のファイルを追加する場合

- プロジェクト・ノード、サブプロジェクト・ノード、ファイル・ノード、ファイルのいずれかを選択し、[ファイル] メニュー→[追加] → [既存のファイルを追加 ...] を選択する。[既存のファイルを追加 ダイアログ](#)がオープンし、追加するファイルを選択する。
- プロジェクト・ノード、サブプロジェクト・ノード、ファイル・ノード、ファイルのいずれかのコンテキスト・メニューの[追加] → [既存のファイルを追加 ...] を選択する。[既存のファイルを追加 ダイアログ](#)がオープンし、追加するファイルを選択する。
- エクスプローラなどでファイルをコピーし、本エリアにフォーカスを移動したのち、[編集] メニュー→[貼り付け] を選択する。
- エクスプローラなどからファイルをドラッグし、本エリア上のファイルを追加したい位置にドロップする。

備考 エクスプローラなどからファイルをドラッグし、プロジェクト・ツリー下部の空白部分にドロップした場合は、メイン・プロジェクト上にドロップしたものとみなします。

- 新しいファイルを追加する場合

- プロジェクト・ノード、サブプロジェクト・ノード、ファイル・ノードのいずれかを選択し、[ファイル] メニュー→[追加] → [新しいファイルを追加 ...] を選択する。[ファイル追加 ダイアログ](#)がオープンし、新しく作成するファイルを指定する。
- プロジェクト・ノード、サブプロジェクト・ノード、ファイル・ノードのいずれかのコンテキスト・メニューの[追加] → [新しいファイルを追加 ...] を選択する。[ファイル追加 ダイアログ](#)がオープンし、新しく作成するファイルを指定する。

備考 [ファイル追加 ダイアログ](#)で指定した場所に、空のファイルを作成します。

(b) プロジェクトからファイルを外す

以下のいずれかの方法により、プロジェクトからファイルを外すことができます。

ファイル自体はファイル・システム上から削除しません。

- プロジェクトから外すファイルを選択し、[プロジェクト] メニュー→[プロジェクトから外す] を選択する。
- プロジェクトから外すファイルを選択し、コンテキスト・メニューの[プロジェクトから外す] を選択する。

(c) ファイルの移動

以下の方法により、ファイルの移動を行うことができます。

ファイルの移動先はファイル・ノード以下となります。

- 移動するファイルをドラッグし、移動先でドロップする。

備考 1. メイン・プロジェクト内、またはサブプロジェクト内でドロップした場合、ファイルに設定した個別オプションは保持します。

- 異なるプロジェクト間、または同一プロジェクトのメイン・プロジェクトかサブプロジェクトにドロップした場合、ファイルは移動ではなく、コピーします。

なお、ファイルに設定した個別オプションは保持しません。

(d) カテゴリの追加

以下のいずれかの方法により、カテゴリ・ノードの追加を行うことができます。

カテゴリ・ノードの追加先はファイル・ノード以下となります。

- [プロジェクト] メニュー→[新しいカテゴリを追加]を選択する。
- プロジェクト・ノード、サブプロジェクト・ノード、またはファイル・ノードのコンテキスト・メニューの[新しいカテゴリを追加]を選択する。

備考 1. カテゴリ名は、デフォルトで“新しいカテゴリ”となります。

- すでに存在するカテゴリ・ノードと同名のカテゴリ・ノードを追加することもできます。

(e) カテゴリの移動

以下の方法により、カテゴリ・ノードの移動を行うことができます。

カテゴリ・ノードの移動先はファイル・ノード以下となります。

- 移動するカテゴリ・ノードをドラッグし、移動先でドロップする。

備考 1. メイン・プロジェクト内、またはサブプロジェクト内でドロップした場合、カテゴリ・ノードに含まれるファイルに設定した個別オプションは保持します。

- 異なるプロジェクト間、または同一プロジェクトのメイン・プロジェクトかサブプロジェクトにドロップした場合、カテゴリ・ノードは移動ではなく、コピーします。

なお、カテゴリ・ノードに含まれるファイルに設定した個別オプションは保持しません。

(f) フォルダの追加

以下の方法により、エクスプローラなどからフォルダの追加を行うことができます。

フォルダの追加先はファイル・ノード以下となります。

なお、フォルダはカテゴリとして追加します。

- エクスプローラなどからフォルダをドラッグし、移動先でドロップする。[フォルダとファイル追加ダイアログ](#)がオープンし、フォルダに含まれているファイルのうち追加するファイルの種類と、フォルダの階層を指定する。

注意 フォルダとファイルを同時にドラッグして、本エリアにドロップすることはできません。

(g) サブプロジェクトのビルド順の表示編集

サブプロジェクトは、ビルド順に上から表示します。

そのため、サブプロジェクトの表示位置を変更することで、ビルド順も変更することができます。

なお、プロジェクトのビルドは、すべてのサブプロジェクト、メイン・プロジェクトの順で行います。

(h) 標準ビルド・オプションの設定

プロパティ パネルにおいて、標準ビルド・オプションの設定に変更を加えると、プロパティの値を太字表示します。

以下の方法により、現在設定しているビルド・オプションを標準ビルド・オプションとする（太字表示を解除する）ことができます。

- ビルド・ツール・ノードを選択し、コンテキスト・メニューの「現在のビルド・オプションをプロジェクトの標準に設定する」を選択する。

備考 標準ビルド・オプションの設定は、プロジェクト全体（メイン・プロジェクト、およびサブプロジェクト）に対して行います。

(i) ファイル、およびカテゴリのソート

以下の方法により、ファイル、およびカテゴリ・ノードをファイル名順、タイムスタンプ順、ユーザ指定順でソートすることができます。

- ツールバーのいずれかのボタンを選択する。

以下に、各ボタンの説明を示します。

なお、デフォルトでは  を選択します。

ボタン	説明
	カテゴリ・ノード、およびファイルを名前順でソートします。  : 昇順  : 降順  : 昇順
	カテゴリ・ノード、およびファイルをタイムスタンプ順でソートします。  : 降順  : 昇順  : 降順
	カテゴリ・ノードとファイル（依存関係ファイルは除く）をユーザが指定した順で表示します（デフォルト）。 カテゴリ・ノード、およびファイルをドラッグ・アンド・ドロップすることにより、表示順を任意に変更することができます。

(j) 依存関係ファイルの表示

プロジェクトに追加しているソース・ファイルに依存関係ファイルが存在する場合、その依存関係ファイルをソース・ファイルにぶら下げて表示します。

依存関係ファイルが存在しないソース・ファイル	 main.c
依存関係ファイルが存在するソース・ファイル	 main.c  io.h

なお、依存関係ファイルの表示は、以下のタイミングで更新します。

- プロジェクトを読み込んだのち、初めてビルドを実行したとき
- ツールバーの  をクリックしたとき
- [ビルド] メニュー→[依存関係の更新] を選択したとき
- [ビルド] メニュー→[アクティブ・プロジェクトの依存関係の更新] を選択したとき

備考 1. 本機能は、[オプション ダイアログ](#)の[全般-ビルド/デバッグ] カテゴリの[プロジェクト・ツリーに依存関係ファイルを表示する] がチェック状態の場合のみ有効となります。

2. プロジェクト・ツリーに表示している依存関係ファイルの情報は、プロジェクト・ファイルには保存しません。

(k) 編集集中ファイルの表示

プロジェクトに追加しているファイル（依存関係ファイルは除く）を[エディタ パネル](#)で編集し、未保存の場合、ファイル名の後ろに“*”を表示します。

ファイルを保存すると、“*”を消去します。

通常のファイル	 main.c
編集後、未保存のファイル	 main.c*

(l) 個別ビルド・オプションを設定しているソース・ファイルの強調表示

プロジェクトの全体オプションとは異なるビルド・オプション（個別コンパイル・オプション、個別アセンブル・オプション）を設定しているソース・ファイルのアイコンは、通常とは異なるアイコンに変更します。

通常のファイル	 main.c
個別ビルド・オプションを設定しているファイル	 main.c

(m) 読み取り専用属性ファイルの強調表示

プロジェクトに追加している読み取り専用属性ファイルは、イタリック表示します。

通常のファイル	 main.c
読み取り専用属性ファイル	 <i>main.c</i>

(n) 存在しないファイルの強調表示

プロジェクトに追加しているファイルで、存在しないファイルは、グレー表示し、アイコンは淡色表示となります。

通常のファイル	 main.c
存在しないファイル	 main.c

(o) ビルド対象ファイルの強調表示

- ビルド（ラピッド・ビルド）、リビルド、コンパイル、アセンブル時にエラーを検出したファイルは、以下の例のように強調表示します。

エラーもワーニングも検出なかったファイル	 main.c
エラーを検出したファイル	 main.c
ワーニングを検出したファイル	 main.c

備考 1. エラーとワーニングの両方を検出したファイルは、赤色表示します。

2. 強調表示は、ビルド・オプション（全体オプション、または個別オプション）の変更、およびビルド・モードの変更により解除します。

- 以下のファイルは、太字表示します。

- 編集後、コンパイルしていないソース・ファイル
- クリーンを実行した場合のソース・ファイル
- ビルド・ツールのオプションを変更した場合のソース・ファイル
- ビルド・モードを変更した場合のソース・ファイル

備考 プロジェクトを開いた直後は、すべて太字表示となり、ビルドを行うことで太字表示を解除します。

(p) ビルド対象外ファイルの強調表示

ビルドの対象外に設定しているファイルは、以下の例のように強調表示します。

通常のファイル	 main.c
ビルド対象外ファイル	 main.c

(q) オーバーレイ・アイコンの強調表示

プロジェクト、およびプロジェクトに追加しているファイル、カテゴリ（フォルダへのショートカットを設定している場合のみ）に設定されている Windows エクスプローラのオーバーレイ・アイコンは、以下の例のように通常のアイコンの左側に表示します。

通常のプロジェクト	 sample (プロジェクト)
オーバーレイ・アイコンを表示したプロジェクト	 sample (プロジェクト)

注意 上記のオーバーレイ・アイコンは、サンプルとして掲載しています。

お使いのツールによって表示されるアイコンは異なりますので、ご注意ください。

なお、オーバーレイ・アイコンの表示は、以下のタイミングで更新します。

- プロジェクトを読み込んだとき
- ツールバーの  をクリックしたとき
- [編集] メニュー→[最新の情報に更新] を選択したとき

備考 本機能は、**オプション ダイアログ**の[全般 - 表示]カテゴリの[プロジェクト・ツリーに Windows エクスプローラのオーバーレイ・アイコンを表示する]がチェック状態で本製品を起動した場合のみ有効となります。

(r) フォルダへのショートカットを設定しているカテゴリの強調表示

フォルダへのショートカットを設定しているカテゴリは、以下の例のように強調表示します。

通常のカテゴリ	 source
フォルダへのショートカットを設定しているカテゴリ	 source

(s) 変更したプロジェクトの強調表示

プロジェクトに追加しているファイル構成を変更した場合、およびプロジェクトの構成要素のプロパティを変更した場合、プロジェクト名に“*”を付加し、太字表示します。

強調表示は、プロジェクトを保存すると解除します。

通常のプロジェクト	 sample (プロジェクト)
変更したプロジェクト	 sample (プロジェクト)*

(t) アクティブ・プロジェクトの強調表示

アクティブ・プロジェクトには、下線を付加します。

通常のプロジェクト	 sample (プロジェクト)
アクティブ・プロジェクト	 <u>sample (プロジェクト)*</u>

(u) ファイルの強調表示の状態を更新

以下の方法により、ファイル、読み取り専用属性ファイル、存在しないファイル、およびオーバーレイ・アイコンの強調表示の状態を最新の情報に更新することができます。

- ツールバーの  を選択する。

(v) エディタの起動

特定の拡張子を持つファイルを**エディタ パネル**でオープンします。**オプション ダイアログ**で、外部テキスト・エディタを使用する設定をしている場合は、設定している外部テキスト・エディタでオープンします。それ以外のファイルは、ホスト OS で関連付けられているアプリケーションで起動します。

注意 ホスト OS で関連付けられていない拡張子のファイルは表示しません。

以下のいずれかの方法により、エディタをオープンすることができます。

- ファイルをダブルクリックする。
- ファイルを選択し、コンテキスト・メニューの「開く」を選択する。
- ファイルを選択し、[Enter] キーを押下する。

以下に、**エディタ パネル**でオープンできるファイルを示します。

- C ソース・ファイル (*.c)
- アセンブリ・ソース・ファイル (*.asm, *.s)
- シンボル・アドレス・ファイル (*.fsy)
- ヘッダ・ファイル (*.h, *.inc)
- アセンブリ・プログラム用リスト・ファイル (*.lst, *.lis)
- アセンブリ・プログラム用プリプロセッサ展開ファイル (*.exp)
- アセンブル・リスト・ファイル (*.prn)
- リンク順指定ファイル (*.mtls)
- リンク・マップ・ファイル (*.map)
- ライブラリ・リスト・ファイル (*.lbp)
- インテル拡張ヘキサ・ファイル (*.hex)
- モトローラ・S タイプ・ファイル (*.mot)
- テキスト・ファイル (*.txt)

備考 以下のいずれかの方法により、上記以外のファイルも**エディタ パネル**でオープンすることができます。

- ファイルをドラッグし、**エディタ パネル**にドロップする。
- ファイルを選択し、コンテキスト・メニューの「内部エディタで開く ...」を選択する。

[[編集] メニュー (プロジェクト・ツリー パネル専用部分)]

コピー	選択しているファイル、カテゴリ・ノードをクリップ・ボードにコピーします。 ファイル名、カテゴリ名を編集中の場合は、選択している文字列をクリップ・ボードにコピーします。 なお、本メニューは、ファイル（依存関係ファイルは除く）、カテゴリ・ノードを選択している場合のみ有効となります。
貼り付け	クリップ・ボードの内容をプロジェクト・ツリー上で選択しているノードの直下に挿入します。 ファイル名、カテゴリ名を編集中の場合は、クリップ・ボードの内容を挿入します。 なお、本メニューは、クリップボードの内容が同一プロジェクトに存在する場合、ファイル、カテゴリ・ノードを複数選択している場合、およびビルド・ツールが実行中の場合は無効となります。
名前の変更	選択しているプロジェクト、サブプロジェクト、ファイル、カテゴリ・ノードの名前が編集可能な状態になります。[Enter] キーの押下により編集を確定し、[ESC] キーの押下により編集をキャンセルすることができます。 ファイルを選択している場合は、実際のファイル名も変更します。 ファイルを選択し、そのファイルをほかのプロジェクトにも追加している場合は、それらの名前も変更します。 なお、本メニューは、プロジェクト、サブプロジェクト（依存関係ファイルは除く）、ファイル、カテゴリ・ノードを選択している場合のみ有効となります。ただし、ビルド・ツールが実行中の場合は無効となります。

[コンテキスト・メニュー]

(1) プロジェクト・ノードを選択している場合

アクティブ・プロジェクトをビルド	アクティブ・プロジェクトのビルドを行います。 アクティブ・プロジェクトがメイン・プロジェクトの場合、サブプロジェクトのビルドは行いません。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
アクティブ・プロジェクトをリビルド	アクティブ・プロジェクトのリビルドを行います。 アクティブ・プロジェクトがメイン・プロジェクトの場合、サブプロジェクトのリビルドは行いません。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
アクティブ・プロジェクトをクリーン	アクティブ・プロジェクトのクリーンを行います。 アクティブ・プロジェクトがメイン・プロジェクトの場合、サブプロジェクトのクリーンは行いません。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
エクスプローラでフォルダを開く	選択しているプロジェクトのプロジェクト・ファイルが存在しているフォルダをエクスプローラでオープンします。
Windows エクスプローラのメニュー	選択しているプロジェクトのプロジェクト・ファイルに対応する Windows エクスプローラでのコンテキスト・メニューを表示します。
追加	プロジェクトにサブプロジェクト、ファイルを追加するためのカスケード・メニューを表示します。
既存のサブプロジェクトを追加 ...	既存のサブプロジェクトを追加 ダイアログをオープンし、選択したサブプロジェクトをプロジェクトに追加します。
新しいサブプロジェクトを追加 ...	プロジェクト作成 ダイアログをオープンし、作成したサブプロジェクトをプロジェクトに追加します。
既存のファイルを追加 ...	既存のファイルを追加 ダイアログをオープンし、選択したファイルをプロジェクトに追加します。
新しいファイルを追加 ...	ファイル追加 ダイアログをオープンし、選択した種類でファイルを作成し、プロジェクトに追加します。 追加したファイルはファイルの拡張子に割り当てられたアプリケーションでオープンします。
新しいカテゴリを追加	ファイル・ノードの直下にカテゴリ・ノードを追加し、カテゴリ名が編集可能な状態になります。 カテゴリ名は、200 文字まで指定可能です。 カテゴリ名は、デフォルトで“新しいカテゴリ”となります。すでに存在するカテゴリ・ノードと同名のカテゴリ・ノードを追加することもできます。 なお、本メニューは、ビルド・ツールが実行中の場合、およびカテゴリのネスト数が 20 の場合は無効となります。
選択しているプロジェクトをアクティブ・プロジェクトに設定	選択しているプロジェクトをアクティブ・プロジェクトに設定します。

プロジェクトと開発ツールをバックして保存 ...	本製品一式とプロジェクト一式を指定したフォルダにコピーして、1つのフォルダにまとめて保存します。 また、プロジェクトのみをバックして保存することも可能です。
貼り付け	本メニューは常に無効です。
名前の変更	選択しているプロジェクトの名前が編集可能な状態になります。
プロパティ	選択しているプロジェクトのプロパティを プロパティ パネル に表示します。

(2) サブプロジェクト・ノードを選択している場合

アクティブ・プロジェクトをビルド	アクティブ・プロジェクトのビルドを行います。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
アクティブ・プロジェクトをリビルド	アクティブ・プロジェクトのリビルドを行います。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
アクティブ・プロジェクトをクリーン	アクティブ・プロジェクトのクリーンを行います。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
エクスプローラでフォルダを開く	選択しているサブプロジェクトのサブプロジェクト・ファイルが存在しているフォルダをエクスプローラでオープンします。
Windows エクスプローラのメニュー	選択しているサブプロジェクトのサブプロジェクト・ファイルに対応するWindows エクスプローラでのコンテキスト・メニューを表示します。
追加	プロジェクトにサブプロジェクト、ファイル、カテゴリ・ノードを追加するためのカスケード・メニューを表示します。
既存のサブプロジェクトを追加 ...	既存のサブプロジェクトを追加 ダイアログをオープンし、選択したサブプロジェクトをプロジェクトに追加します。 サブプロジェクトにサブプロジェクトを追加することはできません。
新しいサブプロジェクトを追加 ...	プロジェクト作成 ダイアログをオープンし、作成したサブプロジェクトをプロジェクトに追加します。 サブプロジェクトにサブプロジェクトを追加することはできません。
既存のファイルを追加 ...	既存のファイルを追加 ダイアログ をオープンし、選択したファイルをプロジェクトに追加します。
新しいファイルを追加 ...	ファイル追加 ダイアログ をオープンし、選択した種類でファイルを作成し、プロジェクトに追加します。 追加したファイルはファイルの拡張子に割り当てられたアプリケーションでオープンします。
新しいカテゴリを追加	ファイル・ノードの直下にカテゴリ・ノードを追加し、カテゴリ名が編集可能な状態になります。 カテゴリ名は、200文字まで指定可能です。 カテゴリ名は、デフォルトで“新しいカテゴリ”となります。すでに存在するカテゴリ・ノードと同名のカテゴリ・ノードを追加することもできます。 なお、本メニューは、ビルド・ツールが実行中の場合、およびカテゴリのネスト数が20の場合は無効となります。
選択しているサブプロジェクトをアクティブ・プロジェクトに設定	選択しているサブプロジェクトをアクティブ・プロジェクトに設定します。

プロジェクトから外す	選択しているサブプロジェクトをプロジェクトから外します。 サブプロジェクト・ファイル自体はファイル・システム上から削除しません。 選択しているサブプロジェクトがアクティブ・プロジェクトの場合は、プロジェクトから外すことはできません。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
貼り付け	本メニューは常に無効です。
名前の変更	選択しているサブプロジェクトの名前が編集可能な状態になります。
プロパティ	選択しているサブプロジェクトのプロパティを プロパティ パネル に表示します。

(3) ビルド・ツール・ノードを選択している場合

ビルド・プロジェクト	選択しているプロジェクト（メイン・プロジェクト、またはサブプロジェクト）のビルドを行います。サブプロジェクトを追加している場合は、サブプロジェクトのビルドも行います。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
リビルド・プロジェクト	選択しているプロジェクト（メイン・プロジェクト、またはサブプロジェクト）のリビルドを行います。サブプロジェクトを追加している場合は、サブプロジェクトのリビルドも行います。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
クリーン・プロジェクト	選択しているプロジェクト（メイン・プロジェクト、またはサブプロジェクト）のクリーンを行います。サブプロジェクトを追加している場合は、サブプロジェクトのクリーンも行います。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
現在のビルド・オプションを選択しているプロジェクトの標準に設定する	現在のビルド・オプションを選択しているプロジェクトの標準に設定します。サブプロジェクトを追加している場合、サブプロジェクトの設定は行いません。 標準と異なるビルド・オプションを設定した場合、そのプロパティは太字表示します。
ビルド・オプションのインポート ...	ビルド・オプションのインポート ダイアログ をオープンし、選択したプロジェクト・ファイルからビルド・オプションをインポートします。 注
リンク順を設定する ...	リンク順設定 ダイアログ をオープンし、オブジェクト・ファイル、ライブラリ・ファイルの表示、およびリンク順の設定を行います。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
プロパティ	選択しているビルド・ツールのプロパティを プロパティ パネル に表示します。

注 ビルド・オプションのインポート機能についての詳細は、「[2.11.1 他のプロジェクトのビルド・オプションをインポートする](#)」を参照してください。

(4) ファイル・ノードを選択している場合

追加	プロジェクトにファイル、カテゴリ・ノードを追加するためのカスケード・メニューを表示します。
既存のファイルを追加 ...	既存のファイルを追加 ダイアログ をオープンし、選択したファイルをプロジェクトに追加します。追加先は、本ノードの直下です。 追加したファイルはファイルの拡張子に割り当てられたアプリケーションでオープンします。
新しいファイルを追加 ...	ファイル追加 ダイアログ をオープンし、選択した種類でファイルを作成し、プロジェクトに追加します。追加先は、本ノードの直下です。 追加したファイルはファイルの拡張子に割り当てられたアプリケーションでオープンします。
新しいカテゴリを追加	本ノードの直下にカテゴリ・ノードを追加し、カテゴリ名が編集可能な状態になります。 カテゴリ名は、200 文字まで指定可能です。 カテゴリ名は、デフォルトで“新しいカテゴリ”となります。すでに存在するカテゴリ・ノードと同名のカテゴリ・ノードを追加することもできます。 なお、本メニューは、ビルド・ツールが実行中の場合、およびカテゴリのネスト数が 20 の場合は無効となります。
エクスプローラでフォルダを開く	本メニューは常に無効です。
Windows エクスプローラのメニュー	本メニューは常に無効です。
プロジェクトから外す	本メニューは常に無効です。
コピー	本メニューは常に無効です。
貼り付け	クリップ・ボードの内容を本ノードの直下に挿入します。 ただし、クリップボードの内容が同一プロジェクトに存在する場合は、無効となります。
名前の変更	本メニューは常に無効です。
プロパティ	本ノードのプロパティを プロパティ パネル に表示します。

(5) ファイルを選択している場合

コンパイル	選択している C ソース・ファイルをコンパイルします。 なお、本メニューは、C ソース・ファイル（ビルド対象外のファイルを除く）を選択している場合のみ表示します。 ただし、ビルド・ツールが実行中の場合は無効となります。
アセンブル	選択しているアセンブリ・ソース・ファイルをアセンブルします。 なお、本メニューは、アセンブリ・ソース・ファイル（ビルド対象外のファイルを除く）を選択している場合のみ表示します。 ただし、ビルド・ツールが実行中の場合は無効となります。
開く	ファイルの拡張子に割り当てられたアプリケーションで選択しているファイルをオープンします（「 (v) エディタの起動 」参照）。
内部エディタで開く ...	エディタ パネル で選択しているファイルをオープンします。

アプリケーションを指定して開く ...	プログラムから開く ダイアログ をオープンし、指定したアプリケーションで選択しているファイルをオープンします。
エクスプローラでフォルダを開く	選択しているファイルが存在しているフォルダをエクスプローラでオープンします。
Windows エクスプローラのメニュー	選択しているファイルに対応する Windows エクスプローラでのコンテキスト・メニューを表示します。
追加	プロジェクトにファイル、カテゴリ・ノードを追加するためのカスケード・メニューを表示します。
既存のファイルを追加 ...	既存のファイルを追加 ダイアログ をオープンし、選択したファイルをプロジェクトに追加します。追加先は、選択しているファイルと同じレベルです。
新しいファイルを追加 ...	ファイル追加 ダイアログ をオープンし、選択した種類でファイルを作成し、プロジェクトに追加します。追加先は、選択しているファイルと同じレベルです。追加したファイルはファイルの拡張子に割り当てられたアプリケーションでオープンします。
新しいカテゴリを追加	選択しているファイルと同じレベルにカテゴリ・ノードを追加し、カテゴリ名が編集可能な状態になります。 カテゴリ名は、200 文字まで指定可能です。 カテゴリ名は、デフォルトで“新しいカテゴリ”となります。すでに存在するカテゴリ・ノードと同名のカテゴリ・ノードを追加することもできます。 なお、本メニューは、ビルド・ツールが実行中の場合、およびカテゴリのネスト数が 20 の場合は無効となります。
プロジェクトから外す	選択しているファイルをプロジェクトから外します。 ファイル自体はファイル・システム上から削除しません。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
コピー	選択しているファイルをクリップ・ボードにコピーします。 ファイル名を編集の場合は、選択している文字列をクリップ・ボードにコピーします。
貼り付け	本メニューは常に無効です。
名前の変更	選択しているファイルの名前が編集可能な状態になります。 実際のファイル名も変更します。 選択しているファイルをほかのプロジェクトにも追加している場合は、それらの名前も変更します。
拡張子の変更 ...	選択しているファイルの拡張子を変更するかどうかを確認するメッセージダイアログを表示します。 ダイアログ上で [はい] をクリックすると、文字列入力 ダイアログをオープンし、指定した拡張子で選択しているファイルの拡張子を変更します。 複数のファイルを選択している場合は、一括して変更します。 なお、本メニューは、ファイル名の変更とプロジェクトから外すことができないファイルを選択している場合、およびビルド・ツールが実行中の場合は無効となります。
プロパティ	選択しているファイルのプロパティを プロパティ パネル に表示します。

(6) ビルド・ツール生成ファイル・ノードを選択している場合

プロパティ	本ノードのプロパティを プロパティ パネル に表示します。
-------	---

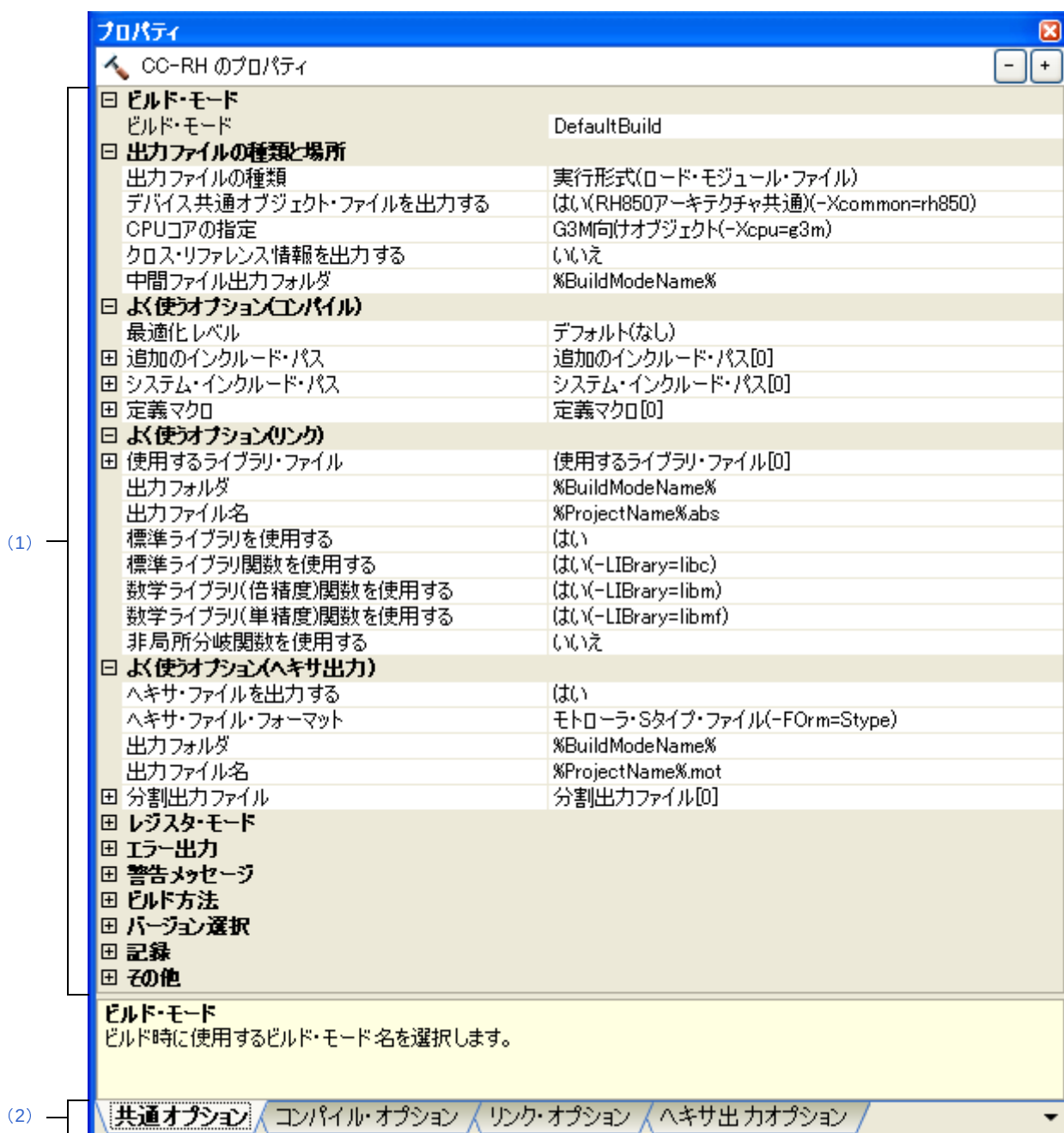
(7) カテゴリ・ノードを選択している場合

追加	プロジェクトにファイル、カテゴリ・ノードを追加するためのカスケード・メニューを表示します。
既存のファイルを追加 ...	既存のファイルを追加 ダイアログ をオープンし、選択したファイルをプロジェクトに追加します。追加先は、本ノードの直下です。 追加したファイルはファイルの拡張子に割り当てられたアプリケーションでオープンします。
新しいファイルを追加 ...	ファイル追加 ダイアログ をオープンし、選択した種類でファイルを作成し、プロジェクトに追加します。追加先は、本ノードの直下です。 追加したファイルはファイルの拡張子に割り当てられたアプリケーションでオープンします。
新しいカテゴリを追加	本ノードの直下にカテゴリ・ノードを追加し、カテゴリ名が編集可能な状態になります。 カテゴリ名は、200 文字まで指定可能です。 カテゴリ名は、デフォルトで“新しいカテゴリ”となります。すでに存在するカテゴリ・ノードと同名のカテゴリ・ノードを追加することもできます。 なお、本メニューは、ビルド・ツールが実行中の場合、およびカテゴリのネスト数が 20 の場合は無効となります。
エクスプローラでフォルダを開く	選択しているカテゴリに設定しているフォルダへのショートカット先をエクスプローラでオープンします。 なお、本メニューは、フォルダへのショートカットを設定していない場合は無効となります。
Windows エクスプローラのメニュー	選択しているカテゴリに設定しているフォルダへのショートカット先に対応する Windows エクスプローラでのコンテキスト・メニューを表示します。 なお、本メニューは、フォルダへのショートカットを設定していない場合は無効となります。
プロジェクトから外す	選択しているカテゴリ・ノードをプロジェクトから外します。 なお、本メニューは、ビルド・ツールが実行中の場合は無効となります。
コピー	選択しているカテゴリ・ノードをクリップ・ボードにコピーします。 カテゴリ名を編集の場合は、選択している文字列をクリップ・ボードにコピーします。
貼り付け	クリップ・ボードの内容を本ノードの直下に挿入します。 ただし、クリップボードの内容が同一プロジェクトに存在する場合は、無効となります。 カテゴリ名を編集の場合は、クリップ・ボードの内容を挿入します。
名前の変更	選択しているカテゴリ・ノードの名前が編集可能な状態になります。
プロパティ	選択しているカテゴリ・ノードのプロパティを プロパティ パネル に表示します。

プロパティ パネル

プロジェクト・ツリー パネルで選択しているビルド・ツール・ノード、ファイル、カテゴリ・ノードについて、カテゴリ別に詳細情報の表示、および設定の変更を行います。

図 A—3 プロパティ パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]

- [[編集] メニュー (プロパティ パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- プロジェクト・ツリー パネルにおいて、ビルド・ツール・ノード、ファイル、カテゴリ・ノードを選択したのち、[表示] メニュー→[プロパティ] を選択、またはコンテキスト・メニュー→[プロパティ] を選択

備考 すでにプロパティ パネルがオープンしている場合、プロジェクト・ツリー パネルにおいて、ビルド・ツール・ノード、ファイル、カテゴリ・ノードを選択することで、選択した項目の詳細情報を表示します。

[各エリアの説明]

(1) 詳細情報表示／変更エリア

プロジェクト・ツリー パネルで選択しているビルド・ツール・ノード、ファイル、カテゴリ・ノードの詳細情報を、カテゴリ別のリスト形式で表示し、設定の変更を直接行うことができるエリアです。

マークは、そのカテゴリ内に含まれているすべての項目を展開表示していることを示し、また、マークは、カテゴリ内の項目を折りたたみ表示していることを示します。展開／折りたたみ表示の切り替えは、このマークのクリック、またはカテゴリ名のダブルクリックにより行うことができます。

マークは、そのプロパティのテキスト・ボックスが16進数入力専用であることを示します。

カテゴリ、およびそれに含まれる項目の表示内容／設定方法についての詳細は、該当するタブの項を参照してください。

(2) タブ選択エリア

タブを選択することにより、詳細情報を表示するカテゴリが切り替わります。

本パネルには、次のタブが存在します（各タブ上における表示内容／設定方法についての詳細は、該当するタブの項を参照してください）。

(a) プロジェクト・ツリー パネルでビルド・ツール・ノードを選択している場合

- [共通オプション] タブ
- [コンパイル・オプション] タブ
- [アセンブル・オプション] タブ
- [リンク・オプション] タブ
- [ヘキサ出力オプション] タブ
- [ライブラリ生成オプション] タブ

(b) プロジェクト・ツリー パネルでファイルを選択している場合

- [ビルド設定] タブ (C ソース・ファイル、アセンブリ・ソース・ファイル、オブジェクト・ファイル、ライブラリ・ファイルの場合)
- [個別コンパイル・オプション] タブ (C ソース・ファイルの場合)
- [個別アセンブル・オプション] タブ (アセンブリ・ソース・ファイルの場合)

- [\[ファイル情報\] タブ](#)

(c) プロジェクト・ツリーパネルでカテゴリ・ノード、ファイル・ノード、ビルド・ツール生成ファイル・ノードを選択している場合

- [\[カテゴリ情報\] タブ](#)

備考 [プロジェクト・ツリーパネル](#)で複数の構成要素を選択している場合は、その構成要素に共通するタブのみ表示します。

プロパティの値の変更は、選択している複数の構成要素に共通に反映します。

[[編集] メニュー（プロパティ パネル専用部分）]

元に戻す	直前に行ったプロパティの値の編集作業を取り消します。
切り取り	プロパティの値を編集中的場合、選択している文字列を切り取ってクリップ・ボードに移動します。
コピー	選択しているプロパティの値文字列をクリップ・ボードにコピーします。
貼り付け	プロパティの値を編集中的場合、クリップ・ボードの内容を挿入します。
削除	プロパティの値を編集中的場合、選択している文字列を削除します。
すべて選択	プロパティの値を編集中的場合、選択しているプロパティの値文字列をすべて選択します。

[コンテキスト・メニュー]

元に戻す	直前に行ったプロパティの値の編集作業を取り消します。
切り取り	プロパティの値を編集中的場合、選択している文字列を切り取ってクリップ・ボードに移動します。
コピー	選択しているプロパティの値文字列をクリップ・ボードにコピーします。
貼り付け	プロパティの値を編集中的場合、クリップ・ボードの内容を挿入します。
削除	プロパティの値を編集中的場合、選択している文字列を削除します。
すべて選択	プロパティの値を編集中的場合、選択しているプロパティの値文字列をすべて選択します。
デフォルトに戻す	選択している項目の設定値をプロジェクトに設定しているデフォルト値に戻します。 ただし、 [個別コンパイル・オプション] タブ 、 [個別アセンブル・オプション] タブ においては、全体オプションの設定値に戻します。
すべてデフォルトに戻す	現在表示しているタブの設定値をすべてプロジェクトに設定しているデフォルト値に戻します。 ただし、 [個別コンパイル・オプション] タブ 、 [個別アセンブル・オプション] タブ においては、全体オプションの設定値に戻します。

[共通オプション] タブ

本タブでは、ビルド・ツールに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

- (1) [ビルド・モード]
- (2) [出力ファイルの種類と場所]
- (3) [よく使うオプション (コンパイル)]
- (4) [よく使うオプション (アセンブル)]
- (5) [よく使うオプション (リンク)]
- (6) [よく使うオプション (ヘキサ出力)]
- (7) [よく使うオプション (ライブラリ生成)]
- (8) [レジスタ・モード]
- (9) [エラー出力]
- (10) [警告メッセージ]
- (11) [ビルド方法]
- (12) [バージョン選択]
- (13) [記録]
- (14) [その他]

備考 [よく使うオプション] カテゴリのプロパティを変更した場合、それらに対応するタブの同名のプロパティの値も連動して変更します。

[共通オプション] タブのカテゴリ	対応するタブ
[よく使うオプション (コンパイル)] カテゴリ	[コンパイル・オプション] タブ
[よく使うオプション (アセンブル)] カテゴリ	[アセンブル・オプション] タブ
[よく使うオプション (リンク)] カテゴリ	[リンク・オプション] タブ
[よく使うオプション (ヘキサ出力)] カテゴリ	[ヘキサ出力オプション] タブ
[よく使うオプション (ライブラリ生成)] カテゴリ	[ライブラリ生成オプション] タブ

図 A—4 プロパティ パネル : [共通オプション] タブ

プロパティ

CC-RH のプロパティ

☐ ビルド・モード	ビルド・モード	DefaultBuild
☐ 出力ファイルの種類と場所	出力ファイルの種類	実行形式(ロード・モジュール・ファイル)
	デバイス共通オブジェクト・ファイルを出力する	(はい)(RH850アーキテクチャ共通)(-Xcommon=rh850)
	CPUコアの指定	G3M向けオブジェクト(-Xcpu=g3m)
	クロス・リファレンス情報を出力する	いいえ
	中間ファイル出力フォルダ	%BuildModeName%
☐ よく使うオプション(コンパイル)	最適化レベル	デフォルト(なし)
	追加のインクルード・パス	追加のインクルード・パス[0]
	システム・インクルード・パス	システム・インクルード・パス[0]
	定義マクロ	定義マクロ[0]
☐ よく使うオプション(リンク)	使用するライブラリ・ファイル	使用するライブラリ・ファイル[0]
	出力フォルダ	%BuildModeName%
	出力ファイル名	%ProjectName%.abs
	標準ライブラリを使用する	(はい)
	標準ライブラリ関数を使用する	(はい)(-LIBrary=libc)
	数学ライブラリ(倍精度)関数を使用する	(はい)(-LIBrary=libm)
	数学ライブラリ(単精度)関数を使用する	(はい)(-LIBrary=libmf)
	非局所分岐関数を使用する	いいえ
☐ よく使うオプション(ヘキサ出力)	ヘキサ・ファイルを出力する	(はい)
	ヘキサ・ファイル・フォーマット	モトローラ・Sタイプ・ファイル(-FOrm=Stype)
	出力フォルダ	%BuildModeName%
	出力ファイル名	%ProjectName%.mot
	分割出力ファイル	分割出力ファイル[0]
☐ レジスタ・モード		
☐ エラー出力		
☐ 警告メッセージ		
☐ ビルド方法		
☐ バージョン選択		
☐ 記録		
☐ その他		

ビルド・モード
ビルド時に使用するビルド・モード名を選択します。

共通オプション / コンパイル・オプション / リンク・オプション / ヘキサ出力オプション

[各カテゴリの説明]

(1) [ビルド・モード]

ビルド・モードに関する詳細情報の表示、および設定の変更を行います。

ビルド・モード	ビルド時に使用するビルド・モードを選択します。 なお、本プロパティには、コンテキスト・メニュー→ [すべてデフォルトに戻す] は適用されません。				
	デフォルト	DefaultBuild			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>DefaultBuild</td><td>プロジェクトの新規作成時にデフォルトで設定するビルド・モードでビルドを行います。</td></tr><tr><td>プロジェクトに登録しているビルド・モード</td><td>プロジェクトに登録しているビルド・モード (DefaultBuild 以外) でビルドを行います。</td></tr></table>	DefaultBuild	プロジェクトの新規作成時にデフォルトで設定するビルド・モードでビルドを行います。	プロジェクトに登録しているビルド・モード
DefaultBuild	プロジェクトの新規作成時にデフォルトで設定するビルド・モードでビルドを行います。				
プロジェクトに登録しているビルド・モード	プロジェクトに登録しているビルド・モード (DefaultBuild 以外) でビルドを行います。				

(2) [出力ファイルの種類と場所]

出力ファイルの種類と場所に関する詳細情報の表示、および設定の変更を行います。

出力ファイルの種類	ビルド時に生成するファイルの種類を選択します。 ここで設定したファイルの種類がデバッグ対象となります。 なお、ライブラリ用のプロジェクト以外の場合は [実行形式 (ロード・モジュール・ファイル)], [実行形式 (ヘキサ・ファイル)] のみを表示します。 ライブラリ用のプロジェクトの場合は [ライブラリ形式] のみを表示します。		
	デフォルト	- ライブラリ用のプロジェクト以外の場合 実行形式 (ロード・モジュール・ファイル) - ライブラリ用のプロジェクトの場合 ライブラリ形式	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	実行形式 (ロード・モジュール・ファイル)	ビルド時にロード・モジュール・ファイルを生成します。 ロード・モジュール・ファイルがデバッグ対象となります。
		実行形式 (ヘキサ・ファイル)	ビルド時にヘキサ・ファイルを生成します。 ヘキサ・ファイルがデバッグ対象となります。 なお、本項目は、 [ヘキサ出力オプション] タブ の [出力ファイル] カテゴリの [ヘキサ・ファイルを出力する] プロパティで [はい] を選択した場合のみ表示します。
	ライブラリ形式	ビルド時にライブラリ・ファイルを生成します。	

デバイス共通オブジェクト・ファイルを出力する	デバイス共通のオブジェクト・ファイルを出力するかどうかを選択します。 ccrh コマンドの -Xcommon オプションに相当します。	
	デフォルト	はい (RH850 アーキテクチャ共通)(-Xcommon=rh850)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (RH850 アーキテクチャ共通)(-Xcommon=rh850)
		はい (V850E3V5 アーキテクチャ共通)(-Xcommon=v850e3v5)
CPU コアの指定	CPU コアを指定します。 ccrh コマンドの -Xcpu オプションに相当します。	
	デフォルト	G3M 向けオブジェクト (-Xcpu=g3m)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	G3M 向けオブジェクト (-Xcpu=g3m)
		G3K 向けオブジェクト (-Xcpu=g3k)
クロス・リファレンス情報を出力する	クロス・リファレンス情報をファイルに出力するかどうかを選択します。 ファイルの出力先は、 [リンク・オプション] タブの [出力ファイル] カテゴリの [出力フォルダ] プロパティで指定したフォルダです。 [出力ファイル名] プロパティで指定したファイルの拡張子を “.cref” で置き換えたファイル名で出力します。 ただし、 [個別コンパイル・オプション] タブの [出力ファイル] カテゴリの [オブジェクト・ファイル名] プロパティを指定している場合は、そのプロパティで指定したファイルの拡張子を “.cref” で置き換えたファイル名で出力します。 また、解析ツールの [強制的にクロス・リファレンス・ファイルを出力する] プロパティで [はい] を選択した場合、本プロパティは [はい (-Xcref)] となります。[強制的にクロス・リファレンス・ファイルを出力する] プロパティが [はい] の場合に、本プロパティを [いいえ] に変更した場合は、ビルド時に [はい (-Xcref)] に変更されます。 ccrh コマンドの -Xcref オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xcref)
		いいえ

中間ファイル出力フォルダ	中間ファイルを出力するフォルダを指定します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの -Xobj_path オプションに相当します。	
	デフォルト	%BuildModeName%
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列

(3) [よく使うオプション（コンパイル）]

コンパイル時によく使うオプションに関する詳細情報の表示、および設定の変更を行います。

最適化レベル	コンパイルの最適化レベルを選択します。 ccrh コマンドの -O オプションに相当します。		
	デフォルト	デフォルト (なし)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	デフォルト (なし)	デバッグに影響しない範囲の最適化 (式の最適化、およびレジスタ割り付けなど) を行います。
		サイズ優先 (-Osize)	オブジェクト・サイズ優先の最適化を行います。 ROM/RAM 容量の削減を重視して、一般的なプログラムに対して有効な最大限の最適化を行います。
		実行速度優先 (-Ospeed)	実行速度優先の最適化を行います。 実行速度の短縮を重視して、一般的なプログラムに対して有効な最大限の最適化を行います。
デバッグ優先 (-Onothing)		デバッグを優先して最適化を行います。 デバッグのしやすさを重視し、デフォルトで実行する最適化を含むすべての最適化を抑止します。	

追加のインクルード・パス	コンパイル時の追加のインクルード・パスを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 指定したインクルード・パスは、CC-RH の標準インクルード・ファイル・フォルダよりも優先して検索します。 パスはプロジェクト・フォルダを基点とします。 本プロパティを省略した場合は、CC-RH の標準インクルード・ファイル・フォルダのみ検索します。 ccrh コマンドの -I オプションに相当します。 指定したインクルード・パスはサブプロパティとして表示します。 なお、プロジェクト・ツリーにインクルード・ファイルを追加すると、インクルード・パスをサブプロパティの一番最初に追加します。 インクルード・パスに大文字、小文字の区別はありません。
デフォルト	追加のインクルード・パス [定義数]
変更方法	[...] ボタンをクリックし、 パス編集ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
指定可能値	259 文字までの文字列 256 個まで指定可能です。

システム・インクルード・パス	コンパイル時にシステムが設定するインクルード・パスの指定順を変更します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 システム・インクルード・パスは、追加のインクルード・パスより低い優先度で検索します。 パスはプロジェクト・フォルダを基点とします。 ccrh コマンドの -I オプションに相当します。 インクルード・パスはサブプロパティとして表示します。	
	デフォルト	システム・インクルード・パス [定義数]
	変更方法	[...] ボタンをクリックし、 システム・インクルード・パス順設定 ダイアログ による編集
	指定可能値	変更不可（インクルード・パスの設定順の変更のみ可能）
定義マクロ	定義したいマクロ名を指定します。 「マクロ名 = 定義値」の形式で 1 行に 1 つずつ指定します。 「= 定義値」の部分は省略可能で、省略した場合、定義値を 1 とします。 ccrh コマンドの -D オプションに相当します。 指定したマクロはサブプロパティとして表示します。	
	デフォルト	定義マクロ [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

(4) [よく使うオプション（アセンブル）]

アセンブル時によく使うオプションに関する詳細情報の表示、および設定の変更を行います。

なお、本カテゴリは、[\[ビルド方法\]](#) カテゴリの「一括ビルドを行う」プロパティで「いいえ」を選択した場合のみ表示します。

追加のインクルード・パス	アセンブル時の追加のインクルード・パスを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 指定したインクルード・パスは、CC-RH の標準インクルード・ファイル・フォルダよりも優先して検索します。 パスはプロジェクト・フォルダを基点とします。 本プロパティを省略した場合は、CC-RH の標準インクルード・ファイル・フォルダのみ検索します。 ccrh コマンドの -I オプションに相当します。 指定したインクルード・パスはサブプロパティとして表示します。 なお、プロジェクト・ツリーにインクルード・ファイルを追加すると、インクルード・パスをサブプロパティの一番最初に追加します。 インクルード・パスに大文字、小文字の区別はありません。	
	デフォルト	追加のインクルード・パス [定義数]
	変更方法	[...] ボタンをクリックし、 パス編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	259 文字までの文字列 256 個まで指定可能です。

システム・インクルード・パス	アセンブル時にシステムが設定するインクルード・パスの指定順を変更します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 システム・インクルード・パスは、追加のインクルード・パスより低い優先度で検索します。 パスはプロジェクト・フォルダを基点とします。 ccrh コマンドの -I オプションに相当します。 インクルード・パスはサブプロパティとして表示します。	
	デフォルト	システム・インクルード・パス [定義数]
	変更方法	[...] ボタンをクリックし、 システム・インクルード・パス順設定 ダイアログ による編集
	指定可能値	変更不可（インクルード・パスの設定順の変更のみ可能）
定義マクロ	定義したいマクロ名を指定します。 「マクロ名 = 定義値」の形式で 1 行に 1 つずつ指定します。 「= 定義値」の部分は省略可能で、省略した場合、定義値を 1 とします。 ccrh コマンドの -D オプションに相当します。 指定したマクロはサブプロパティとして表示します。	
	デフォルト	定義マクロ [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

(5) [よく使うオプション（リンク）]

リンク時によく使うオプションに関する詳細情報の表示、および設定の変更を行います。

なお、本カテゴリは、ライブラリ用のプロジェクトの場合は表示しません。

使用するライブラリ・ファイル	使用するライブラリ・ファイルを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 rlink コマンドの -Library オプションに相当します。 ライブラリ・ファイル名はサブプロパティとして表示します。	
	デフォルト	使用するライブラリ・ファイル [定義数]
	変更方法	[...] ボタンをクリックし、パス編集 ダイアログによる編集 → [参照] ボタンをクリックし、使用するライブラリ・ファイルを指定 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	259 文字までの文字列 65536 個まで指定可能です。
出力フォルダ	出力フォルダを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 rlink コマンドの -Output オプションに相当します。 なお、本プロパティは、[ロード・モジュール・ファイルを分割する] プロパティで [いいえ] を選択した場合のみ表示します。	
	デフォルト	%BuildModeName%
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、フォルダの参照 ダイアログによる編集
	指定可能値	247 文字までの文字列

出力ファイル名	出力ファイル名を指定します。 拡張子を省略した場合は、“.abs”を自動的に付加します。 次のプレースホルダに対応しています。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %ProjectName% : プロジェクト名に置換します。 rlink コマンドの -Output オプションに相当します。 なお、本プロパティは、[ロード・モジュール・ファイルを分割する] プロパティで [いいえ] を選択した場合のみ表示します。		
	デフォルト	%ProjectName%.abs	
	変更方法	テキスト・ボックスによる直接入力	
	指定可能値	259 文字までの文字列	
標準ライブラリを使用する	コンパイラが提供する標準ライブラリを使用するかどうかを選択します。 rlink コマンドの -LibRary オプションに相当します。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	標準ライブラリを使用します。
	いいえ	標準ライブラリを使用しません。	
標準ライブラリ関数を使用する	標準ライブラリ関数を使用するかどうかを選択します。 [数学ライブラリ (倍精度) 関数を使用する]、または [数学ライブラリ (単精度) 関数を使用する] プロパティで [はい] を選択した場合、本プロパティは [はい] となります。 rlink コマンドの -LibRary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。		
	デフォルト	はい (-LibRary=libc)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-LibRary=libc)	標準ライブラリ関数を使用します。
	いいえ	標準ライブラリ関数を使用しません。	
数学ライブラリ (倍精度) 関数を使用する	数学ライブラリ (倍精度) 関数を使用するかどうかを選択します。 [標準ライブラリ関数を使用する] プロパティで [いいえ] を選択した場合、本プロパティは [いいえ] となります。 rlink コマンドの -LibRary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。		
	デフォルト	はい (-LibRary=libm)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-LibRary=libm)	数学ライブラリ (倍精度) 関数を使用します。
	いいえ	数学ライブラリ (倍精度) 関数を使用しません。	

数学ライブラリ (単精度) 関数を使用する	数学ライブラリ (単精度) 関数を使用するかどうかを選択します。 [標準ライブラリ関数を使用する] プロパティで [いいえ] を選択した場合、本プロパティは [いいえ] となります。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	はい (-LIBrary=libmf)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libmf) 数学ライブラリ (単精度) 関数を使用します。
		いいえ 数学ライブラリ (単精度) 関数を使用しません。
非局所分岐関数を使用する	非局所分岐関数を使用するかどうかを選択します。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libsetjmp) 非局所分岐関数を使用します。
		いいえ 非局所分岐関数を使用しません。

(6) [よく使うオプション (ヘキサ出力)]

ヘキサ出力時によく使うオプションに関する詳細情報の表示、および設定の変更を行います。

なお、本カテゴリは、ライブラリ用のプロジェクトの場合は表示しません。

ヘキサ・ファイルを出力する	ヘキサ・ファイルを出力するかどうかを選択します。 rlink コマンドの -FOrm オプションに相当します。	
	デフォルト	はい
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい ヘキサ・ファイルを出力します。
		いいえ ヘキサ・ファイルを出力しません。

ヘキサ・ファイル・フォーマット	出力するヘキサ・ファイルのフォーマットを選択します。 rlink コマンドの -FOrm オプションに相当します。 なお、本プロパティは、[ヘキサ・ファイルを出力する] プロパティで [はい] を選択した場合のみ表示します。		
	デフォルト	モトローラ・S タイプ・ファイル (-FOrm=Stype)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)	インテル拡張ヘキサ・ファイルを出力します。
		モトローラ・S タイプ・ファイル (-FOrm=Stype)	モトローラ・S タイプ・ファイルを出力します。
バイナリ・ファイル (-FOrm=Binary)		バイナリ・ファイルを出力します。	
出力フォルダ	ヘキサ・ファイルの出力フォルダを指定します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 rlink コマンドの -OOutput オプションに相当します。 なお、本プロパティは、[ヘキサ・ファイルを出力する] プロパティで [はい] を選択した場合のみ表示します。		
	デフォルト	%BuildModeName%	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集	
	指定可能値	247 文字までの文字列	

出力ファイル名	ヘキサ・ファイル名を指定します。 拡張子を省略した場合は、[ヘキサ・ファイル・フォーマット] プロパティの選択に依存して、自動的に付加します。 [インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)] を選択している場合 : .hex [モトローラ・S タイプ・ファイル (-FOrm=Stype)] を選択している場合 : .mot [バイナリ・ファイル (-FOrm=Binary)] を選択している場合 : .bin 次のプレースホルダに対応しています。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %ProjectName% : プロジェクト名に置換します。 rlink コマンドの -Output オプションに相当します。 なお、本プロパティは、[ヘキサ・ファイルを出力する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	%ProjectName%.mot
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	259 文字までの文字列

分割出力ファイル	分割出力ファイルを指定します。 「ファイル名=先頭アドレス-終アドレス」、または「ファイル名=セクション名」の形式で1行に1ずつ指定します。 セクション名を複数指定する場合は、「ファイル名=セクション名:セクション名」のように、コロンで区切って指定します（例：file1.mot=sec1:sec2）。 アドレスは16進数で指定します（例：file2.mot=400-4ff）。 拡張子を省略した場合は、[ヘキサ・ファイル・フォーマット] プロパティの選択に依存して、自動的に付加します。 [インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)] を選択している場合：.hex [モトローラ・Sタイプ・ファイル (-FOrm=Stype)] を選択している場合：.mot [バイナリ・ファイル (-FOrm=Binary)] を選択している場合：.bin 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 rlink コマンドの -Output オプションに相当します。 分割出力ファイル名はサブプロパティとして表示します。 なお、本プロパティは、[ヘキサ・ファイルを出力する] プロパティで [はい] を選択した場合のみ表示します。
デフォルト	分割出力ファイル [定義数]
変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
指定可能値	259 文字までの文字列 65535 個まで指定可能です。

(7) [よく使うオプション (ライブラリ生成)]

ライブラリ生成時によく使うオプションに関する詳細情報の表示、および設定の変更を行います。

なお、本カテゴリは、ライブラリ用のプロジェクトの場合のみ表示します。

出力ファイル・フォーマット	出力ファイルのフォーマットを選択します。 rlink コマンドの -FOrm オプションに相当します。	
	デフォルト	ユーザ・ライブラリ・ファイル (-FOrm=Library=U)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	ユーザ・ライブラリ・ファイル (-FOrm=Library=U)
		システム・ライブラリ・ファイル (-FOrm=Library=S)
		リロケータブル・ファイル (-FOrm=Relocate)
出力フォルダ	出力フォルダを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 rlink コマンドの -Output オプションに相当します。	
	デフォルト	%BuildModeName%
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列
出力ファイル名	出力ファイル名を指定します。 拡張子を省略した場合は、[出力ファイル・フォーマット] プロパティの選択に依存して、自動的に付加します。 [ユーザ・ライブラリ・ファイル (-FOrm=Library=U)] を選択している場合 : .lib [システム・ライブラリ・ファイル (-FOrm=Library=S)] を選択している場合 : .lib [リロケータブル・ファイル (-FOrm=Relocate)] を選択している場合 : .rel 次のプレースホルダに対応しています。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %ProjectName% : プロジェクト名に置換します。 rlink コマンドの -Output オプションに相当します。	
	デフォルト	%ProjectName%.lib
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	259 文字までの文字列

標準ライブラリを使用する	コンパイラが提供する標準ライブラリを使用するかどうかを選択します。 rlink コマンドの -LIBrary オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい 標準ライブラリを使用します。 いいえ 標準ライブラリを使用しません。
標準ライブラリ関数を使用する	標準ライブラリ関数を使用するかどうかを選択します。 [数学ライブラリ (倍精度) 関数を使用する]、または [数学ライブラリ (単精度) 関数を使用する] プロパティで [はい] を選択した場合、本プロパティは [はい] となります。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libc) 標準ライブラリ関数を使用します。 いいえ 標準ライブラリ関数を使用しません。
数学ライブラリ (倍精度) 関数を使用する	数学ライブラリ (倍精度) 関数を使用するかどうかを選択します。 [標準ライブラリ関数を使用する] プロパティで [いいえ] を選択した場合、本プロパティは [いいえ] となります。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libmf) 数学ライブラリ (倍精度) 関数を使用します。 いいえ 数学ライブラリ (倍精度) 関数を使用しません。
数学ライブラリ (単精度) 関数を使用する	数学ライブラリ (単精度) 関数を使用するかどうかを選択します。 [標準ライブラリ関数を使用する] プロパティで [いいえ] を選択した場合、本プロパティは [いいえ] となります。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libmf) 数学ライブラリ (単精度) 関数を使用します。 いいえ 数学ライブラリ (単精度) 関数を使用しません。

非局所分岐関数を使用する	非局所分岐関数を使用するかどうかを選択します。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libsetjmp) 非局所分岐関数を使用します。 いいえ 非局所分岐関数を使用しません。

(8) [レジスタ・モード]

レジスタ・モードに関する詳細情報の表示、および設定の変更を行います。

レジスタ・モード	ソフトウェア・レジスタ・バンク機能のレジスタ・モード（コンパイラが使用するレジスタの本数） ^注 を選択します。 ccrh コマンドの -Xreg_mode オプションに相当します。	
	デフォルト	32 レジスタ・モード (なし)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	32 レジスタ・モード (なし) レジスタ・モードを 32 とします。 22 レジスタ・モード (-Xreg_mode=22) レジスタ・モードを 22 とします。 汎用レジスタ・モード (-Xreg_mode=common) レジスタ・モードを 22 とします。 レジスタ・モードに依存しないオブジェクト・ファイルを生成するために使用します。
r2 レジスタを予約する	r2 レジスタを予約するかどうかを選択します。 ccrh コマンドの -Xreserve_r2 オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xreserve_r2) r2 レジスタを予約し、コンパイラでは r2 レジスタを使用しないコードを生成します。 いいえ r2 レジスタの予約を行わずに、コンパイラで使用します。

ep レジスタの扱い	ep レジスタの扱い方を選択します。 プロジェクト内の #pragma section 指令において、以下の属性指定文字を使用した場合は、[プロジェクト全体で固定する (-Xep=fix)] を選択してください。 ep_auto, ep_disp4, ep_disp5, ep_disp7, ep_disp8, ep_disp16, ep_disp23 なお、 [コンパイル・オプション] タブの [最適化 (詳細)] カテゴリの [外部変数アクセス最適化を行う] プロパティで [はい] を選択した場合は [callee-save として扱う (なし)] のみ表示します。 ccrh コマンドの -Xep オプションに相当します。		
	デフォルト	callee-save として扱う (なし)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	callee-save として扱う (なし)	ep レジスタの値をプロジェクト全体で固定します。
		プロジェクト全体で固定する (-Xep=fix)	ep レジスタを関数呼び出し前後で値を保証するレジスタとして扱います。

注 CC-RH が提供しているレジスタ・モードを以下に示します。

レジスタ・モード	作業用レジスタ	レジスタ変数用レジスタ
common	r10 ~ r14	r25 ~ r29
22 レジスタ・モード	r10 ~ r14	r25 ~ r29
32 レジスタ・モード	r10 ~ r19	r20 ~ r29

(9) [エラー出力]

エラー出力に関する詳細情報の表示、および設定の変更を行います。

エラー・メッセージ・ ファイルを出力する	エラー・メッセージ・ファイルを出力するかどうかを選択します。 ccrh コマンドの -Xerror_file オプションに相当します。 なお、本プロパティの選択にかかわらず、エラー・メッセージは 出力 パネル に表示します。				
	デフォルト	いいえ			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-Xerror_file)</td><td>エラー・メッセージ・ファイルを出力します。</td></tr><tr><td>いいえ</td><td>エラー・メッセージ・ファイルを出力しません。</td></tr></table>	はい (-Xerror_file)	エラー・メッセージ・ファイルを出力します。	いいえ
はい (-Xerror_file)	エラー・メッセージ・ファイルを出力します。				
いいえ	エラー・メッセージ・ファイルを出力しません。				

エラー・メッセージ・ ファイル出力フォルダ	エラー・メッセージ・ファイルの出力先フォルダを指定します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの -Xerror_file オプションに相当します。 なお、本プロパティは、[エラー・メッセージ・ファイルを出力する] プロパティで [はい (-Xerror_file)] を選択した場合のみ表示します。	
	デフォルト	%BuildModeName%
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列
エラー・メッセージ・ ファイル名	エラー・メッセージ・ファイルの名前を指定します。 拡張子は自由に指定可能です。 次のプレースホルダに対応しています。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %ProjectName% : プロジェクト名に置換します。 空欄の場合は、%ProjectName%.err を指定したものとみなします。 ccrh コマンドの -Xerror_file オプションに相当します。 なお、本プロパティは、[エラー・メッセージ・ファイルを出力する] プロパティで [はい (-Xerror_file)] を選択した場合のみ表示します。	
	デフォルト	%ProjectName%.err
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	259 文字までの文字列

(10) [警告メッセージ]

警告メッセージに関する詳細情報の表示、および設定の変更を行います。

表示させない警告メッ セージ	表示させない警告メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例： 02042,02107）。 また、ハイフンを使用して、区間設定を行うこともできます（例：02222-02554,02699- 02782）。 ccrh コマンドの -Xno_warning オプションに相当します。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集
	指定可能値	2048 文字までの文字列

(11) [ビルド方法]

ビルド方法に関する詳細情報の表示、および設定の変更を行います。

一括ビルドを行う	複数のファイルを同時にコンパイル／アセンブル／リンクしてロード・モジュール・ファイルを生成するかどうかを選択します。 ただし、個別オプションを設定しているファイル、およびビルド前実行の対象となっているファイルは、一括ビルドの対象から除きます。 一括ビルドについての詳細は、「 2.12.5 複数のファイルのコンパイル／アセンブル／リンクを同時に実行する 」を参照してください。	
	デフォルト	はい
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい 複数のファイルを同時にコンパイル／アセンブル／リンクします。 アセンブリ・ソース・ファイル（個別オプションを設定しているファイルを除く）のアセンブルには、 [コンパイル・オプション] タブの設定を使用します。
		いいえ ファイル単位でコンパイル／アセンブル／リンクを行います。
インクルード・ファイルが存在しないソースの扱い	ソース・ファイルがインクルードしているファイルが存在しない場合、そのソース・ファイルを再コンパイル／アセンブルするかどうかを選択します。	
	デフォルト	再コンパイル／アセンブルする
	変更方法	ドロップダウン・リストによる選択
	指定可能値	再コンパイル／アセンブルする インクルード・ファイルが存在しない場合、ソース・ファイルを再コンパイル／アセンブルします。
		再コンパイル／アセンブルしない インクルード・ファイルが存在しない場合でも、ソース・ファイル再コンパイル／アセンブルしません。

(12) [バージョン選択]

ビルド・ツールのバージョン選択に関する詳細情報の表示、および設定の変更を行います。

使用するコンパイラ・パッケージのインストール・フォルダ	使用するコンパイラ・パッケージをインストールしているフォルダを表示します。	
	デフォルト	インストール・フォルダ名
	変更方法	変更不可
使用するコンパイラ・パッケージのバージョン	使用するコンパイラ・パッケージのバージョンを選択します。 この設定はすべてのビルド・モードで共通です。	
	デフォルト	常にインストール済みの最新版
	変更方法	ドロップダウン・リストによる選択
	指定可能値	常にインストール済みの最新版 インストールしているコンパイラ・パッケージの内、最新バージョンを使用します。
		インストールしているコンパイラ・パッケージのバージョン 選択したバージョンのコンパイラ・パッケージを使用します。

インストール済みのコンパイラ・パッケージの最新バージョン	[使用するコンパイラ・パッケージのバージョン] プロパティで [常にインストール済みの最新版] を選択した場合に使用するコンパイラ・パッケージのバージョンを表示します。 この設定はすべてのビルド・モードで共通です。 なお、本プロパティは、[使用するコンパイラ・パッケージのバージョン] プロパティで [常にインストール済みの最新版] を選択した場合のみ表示します。	
	デフォルト	インストールしているコンパイラ・パッケージの最新バージョン
	変更方法	変更不可

(13) [記録]

記録に関する詳細情報の表示、および設定の変更を行います。

メモ	このビルド・ツールにメモを追加します。 1 行に 1 項目ずつ指定します。 この設定はすべてのビルド・モードで共通です。 追加したメモはサブプロパティとして表示します。	
	デフォルト	メモ [項目数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

(14) [その他]

ビルド・ツールに関するその他の詳細情報の表示、および設定の変更を行います。

出力メッセージ・フォーマット	ビルド中のメッセージのフォーマットを指定します。 対象となるのは、使用するビルド・ツール、およびプラグインによって追加されたコマンドの出力メッセージです。 [ビルド前に実行するコマンド] プロパティ、および [ビルド後に実行するコマンド] プロパティなどで指定したコマンドの出力メッセージは対象外です。 次のプレースホルダに対応しています。 %Program% : 実行中のプログラム名に置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %TargetFiles% : コンパイル／アセンブル中のファイル名、またはリンク後の出力ファイル名に置換します。 空欄の場合は、%Program% %Options% を自動的に設定します。		
	デフォルト	%TargetFiles%	
	変更方法	テキスト・ボックスによる直接入力（256 文字までの文字列）、またはドロップダウン・リストによる選択	
	指定可能値	%TargetFiles%	出力メッセージにファイル名を表示します。
		%TargetFiles%: %Options%	出力メッセージにファイル名とコマンド・ライン・オプションを表示します。
		%Program% %Options%	出力メッセージにプログラム名とコマンド・ライン・オプションを表示します。
ビルド・オプション一覧表示フォーマット	ビルド・オプション一覧（「2. 11. 4 ビルド・オプションを一覧表示する」参照）の表示フォーマットを指定します。 対象となるのは、使用するビルド・ツール、およびプラグインによって追加されたコマンドのオプションです。 [ビルド前に実行するコマンド] プロパティ、および [ビルド後に実行するコマンド] プロパティなどで指定したコマンドのオプションは対象外です。 次のプレースホルダに対応しています。 %Program% : 実行中のプログラム名に置換します。 %Options% : ビルド時のコマンド・ライン・オプションに置換します。 %TargetFiles% : コンパイル／アセンブル中のファイル名、またはリンク後の出力ファイル名に置換します。 空欄の場合は、%TargetFiles% : %Program% %Options% を自動的に設定します。		
	デフォルト	%TargetFiles% : %Program% %Options%	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力ダイアログ による編集	
	指定可能値	256 文字までの文字列	

ビルド前に実行するコマンド	ビルド処理前に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2行目から最終行までの内容を Python コンソールのスクリプトと判断し、ビルド処理前に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。
デフォルト	ビルド前に実行するコマンド [定義数]
変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
指定可能値	1023 文字までの文字列 64 個まで指定可能です。

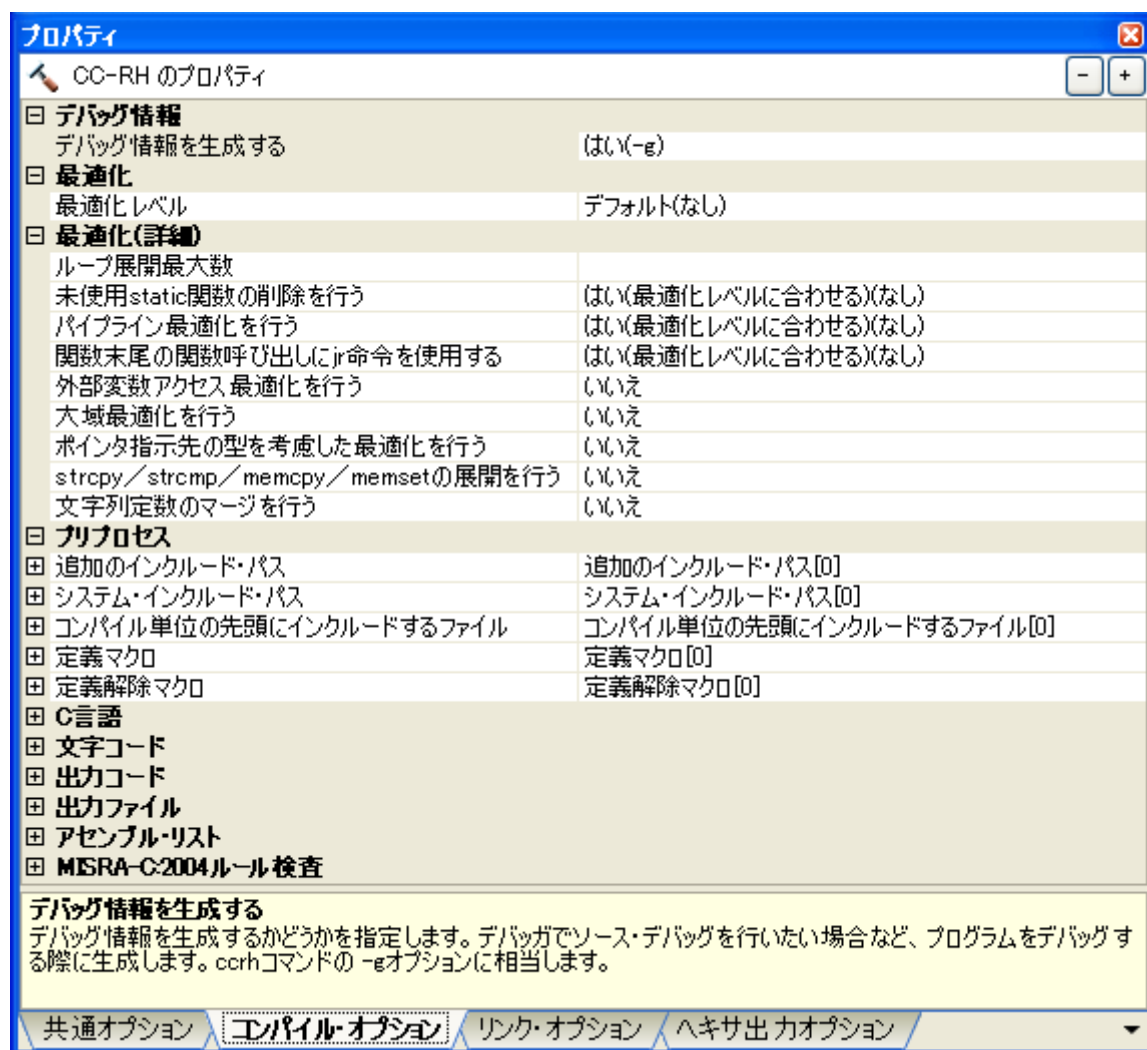
ビルド後に実行するコマンド	ビルド処理後に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、ビルド処理後に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。	
	デフォルト	ビルド後に実行するコマンド [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	1023 文字までの文字列 64 個まで指定可能です。
その他の追加オプション	その他に追加するオプションを入力します。 なお、ここで設定したオプションは、ccrh のオプション群の最後に付加します。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集
	指定可能値	259 文字までの文字列

[コンパイル・オプション] タブ

本タブでは、コンパイル・フェーズに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

- (1) [デバッグ情報]
- (2) [最適化]
- (3) [最適化 (詳細)]
- (4) [プリプロセス]
- (5) [C 言語]
- (6) [文字コード]
- (7) [出力コード]
- (8) [出力ファイル]
- (9) [アセンブル・リスト]
- (10) [MISRA-C:2004 ルール検査]
- (11) [その他]

図 A—5 プロパティ パネル：[コンパイル・オプション] タブ



[各カテゴリの説明]

(1) [デバッグ情報]

デバッグ情報に関する詳細情報の表示、および設定の変更を行います。

デバッグ情報を生成する	デバッグ情報を生成するかどうかを選択します。 出力ファイル中にソース・デバッグ用の情報を出力することにより、デバッガでのソース・デバッグが可能となります。 ccrh コマンドの -g オプションに相当します。		
	デフォルト	はい (-g)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-g)	デバッグ情報を生成します。
		いいえ	デバッグ情報を生成しません。

(2) [最適化]

最適化に関する詳細情報の表示、および設定の変更を行います。

最適化レベル	コンパイルの最適化レベルを選択します。 ccrh コマンドの -O オプションに相当します。		
	デフォルト	デフォルト (なし)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	デフォルト (なし)	デバッグに影響しない範囲の最適化 (式の最適化、およびレジスタ割り付けなど) を行います。
		サイズ優先 (-Osize)	オブジェクト・サイズ優先の最適化を行います。 ROM/RAM 容量の削減を重視して、一般的なプログラムに対して有効な最大限の最適化を行います。
		実行速度優先 (-Ospeed)	実行速度優先の最適化を行います。 実行速度の短縮を重視して、一般的なプログラムに対して有効な最大限の最適化を行います。
デバッグ優先 (-Onothing)		デバッグを優先して最適化を行います。 デバッグのしやすさを重視し、デフォルトで実行する最適化を含むすべての最適化を抑止します。	

(3) [最適化 (詳細)]

最適化に関する詳細情報の表示、および設定の変更を行います。

ループ展開最大数	for、while などのループを展開する最大数を指定します。 0、または 1 を指定した場合は、展開を抑止します。 空欄の場合は、4 を指定したものとみなします。 ccrh コマンドの -Ounroll オプションに相当します。		
	デフォルト	空欄	
	変更方法	テキスト・ボックスによる直接入力	
	指定可能値	0 ～ 999 (10 進数)、または空欄	
未使用 static 関数の削除 を行う	呼び出されない static 関数の削除を行うかどうかを選択します。 ccrh コマンドの -Odelete_static_func オプションに相当します。		
	デフォルト	はい (最適化レベルに合わせる) (なし)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (最適化レベルに合わせる) (なし)	[最適化レベル] プロパティにあわせて最適化を行います。
		はい (-Odelete_static_func)	呼び出されない static 関数の削除を行います。
		いいえ (-Odelete_static_func=off)	呼び出されない static 関数の削除を行いません。

関数のインライン展開を行う	関数を呼び出し箇所にインライン展開するかどうかを選択します。 ccrh コマンドの -Oinline オプションに相当します。 なお、本プロパティは、[最適化レベル] プロパティで [サイズ優先 (-Osize)]、または [実行速度優先 (-Ospeed)] を選択した場合のみ表示します。		
	デフォルト	はい (最適化レベルに合わせる) (なし)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (最適化レベルに合わせる) (なし)	[最適化レベル] プロパティにあわせて最適化を行います。
		はい (指定関数のみ) (-Oinline=1)	#pragma inline 指定した関数を呼び出し箇所にインライン展開します。
		はい (自動判別) (-Oinline=2)	自動的にインライン展開対象の関数を判別して展開します。
はい (コード・サイズが増加しないよう自動判別) (-Oinline=3)		コード・サイズがなるべく増加しない範囲で、自動的にインライン展開対象の関数を判別して展開します。	
	いいえ (-Oinline=0)	#pragma inline 指定した関数を含めて、すべてのインライン展開を抑止します。	
インライン展開サイズの最大増加率	コード・サイズが何%増加するまでインライン展開を行うかを指定します (例: “100” を指定した場合、コード・サイズが 100% (2 倍) 増加するまでインライン展開を行います)。 ccrh コマンドの -Oinline_size オプションに相当します。 なお、本プロパティは、[関数のインライン展開を行う] プロパティで [はい (自動判別) (-Oinline=2)] を選択した場合、または [関数のインライン展開を行う] プロパティで [はい (最適化レベルに合わせる)] かつ [最適化レベル] プロパティで [実行速度優先 (-Ospeed)] を選択した場合のみ表示します。		
	デフォルト	100	
	変更方法	テキスト・ボックスによる直接入力	
	指定可能値	0 ~ 65535 (10 進数)	
	パイプライン最適化を行う	機械語レベルで命令の並べ替えを行い、プログラムの実行性能を引き出すかどうかを選択します。 ccrh コマンドの -Opipeline オプションに相当します。	
デフォルト		はい (最適化レベルに合わせる) (なし)	
変更方法		ドロップダウン・リストによる選択	
指定可能値		はい (最適化レベルに合わせる) (なし)	[最適化レベル] プロパティにあわせて最適化を行います。
		はい (-Opipeline)	パイプライン最適化を行います。
	いいえ (-Opipeline=off)	パイプライン最適化を行いません。	

関数末尾の関数呼び出しに jr 命令を使用する	関数の末尾が関数呼び出しの場合に、jarl 命令の代わりに jr 命令を優先的に使用するかどうかを選択します。 ccrh コマンドの -Otail_call オプションに相当します。	
	デフォルト	はい (最適化レベルに合わせる) (なし)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (最適化レベルに合わせる) (なし) [最適化レベル] プロパティにあわせて最適化を行います。
		はい (-Otail_call) 関数の末尾が関数呼び出しの場合に、jarl 命令の代わりに jr 命令を優先的に使用します。 lp の退避／復帰コードを削除し、コード・サイズを小さくすることができます。 ただし、一部のデバッグ機能を使用することができなくなります。
外部変数アクセス最適化を行う	外部変数アクセス最適化を行うかどうかを選択します。 ccrh コマンドの -Osmap 、および -Omap オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (モジュール内で最適化) (-Osmap) コンパイル対象ファイル内で定義された外部変数、または静的変数についてベース・アドレスを設定し、アクセスをベース・アドレス相対で行うコードを生成します。
		はい (モジュール間で最適化) (-Omap) 外部シンボル割り付け情報ファイルを生成します。 その情報を元に、再コンパイルを行い、外部変数、または静的変数のアクセスをベース・アドレス相対で行うコードを生成します。 なお、本項目は、 [共通オプション] タブの [ビルド方法] カテゴリの「一括ビルドを行う」プロパティで「いいえ」を選択した場合のみ表示されます。 また、本項目を選択しているときに「一括ビルドを行う」プロパティで「はい」を選択した場合、本項目は「いいえ」に変更されます。
	いいえ 外部変数アクセス最適化を行いません。	

大域最適化を行う	大域最適化（関数の統合など）を行うレベルを指定します。 なお、 [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [いいえ] を選択した場合は [はい (レベル 1)(最適化を行う)](-Xintermodule) 、および [いいえ] のみ表示します。 ccrh コマンドの -Xwhole_program 、 -Xmerge_files 、および -Xintermodule オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (レベル 3)(入力ソースがプログラム全体として最適化する)(-Xwhole_program) ソース・ファイルがプログラム全体であることを仮定して、大域最適化を行います。 ただし、前提条件を満たさなかった場合の動作は保証されません。 前提条件の詳細については、「 -Xwhole_program 」を参照してください。
		はい (レベル 2)(複数ファイルをマージして最適化する)(-Xmerge_files, -Xintermodule) 複数の C ソース・ファイルをマージした上で、大域最適化を行います。 なお、本項目は、プロジェクトに 2 個、またはそれ以上のソース・ファイルを追加している場合のみ表示します。
		はい (レベル 1)(最適化を行う)(-Xintermodule) ソース・ファイルごとに、大域最適化を行います。
		いいえ 大域最適化を行いません。
ポインタ指示先の型を考慮した最適化を行う	ANSI 規格に基づくポインタ指示先の型を考慮した最適化を行うかどうかを選択します。 ccrh コマンドの -Xalias オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xalias=ansi) ポインタ指示先の型を考慮した最適化を行います。 一般には、オブジェクト性能が向上しますが、 [いいえ] を選択した場合と比べて実行結果が異なる場合があります。
		いいえ ポインタ指示先の型を考慮した最適化を行いません。

strcpy / strcmp / memcpy / memset の展 開を行う	配列（文字列を含む）、および構造体の整列条件を4バイトとし、関数 strcpy(), strcmp(), memcpy(), memset() の呼び出しをインライン展開するかどうかを選択します。 生成するプログラムの実行速度は高速になりますが、コード・サイズは増大します。 ccrh コマンドの -Xinline_strcpy オプションに相当します。 なお、本プロパティは、 [出力コード] カテゴリの「構造体パッキングを行う」プロパティで「いいえ」を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xinline_strcpy) 関数 strcpy(), strcmp(), memcpy(), memset() の呼び出しをインライン展開します。
		いいえ 関数 strcpy(), strcmp(), memcpy(), memset() のインライン展開を行いません。
文字列定数のマージを行 う	ソース・ファイル内で同じ文字列定数が複数存在する場合、これらをまとめて1つの領域に割り付けるかどうかを選択します。 ccrh コマンドの -Xmerge_string オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xmerge_string) ソース・ファイル内で複数存在する同じ文字列定数をまとめて1つの領域に割り付けます。
		いいえ ソース・ファイル内で複数存在する同じ文字列定数をそれぞれ別々の領域に割り付けます。

(4) [プリプロセス]

プリプロセスに関する詳細情報の表示、および設定の変更を行います。

追加のインクルード・パス	コンパイル時の追加のインクルード・パスを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 指定したインクルード・パスは、CC-RH の標準インクルード・ファイル・フォルダよりも優先して検索します。 パスはプロジェクト・フォルダを基点とします。 本プロパティを省略した場合は、CC-RH の標準インクルード・ファイル・フォルダのみ検索します。 ccrh コマンドの -I オプションに相当します。 指定したインクルード・パスはサブプロパティとして表示します。 なお、プロジェクト・ツリーにインクルード・ファイルを追加すると、インクルード・パスをサブプロパティの一番最初に追加します。 インクルード・パスに大文字、小文字の区別はありません。	
	デフォルト	追加のインクルード・パス [定義数]
	変更方法	[...] ボタンをクリックし、 パス編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	259 文字までの文字列 256 個まで指定可能です。

システム・インクルード・パス	コンパイル時にシステムが設定するインクルード・パスの指定順を変更します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 システム・インクルード・パスは、追加のインクルード・パスより低い優先度で検索します。 パスはプロジェクト・フォルダを基点とします。 ccrh コマンドの -i オプションに相当します。 インクルード・パスはサブプロパティとして表示します。	
	デフォルト	システム・インクルード・パス [定義数]
	変更方法	[...] ボタンをクリックし、 システム・インクルード・パス順設定 ダイアログ による編集
	指定可能値	変更不可（インクルード・パスの設定順の変更のみ可能）
コンパイル単位の前頭にインクルードするファイル	コンパイル単位の前頭にインクルードするファイルを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 なお、パスはプロジェクト・フォルダを基点とします。 ccrh コマンドの -Xpreinclude オプションに相当します。 指定したインクルード・ファイル名はサブプロパティとして表示します。	
	デフォルト	コンパイル単位の前頭にインクルードするファイル [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	247 文字までの文字列 256 個まで指定可能です。

定義マクロ	定義したいマクロ名を指定します。 「マクロ名 = 定義値」の形式で 1 行に 1 つずつ指定します。 「= 定義値」の部分は省略可能で、省略した場合、定義値を 1 とします。 ccrh コマンドの -D オプションに相当します。 指定したマクロはサブプロパティとして表示します。				
	デフォルト	定義マクロ [定義数]			
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能			
	指定可能値	256 文字までの文字列 256 個まで指定可能です。			
定義解除マクロ	定義解除したいマクロ名を指定します。 「マクロ名」の形式で 1 行に 1 つずつ指定します。 ccrh コマンドの -U オプションに相当します。 指定したマクロはサブプロパティとして表示します。				
	デフォルト	定義解除マクロ [定義数]			
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能			
	指定可能値	256 文字までの文字列 256 個まで指定可能です。			
プリプロセス結果に C ソース・コメントを出力 する	プリプロセス結果のファイルに、C ソースのコメントを出力するかどうかを選択します。 ccrh コマンドの -Xpreprocess オプションに相当します。 なお、本プロパティは、 [出力ファイル] カテゴリの [プリプロセス処理したソースを出力する] プロパティで [はい (-P)] を選択した場合のみ表示します。				
	デフォルト	いいえ			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-Xpreprocess=comment)</td><td>プリプロセス結果のファイルに、C ソースのコメントを出力します。</td></tr><tr><td>いいえ</td><td>プリプロセス結果のファイルに、C ソースのコメントを出力しません。</td></tr></table>	はい (-Xpreprocess=comment)	プリプロセス結果のファイルに、C ソースのコメントを出力します。	いいえ
はい (-Xpreprocess=comment)	プリプロセス結果のファイルに、C ソースのコメントを出力します。				
いいえ	プリプロセス結果のファイルに、C ソースのコメントを出力しません。				
プリプロセス結果に行番 号情報を出力する	プリプロセス結果のファイルに、C ソースの行番号情報を出力するかどうかを選択します。 ccrh コマンドの -Xpreprocess オプションに相当します。 なお、本プロパティは、 [出力ファイル] カテゴリの [プリプロセス処理したソースを出力する] プロパティで [はい (-P)] を選択した場合のみ表示します。				
	デフォルト	いいえ			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-Xpreprocess=line)</td><td>プリプロセス結果のファイルに、C ソースの行番号情報を出力します。</td></tr><tr><td>いいえ</td><td>プリプロセス結果のファイルに、C ソースの行番号情報を出力しません。</td></tr></table>	はい (-Xpreprocess=line)	プリプロセス結果のファイルに、C ソースの行番号情報を出力します。	いいえ
はい (-Xpreprocess=line)	プリプロセス結果のファイルに、C ソースの行番号情報を出力します。				
いいえ	プリプロセス結果のファイルに、C ソースの行番号情報を出力しません。				

(5) [C 言語]

C 言語に関する詳細情報の表示、および設定の変更を行います。

ANSI 規格に厳密に合わせてコンパイルする	C ソース・プログラムを ANSI 規格に厳密にあわせて処理し、規格に反する記述に対してエラーや警告を出力するかどうかを選択します。 ccrh コマンドの -Xansi オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xansi) C ソース・プログラムを ANSI 規格に厳密にあわせて処理し、規格に反する記述に対してエラーや警告を出力します。
		いいえ 従来の C 言語の仕様との両立性を持たせ、警告を出力して処理を続行します。
enum の型	列挙型に対して、どの整数型として扱うかを選択します。 ccrh コマンドの -Xenum_type オプションに相当します。	
	デフォルト	signed int(なし)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	signed int(なし) 列挙型に対して、int 型として扱います。
		自動 (-Xenum_type=auto) 各列挙型について、その型のすべての列挙子の値を表現可能な最小の整数型として扱います。
外部変数を volatile 化する	すべての外部変数を volatile 宣言したものとして扱うかどうかを選択します。 ccrh コマンドの -Xvolatile オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xvolatile) すべての外部変数を volatile 宣言したものとして扱います。
		いいえ volatile 修飾のある変数のみを volatile 宣言したものとして扱います。
C プログラムの互換性をチェックする	C プログラムの互換性をチェックするかどうかを選択します。 ccrh コマンドの -Xcheck オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (SuperH RISC engine C/C++ コンパイラ用) (-Xcheck=shc) SuperH ファミリー用 C/C++ コンパイラとの互換性をチェックします。
		いいえ 既存プログラムとの互換性をチェックしません。

(6) [文字コード]

文字コードに関する詳細情報の表示、および設定の変更を行います。

文字コード	ソース・ファイル中の日本語／中国語のコメント、文字列に対して、使用する文字コードを選択します。		
	ccrh コマンドの -Xcharacter_set オプションに相当します。		
	デフォルト	自動 (なし)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	自動 (なし)	ソース・ファイル中の日本語の文字コードを SJIS と解釈します。
		SJIS(-Xcharacter_set=sjis)	ソース・ファイル中の日本語の文字コードを SJIS と解釈します。
		EUC(-Xcharacter_set=euc_jp)	ソース・ファイル中の日本語の文字コードを EUC と解釈します。
		UFT-8(-Xcharacter_set=utf8)	ソース・ファイル中の日本語の文字コードを UFT-8 と解釈します。
		Big5(-Xcharacter_set=big5)	ソース・ファイル中の中国語の文字コードを繁体字中国語と解釈します。
GB2312(-Xcharacter_set=gb2312)		ソース・ファイル中の中国語の文字コードを簡体字中国語と解釈します。	
処理しない (-Xcharacter_set=none)		ソース・ファイル中の日本語／中国語の文字コードを解釈しません。	

(7) [出力コード]

出力コードに関する詳細情報の表示、および設定の変更を行います。

構造体パッキングを行う

構造体パッキングを行うかどうかを選択します。 構造体メンバをメンバ型に応じてアライメントすることなく、指定したアライメント値を用いることができます。 ccrh コマンドの -Xpack オプションに相当します。		
デフォルト	いいえ	
変更方法	ドロップダウン・リストによる選択	
指定可能値	1 バイト (-Xpack=1)	構造体メンバを 1 バイトのアライメントで整列します。
	2 バイト (-Xpack=2)	構造体メンバを 2 バイトのアライメントで整列します。
	4 バイト (-Xpack=4)	構造体メンバを 4 バイトのアライメントで整列します。
	いいえ	構造体パッキングを行いません。

ビット・フィールド・メンバの並び順	ビット・フィールド・メンバの並び順を選択します。 ccrh コマンドの -Xbit_order オプションに相当します。			
	デフォルト	右から割り付け (なし)		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	左から割り付け (-Xbit_order=left)	上位ビットからメンバを割り付けます。	
右から割り付け (なし)		下位ビットからメンバを割り付けます。		
アセンブリ・ソース・ファイルにコメントを出力する	出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力するかどうかを選択します。 ccrh コマンドの -Xpass_source オプションに相当します。 なお、本プロパティは、 [出力ファイル] カテゴリの [アセンブリ・ソース・ファイルを出力する] プロパティで [はい (-Xasm_path)] を選択した場合、または [アセンブル・リスト] カテゴリの [アセンブル・リスト・ファイルを出力する] プロパティで [はい (-Xasm_option=-Xprn_path)] を選択した場合のみ表示します。			
	デフォルト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (-Xpass_source)	出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力します。	
		いいえ	出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力しません。	

switch 文の出力コードの 選択	プログラム中の switch 文のコード出力形式を選択します。 ccrh コマンドの -Xswitch オプションに相当します。		
	デフォルト	自動選択 (なし)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	自動選択 (なし)	ccrh コマンドが最適な出力形式を自動的に選択します。
		if-else(-Xswitch=ifelse)	プログラム中の switch 文を case 文の並びに沿った if-else 文と同じ形式で出力します。 頻度が多い順に case 文を書いているときやラベル数が少ないときに、本項目を選択します。 上から順に比較するので、頻繁に合致する case 文を先に記述すると余計な比較が減り、実行速度向上につながります。
バイナリ・サーチ (-Xswitch=binary)		プログラム中の switch 文をバイナリ・サーチ形式で出力します。 バイナリ・サーチ・アルゴリズムに用いて合致する case 文を探します。 ラベル数が多いときに本項目を選択すると、どの case 文も同じくらいの速さで見つけることができます。	
	テーブル分岐 (-Xswitch=table)	プログラム中の switch 文をテーブル・ジャンプ形式で出力します。 case 文の値を基にインデックス化したテーブルを参照し、switch 文の値により case ラベルを選択して処理を行います。 どの case 文にも同じくらい速く分岐します。 ただし、case 値が連続していないときは無駄な領域ができます。	
浮動小数点演算方法	浮動小数点演算に対して、ランタイム・ライブラリの呼び出し命令を生成するか、FPU (浮動小数点ユニット) の浮動小数点演算命令を生成するかを選択します。 ccrh コマンドの -Xfloat オプションに相当します。 なお、本プロパティは、 【共通オプション】 タブの 【出力ファイルの種類と場所】 カテゴリの 【CPU コアの指定】 プロパティで 【G3M 向けオブジェクト (-Xcpu=g3m)】 を選択した場合のみ表示します。		
	デフォルト	自動選択 (なし)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	自動選択 (なし)	浮動小数点演算命令を生成します。
		ソフトウェアで行う (-Xfloat=soft)	浮動小数点演算に対して、ランタイム・ライブラリの呼び出し命令を生成します。
FPU で行う (-Xfloat=fpu)		浮動小数点演算に対して、FPU の浮動小数点演算命令を生成します。	

div / divu 除算命令を生成する	除算に対して、divq、および divqu 命令を生成する代わりに、div、および divu 命令を生成するかどうかを選択します。 divq、および divqu 命令は高速ですが、実行サイクル数がオペランドの値により変わります。 ccrh コマンドの -Xdiv オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xdiv) 除算に対して、div、および divu 命令を生成します。
		いいえ 除算に対して、divq、および divqu 命令を生成します。
除算時の OV フラグ・チェック・コードを生成する	除算命令の後に OV フラグのチェックを行い、OV フラグが 1 のときに FE レベル・ソフトウェア例外を発生させるコード (fetrp 命令) を生成するかどうかを選択します。 ccrh コマンドの -Xcheck_div_ov オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xcheck_div_ov) 除算時に OV フラグのチェックを行うコードを生成します。
		いいえ 除算時に OV フラグのチェックを行わないコードを生成します。
除算時例外の fetrap 命令ベクタ番号	OV フラグが 1 のときに生成する fetrap 命令のベクタ番号を指定します。 ccrh コマンドの -Xcheck_div_ov オプションに相当します。 なお、本プロパティは、[除算時の OV フラグ・チェック・コードを生成する] プロパティで [はい (-Xcheck_div_ov)] を選択した場合のみ表示します。	
	デフォルト	1
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	1 ~ 15 (10 進数)
積和演算命令を生成する	単精度浮動小数点積和演算に対して、積和演算命令 (fmaf.s, fmsf.s, fnmaf.s, fnmsf.s) を生成するかどうかを選択します。 ccrh コマンドの -Xuse_fmaf オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xuse_fmaf) 単精度浮動小数点積和演算に対して、積和演算命令を生成します。 実行速度は上がりますが、演算精度が変わります。
		いいえ 積和演算命令を生成しません。

浮動小数点比較で無効演算例外を発生させる	浮動小数点比較において、比較する値に非数が含まれている場合に無効演算例外を発生する比較条件を使用するコードを生成するかどうかを選択します。 ccrh コマンドの -Xunordered_cmpf オプションに相当します。				
	デフォルト	いいえ			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-Xunordered_cmpf)</td><td>浮動小数点比較において、比較する値に非数が含まれている場合に無効演算例外を発生する比較条件を使用するコードを生成します。</td></tr><tr><td>いいえ</td><td>浮動小数点比較において、無効演算例外の検出を行いません。</td></tr></table>	はい (-Xunordered_cmpf)	浮動小数点比較において、比較する値に非数が含まれている場合に無効演算例外を発生する比較条件を使用するコードを生成します。	いいえ
はい (-Xunordered_cmpf)	浮動小数点比較において、比較する値に非数が含まれている場合に無効演算例外を発生する比較条件を使用するコードを生成します。				
いいえ	浮動小数点比較において、無効演算例外の検出を行いません。				
32 ビット分岐命令を使用する	jarl、および jr 命令に対して、far jump 機能を使用するかどうかを選択します。 far jump 機能を使用することにより、jarl、および jr 命令を jarl32、および jr32 命令とみなしてアセンブルを行います。 ccrh コマンドの -Xasm_option=-Xasm_far_jump オプションに相当します。 なお、本プロパティは、 【共通オプション】タブの【ビルド方法】 カテゴリの 【一括ビルドを行う】 プロパティで 【はい】 を選択した場合のみ表示します。				
	デフォルト	いいえ			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-Xasm_option=-Xasm_far_jump)</td><td>jarl、および jr 命令を jarl32、および jr32 命令とみなしてアセンブルを行います。</td></tr><tr><td>いいえ</td><td>jarl、および jr 命令としてアセンブルを行います。</td></tr></table>	はい (-Xasm_option=-Xasm_far_jump)	jarl、および jr 命令を jarl32、および jr32 命令とみなしてアセンブルを行います。	いいえ
はい (-Xasm_option=-Xasm_far_jump)	jarl、および jr 命令を jarl32、および jr32 命令とみなしてアセンブルを行います。				
いいえ	jarl、および jr 命令としてアセンブルを行います。				
関数呼び出し命令を指定する	関数呼び出しの分岐に対して、生成する命令を選択します。 ccrh コマンドの -Xcall_jump オプションに相当します。				
	デフォルト	jarl および jr 命令を生成 (なし)			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>jarl32 および jr32 命令を生成 (-Xcall_jump=32)</td><td>関数呼び出しの分岐に対して、jarl32、および jr32 命令を生成します。</td></tr><tr><td>jarl および jr 命令を生成 (なし)</td><td>関数呼び出しの分岐に対して、jarl、および jr 命令を生成します。</td></tr></table>	jarl32 および jr32 命令を生成 (-Xcall_jump=32)	関数呼び出しの分岐に対して、jarl32、および jr32 命令を生成します。	jarl および jr 命令を生成 (なし)
jarl32 および jr32 命令を生成 (-Xcall_jump=32)	関数呼び出しの分岐に対して、jarl32、および jr32 命令を生成します。				
jarl および jr 命令を生成 (なし)	関数呼び出しの分岐に対して、jarl、および jr 命令を生成します。				
Far Jump ファイル名	Far Jump ファイル名を指定します。 Far Jump ファイルには、ファイルに記述した関数への分岐命令に対して、jarl32、および jr32 命令を使用したコードを出力します。 関数本体が jarl、および jr 命令では分岐できない範囲 (± 2M バイト以上) にあり、ccrh コマンドがエラーを出力する場合、Far Jump ファイルを用いてコンパイルし直します。 なお、拡張子は “.fjp” としてください。 ccrh コマンドの -Xfar_jump オプションに相当します。				
	デフォルト	Far Jump ファイル名			
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、Far Jump ファイルを指定 ダイアログ による編集			
	指定可能値	259 文字までの文字列			

生成するプログラムの種類	生成するプログラムの種類を選択します。 ccrh コマンドの -Xmulti_level オプションに相当します。 なお、本プロパティは、マルチコア用プロジェクトの場合のみ表示します。		
	デフォルト	マルチコア用のプログラムを生成する (-Xmulti_level=1)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	シングルコア用のプログラムを生成する (なし)	シングルコア用プログラムを生成します。 プログラム中の #pragma pmodule 指定は無視します。
		マルチコア用のプログラムを生成する (-Xmulti_level=1)	マルチコア用プログラムを生成します。 プログラム中の #pragma pmodule 指定が有効となり、セクション名の末尾に PM 番号が付加されます。

(8) [出力ファイル]

出力ファイルに関する詳細情報の表示、および設定の変更を行います。

アセンブリ・ソース・ファイルを出力する	C ソースのコンパイル結果のアセンブリ・ソース・ファイルを出力するかどうかを選択します。 ccrh コマンドの -Xasm_path オプションに相当します。				
	デフォルト	いいえ			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-Xasm_path)</td><td>C ソースのコンパイル結果のアセンブリ・ソース・ファイルを出力します。</td></tr><tr><td>いいえ</td><td>C ソースのコンパイル結果のアセンブリ・ソース・ファイルを出力しません。</td></tr></table>	はい (-Xasm_path)	C ソースのコンパイル結果のアセンブリ・ソース・ファイルを出力します。	いいえ
はい (-Xasm_path)	C ソースのコンパイル結果のアセンブリ・ソース・ファイルを出力します。				
いいえ	C ソースのコンパイル結果のアセンブリ・ソース・ファイルを出力しません。				
アセンブリ・ソース・ファイル出力フォルダ	アセンブリ・ソース・ファイルの出力先フォルダを指定します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% ： ビルド・モード名に置換します。 アセンブリ・ソース・ファイルは、C ソース・ファイルの拡張子を“.asm”で置き換えたファイル名で出力します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの -Xasm_path オプションに相当します。 なお、本プロパティは、[アセンブリ・ソース・ファイルを出力する] プロパティで [はい (-Xasm_path)] を選択した場合のみ表示します。				
	デフォルト	%BuildModeName%			
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集			
	指定可能値	247 文字までの文字列			

プリプロセス処理したソースを出力する	ソース・ファイルに対して、プリプロセス処理を実行した結果をファイルに出力するかどうかを選択します。 ccrh コマンドの -P オプションに相当します。				
	デフォルト	いいえ			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-P)</td><td>ソース・ファイルに対して、プリプロセス処理を実行した結果をファイルに出力します。</td></tr><tr><td>いいえ</td><td>ソース・ファイルに対して、プリプロセス処理を実行した結果をファイルに出力しません。</td></tr></table>	はい (-P)	ソース・ファイルに対して、プリプロセス処理を実行した結果をファイルに出力します。	いいえ
はい (-P)	ソース・ファイルに対して、プリプロセス処理を実行した結果をファイルに出力します。				
いいえ	ソース・ファイルに対して、プリプロセス処理を実行した結果をファイルに出力しません。				
プリプロセス処理したソース・ファイル出力フォルダ	プリプロセス処理したソース・ファイルの出力先フォルダを指定します。 ファイルは、ソース・ファイルの拡張子を“.i”で置き換えた名前で出力します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの -Xprep_path オプションに相当します。 なお、本プロパティは、[プリプロセス処理したソースを出力する] プロパティで [はい (-P)] を選択した場合のみ表示します。				
	デフォルト	%BuildModeName%			
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集			
	指定可能値	247 文字までの文字列			

(9) [アセンブル・リスト]

アセンブル・リストに関する詳細情報の表示、および設定の変更を行います。

アセンブル・リスト・ ファイルを出力する	アセンブル・リスト・ファイルを出力するかどうかを選択します。 ccrh コマンドの -Xasm_option=-Xprn_path オプションに相当します。				
	デフォルト	いいえ			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-Xasm_option=-Xprn_path)</td><td>アセンブル・リスト・ファイルを出力します。</td></tr><tr><td>いいえ</td><td>アセンブル・リスト・ファイルを出力しません。</td></tr></table>	はい (-Xasm_option=-Xprn_path)	アセンブル・リスト・ファイルを出力します。	いいえ
はい (-Xasm_option=-Xprn_path)	アセンブル・リスト・ファイルを出力します。				
いいえ	アセンブル・リスト・ファイルを出力しません。				

アセンブル・リスト・ ファイル出力フォルダ	アセンブル・リスト・ファイルの出力先フォルダを指定します。 アセンブル・リスト・ファイルは、ソース・ファイル名の拡張子を“.prn”で置き換えた名前でも出力します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの <code>-Xasm_option=-Xprn_path</code> オプションに相当します。 なお、本プロパティは、[アセンブル・リスト・ファイルを出力する] プロパティで [はい (-Xasm_option=-Xprn_path)] を選択した場合のみ表示します。	
	デフォルト	%BuildModeName%
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列

(10) [MISRA-C:2004 ルール検査]

MISRA-C:2004 ルール検査に関する詳細情報の表示、および設定の変更を行います。

適用するルール	適用する MISRA-C:2004 ルールを選択します。 ccrh コマンドの -Xmisra2004 オプションに相当します。		
	デフォルト	適用ルールなし (なし)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	すべてのルールを適用 (-Xmisra2004=all)	サポートしているすべてのルールをチェック対象とします。
		指定したルール番号を適用 (-Xmisra2004=apply)	サポートしているルールのうち、指定したルール番号をチェック対象とします。
		指定したルール番号を除外 (-Xmisra2004=ignore)	サポートしているルールのうち、指定したルール番号以外のルールをチェック対象とします。
		必須ルールを適用 (-Xmisra2004=required)	サポートしているルールのうち、ルールの分類が “required” になっているルールをチェック対象とします。
		必須ルールと指定したルールを適用 (-Xmisra2004=required_add)	サポートしているルールのうち、ルールの分類が “required” になっているルールと指定したルール番号をチェック対象とします。
		必須ルールから指定したルール番号を除外 (-Xmisra2004=required_remove)	サポートしているルールのうち、ルールの分類が “required” になっているルールから指定したルール番号を除いたルール番号をチェック対象とします。
指定ファイルに記載されたルール番号を適用 (-misra2004=<ファイル名>)		サポートしているルールのうち、指定したファイル名に記載したルール番号をチェック対象とします。	
適用ルールなし (なし)	MISRA-C:2004 ルールを適用しません。		
ルール番号記載ファイル	ルール番号記載ファイル（MISRA-C:2004 ルール・ファイル）を指定します。 次のプレースホルダに対応しています。 %BuildModeName% ： ビルド・モード名に置換します。 %MicomToolPath% ： 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectName% ： プロジェクト名に置換します。 ccrh コマンドの -Xmisra2004 オプションに相当します。 なお、本プロパティは、[適用するルール] プロパティで [指定ファイルに記載されたルール番号を適用 (-misra2004=<ファイル名>)] を選択した場合のみ表示します。		
	デフォルト	空欄	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 MISRA-C:2004 ルール・ファイルの指定 ダイアログ による編集	
	指定可能値	259 文字までの文字列	

ルール番号	チェック対象とするルール番号を指定します。 ルール番号は、必ず 10 進数で 1 つ以上指定してください。 ccrh コマンドの -Xmisra2004 オプションに相当します。 なお、本プロパティは、[適用するルール] プロパティで [指定したルール番号を適用 (-Xmisra2004=apply)] を選択した場合のみ表示します。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 ルール番号の指定 ダイアログ による編集
	指定可能値	259 文字までの文字列
除外するルール番号	チェック対象から除外するルール番号を指定します。 ルール番号は、必ず 10 進数で 1 つ以上指定してください。 ccrh コマンドの -Xmisra2004 オプションに相当します。 なお、本プロパティは、[適用するルール] プロパティで [指定したルール番号を除外 (-Xmisra2004=ignore)] を選択した場合のみ表示します。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 ルール番号の指定 ダイアログ による編集
	指定可能値	259 文字までの文字列
必須ルールの他にチェックするルール番号	必須ルールのほかにチェック対象とするルール番号を指定します。 ルール番号は、必ず 10 進数で 1 つ以上指定してください。 ccrh コマンドの -Xmisra2004 オプションに相当します。 なお、本プロパティは、[適用するルール] プロパティで [必須ルールと指定したルールを適用 (-Xmisra2004=required_add)] を選択した場合のみ表示します。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 ルール番号の指定 ダイアログ による編集
	指定可能値	259 文字までの文字列
必須ルールから除外するルール番号	必須ルールのうち、チェック対象から除外するルール番号を指定します。 ルール番号は、必ず 10 進数で 1 つ以上指定してください。 ccrh コマンドの -Xmisra2004 オプションに相当します。 なお、本プロパティは、[適用するルール] プロパティで [必須ルールから指定したルール番号を除外 (-Xmisra2004=required_remove)] を選択した場合のみ表示します。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 ルール番号の指定 ダイアログ による編集
	指定可能値	259 文字までの文字列

ルール・チェック対象外のファイル	MISRA-C:2004 ルールのチェック対象外のファイルを指定します。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 ccrh コマンドの <code>-Xignore_files_misra</code> オプションに相当します。 なお、本プロパティは、以下の場合のみ表示します。 - [適用するルール] プロパティで [すべてのルールを適用] を選択した場合 - [適用するルール] プロパティで [必須ルールを適用] を選択した場合 - [適用するルール] プロパティで [指定したルール番号を適用] を選択し、かつ [ルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [指定したルール番号を除外] を選択し、かつ [除外するルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [必須ルールと指定したルールを適用] を選択し、かつ [必須ルールの他にチェックするルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [必須ルールから指定したルール番号を除外] を選択し、かつ [必須ルールから除外するルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [指定ファイルに記載されたルール番号を適用] を選択し、かつ [ルール番号記載ファイル] プロパティにルール番号記載ファイルを指定した場合	
	デフォルト	ルール・チェック対象外のファイル [定義数]
	変更方法	[...] ボタンをクリックし、 パス編集 ダイアログ をオープン → [参照] ボタンをクリックし、 対象外ファイルの追加 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	259 文字までの文字列

拡張キーワードや拡張仕様をメッセージ出力する	拡張キーワードや拡張仕様をメッセージ出力するかどうかを選択します。 ccrh コマンドの <code>-Xcheck_language_extention</code> オプションに相当します。 なお、本プロパティは、以下の場合のみ表示します。 - [適用するルール] プロパティで [すべてのルールを適用] を選択した場合 - [適用するルール] プロパティで [必須ルールを適用] を選択した場合 - [適用するルール] プロパティで [指定したルール番号を適用] を選択し、かつ [ルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [指定したルール番号を除外] を選択し、かつ [除外するルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [必須ルールと指定したルールを適用] を選択し、かつ [必須ルールの他にチェックするルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [必須ルールから指定したルール番号を除外] を選択し、かつ [必須ルールから除外するルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [指定ファイルに記載されたルール番号を適用] を選択し、かつ [ルール番号記載ファイル] プロパティにルール番号記載ファイルを指定した場合	
デフォルト	いいえ	
変更方法	ドロップダウン・リストによる選択	
指定可能値	はい (-Xcheck_language_extension)	C 言語規格から独自に拡張した言語仕様のために MISRA-C:2004 ルール・チェックが部分抑止される場合にルール・チェックを有効にして、メッセージを出力します。
	いいえ	言語拡張により部分抑止される MISRA-C:2004 ルール・チェックを無効にします。

(11) [その他]

コンパイルに関するその他の詳細情報の表示、および設定の変更を行います。

なお、本カテゴリは、[\[共通オプション\]](#) タブの [\[ビルド方法\]](#) カテゴリの [一括ビルドを行う] プロパティで [いいえ] を選択した場合のみ表示します。

コンパイル前に実行する コマンド	コンパイル処理前に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %CompiledFile% : コンパイル時の出力ファイルの絶対パスに置換します。 %InputFile% : コンパイル対象ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、コンパイル処理前に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。
デフォルト	コンパイル前に実行するコマンド[定義数]
変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
指定可能値	1023 文字までの文字列 64 個まで指定可能です。

コンパイル後に実行する コマンド	コンパイル処理後に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %CompiledFile% : コンパイル時の出力ファイルの絶対パスに置換します。 %InputFile% : コンパイル対象ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、コンパイル処理後に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。						
	<table border="1"> <tr> <td data-bbox="536 1238 671 1272">デフォルト</td><td data-bbox="671 1238 1394 1272">コンパイル後に実行するコマンド [定義数]</td></tr> <tr> <td data-bbox="536 1283 671 1361">変更方法</td><td data-bbox="671 1283 1394 1361">[...] ボタンをクリックし、テキスト編集 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能</td></tr> <tr> <td data-bbox="536 1373 671 1451">指定可能値</td><td data-bbox="671 1373 1394 1451">1023 文字までの文字列 64 個まで指定可能です。</td></tr> </table>	デフォルト	コンパイル後に実行するコマンド [定義数]	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能	指定可能値	1023 文字までの文字列 64 個まで指定可能です。
デフォルト	コンパイル後に実行するコマンド [定義数]						
変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能						
指定可能値	1023 文字までの文字列 64 個まで指定可能です。						
その他の追加オプション	その他に追加するコンパイル・オプションを入力します。 ここで設定したオプションは、コンパイル・オプション群の最後に付加します。 <table border="1"> <tr> <td data-bbox="536 1552 671 1585">デフォルト</td><td data-bbox="671 1552 1394 1585">空欄</td></tr> <tr> <td data-bbox="536 1597 671 1675">変更方法</td><td data-bbox="671 1597 1394 1675">テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログによる編集</td></tr> <tr> <td data-bbox="536 1686 671 1722">指定可能値</td><td data-bbox="671 1686 1394 1722">259 文字までの文字列</td></tr> </table>	デフォルト	空欄	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集	指定可能値	259 文字までの文字列
デフォルト	空欄						
変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集						
指定可能値	259 文字までの文字列						

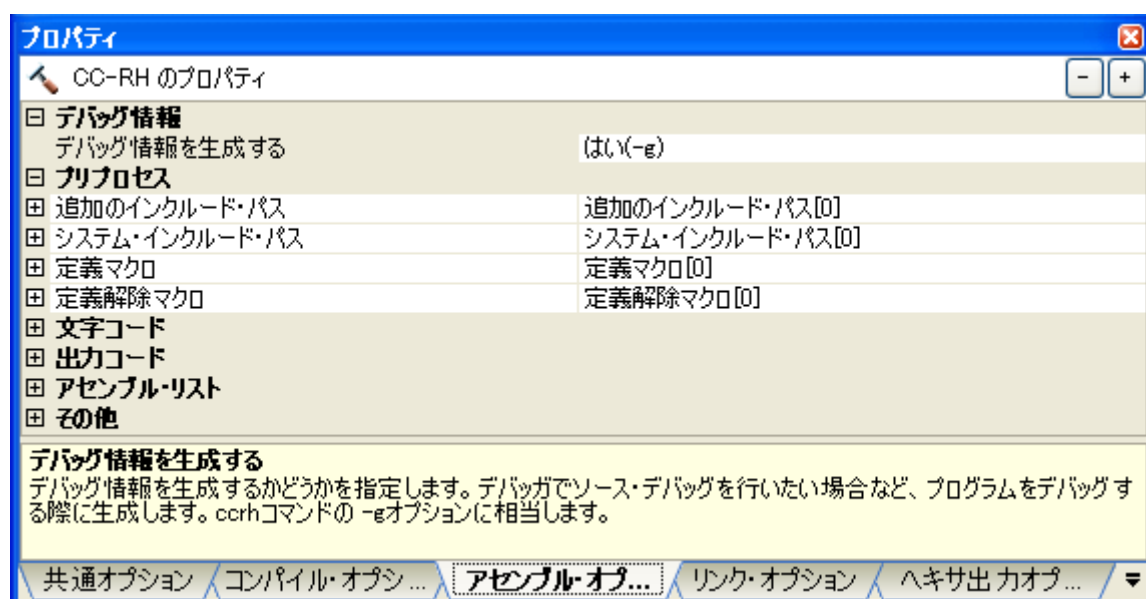
[アセンブル・オプション] タブ

本タブでは、アセンブル・フェーズに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

- (1) [デバッグ情報]
- (2) [プリプロセス]
- (3) [文字コード]
- (4) [出力コード]
- (5) [アセンブル・リスト]
- (6) [その他]

注意 本タブは、[共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [いいえ] を選択した場合のみ表示します。

図 A—6 プロパティ パネル：[アセンブル・オプション] タブ



[各カテゴリの説明]

(1) [デバッグ情報]

デバッグ情報に関する詳細情報の表示、および設定の変更を行います。

デバッグ情報を生成する	デバッグ情報を生成するかどうかを選択します。 出力ファイル中にソース・デバッグ用の情報を出力することにより、デバッガでのソース・デバッグが可能となります。 ccrh コマンドの -g オプションに相当します。	
	デフォルト	はい (-g)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-g) デバッグ情報を生成します。 いいえ デバッグ情報を生成しません。

(2) [プリプロセス]

プリプロセスに関する詳細情報の表示、および設定の変更を行います。

追加のインクルード・パス	アセンブル時の追加のインクルード・パスを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 指定したインクルード・パスは、CC-RH の標準インクルード・ファイル・フォルダよりも優先して検索します。 パスはプロジェクト・フォルダを基点とします。 本プロパティを省略した場合は、CC-RH の標準インクルード・ファイル・フォルダのみ検索します。 ccrh コマンドの -I オプションに相当します。 指定したインクルード・パスはサブプロパティとして表示します。 なお、プロジェクト・ツリーにインクルード・ファイルを追加すると、インクルード・パスをサブプロパティの一番最初に追加します。 インクルード・パスに大文字、小文字の区別はありません。	
	デフォルト	追加のインクルード・パス [定義数]
	変更方法	[...] ボタンをクリックし、 パス編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	259 文字までの文字列 256 個まで指定可能です。

システム・インクルード・パス	アセンブル時にシステムが設定するインクルード・パスの指定順を変更します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 システム・インクルード・パスは、追加のインクルード・パスより低い優先度で検索します。 パスはプロジェクト・フォルダを基点とします。 ccrh コマンドの -I オプションに相当します。 インクルード・パスはサブプロパティとして表示します。	
	デフォルト	システム・インクルード・パス [定義数]
	変更方法	[...] ボタンをクリックし、 システム・インクルード・パス順設定 ダイアログ による編集
	指定可能値	変更不可（インクルード・パスの設定順の変更のみ可能）
定義マクロ	定義したいマクロ名を指定します。 「マクロ名 = 定義値」の形式で 1 行に 1 つずつ指定します。 「= 定義値」の部分は省略可能で、省略した場合、定義値を 1 とします。 ccrh コマンドの -D オプションに相当します。 指定したマクロはサブプロパティとして表示します。	
	デフォルト	定義マクロ [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。
定義解除マクロ	定義解除したいマクロ名を指定します。 「マクロ名」の形式で 1 行に 1 つずつ指定します。 ccrh コマンドの -U オプションに相当します。 指定したマクロはサブプロパティとして表示します。	
	デフォルト	定義解除マクロ [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

(3) [文字コード]

文字コードに関する詳細情報の表示、および設定の変更を行います。

文字コード	ソース・ファイル中の日本語／中国語のコメント、文字列に対して、使用する文字コードを選択します。 ccrh コマンドの -Xcharacter_set オプションに相当します。		
	デフォルト	自動 (なし)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	自動 (なし)	ソース・ファイル中の日本語の文字コードを SJIS と解釈します。
		SJIS(-Xcharacter_set=sjis)	ソース・ファイル中の日本語の文字コードを SJIS と解釈します。
		EUC(-Xcharacter_set=euc_jp)	ソース・ファイル中の日本語の文字コードを EUC と解釈します。
		UFT-8(-Xcharacter_set=utf8)	ソース・ファイル中の日本語の文字コードを UFT-8 と解釈します。
		Big5(-Xcharacter_set=big5)	ソース・ファイル中の中国語の文字コードを繁体字中国語と解釈します。
		GB2312(-Xcharacter_set=gb2312)	ソース・ファイル中の中国語の文字コードを簡体字中国語と解釈します。
処理しない (-Xcharacter_set=none)	ソース・ファイル中の日本語／中国語の文字コードを解釈しません。		

(4) [出力コード]

出力コードに関する詳細情報の表示、および設定の変更を行います。

32 ビット分岐命令を使用する	jarl, および jr 命令に対して、far jump 機能を使用するかどうかを選択します。 far jump 機能を使用することにより、jarl, および jr 命令を jarl32, および jr32 命令とみなしてアセンブルを行います。 ccrh コマンドの -Xasm_option=-Xasm_far_jump オプションに相当します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-Xasm_option=-Xasm_far_jump)	jarl, および jr 命令を jarl32, および jr32 命令とみなしてアセンブルを行います。
		いいえ	jarl, および jr 命令としてアセンブルを行います。

(5) [アセンブル・リスト]

アセンブル・リストに関する詳細情報の表示、および設定の変更を行います。

アセンブル・リスト・ファイルを出力する	アセンブル・リスト・ファイルを出力するかどうかを選択します。 ccrh コマンドの <code>-Xasm_option=-Xprn_path</code> オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xasm_option=-Xprn_path)
		いいえ
アセンブル・リスト・ファイル出力フォルダ	アセンブル・リスト・ファイルの出力先フォルダを指定します。 アセンブル・リスト・ファイルは、ソース・ファイル名の拡張子を“.prn”で置き換えた名前で出力します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの <code>-Xasm_option=-Xprn_path</code> オプションに相当します。 なお、本プロパティは、[アセンブル・リスト・ファイルを出力する] プロパティで [はい (-Xasm_option=-Xprn_path)] を選択した場合のみ表示します。	
	デフォルト	%BuildModeName%
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列

(6) [その他]

アセンブルに関するその他の詳細情報の表示、および設定の変更を行います。

アセンブル前に実行する コマンド	アセンブル処理前に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %AssembledFile% : アセンブル時の出力ファイルの絶対パスに置換します。 %BuildModeName% : ビルド・モード名に置換します。 %InputFile% : アセンブル対象ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、アセンブル処理前に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。	
	デフォルト	アセンブル前に実行するコマンド[定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	1023 文字までの文字列 64 個まで指定可能です。

アセンブル後に実行する コマンド	アセンブル処理後に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %AssembledFile% : アセンブル時の出力ファイルの絶対パスに置換します。 %BuildModeName% : ビルド・モード名に置換します。 %InputFile% : アセンブル対象ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、アセンブル処理後に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。	
	デフォルト	アセンブル後に実行するコマンド [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	1023 文字までの文字列 64 個まで指定可能です。
その他の追加オプション	その他に追加するアセンブル・オプションを入力します。 ここで設定したオプションは、アセンブル・オプション群の最後に付加します。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集
	指定可能値	259 文字までの文字列

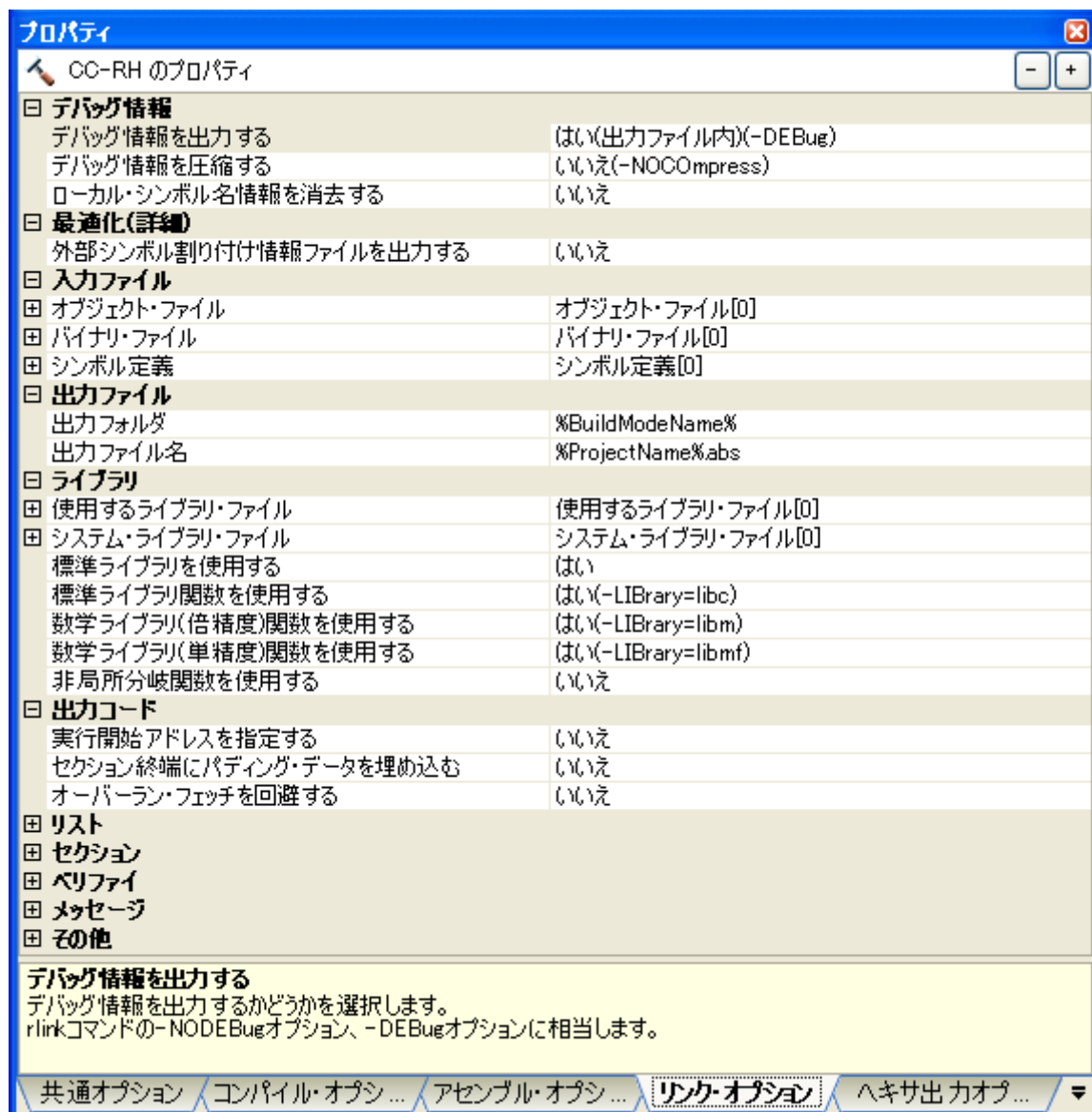
[リンク・オプション] タブ

本タブでは、リンク・フェーズに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

- (1) [デバッグ情報]
- (2) [最適化 (詳細)]
- (3) [入力ファイル]
- (4) [出力ファイル]
- (5) [ライブラリ]
- (6) [出力コード]
- (7) [リスト]
- (8) [セクション]
- (9) [ベリファイ]
- (10) [メッセージ]
- (11) [その他]

注意 本タブは、ライブラリ用のプロジェクトの場合は表示しません。

図 A—7 プロパティ パネル: [リンク・オプション] タブ



[各カテゴリの説明]

(1) [デバッグ情報]

デバッグ情報に関する詳細情報の表示、および設定の変更を行います。

デバッグ情報を出力する	デバッグ情報を出力するかどうかを選択します。 rlink コマンドの -DEBug 、および -NODebug オプションに相当します。		
	デフォルト	はい (出力ファイル内) (-DEBug)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (出力ファイル内) (-DEBug)	デバッグ情報を出力します。
		いいえ (-NODebug)	デバッグ情報を出力しません。

デバッグ情報を圧縮する	デバッグ情報を圧縮するかどうかを選択します。 rlink コマンドの -Compress 、および -NOCOmpress オプションに相当します。 なお、本プロパティは、[デバッグ情報を出力する] プロパティで [はい (出力ファイル内)(-DEBug)] を選択した場合のみ表示します。			
	デフォルト	いいえ (-NOCOmpress)		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (-Compress)	デバッグ情報を圧縮します。 デバッグのロード速度が速くなります。	
		いいえ (-NOCOmpress)	デバッグ情報を圧縮しません。 リンク時間が短くなります。	
ローカル・シンボル名情報を消去する	ローカル・シンボル名情報を消去するかどうかを選択します。 rlink コマンドの -Hide オプションに相当します。			
	デフォルト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (-Hide)	ローカル・シンボル名情報を消去します。	
		いいえ	ローカル・シンボル名情報の消去を行いません。	

(2) [最適化 (詳細)]

最適化に関する詳細情報の表示、および設定の変更を行います。

外部シンボル割り付け情報ファイルを出力する	外部シンボル割り付け情報ファイルを出力するかどうかを選択します。 [コンパイル・オプション] タブの [最適化 (詳細)] カテゴリの [外部変数アクセス最適化を行う] プロパティで [はい (モジュール間で最適化)(-Omap)] を選択した場合、本プロパティは [はい (-Map)] となります。 rlink コマンドの -MAP オプション に相当します。			
	デフォルト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (-Map)	外部シンボル割り付け情報ファイルを出力します。	
		いいえ	外部シンボル割り付け情報ファイルを出力しません。	

(3) [入力ファイル]

入力ファイルに関する詳細情報の表示、および設定の変更を行います。

オブジェクト・ファイル	オブジェクト・ファイルを指定します。 「ライブラリ(モジュール)」の形式で1行に1つずつ指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 rlink コマンドの -Input オプションに相当します。 オブジェクト・ファイル名はサブプロパティとして表示します。	
	デフォルト	オブジェクト・ファイル [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	1024 文字までの文字列 256 個まで指定可能です。

バイナリ・ファイル	バイナリ・ファイルを指定します。 「ファイル名 (セクション名 [: アライメント数] [/ セクション属性], シンボル名)」の形式で 1 行に 1 つずつ指定します。 「: アライメント数」, 「/ セクション属性」, 「, シンボル名」の部分は省略可能です。 アライメント数に指定可能な値は、1、2、4、8、16、32 のいずれかです。 省略した場合は、1 を指定したものとみなします。 セクション属性に指定可能なものは、“CODE”、または“DATA”です。 省略した場合は、書き込み、読み取り、実行、すべての属性が有効になります。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 rlink コマンドの -Binary オプションに相当します。 バイナリ・ファイル名はサブプロパティとして表示します。	
	デフォルト	バイナリ・ファイル [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	1024 文字までの文字列 256 個まで指定可能です。
シンボル定義	シンボルを定義します。 「シンボル名 = シンボル名」, または「シンボル名 = 数値」の形式で 1 行に 1 つずつ指定します。 数値は 16 進数で指定します。 rlink コマンドの -DEFine オプションに相当します。 シンボル名はサブプロパティとして表示します。	
	デフォルト	シンボル定義 [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

(4) [出力ファイル]

出力ファイルに関する詳細情報の表示、および設定の変更を行います。

出力フォルダ	出力フォルダを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 rlink コマンドの -Output オプションに相当します。	
	デフォルト	%BuildModeName%
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列
出力ファイル名	出力ファイル名を指定します。 拡張子を省略した場合は、“.abs”を自動的に付加します。 次のプレースホルダに対応しています。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %ProjectName% : プロジェクト名に置換します。 rlink コマンドの -Output オプションに相当します。	
	デフォルト	%ProjectName%.abs
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	259 文字までの文字列

(5) [ライブラリ]

ライブラリに関する詳細情報の表示、および設定の変更を行います。

使用するライブラリ・ファイル	使用するライブラリ・ファイルを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 rlink コマンドの -LIBrary オプションに相当します。 ライブラリ・ファイル名はサブプロパティとして表示します。		
	デフォルト	使用するライブラリ・ファイル [定義数]	
	変更方法	[...] ボタンをクリックし、 パス編集 ダイアログ による編集 → [参照] ボタンをクリックし、 使用するライブラリ・ファイルを指定ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能	
	指定可能値	259 文字までの文字列 65536 個まで指定可能です。	
システム・ライブラリ・ファイル	システム・ライブラリ・ファイルを表示します。 rlink コマンドの -LIBrary オプションに相当します。 システム・ライブラリ・ファイル名はサブプロパティとして表示します。		
	デフォルト	システム・ライブラリ・ファイル [定義数]	
	指定可能値	変更不可	
標準ライブラリを使用する	コンパイラが提供する標準ライブラリを使用するかどうかを選択します。 rlink コマンドの -LIBrary オプションに相当します。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	標準ライブラリを使用します。
		いいえ	標準ライブラリを使用しません。

標準ライブラリ関数を使用する	標準ライブラリ関数を使用するかどうかを選択します。 [数学ライブラリ (倍精度) 関数を使用する]。または [数学ライブラリ (単精度) 関数を使用する] プロパティで [はい] を選択した場合、本プロパティは [はい] となります。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	はい (-LIBrary=libc)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libc) 標準ライブラリ関数を使用します。 いいえ 標準ライブラリ関数を使用しません。
数学ライブラリ (倍精度) 関数を使用する	数学ライブラリ (倍精度) 関数を使用するかどうかを選択します。 [標準ライブラリ関数を使用する] プロパティで [いいえ] を選択した場合、本プロパティは [いいえ] となります。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	はい (-LIBrary=libm)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libm) 数学ライブラリ (倍精度) 関数を使用します。 いいえ 数学ライブラリ (倍精度) 関数を使用しません。
数学ライブラリ (単精度) 関数を使用する	数学ライブラリ (単精度) 関数を使用するかどうかを選択します。 [標準ライブラリ関数を使用する] プロパティで [いいえ] を選択した場合、本プロパティは [いいえ] となります。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	はい (-LIBrary=libmf)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libmf) 数学ライブラリ (単精度) 関数を使用します。 いいえ 数学ライブラリ (単精度) 関数を使用しません。
非局所分岐関数を使用する	非局所分岐関数を使用するかどうかを選択します。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libsetjmp) 非局所分岐関数を使用します。 いいえ 非局所分岐関数を使用しません。

(6) [出力コード]

出力コードに関する詳細情報の表示、および設定の変更を行います。

実行開始アドレスを指定する	実行開始アドレスを外部定義シンボル、またはアドレスで指定するかどうかを選択します。 rlink コマンドの -ENTry オプションに相当します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-ENTry)	実行開始アドレスを外部定義シンボル、またはアドレスで指定します。
		いいえ	実行開始アドレスを外部定義シンボル、またはアドレスで指定しません。
実行開始アドレス	実行開始アドレスを指定します。 「シンボル名」、または「アドレス」の形式で指定します。 アドレスは 16 進数で指定します。 rlink コマンドの -ENTry オプションに相当します。 実行開始アドレスはサブプロパティとして表示します。 なお、本プロパティは、[実行開始アドレスを指定する] プロパティで [はい (-ENTry)] を選択した場合のみ表示します。		
	デフォルト	空欄	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集	
	指定可能値	32767 文字までの文字列	
	セクション終端にパディング・データを埋め込む	セクション終端にパディング・データを埋め込むかどうかを選択します。 rlink コマンドの -PADDING オプションに相当します。	
デフォルト		いいえ	
変更方法		ドロップダウン・リストによる選択	
指定可能値		はい (-PADDING)	セクション・サイズが、セクションのアライメントの倍数となるように、セクションの終端にデータを埋め込みます。
		いいえ	セクション終端にパディング・データを埋め込みません。
オーバーラン・フェッチを回避する	オーバーラン・フェッチに伴う空き領域の読み出しを回避するかどうかを選択します。 rlink コマンドの -OVERRUN_FETCH オプションに相当します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-OVERRUN_FETCH)	セクション間、および終端コード・セクションの直後に 128 バイト以上の空き領域が存在する場合、空き領域に対して 128 バイトの NOP 命令を挿入します。
		いいえ	オーバーラン・フェッチに伴う空き領域の読み出しの回避を指定しません。

(7) [リスト]

リストに関する詳細情報の表示、および設定の変更を行います。

リンク・マップ・ファイル を出力する	リンク・マップ・ファイルを出力するかどうかを選択します。 rlink コマンドの -LIST 、 および -SHow オプションに相当します。		
	デフォルト	はい (リスト内容 = 選択)(-LIST)	
	変更方法	ドロップダウン・ リストによる選択	
	指定可能値	はい (リスト内容 = 指定 なし)(-LIST -SHow)	リンク・マップ・ ファイルに出力フォー マットに従った情報を出力します。
		はい (リスト内容 = すべ て)(-LIST -SHow=ALL)	リンク・マップ・ ファイルに出力フォー マットに従ったすべての情報を出力します。
		はい (リスト内容 = 選択)(-LIST)	リンク・マップ・ ファイルに指定した情報 を出力します。
いいえ	リンク・マップ・ ファイルを出力しません。		
シンボル情報を出力する	シンボル情報（シンボル・アドレス、 サイズ、 種別、 最適化内容）を出力するかどうか を選択します。 rlink コマンドの -SHow=SYmbol オプションに相当します。 なお、 本プロパティは、 [リンク・マップ・ファイルを出力する] プロパティで [はい (リスト内容 = 選択)(-LIST)] を選択した場合のみ表示します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・ リストによる選択	
	指定可能値	はい (-SHow=SYmbol)	シンボル情報を出力します。
		いいえ	シンボル情報を出力しません。
	シンボルの参照回数を出 力する	シンボルの参照回数を出力するかどうかを選択します。 rlink コマンドの -SHow=Reference オプションに相当します。 なお、 本プロパティは、 [リンク・マップ・ファイルを出力する] プロパティで [はい (リスト内容 = 選択)(-LIST)] を選択した場合のみ表示します。	
デフォルト		いいえ	
変更方法		ドロップダウン・ リストによる選択	
指定可能値		はい (-SHow=Reference)	シンボルの参照回数を出力します。
		いいえ	シンボルの参照回数を出力しません。
クロス・リファレンス情 報を出力する		クロス・リファレンス情報を出力するかどうかを選択します。 rlink コマンドの -SHow=XReference オプションに相当します。 なお、 本プロパティは、 [リンク・マップ・ファイルを出力する] プロパティで [はい (リスト内容 = 選択)(-LIST)] を選択した場合のみ表示します。	
	デフォルト	いいえ	
	変更方法	ドロップダウン・ リストによる選択	
	指定可能値	はい (-SHow=Xreference)	クロス・リファレンス情報を出力します。
		いいえ	クロス・リファレンス情報を出力しません。

セクションの合計サイズを出力する	セクションの合計サイズを出力するかどうかを選択します。 rlink コマンドの -SHow=Total_size オプションに相当します。 なお、本プロパティは、[リンク・マップ・ファイルを出力する] プロパティで [はい(リスト内容 = 選択)(-LIST)] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-SHow=Total_size) ROM 配置対象、RAM 配置対象ごとに、セクションの合計サイズを出力します。 いいえ セクションの合計サイズを出力しません。

(8) [セクション]

セクションに関する詳細情報の表示、および設定の変更を行います。

セクションの開始アドレス	セクションの開始アドレスを指定します。 rlink コマンドの -START オプションに相当します。	
	デフォルト	VECT,.const,.text,.data/00000000,RESET/01000000,.data.R,.bss,.stack.bss/FEBF8000
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 セクション設定 ダイアログ による編集
	指定可能値	32767 文字までの文字列
スタートアップの属するセクション	スタートアップの属するセクションを指定します。 1 行に 1 つずつ指定します。 本プロパティを指定することにより、マルチコア用ブート・ローダから呼び出すスタートアップのシンボルがブート・ローダ・プロジェクトに登録されます。 rlink コマンドの -FSymbol オプションに相当します。 セクション名はサブプロパティとして表示します。 なお、本プロパティは、ブート・ローダ・ノードの [ブート・ローダ] タブの [構成するアプリケーション・プロジェクト] プロパティに設定しているプロジェクトの場合のみ表示します。	
	デフォルト	スタートアップの属するセクション [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	32767 文字までの文字列 65535 個まで指定可能です。
外部定義シンボルをファイル出力するセクション	外部定義シンボルをファイル出力するセクションを指定します。 1 行に 1 つずつ指定します。 rlink コマンドの -FSymbol オプションに相当します。 セクション名はサブプロパティとして表示します。	
	デフォルト	外部定義シンボルをファイル出力するセクション [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	32767 文字までの文字列 65535 個まで指定可能です。

セクション・アライメント	アライメント数を 16 バイトに変更するセクションを指定します。 1 行に 1 つずつ指定します。 rlink コマンドの -ALIGNED_SECTION オプションに相当します。 セクション名はサブプロパティとして表示します。	
	デフォルト	セクション・アライメント [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	32767 文字までの文字列 65535 個まで指定可能です。
ROM から RAM ヘマッ プするセクション	ROM から RAM ヘマッパするセクションを指定します。 「ROM セクション名=RAM セクション名」の形式で 1 行に 1 つずつ指定します。 rlink コマンドの -ROm オプションに相当します。 セクション名はサブプロパティとして表示します。	
	デフォルト	ROM から RAM ヘマッパするセクション [定義数] サブプロパティには “.data=.data.R” が指定されます。
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	32767 文字までの文字列 65535 個まで指定可能です。

(9) [ベリファイ]

ベリファイに関する詳細情報の表示、および設定の変更を行います。

セクションの割り付けア ドレスをチェックする	セクションの割り付けアドレスの整合性をチェックするかどうかを選択します。 rlink コマンドの -CPu オプションに相当します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-CPu) セクションの割り付けアドレスの整合性をチェックし ます。
		いいえ セクションの割り付けアドレスの整合性をチェックし ません。

メモリ種別のアドレス範囲	メモリ種別のアドレス範囲を指定します。 「メモリ種別 = 先頭アドレス - 終了アドレス」の形式で 1 行に 1 つずつ指定します。 メモリ種別に指定可能なものは、“ROm”、“RAm”、“FIX” のいずれかです。 先頭アドレス、終了アドレスは 16 進数で指定します。 rlink コマンドの -CPu オプションに相当します。 メモリ種別のアドレス範囲はサブプロパティとして表示します。 なお、本プロパティは、[セクションの割り付けアドレスをチェックする] プロパティで [はい (-CPu)] を選択した場合のみ表示します。	
	デフォルト	メモリ種別のアドレス範囲 [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	32767 文字までの文字列 65535 個まで指定可能です。

(10) [メッセージ]

メッセージに関する詳細情報の表示、および設定の変更を行います。

インフォメーション・メッセージ出力を有効にする	インフォメーション・メッセージの出力を有効にするかどうかを選択します。 rlink コマンドの -Message 、および -NOMessage オプションに相当します。				
	デフォルト	いいえ (-NOMessage)			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-Message)</td><td>インフォメーション・メッセージを出力します。</td></tr><tr><td>いいえ (-NOMessage)</td><td>インフォメーション・メッセージの出力を抑止します。</td></tr></table>	はい (-Message)	インフォメーション・メッセージを出力します。	いいえ (-NOMessage)
はい (-Message)	インフォメーション・メッセージを出力します。				
いいえ (-NOMessage)	インフォメーション・メッセージの出力を抑止します。				
抑止するインフォメーション・メッセージの番号	出力を抑止するインフォメーション・メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例：4,200）。 また、ハイフンを使用して、メッセージ番号の範囲を指定することもできます（例：4,200-203,1300）。 rlink コマンドの -NOMessage オプションに相当します。 なお、本プロパティは、[インフォメーション・メッセージ出力を有効にする] プロパティで [いいえ (-NOMessage)] を選択した場合のみ表示します。				
	デフォルト	空欄			
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集			
	指定可能値	2048 文字までの文字列			

参照されない定義シンボルを通知する	参照されない定義シンボルを通知するかどうかを選択します。 rlink コマンドの -MSg_unused オプションに相当します。 なお、本プロパティは、[インフォメーション・メッセージ出力を有効にする] プロパティで [はい (-Message)] を選択した場合、または [抑止するインフォメーション・メッセージの番号] プロパティを指定した場合のみ表示します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-MSg_unused)	参照されない定義シンボルを通知します。
		いいえ	参照されない定義シンボルを通知しません。
ワーニング・メッセージをインフォメーション・メッセージに変更する	ワーニングのメッセージ種別をインフォメーションに変更するかどうかを選択します。 rlink コマンドの -CHange_message オプションに相当します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (すべて)(-CHange_message=Information)	すべてのワーニングのメッセージ種別をインフォメーションに変更します。
		はい (メッセージ番号指定)(-CHange_message=Information=<メッセージ番号>)	メッセージ種別をインフォメーションに変更するワーニングのメッセージ番号を指定します。
いいえ		ワーニングのメッセージ種別の変更を行いません。	
ワーニング・メッセージの番号	ワーニング・メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例：4,200）。 また、ハイフンを使用して、メッセージ番号の範囲を指定することもできます（例：4,200-203,1300）。 rlink コマンドの -CHange_message オプションに相当します。 なお、本プロパティは、[ワーニング・メッセージをインフォメーション・メッセージに変更する] プロパティで [はい (メッセージ番号指定)(-CHange_message=Information=<メッセージ番号>)] を選択した場合のみ表示します。		
	デフォルト	空欄	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集	
	指定可能値	2048 文字までの文字列	

インフォメーション・メッセージをワーニング・メッセージに変更する	インフォメーションのメッセージ種別 をワーニングに変更するかどうかを選択します。 rlink コマンドの -CHange_message オプションに相当します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (すべて)(-CHange_message=Warning)	すべてのインフォメーションのメッセージ種別 をワーニングに変更します。
		はい (メッセージ番号指定)(-CHange_message=Warning=<メッセージ番号 >)	メッセージ種別をワーニングに変更するインフォメーションのメッセージ番号を指定します。
いいえ	インフォメーションのメッセージ種別の変更を行いません。		
インフォメーション・メッセージの番号	インフォメーション・メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例：4,200）。 また、ハイフンを使用して、メッセージ番号の範囲を指定することもできます（例：4,200-203,1300）。 rlink コマンドの -CHange_message オプションに相当します。 なお、本プロパティは、[ワーニング・メッセージをインフォメーション・メッセージに変更する] プロパティで [はい (メッセージ番号指定)(-CHange_message=Information=<メッセージ番号 >)] を選択した場合のみ表示します。		
	デフォルト	空欄	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集	
	指定可能値	2048 文字までの文字列	
	インフォメーション、ワーニング・メッセージをエラー・メッセージに変更する	インフォメーション、 およびワーニングのメッセージ種別 をエラーに変更するかどうかを選択します。 rlink コマンドの -CHange_message オプションに相当します。	
デフォルト		いいえ	
変更方法		ドロップダウン・リストによる選択	
指定可能値		はい (すべて)(-CHange_message=Error)	すべてのインフォメーション、 およびワーニングのメッセージ種別 をエラーに変更します。
		はい (メッセージ番号指定)(-CHange_message=Error=<メッセージ番号 >)	メッセージ種別をエラーに変更するインフォメーション、 およびワーニングのメッセージ番号を指定します。
いいえ	インフォメーション、 およびワーニングのメッセージ種別の変更を行いません。		

インフォメーション、 ワーニング・メッセージ の番号	インフォメーション、およびワーニング・メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例：4,200）。 また、ハイフンを使用して、メッセージ番号の範囲を指定することもできます（例： 4,200-203,1300）。 rlink コマンドの -CHange_message オプションに相当します。 なお、本プロパティは、[ワーニング・メッセージをインフォメーション・メッセージに 変更する] プロパティで [はい (メッセージ番号指定) (-CHange_message=Error=< メッセージ番号 >)] を選択した場合のみ表示します。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集
	指定可能値	2048 文字までの文字列

(11) [その他]

リンクに関するその他の詳細情報の表示、および設定の変更を行います。

スタック情報ファイルを出力する	スタック情報ファイルを出力するかどうかを選択します。 rlink コマンドの -STACK オプションに相当します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-STACK)	スタック情報ファイルを出力します。
		いいえ	スタック情報ファイルを出力しません。

メモリ使用量を削減する	メモリ使用量を削減するかどうかを選択します。 rlink コマンドの -MEMory オプションに相当します。 なお、本プロパティは、 以下の場合のみ表示します。 - 【デバッグ情報】 カテゴリの 【デバッグ情報を出力する】 プロパティで 【いいえ (-NODEBug)】 、または 【デバッグ情報を圧縮する】 プロパティで 【いいえ (-NOCompress)】 を選択した場合 - 【最適化 (詳細)】 カテゴリの 【外部シンボル割り付け情報ファイルを出力する】 プロパティで 【いいえ】 を選択した場合 - 以下の条件のうち、 いずれか 1 つが成立する場合 (1) 【リスト】 カテゴリの 【リンク・マップ・ファイルを出力する】 プロパティで 【いいえ】 を選択した場合 (2) 【リスト】 カテゴリの 【リンク・マップ・ファイルを出力する】 プロパティで 【はい (リスト内容 = 指定なし)(-LISt -SHoW)】 を選択した場合 (3) 【リスト】 カテゴリの 【リンク・マップ・ファイルを出力する】 プロパティで 【はい (リスト内容 = 選択)(-LISt)】 、 【シンボルの参照回数を出力する】 プロパティで 【いいえ】 、 および 【クロス・リファレンス情報を出力する】 プロパティで 【いいえ】 を選択した場合 - 【スタック情報ファイル出力する】 プロパティで 【いいえ】 を選択した場合		
	デフォルト	いいえ (-MEMory=High)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-MEMory=Low)	メモリ使用量を削減します。 大規模なプロジェクトをリンクした際、最適化リンクのメモリ使用量が稼動マシンの実装メモリ量を超えてしまい、動作が遅くなっているような場合に選択してください。
		いいえ (-MEMory=High)	従来通りの処理を行います。
合計セクション・サイズを表示する	リンク後の合計セクション・サイズを表示するかどうかを選択します。 rlink コマンドの -Total_size オプションに相当します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-Total_size)	リンク後の合計セクション・サイズを表示します。
		いいえ	リンク後の合計セクション・サイズを表示しません。
コピーライト情報を表示する	コピーライト情報を表示するかどうかを選択します。 rlink コマンドの -LOgo 、 および -NOLOgo オプションに相当します。 なお、本プロパティは、 【共通オプション】 タブの 【ビルド方法】 カテゴリの 【一括ビルドを行う】 プロパティで 【いいえ】 を選択した場合のみ表示します。		
	デフォルト	いいえ (-NOLOgo)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-LOgo)	コピーライト情報を表示します。
		いいえ (-NOLOgo)	コピーライト情報の表示を抑止します。

リンク前に実行するコマンド	リンク処理前に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %LinkedFile% : リンク処理時の出力ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、リンク処理前に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。 なお、本プロパティは、[共通オプション] タブの[ビルド方法]カテゴリの[一括ビルドを行う] プロパティで [いいえ] を選択した場合のみ表示します。
デフォルト	リンク前に実行するコマンド [定義数]
変更方法	[...] ボタン をクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
指定可能値	1023 文字までの文字列 64 個まで指定可能です。

リンク後に実行するコマンド	リンク処理後に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %LinkedFile% : リンク処理時の出力ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、リンク処理後に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。 なお、本プロパティは、[共通オプション] タブの[ビルド方法]カテゴリの[一括ビルドを行う]プロパティで[いいえ]を選択した場合のみ表示します。	
	デフォルト	リンク後に実行するコマンド[定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	1023 文字までの文字列 64 個まで指定可能です。
その他の追加オプション	その他に追加するリンクのオプションを入力します。 なお、ここで設定したオプションは、リンクのオプション群の最後に付加されます。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集
	指定可能値	259 文字までの文字列

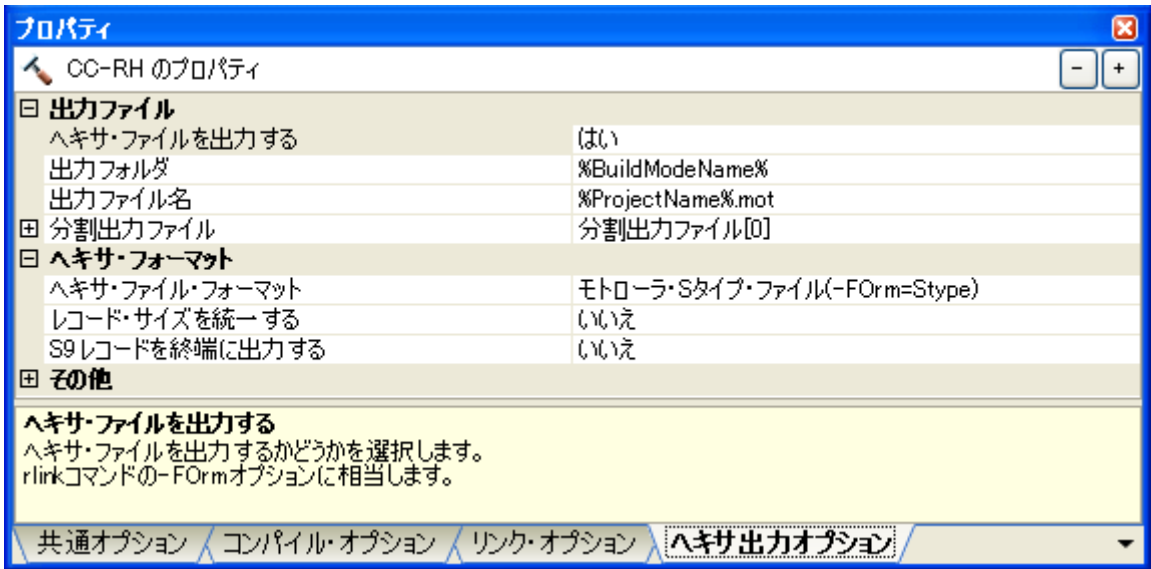
[ヘキサ出力オプション] タブ

本タブでは、ヘキサ出力フェーズに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

- (1) [出力ファイル]
- (2) [ヘキサ・フォーマット]
- (3) [その他]

注意 本タブは、ライブラリ用のプロジェクトの場合は表示しません。

図 A—8 プロパティ パネル : [ヘキサ出力オプション] タブ



[各カテゴリの説明]

(1) [出力ファイル]

出力ファイルに関する詳細情報の表示、および設定の変更を行います。

ヘキサ・ファイルを出力する	ヘキサ・ファイルを出力するかどうかを選択します。 rlink コマンドの -FOrm オプションに相当します。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	ヘキサ・ファイルを出力します。
		いいえ	ヘキサ・ファイルを出力しません。

出力フォルダ	ヘキサ・ファイルの出力フォルダを指定します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 rlink コマンドの -Output オプションに相当します。 なお、本プロパティは、[ヘキサ・ファイルを出力する] プロパティで [はい] を選択した場合のみ表示します。 <table border="1" data-bbox="536 1070 1394 1249"> <tr> <td>デフォルト</td><td>%BuildModeName%</td></tr> <tr> <td>変更方法</td><td>テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、フォルダの参照 ダイアログによる編集</td></tr> <tr> <td>指定可能値</td><td>247 文字までの文字列</td></tr> </table>	デフォルト	%BuildModeName%	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集	指定可能値	247 文字までの文字列
デフォルト	%BuildModeName%						
変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集						
指定可能値	247 文字までの文字列						
出力ファイル名	ヘキサ・ファイル名を指定します。 本プロパティは必ず指定してください。 拡張子を省略した場合は、[ヘキサ・フォーマット] カテゴリの [ヘキサ・ファイル・フォーマット] プロパティの選択に依存して、自動的に付加します。 [インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)] を選択している場合 : .hex [モトローラ・S タイプ・ファイル (-FOrm=Stype)] を選択している場合 : .mot [バイナリ・ファイル (-FOrm=Binary)] を選択している場合 : .bin 次のプレースホルダに対応しています。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %ProjectName% : プロジェクト名に置換します。 rlink コマンドの -Output オプションに相当します。 なお、本プロパティは、[ヘキサ・ファイルを出力する] プロパティで [はい] を選択した場合のみ表示します。 <table border="1" data-bbox="536 1821 1394 1960"> <tr> <td>デフォルト</td><td>%ProjectName%.mot</td></tr> <tr> <td>変更方法</td><td>テキスト・ボックスによる直接入力</td></tr> <tr> <td>指定可能値</td><td>259 文字までの文字列</td></tr> </table>	デフォルト	%ProjectName%.mot	変更方法	テキスト・ボックスによる直接入力	指定可能値	259 文字までの文字列
デフォルト	%ProjectName%.mot						
変更方法	テキスト・ボックスによる直接入力						
指定可能値	259 文字までの文字列						

分割出力ファイル	分割出力ファイルを指定します。 「ファイル名 = 先頭アドレス - 終了アドレス」（先頭アドレス、終了アドレス：出力範囲の先頭アドレス、終了アドレス）、または「ファイル名 = セクション名」（セクション名：出力するセクション名）の形式で 1 行に 1 つずつ指定します。 セクション名を複数指定する場合は、「ファイル名 = セクション名 : セクション名」のように、コロンで区切って指定します（例：file1.mot=sec1:sec2）。 アドレスは 16 進数で指定します（例：file2.mot=400-4ff）。 拡張子を省略した場合は、[ヘキサ・フォーマット] カテゴリの [ヘキサ・ファイル・フォーマット] プロパティの選択に依存して、自動的に付加します。 [インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)] を選択している場合：.hex [モトローラ・S タイプ・ファイル (-FOrm=Stype)] を選択している場合：.mot [バイナリ・ファイル (-FOrm=Binary)] を選択している場合：.bin 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 mlink コマンドの -Output オプションに相当します。 分割出力ファイル名はサブプロパティとして表示します。 なお、本プロパティは、[ヘキサ・ファイルを出力する] プロパティで [はい] を選択した場合のみ表示します。
デフォルト	分割出力ファイル [定義数]
変更方法	[...] ボタン をクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
指定可能値	259 文字までの文字列 65535 個まで指定可能です。

オブジェクト結合機能を使用する	構成アプリケーションとして設定した各プロジェクトのヘキサ・ファイルを1つに結合するかどうかを選択します。 rlink コマンドの -FOrm オプション に相当します。 なお、本プロパティは、以下の場合のみ表示します。 - マルチコア用ブート・ローダ・プロジェクトの場合 - ブート・ローダ・プロジェクト、およびすべての構成アプリケーション・プロジェクトの [ヘキサ・ファイルを出力する] プロパティで [はい]、および [ヘキサ・フォーマット] カテゴリの [ヘキサ・ファイル・フォーマット] プロパティで同一の項目を選択している場合	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい 構成アプリケーションとして設定した各プロジェクトのヘキサ・ファイルを1つに結合します。 いいえ プロジェクトごとにヘキサ・ファイルを出力します。
結合ヘキサ・ファイルの出力フォルダ	結合したヘキサ・ファイルの出力フォルダを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 rlink コマンドの -OUTput オプションに相当します。 なお、本プロパティは、[オブジェクト結合機能を使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	%BuildModeName%_merged
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列

(2) [ヘキサ・フォーマット]

ヘキサ・フォーマットに関する詳細情報の表示、および設定の変更を行います。

なお、本カテゴリは、**[出力ファイル]** カテゴリの [ヘキサ・ファイルを出力する] プロパティで [はい] を選択した場合のみ表示します。

ヘキサ・ファイル・フォーマット	出力するヘキサ・ファイルのフォーマットを選択します。 rlink コマンドの -FOrm オプションに相当します。		
	デフォルト	モトローラ・S タイプ・ファイル (-FOrm=Stype)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)	インテル拡張ヘキサ・ファイルを出力します。
		モトローラ・S タイプ・ファイル (-FOrm=Stype)	モトローラ・S タイプ・ファイルを出力します。
バイナリ・ファイル (-FOrm=Binary)		バイナリ・ファイルを出力します。	
レコード・サイズを統一する 【インテル拡張ヘキサ・ファイル】	アドレス範囲に関係なく、一定のデータ・レコードで出力するかどうかを選択します。 rlink コマンドの -RECORD オプションに相当します。 なお、本プロパティは、[ヘキサ・ファイル・フォーマット] プロパティで [インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)] を選択した場合のみ表示します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (インテル・ヘキサ・レコード)(-REcord=H16)	インテル・ヘキサ・レコードを出力します。
		はい (インテル拡張ヘキサ・レコード)(-REcord=H20)	インテル拡張ヘキサ・レコードを出力します。
はい (インテル 32 ビット・ヘキサ・レコード)(-REcord=H32)		インテル 32 ビット・ヘキサ・レコードを出力します。	
いいえ	それぞれのアドレスにあわせて、混在したデータ・レコードを出力します。		
レコード・サイズを統一する 【モトローラ・S タイプ・ファイル】	アドレス範囲に関係なく、一定のデータ・レコードで出力するかどうかを選択します。 rlink コマンドの -RECORD オプションに相当します。 なお、本プロパティは、[ヘキサ・ファイル・フォーマット] プロパティで [モトローラ・S タイプ・ファイル (-FOrm=Stype)] を選択した場合のみ表示します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (S1 レコード)(-REcord=S1)	S1 レコードを出力します。
		はい (S2 レコード)(-REcord=S2)	S2 レコードを出力します。
はい (S3 レコード)(-REcord=S3)		S3 レコードを出力します。	
いいえ	それぞれのアドレスにあわせて、混在したデータ・レコードを出力します。		

出力範囲のメモリの空き領域をデータで充てんする	出力範囲のメモリの空き領域をデータで充てんするかどうかを選択します。 rlink コマンドの -SPace オプションに相当します。 なお、本プロパティは、 [出力ファイル] カテゴリの [分割出力ファイル] プロパティにファイルを指定した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (乱数) (-SPace=Random) 空き領域を乱数で充てんします。
		はい (データ指定) (-SPace=<数値>) 空き領域を指定した 16 進数の数値で充てんします。
		いいえ 空き領域の充てんを行いません。
空き領域出力データ	空き領域を充てんするデータを 16 進数の数値で指定します。 rlink コマンドの -SPace オプションに相当します。 なお、本プロパティは、 [出力範囲のメモリの空き領域をデータで充てんする] プロパティで [はい (データ指定) (-SPace=<数値>)] を選択した場合のみ表示します。	
	デフォルト	FF
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	0 ~ FFFFFFFF (16 進数)
データ・レコードのバイト数を指定する	データ・レコードのバイト数の最大値を指定するかどうかを選択します。 rlink コマンドの -BYte_count オプションに相当します。 なお、本プロパティは、 [ヘキサ・ファイル・フォーマット] プロパティで [インテル拡張ヘキサ・ファイル (-FOrm=Hexadecimal)] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-BYte_count) データ・レコードのバイト数の最大値を指定します。
		いいえ データ・レコードのバイト数の最大値を 0xFF とします。
データ・レコードのバイト数最大値	データ・レコードのバイト数の最大値を指定します。 rlink コマンドの -BYte_count オプションに相当します。 なお、本プロパティは、 [データ・レコードのバイト数を指定する] プロパティで [はい (-BYte_count)] を選択した場合のみ表示します。	
	デフォルト	FF
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	1 ~ FF (16 進数)

S9 レコードを端末に出力する	S9 レコードを端末に出力するかどうかを選択します。 rlink コマンドの -S9 オプションに相当します。 なお、本プロパティは、[ヘキサ・ファイル・フォーマット] プロパティで [モトローラ・S タイプ・ファイル (-FOrm=Stype)] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-S9) S9 レコードを端末に出力します。 いいえ S9 レコードを端末に出力しません。

(3) [その他]

ヘキサ出力に関するその他の詳細情報の表示、および設定の変更を行います。

なお、本カテゴリは、[\[出力ファイル\]](#) カテゴリの [ヘキサ・ファイルを出力する] プロパティで [はい] を選択した場合のみ表示します。

その他の追加オプション	その他に追加するヘキサ出力オプションを入力します。 なお、ここで設定したオプションは、ヘキサ出力オプション群の最後に付加されます。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集
	指定可能値	259 文字までの文字列

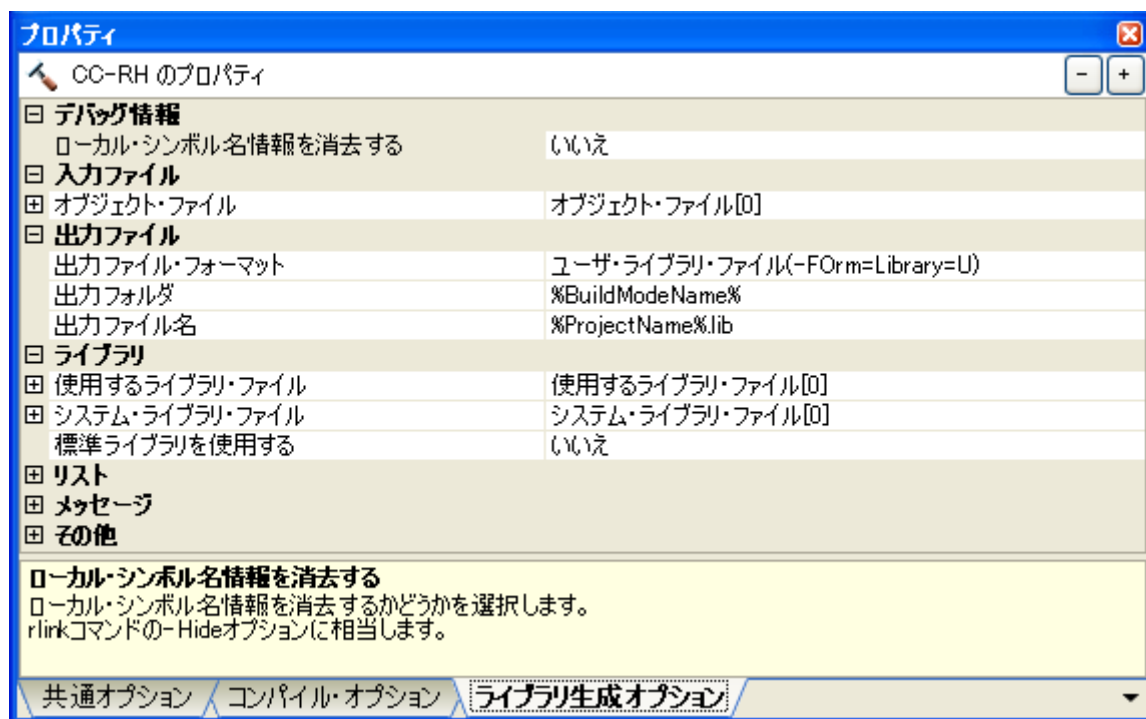
[ライブラリ生成オプション] タブ

本タブでは、ライブラリ生成フェーズに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

- (1) [デバッグ情報]
- (2) [入力ファイル]
- (3) [出力ファイル]
- (4) [ライブラリ]
- (5) [リスト]
- (6) [メッセージ]
- (7) [その他]

注意 本タブは、ライブラリ用のプロジェクトの場合のみ表示します。

図 A—9 プロパティ パネル : [ライブラリ生成オプション] タブ



[各カテゴリの説明]

(1) [デバッグ情報]

デバッグ情報に関する詳細情報の表示、および設定の変更を行います。

デバッグ情報を出力する	デバッグ情報を出力するかどうかを選択します。 rlink コマンドの -DEBug 、および -NODEBug オプションに相当します。 なお、本プロパティは、 [出力ファイル] カテゴリの [出力ファイル・フォーマット] プロパティで [リロケートブル・ファイル (-FOrm=Relocate)] を選択した場合のみ表示します。		
	デフォルト	はい (出力ファイル内) (-DEBug)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (出力ファイル内) (-DEBug)	デバッグ情報を出力します。
		いいえ (-NODEBug)	デバッグ情報を出力しません。
ローカル・シンボル名情報を消去する	ローカル・シンボル名情報を消去するかどうかを選択します。 rlink コマンドの -Hide オプションに相当します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-Hide)	ローカル・シンボル名情報を消去します。
		いいえ	ローカル・シンボル名情報の消去を行いません。

(2) [入力ファイル]

入力ファイルに関する詳細情報の表示、および設定の変更を行います。

オブジェクト・ファイル	オブジェクト・ファイルを指定します。 「ライブラリ(モジュール)」の形式で1行に1つずつ指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 rlink コマンドの -Input オプションに相当します。 オブジェクト・ファイル名はサブプロパティとして表示します。	
	デフォルト	オブジェクト・ファイル [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	1024 文字までの文字列 256 個まで指定可能です。

バイナリ・ファイル	バイナリ・ファイルを指定します。 「ファイル名 (セクション名 [アライメント数] / セクション属性 [, シンボル名])」の形式で 1 行に 1 つずつ指定します。 「: アライメント数」, 「/ セクション属性」, 「, シンボル名」の部分は省略可能です。 アライメント数に指定可能な値は、1、2、4、8、16、32 のいずれかです。 省略した場合は、1 を指定したものとみなします。 セクション属性に指定可能なものは、“CODE”、または“DATA”です。 省略した場合は、書き込み、読み取り、実行、すべての属性が有効になります。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 rlink コマンドの -Binary オプションに相当します。 バイナリ・ファイル名はサブプロパティとして表示します。 なお、本プロパティは、[出力ファイル] カテゴリの [出力ファイル・フォーマット] プロパティで [リロケートブル・ファイル (-FOrm=Relocate)] を選択した場合のみ表示します。	
	デフォルト	バイナリ・ファイル [定義数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	1024 文字までの文字列 256 個まで指定可能です。

(3) [出力ファイル]

出力ファイルに関する詳細情報の表示、および設定の変更を行います。

出力ファイル・フォーマット	出力ファイルのフォーマットを選択します。 rlink コマンドの -FOrm オプションに相当します。		
	デフォルト	ユーザ・ライブラリ・ファイル (-FOrm=Library=U)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	ユーザ・ライブラリ・ファイル (-FOrm=Library=U)	ユーザ・ライブラリ・ファイルを出力します。
		システム・ライブラリ・ファイル (-FOrm=Library=S)	システム・ライブラリ・ファイルを出力します。
リロケートブル・ファイル (-FOrm=Relocate)		リロケートブル・ファイルを出力します。	

出力フォルダ	出力フォルダを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 rlink コマンドの -Output オプションに相当します。	
	デフォルト	%BuildModeName%
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列
出力ファイル名	出力ファイル名を指定します。 拡張子を省略した場合は、[出力ファイル・フォーマット] プロパティの選択に依存して、自動的に付加します。 [ユーザ・ライブラリ・ファイル (-FOrm=Library=U)] を選択している場合 : .lib [システム・ライブラリ・ファイル (-FOrm=Library=S)] を選択している場合 : .lib [リロケータブル・ファイル (-FOrm=Relocate)] を選択している場合 : .rel 次のプレースホルダに対応しています。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %ProjectName% : プロジェクト名に置換します。 rlink コマンドの -Output オプションに相当します。	
	デフォルト	%ProjectName%.lib
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	259 文字までの文字列

(4) [ライブラリ]

ライブラリに関する詳細情報の表示、および設定の変更を行います。

使用するライブラリ・ファイル	使用するライブラリ・ファイルを指定します。 次のプレースホルダに対応しています。		
	%ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。		
	%ActiveProjectName% : アクティブ・プロジェクト名に置換します。		
	%BuildModeName% : ビルド・モード名に置換します。		
	%MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。		
使用するライブラリ・ファイル	%MainProjectName% : メイン・プロジェクト名に置換します。		
	%MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。		
	%ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。		
	%ProjectName% : プロジェクト名に置換します。		
	%TempDir% : テンポラリ・フォルダの絶対パスに置換します。		
	%WinDir% : Windows システム・フォルダの絶対パスに置換します。		
	rlink コマンドの -LIBrary オプションに相当します。		
	ライブラリ・ファイル名はサブプロパティとして表示します。		
	デフォルト	使用するライブラリ・ファイル [定義数]	
	変更方法	[...] ボタンをクリックし、 パス編集 ダイアログ による編集 → [参照] ボタンをクリックし、 使用するライブラリ・ファイルを指定ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能	
指定可能値	259 文字までの文字列 65536 個まで指定可能です。		
システム・ライブラリ・ファイル	システム・ライブラリ・ファイルを表示します。 rlink コマンドの -LIBrary オプションに相当します。 システム・ライブラリ・ファイル名はサブプロパティとして表示します。		
	デフォルト	システム・ライブラリ・ファイル [定義数]	
	指定可能値	変更不可	
	標準ライブラリを使用する	コンパイラが提供する標準ライブラリを使用するかどうかを選択します。 rlink コマンドの -LIBrary オプションに相当します。	
デフォルト		いいえ	
変更方法		ドロップダウン・リストによる選択	
指定可能値		はい	標準ライブラリを使用します。
		いいえ	標準ライブラリを使用しません。

標準ライブラリ関数を使用する	標準ライブラリ関数を使用するかどうかを選択します。 [数学ライブラリ (倍精度) 関数を使用する]。または [数学ライブラリ (単精度) 関数を使用する] プロパティで [はい] を選択した場合、本プロパティは [はい] となります。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libc) 標準ライブラリ関数を使用します。 いいえ 標準ライブラリ関数を使用しません。
数学ライブラリ (倍精度) 関数を使用する	数学ライブラリ (倍精度) 関数を使用するかどうかを選択します。 [標準ライブラリ関数を使用する] プロパティで [いいえ] を選択した場合、本プロパティは [いいえ] となります。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libmf) 数学ライブラリ (倍精度) 関数を使用します。 いいえ 数学ライブラリ (倍精度) 関数を使用しません。
数学ライブラリ (単精度) 関数を使用する	数学ライブラリ (単精度) 関数を使用するかどうかを選択します。 [標準ライブラリ関数を使用する] プロパティで [いいえ] を選択した場合、本プロパティは [いいえ] となります。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libmf) 数学ライブラリ (単精度) 関数を使用します。 いいえ 数学ライブラリ (単精度) 関数を使用しません。
非局所分岐関数を使用する	非局所分岐関数を使用するかどうかを選択します。 rlink コマンドの -LIBrary オプションに相当します。 なお、本プロパティは、[標準ライブラリを使用する] プロパティで [はい] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-LIBrary=libsetjmp) 非局所分岐関数を使用します。 いいえ 非局所分岐関数を使用しません。

(5) [リスト]

リストに関する詳細情報の表示、および設定の変更を行います。

リンク・マップ・ファイル を出力する	ライブラリ・リスト・ファイルを出力するかどうかを選択します。 rlink コマンドの -LIST 、および -SHow オプションに相当します。			
	デフォルト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (リスト内容 = 指定なし)(-LIST -SHow)	ライブラリ・リスト・ファイルに出力フォーマットに従った情報を出力します。	
		はい (リスト内容 = すべて)(-LIST -SHow=ALL)	ライブラリ・リスト・ファイルに出力フォーマットに従ったすべての情報を出力します。	
はい (リスト内容 = 選択)(-LIST)		ライブラリ・リスト・ファイルに指定した情報を出力します。		
	いいえ	ライブラリ・リスト・ファイルを出力しません。		
シンボル情報を出力する	シンボル情報 (モジュール内シンボル名) を出力するかどうかを選択します。 rlink コマンドの -SHow=SYmbol オプションに相当します。 なお、本プロパティは、[リンク・マップ・ファイルを出力する] プロパティで [はい (リスト内容 = 選択)(-LIST)] を選択した場合のみ表示します。			
	デフォルト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (-SHow=SYmbol)	シンボル情報を出力します。	
		いいえ	シンボル情報を出力しません。	
モジュール内セクション 一覧を出力する	モジュール内セクション名の一覧を出力するかどうかを選択します。 rlink コマンドの -SHow=SEction オプションに相当します。 なお、本プロパティは、[リンク・マップ・ファイルを出力する] プロパティで [はい (リスト内容 = 選択)(-LIST)]、および [出力ファイル] カテゴリの [出力ファイル・フォーマット] プロパティで [ユーザ・ライブラリ・ファイル (-FOrm=Library=U)]、または [システム・ライブラリ・ファイル (-FOrm=Library=S)] を選択した場合のみ表示します。			
	デフォルト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (-SHow=SEction)	モジュール内セクション名の一覧を出力します。	
		いいえ	モジュール内セクション名の一覧を出力しません。	

クロス・リファレンス情報 を出力する	クロス・リファレンス情報を出力するかどうかを選択します。 rlink コマンドの -SHow=XReference オプションに相当します。 なお、本プロパティは、[リンク・マップ・ファイルを出力する] プロパティで [はい (リスト内容 = 選択)](-LISt)], および [出力ファイル] カテゴリの [出力ファイル・フォーマット] プロパティで [リロケータブル・ファイル (-FOrm=Relocate)] を選択した場合のみ表示します。			
	デフォルト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (-SHow=Xreference)	クロス・リファレンス情報を出力します。	
		いいえ	クロス・リファレンス情報を出力しません。	
セクションの合計サイズ を出力する	セクションの合計サイズを出力するかどうかを選択します。 rlink コマンドの -SHow=Total_size オプションに相当します。 なお、本プロパティは、[リンク・マップ・ファイルを出力する] プロパティで [はい (リスト内容 = 選択)](-LISt)], および [出力ファイル] カテゴリの [出力ファイル・フォーマット] プロパティで [リロケータブル・ファイル (-FOrm=Relocate)] を選択した場合のみ表示します。			
	デフォルト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (-SHow=Total_size)	ROM 配置対象、RAM 配置対象ごとに、セクションの合計サイズを出力します。	
		いいえ	セクションの合計サイズを出力しません。	

(6) [メッセージ]

メッセージに関する詳細情報の表示、および設定の変更を行います。

インフォメーション・メッセージ出力を有効にする	インフォメーション・メッセージの出力を有効にするかどうかを選択します。 rlink コマンドの -Message 、 -NOMessage オプションに相当します。		
	デフォルト	いいえ (-NOMessage)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-Message)	インフォメーション・メッセージを出力します。
		いいえ (-NOMessage)	インフォメーション・メッセージの出力を抑止します。

抑止するインフォメーション・メッセージの番号	出力を抑止するインフォメーション・メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例：4,200）。 また、ハイフンを使用して、メッセージ番号の範囲を指定することもできます（例：4,200-203,1300）。 rlink コマンドの -NOMessage オプションに相当します。 なお、本プロパティは、[インフォメーション・メッセージ出力を有効にする] プロパティで [いいえ (-NOMessage)] を選択した場合のみ表示します。			
	デフォルト	空欄		
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集		
	指定可能値	2048 文字までの文字列		
ワーニング・メッセージをインフォメーション・メッセージに変更する	ワーニングのメッセージ種別をインフォメーションに変更するかどうかを選択します。 rlink コマンドの -CHange_message オプションに相当します。			
	デフォルト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (すべて)(-CHange_message=Information)	すべてのワーニングのメッセージ種別をインフォメーションに変更します。	
		はい (メッセージ番号指定)(-CHange_message=Information=<メッセージ番号>)	メッセージ種別をインフォメーションに変更するワーニングのメッセージ番号を指定します。	
いいえ		ワーニングのメッセージ種別の変更を行いません。		
ワーニング・メッセージの番号	ワーニング・メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例：4,200）。 また、ハイフンを使用して、メッセージ番号の範囲を指定することもできます（例：4,200-203,1300）。 rlink コマンドの -CHange_message オプションに相当します。 なお、本プロパティは、[ワーニング・メッセージをインフォメーション・メッセージに変更する] プロパティで [はい (メッセージ番号指定)(-CHange_message=Information=<メッセージ番号>)] を選択した場合のみ表示します。			
	デフォルト	空欄		
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集		
	指定可能値	2048 文字までの文字列		

インフォメーション・メッセージをワーニング・メッセージに変更する	インフォメーションのメッセージ種別をワーニングに変更するかどうかを選択します。 rlink コマンドの -CHange_message オプションに相当します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (すべて)(-CHange_message=Warning)	すべてのインフォメーションのメッセージ種別をワーニングに変更します。
		はい (メッセージ番号指定)(-CHange_message=Warning=<メッセージ番号 >)	メッセージ種別をワーニングに変更するインフォメーションのメッセージ番号を指定します。
	いいえ	インフォメーションのメッセージ種別の変更を行いません。	
インフォメーション・メッセージの番号	インフォメーション・メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例：4,200）。 また、ハイフンを使用して、メッセージ番号の範囲を指定することもできます（例：4,200-203,1300）。 rlink コマンドの -CHange_message オプションに相当します。 なお、本プロパティは、[ワーニング・メッセージをインフォメーション・メッセージに変更する] プロパティで [はい (メッセージ番号指定)(-CHange_message=Information=<メッセージ番号 >)] を選択した場合のみ表示します。		
	デフォルト	空欄	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集	
	指定可能値	2048 文字までの文字列	
	インフォメーション、ワーニング・メッセージをエラー・メッセージに変更する	インフォメーション、およびワーニングのメッセージ種別をエラーに変更するかどうかを選択します。 rlink コマンドの -CHange_message オプションに相当します。	
デフォルト		いいえ	
変更方法		ドロップダウン・リストによる選択	
指定可能値		はい (すべて)(-CHange_message=Error)	すべてのインフォメーション、およびワーニングのメッセージ種別をエラーに変更します。
		はい (メッセージ番号指定)(-CHange_message=Error=<メッセージ番号 >)	メッセージ種別をエラーに変更するインフォメーション、およびワーニングのメッセージ番号を指定します。
	いいえ	インフォメーション、およびワーニングのメッセージ種別の変更を行いません。	

インフォメーション、 ワーニング・メッセージ の番号	インフォメーション、およびワーニング・メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例：4,200）。また、ハイフンを使用して、メッセージ番号の範囲を指定することもできます（例：4,200-203,1300）。 rlink コマンドの -CHange_message オプションに相当します。 なお、本プロパティは、[ワーニング・メッセージをインフォメーション・メッセージに変更する] プロパティで [はい (メッセージ番号指定) (-CHange_message=Error=<メッセージ番号>)] を選択した場合のみ表示します。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集
	指定可能値	2048 文字までの文字列

(7) [その他]

ライブラリ生成に関するその他の詳細情報の表示、および設定の変更を行います。

メモリ使用量を削減する	メモリ使用量を削減するかどうかを選択します。 rlink コマンドの -MEMory オプションに相当します。 なお、本プロパティは、以下の場合のみ表示します。 - [デバッグ情報] カテゴリの [ローカル・シンボル名情報を消去する] プロパティで [いいえ] を選択した場合 - [出力ファイル] カテゴリの [出力ファイル・フォーマット] プロパティで [ユーザ・ライブラリ・ファイル (-FOrm=Library=U)]、または [システム・ライブラリ・ファイル (-FOrm=Library=S)] を選択した場合	
	デフォルト	いいえ (-MEMory=High)
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-MEMory=Low)
		メモリ使用量を削減します。 大規模なプロジェクトをリンクした際、最適化リンクのメモリ使用量が稼動マシンの実装メモリ量を超えてしまい、動作が遅くなっているような場合に選択してください。
合計セクション・サイズ を表示する	リンク後の合計セクション・サイズを表示するかどうかを選択します。 rlink コマンドの -Total_size オプションに相当します。 なお、本プロパティは、[出力ファイル] カテゴリの [出力ファイル・フォーマット] プロパティで [リロケートブル・ファイル (-FOrm=Relocate)] を選択した場合のみ表示します。	
	デフォルト	いいえ
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Total_size)
		リンク後の合計セクション・サイズを表示します。
	いいえ	リンク後の合計セクション・サイズを表示しません。

コピーライト情報を表示する	コピーライト情報を表示するかどうかを選択します。 rlink コマンドの -Logo 、および -NOLogo オプションに相当します。 なお、本プロパティは、 [共通オプション] タブ の [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [いいえ] を選択した場合のみ表示します。				
	デフォルト	いいえ (-NOLogo)			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-Logo)</td><td>コピーライト情報を表示します。</td></tr><tr><td>いいえ (-NOLogo)</td><td>コピーライト情報の表示を抑止します。</td></tr></table>	はい (-Logo)	コピーライト情報を表示します。	いいえ (-NOLogo)
はい (-Logo)	コピーライト情報を表示します。				
いいえ (-NOLogo)	コピーライト情報の表示を抑止します。				
ライブラリ生成前に実行するコマンド	ライブラリ生成処理前に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %LibraryFile% : ライブラリ生成処理時の出力ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、ライブラリ生成処理前に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。 なお、本プロパティは、 [共通オプション] タブ の [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [いいえ] を選択した場合のみ表示します。				
	デフォルト	ライブラリ生成前に実行するコマンド [定義数]			
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能			
	指定可能値	1023 文字までの文字列 64 個まで指定可能です。			

ライブラリ生成後に実行するコマンド	ライブラリ生成処理後に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %LibraryFile% : ライブラリ生成処理時の出力ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、ライブラリ生成処理後に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。 なお、本プロパティは、[共通オプション] タブの[ビルド方法]カテゴリの「一括ビルドを行う」プロパティで「いいえ」を選択した場合のみ表示します。
デフォルト	ライブラリ生成後に実行するコマンド [定義数]
変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
指定可能値	1023 文字までの文字列 64 個まで指定可能です。
その他の追加オプション	その他に追加するライブラリ生成オプションを入力します。 なお、ここで設定したオプションは、ライブラリ生成オプション群の最後に付加されます。
デフォルト	空欄
変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集
指定可能値	259 文字までの文字列

[ビルド設定] タブ

本タブでは、各 C ソース・ファイル、アセンブリ・ソース・ファイル、オブジェクト・ファイル、ライブラリ・ファイルに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

(1) [ビルド]

図 A—10 プロパティ パネル：[ビルド設定] タブ（C ソース・ファイルを選択した場合）

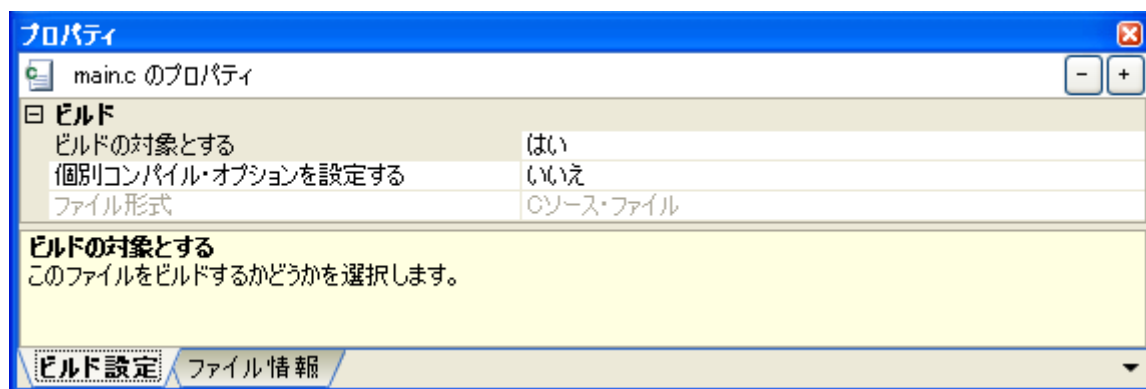


図 A—11 プロパティ パネル：[ビルド設定] タブ（アセンブリ・ソース・ファイルを選択した場合）

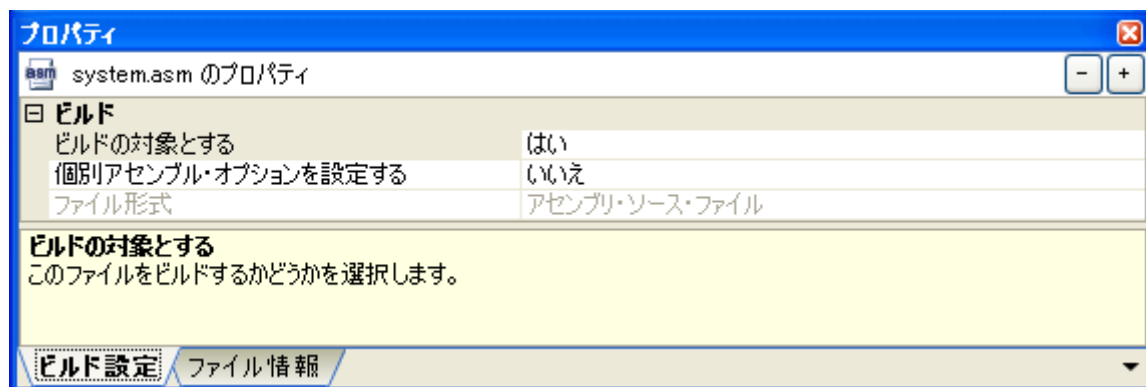


図 A—12 プロパティ パネル：[ビルド設定] タブ（オブジェクト・ファイルを選択した場合）

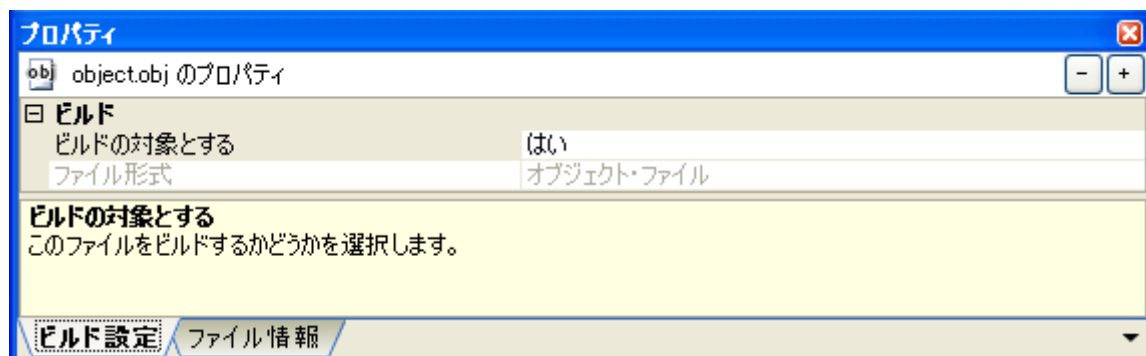
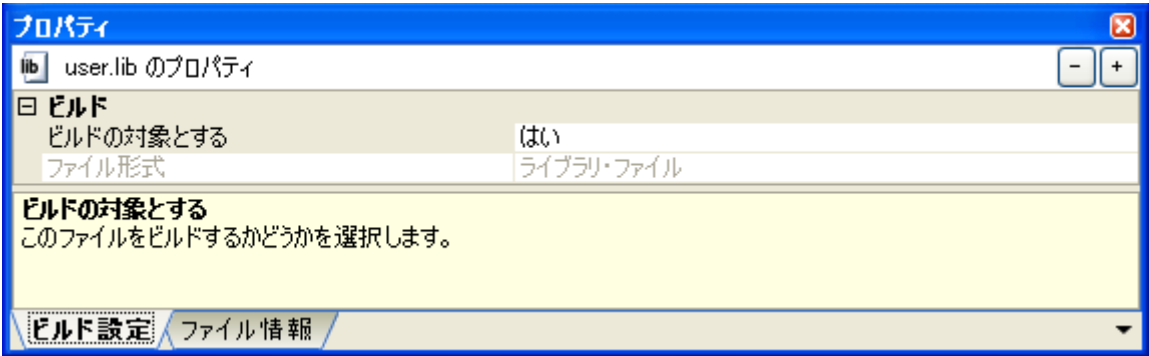


図 A—13 プロパティ パネル : [ビルド設定] タブ (ライブラリ・ファイルを選択した場合)



[各カテゴリの説明]

(1) [ビルド]

ビルドに関する詳細情報の表示、および設定の変更を行います。

ビルドの対象とする	選択しているファイルをビルド対象とかどうかを選択します。		
	デフォルト	はい	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	選択しているファイルをビルド対象とします。
		いいえ	選択しているファイルをビルド対象としません。
個別コンパイル・オプションを設定する	選択している C ソース・ファイルにプロジェクトの設定とは異なるコンパイル・オプションを設定するかどうかを選択します。 [コンパイル・オプション] タブの [最適化 (詳細)] カテゴリの [大域最適化を行う] プロパティで [はい (レベル 3)(入力ソースがプログラム全体として最適化する)(-Xwhole_program)] を選択した場合、本プロパティはグレー表示となり、[いいえ] に変更されます。 なお、本プロパティは、プロジェクト・ツリーで C ソース・ファイルを選択し、本タブの [ビルドの対象とする] プロパティで [はい] を選択した場合のみ表示します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	選択している C ソース・ファイルにプロジェクトの設定とは異なるオプションを設定します。
		いいえ	選択している C ソース・ファイルにプロジェクトの設定とは異なるオプションを設定しません。

個別アセンブル・オプションを設定する	選択しているアセンブリ・ソース・ファイルにプロジェクトの設定とは異なるアセンブル・オプションを設定するかどうかを選択します。 [コンパイル・オプション] タブの [最適化 (詳細)] カテゴリの [大域最適化を行う] プロパティで [はい (レベル 3)(入力ソースがプログラム全体として最適化する)(-Xwhole_program)] を選択した場合、本プロパティはグレー表示となり、[いいえ] に変更されます。 なお、本プロパティは、プロジェクト・ツリーでアセンブリ・ソース・ファイルを選択し、本タブの [ビルドの対象とする] プロパティで [はい] を選択した場合のみ表示します。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	選択しているアセンブリ・ソース・ファイルにプロジェクトの設定とは異なるオプションを設定します。
		いいえ	選択しているアセンブリ・ソース・ファイルにプロジェクトの設定とは異なるオプションを設定しません。
ファイル形式	選択しているファイルの形式を表示します。		
	デフォルト	C ソース・ファイル (C ソース・ファイルを選択している場合) アセンブリ・ソース・ファイル (アセンブリ・ソース・ファイルを選択している場合) オブジェクト・ファイル (オブジェクト・ファイルを選択している場合) ライブラリ・ファイル (ライブラリ・ファイルを選択している場合)	
	変更方法	変更不可	

【個別コンパイル・オプション】 タブ

本タブでは、1つのCソース・ファイルに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

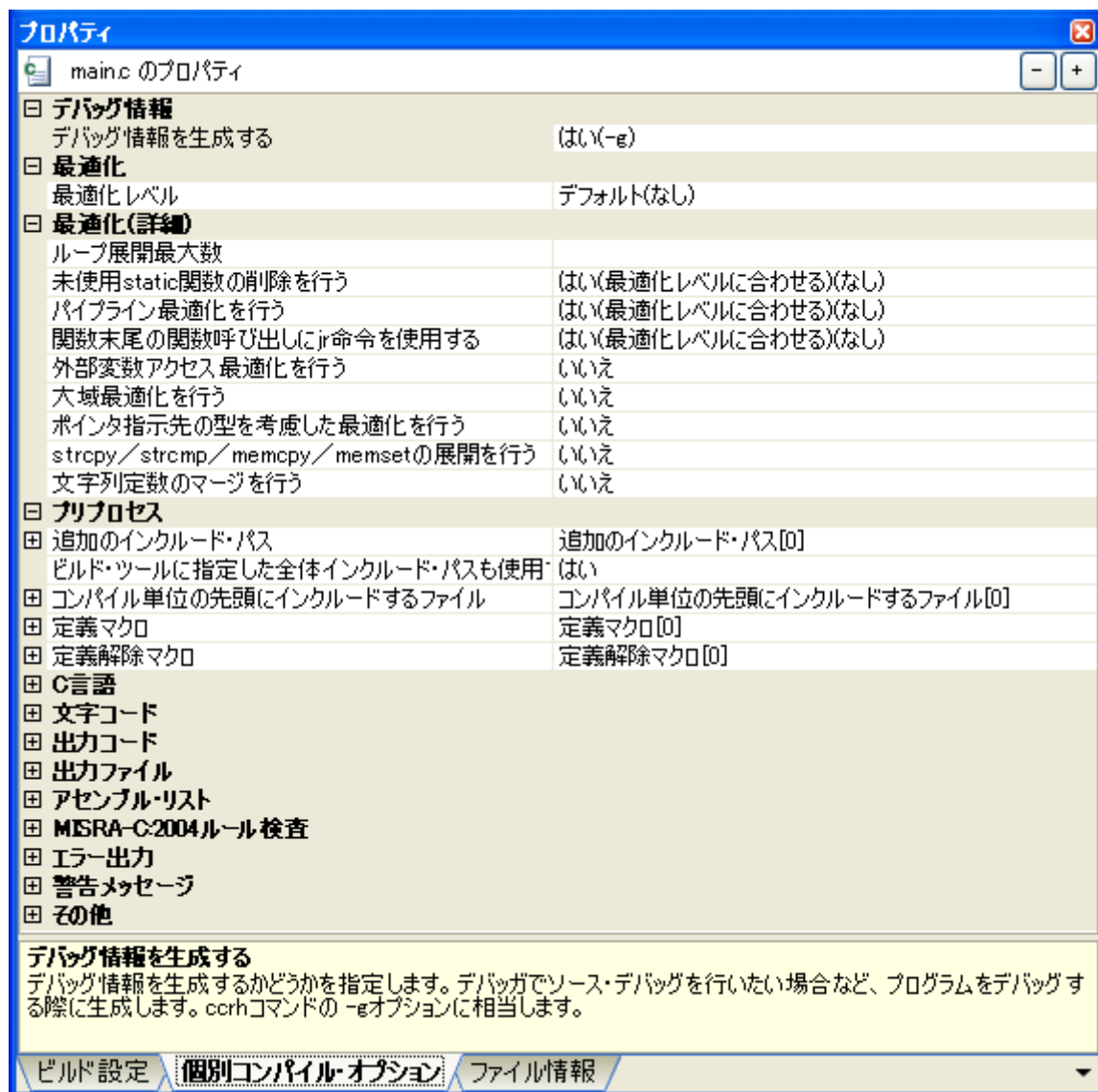
なお、本タブは、[\[共通オプション\] タブ](#)、および [\[コンパイル・オプション\] タブ](#)の設定内容を継承します。

これらのタブと異なる値を設定した場合は、プロパティが太字表示となります。

- (1) [\[デバッグ情報\]](#)
- (2) [\[最適化\]](#)
- (3) [\[最適化 \(詳細\)\]](#)
- (4) [\[プリプロセス\]](#)
- (5) [\[C 言語\]](#)
- (6) [\[文字コード\]](#)
- (7) [\[出力コード\]](#)
- (8) [\[出力ファイル\]](#)
- (9) [\[アセンブル・リスト\]](#)
- (10) [\[MISRA-C:2004 ルール検査\]](#)
- (11) [\[エラー出力\]](#)
- (12) [\[警告メッセージ\]](#)
- (13) [\[その他\]](#)

備考 本タブは、[\[ビルド設定\] タブ](#)の [\[ビルド\]](#) カテゴリの [\[個別コンパイル・オプションを設定する\]](#) プロパティで [\[はい\]](#) を選択した場合のみ表示します。

図 A—14 プロパティ パネル：[個別コンパイル・オプション] タブ



[各カテゴリの説明]

(1) [デバッグ情報]

デバッグ情報に関する詳細情報の表示、および設定の変更を行います。

デバッグ情報を生成する

デバッグ情報を生成するかどうかを選択します。

出力ファイル中にソース・デバッグ用の情報を出力することにより、デバッガでのソース・デバッグが可能となります。

ccrh コマンドの **-g** オプションに相当します。

デフォルト	コンパイル・オプションの設定値	
変更方法	ドロップダウン・リストによる選択	
指定可能値	はい (-g)	デバッグ情報を生成します。
	いいえ	デバッグ情報を生成しません。

(2) [最適化]

最適化に関する詳細情報の表示、および設定の変更を行います。

最適化レベル	コンパイルの最適化レベルを選択します。 ccrh コマンドの -O オプションに相当します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	デフォルト (なし)	デバッグに影響しない範囲の最適化（式の最適化、およびレジスタ割り付けなど）を行います。
		サイズ優先 (-Osize)	オブジェクト・サイズ優先の最適化を行います。 ROM/RAM 容量の削減を重視して、一般的なプログラムに対して有効な最大限の最適化を行います。
		実行速度優先 (-Ospeed)	実行速度優先の最適化を行います。 実行速度の短縮を重視して、一般的なプログラムに対して有効な最大限の最適化を行います。
		デバッグ優先 (-Onothing)	デバッグを優先して最適化を行います。 デバッグのしやすさを重視し、デフォルトで実行する最適化を含むすべての最適化を抑止します。

(3) [最適化 (詳細)]

最適化に関する詳細情報の表示、および設定の変更を行います。

ループ展開最大数	for、while などのループを展開する最大数を指定します。 0、または 1 を指定した場合は、展開を抑止します。 空欄の場合は、4 を指定したものとみなします。 ccrh コマンドの -Ounroll オプションに相当します。			
	デフォルト	コンパイル・オプションの設定値		
	変更方法	テキスト・ボックスによる直接入力		
	指定可能値	0 ～ 999 (10 進数)、または空欄		
未使用 static 関数の削除 を行う	呼び出されない static 関数の削除を行うかどうかを選択します。 ccrh コマンドの -Odelete_static_func オプションに相当します。			
	デフォルト	コンパイル・オプションの設定値		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (最適化レベルに合 わせる) (なし)	[最適化レベル] プロパティにあわせて最適 化を行います。	
		はい (- Odelete_static_func)	呼び出されない static 関数の削除を行います。	
		いいえ (- Odelete_static_func=off)	呼び出されない static 関数の削除を行いま せん。	

関数のインライン展開を行う	関数を呼び出し箇所にインライン展開するかどうかを選択します。 ccrh コマンドの -Oinline オプションに相当します。 なお、本プロパティは、[最適化レベル] プロパティで [サイズ優先 (-Osize)]、または [実行速度優先 (-Ospeed)] を選択した場合のみ表示します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (最適化レベルに合わせる)(なし)	[最適化レベル] プロパティにあわせて最適化を行います。
		はい (指定関数のみ)(-Oinline=1)	#pragma inline 指定した関数を呼び出し箇所にインライン展開します。
		はい (自動判別)(-Oinline=2)	自動的にインライン展開対象の関数を判別して展開します。
はい (コード・サイズが増加しないよう自動判別)(-Oinline=3)		コード・サイズがなるべく増加しない範囲で、自動的にインライン展開対象の関数を判別して展開します。	
	いいえ (-Oinline=0)	#pragma inline 指定した関数を含めて、すべてのインライン展開を抑止します。	
インライン展開サイズの最大増加率	コード・サイズが何%増加するまでインライン展開を行うかを指定します（例：“100”を指定した場合、コード・サイズが 100%（2 倍）増加するまでインライン展開を行います）。 ccrh コマンドの -Oinline_size オプションに相当します。 なお、本プロパティは、[関数のインライン展開を行う] プロパティで [はい (自動判別) (-Oinline=2)] を選択した場合、または [関数のインライン展開を行う] プロパティで [はい (最適化レベルに合わせる)] かつ [最適化レベル] プロパティで [実行速度優先 (-Ospeed)] を選択した場合のみ表示します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	テキスト・ボックスによる直接入力	
	指定可能値	0 ～ 65535（10 進数）	
	パイプライン最適化を行う	機械語レベルで命令の並べ替えを行い、プログラムの実行性能を引き出すかどうかを選択します。 ccrh コマンドの -Opipeline オプションに相当します。	
デフォルト		コンパイル・オプションの設定値	
変更方法		ドロップダウン・リストによる選択	
指定可能値		はい (最適化レベルに合わせる)(なし)	[最適化レベル] プロパティにあわせて最適化を行います。
		はい (-Opipeline)	パイプライン最適化を行います。
	いいえ (-Opipeline=off)	パイプライン最適化を行いません。	

関数末尾の関数呼び出しに jr 命令を使用する	関数の末尾が関数呼び出しの場合に、jarl 命令の代わりに jr 命令を優先的に使用するかどうかを選択します。 ccrh コマンドの -Otail_call オプションに相当します。			
	デフォルト	コンパイル・オプションの設定値		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (最適化レベルに合わせる)(なし)	[最適化レベル] プロパティにあわせて最適化を行います。	
		はい (-Otail_call)	関数の末尾が関数呼び出しの場合に、jarl 命令の代わりに jr 命令を優先的に使用します。 lp の退避／復帰コードを削除し、コード・サイズを小さくすることができます。 ただし、一部のデバッグ機能を使用することができなくなります。	
いいえ (-Otail_call=off)		関数の末尾が関数呼び出しの場合に、jarl 命令を使用します。		
外部変数アクセス最適化を行う	外部変数アクセス最適化を行うかどうかを選択します。 ccrh コマンドの -Osmap 、および -Omap オプションに相当します。			
	デフォルト	コンパイル・オプションの設定値		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (モジュール内で最適化)(-Osmap)	コンパイル対象ファイル内で定義された外部変数、または静的変数についてベース・アドレスを設定し、アクセスをベース・アドレス相対で行うコードを生成します。	
		はい (モジュール間で最適化)(-Omap)	外部シンボル割り付け情報ファイルを生成します。 その情報を元に、再コンパイルを行い、外部変数、または静的変数のアクセスをベース・アドレス相対で行うコードを生成します。 なお、本項目は、 [共通オプション] タブの [ビルド方法] カテゴリの「一括ビルドを行う」プロパティで「いいえ」を選択した場合のみ表示されます。 また、本項目を選択しているときに「一括ビルドを行う」プロパティで「はい」を選択した場合、本項目は「いいえ」に変更されます。	
	いいえ	外部変数アクセス最適化を行いません。		

大域最適化を行う	大域最適化（関数の統合など）を行うレベルを指定します。 ccrh コマンドの -Xintermodule オプションに相当します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	いいえ はい (レベル 1)(最適化を行う)(-Xintermodule) 大域最適化を行いません。 ソース・ファイルごとに、大域最適化を行います。
ポインタ指示先の型を考慮した最適化を行う	ANSI 規格に基づくポインタ指示先の型を考慮した最適化を行うかどうかを選択します。 ccrh コマンドの -Xalias オプションに相当します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xalias=ansi) ポインタ指示先の型を考慮した最適化を行います。 一般には、オブジェクト性能が向上しますが、[いいえ] を選択した場合と比べて実行結果が異なる場合があります。 いいえ ポインタ指示先の型を考慮した最適化を行いません。
strcpy / strcmp / memcpy / memset の展開を行う	配列（文字列を含む）、および構造体の整列条件を 4 バイトとし、関数 strcpy(), strcmp(), memcpy(), memset() の呼び出しをインライン展開するかどうかを選択します。 生成するプログラムの実行速度は高速になりますが、コード・サイズは増大します。 ccrh コマンドの -Xinline_strcpy オプションに相当します。 なお、本プロパティは、[コンパイル・オプション] タブ の [出力コード] カテゴリの [構造体パッキングを行う] プロパティで [いいえ] を選択した場合のみ表示します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xinline_strcpy) 関数 strcpy(), strcmp(), memcpy(), memset() の呼び出しをインライン展開します。 いいえ 関数 strcpy(), strcmp(), memcpy(), memset() のインライン展開を行いません。
文字列定数のマージを行う	ソース・ファイル内で同じ文字列定数が複数存在する場合、これらをまとめて 1 つの領域に割り付けるかどうかを選択します。 ccrh コマンドの -Xmerge_string オプションに相当します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xmerge_string) ソース・ファイル内で複数存在する同じ文字列定数をまとめて 1 つの領域に割り付けます。 いいえ ソース・ファイル内で複数存在する同じ文字列定数をそれぞれを別々の領域に割り付けます。

(4) [プリプロセス]

プリプロセスに関する詳細情報の表示、および設定の変更を行います。

追加のインクルード・パス	コンパイル時の追加のインクルード・パスを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 指定したインクルード・パスは、CC-RH の標準インクルード・ファイル・フォルダよりも優先して検索します。 パスはプロジェクト・フォルダを基点とします。 本プロパティを省略した場合は、CC-RH の標準インクルード・ファイル・フォルダのみ検索します。 ccrh コマンドの -I オプションに相当します。 指定したインクルード・パスはサブプロパティとして表示します。 インクルード・パスに大文字、小文字の区別はありません。	
	デフォルト	追加のインクルード・パス [定義数]
	変更方法	[...] ボタンをクリックし、 パス編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	259 文字までの文字列 256 個まで指定可能です。
ビルド・ツールに指定した全体インクルード・パスも使用する	使用するビルド・ツールの [コンパイル・オプション] タブの [プリプロセス] カテゴリの [追加のインクルード・パス] プロパティで指定したインクルード・パスも使用してコンパイルするかどうかを選択します。 インクルード・パスは、以下の順で追加します。 - 本タブの [追加のインクルード・パス] プロパティで指定したパス - [コンパイル・オプション] タブの [追加のインクルード・パス] プロパティで指定したパス - [コンパイル・オプション] タブの [システム・インクルード・パス] プロパティに表示しているパス ccrh コマンドの -I オプションに相当します。	
	デフォルト	はい
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい 使用するビルド・ツールのプロパティで指定したインクルード・パスも使用してコンパイルします。
		いいえ 使用するビルド・ツールのプロパティで指定したインクルード・パスを使用しません。

コンパイル単位の先頭にインクルードするファイル	コンパイル単位の先頭にインクルードするファイルを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 なお、パスはプロジェクト・フォルダを基点とします。 ccrh コマンドの -Xpreinclude オプションに相当します。 指定したインクルード・ファイル名はサブプロパティとして表示します。 <table border="1"> <tr> <td>デフォルト</td><td>コンパイル・オプションの設定値</td></tr> <tr> <td>変更方法</td><td>[...] ボタンをクリックし、テキスト編集 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能</td></tr> <tr> <td>指定可能値</td><td>247 文字までの文字列 256 個まで指定可能です。</td></tr> </table>	デフォルト	コンパイル・オプションの設定値	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能	指定可能値	247 文字までの文字列 256 個まで指定可能です。
デフォルト	コンパイル・オプションの設定値						
変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能						
指定可能値	247 文字までの文字列 256 個まで指定可能です。						
定義マクロ	定義したいマクロ名を指定します。 「マクロ名 = 定義値」の形式で 1 行に 1 つずつ指定します。 「= 定義値」の部分は省略可能で、省略した場合、定義値を 1 とします。 ccrh コマンドの -D オプションに相当します。 指定したマクロはサブプロパティとして表示します。 <table border="1"> <tr> <td>デフォルト</td><td>コンパイル・オプションの設定値</td></tr> <tr> <td>変更方法</td><td>[...] ボタンをクリックし、テキスト編集 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能</td></tr> <tr> <td>指定可能値</td><td>256 文字までの文字列 256 個まで指定可能です。</td></tr> </table>	デフォルト	コンパイル・オプションの設定値	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能	指定可能値	256 文字までの文字列 256 個まで指定可能です。
デフォルト	コンパイル・オプションの設定値						
変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能						
指定可能値	256 文字までの文字列 256 個まで指定可能です。						
定義解除マクロ	定義解除したいマクロ名を指定します。 「マクロ名」の形式で 1 行に 1 つずつ指定します。 ccrh コマンドの -U オプションに相当します。 指定したマクロはサブプロパティとして表示します。 <table border="1"> <tr> <td>デフォルト</td><td>コンパイル・オプションの設定値</td></tr> <tr> <td>変更方法</td><td>[...] ボタンをクリックし、テキスト編集 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能</td></tr> <tr> <td>指定可能値</td><td>256 文字までの文字列 256 個まで指定可能です。</td></tr> </table>	デフォルト	コンパイル・オプションの設定値	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能	指定可能値	256 文字までの文字列 256 個まで指定可能です。
デフォルト	コンパイル・オプションの設定値						
変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能						
指定可能値	256 文字までの文字列 256 個まで指定可能です。						

プリプロセス結果に C ソース・コメントを出力 する	プリプロセス結果のファイルに、C ソースのコメントを出力するかどうかを選択します。 ccrh コマンドの -Xpreprocess オプションに相当します。 なお、本プロパティは、 [出力ファイル] カテゴリの「プリプロセス処理したソースを出力する」プロパティで「はい (-P)」を選択した場合のみ表示します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xpreprocess=comment) プリプロセス結果のファイルに、C ソースのコメントを出力します。
		いいえ プリプロセス結果のファイルに、C ソースのコメントを出力しません。
プリプロセス結果に行番 号情報を出力する	プリプロセス結果のファイルに、C ソースの行番号情報を出力するかどうかを選択します。 ccrh コマンドの -Xpreprocess オプションに相当します。 なお、本プロパティは、 [出力ファイル] カテゴリの「プリプロセス処理したソースを出力する」プロパティで「はい (-P)」を選択した場合のみ表示します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xpreprocess=line) プリプロセス結果のファイルに、C ソースの行番号情報を出力します。
		いいえ プリプロセス結果のファイルに、C ソースの行番号情報を出力しません。

(5) [C 言語]

C 言語に関する詳細情報の表示、および設定の変更を行います。

ANSI 規格に厳密に合 わせてコンパイルする	C ソース・プログラムを ANSI 規格に厳密にあわせて処理し、規格に反する記述に対してエラーや警告を出力するかどうかを選択します。 ccrh コマンドの -Xansi オプションに相当します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xansi) C ソース・プログラムを ANSI 規格に厳密にあわせて処理し、規格に反する記述に対してエラーや警告を出力します。
		いいえ 従来の C 言語の仕様との両立性を持たせ、警告を出力して処理を続行します。

外部変数を volatile 化する	すべての外部変数を volatile 宣言したものとして扱うかどうかを選択します。 ccrh コマンドの -Xvolatile オプションに相当します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-Xvolatile)	すべての外部変数を volatile 宣言したものとして扱います。
		いいえ	volatile 修飾のある変数のみを volatile 宣言したものとして扱います。
C プログラムの互換性をチェックする	C プログラムの互換性をチェックするかどうかを選択します。 ccrh コマンドの -Xcheck オプションに相当します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (SuperH RISC engine C/C++ コンパイラ用) (-Xcheck=shc)	SuperH ファミリ用 C/C++ コンパイラとの互換性をチェックします。
		いいえ	既存プログラムとの互換性をチェックしません。

(6) [文字コード]

文字コードに関する詳細情報の表示、および設定の変更を行います。

文字コード	ソース・ファイル中の日本語／中国語のコメント、文字列に対して、使用する文字コードを選択します。 ccrh コマンドの -Xcharacter_set オプションに相当します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	自動 (なし)	ソース・ファイル中の日本語の文字コードを SJIS と解釈します。
		SJIS(-Xcharacter_set=sjis)	ソース・ファイル中の日本語の文字コードを SJIS と解釈します。
		EUC(-Xcharacter_set=euc_jp)	ソース・ファイル中の日本語の文字コードを EUC と解釈します。
		UFT-8(-Xcharacter_set=utf8)	ソース・ファイル中の日本語の文字コードを UFT-8 と解釈します。
		Big5(-Xcharacter_set=big5)	ソース・ファイル中の中国語の文字コードを繁体字中国語と解釈します。
		GB2312(-Xcharacter_set=gb2312)	ソース・ファイル中の中国語の文字コードを簡体字中国語と解釈します。
		処理しない (-Xcharacter_set=none)	ソース・ファイル中の日本語／中国語の文字コードを解釈しません。

(7) [出力コード]

出力コードに関する詳細情報の表示、および設定の変更を行います。

アセンブリ・ソース・ファイルにコメントを出力する	出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力するかどうかを選択します。 ccrh コマンドの -Xpass_source オプションに相当します。 なお、本プロパティは、 [出力ファイル] カテゴリの [アセンブリ・ソース・ファイルを出力する] プロパティで [はい (-Xasm_path)] を選択した場合、または [アセンブル・リスト] カテゴリの [アセンブル・リスト・ファイルを出力する] プロパティで [はい (-Xasm_option=-Xprn_path)] を選択した場合のみ表示します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xpass_source) 出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力します。
		いいえ 出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力しません。

switch 文の出力コードの 選択	プログラム中の switch 文のコード出力形式を選択します。 ccrh コマンドの -Xswitch オプションに相当します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	自動選択 (なし)	ccrh コマンドが最適な出力形式を自動的に選択します。
		if-else(-Xswitch=ifelse)	プログラム中の switch 文を case 文の並びに沿った if-else 文と同じ形式で出力します。 頻度が多い順に case 文を書いているときやラベル数が少ないときに、本項目を選択します。 上から順に比較するので、頻繁に合致する case 文を先に記述すると余計な比較が減り、実行速度向上につながります。
バイナリ・サーチ (-Xswitch=binary)		プログラム中の switch 文をバイナリ・サーチ形式で出力します。 バイナリ・サーチ・アルゴリズムに用いて合致する case 文を探します。 ラベル数が多いときに本項目を選択すると、どの case 文も同じくらいの速さで見つけることができます。	
	テーブル分岐 (-Xswitch=table)	プログラム中の switch 文をテーブル・ジャンプ形式で出力します。 case 文の値を基にインデックス化したテーブルを参照し、switch 文の値により case ラベルを選択して処理を行います。 どの case 文にも同じくらい速く分岐します。 ただし、case 値が連続していないときは無駄な領域ができます。	
浮動小数点演算方法	浮動小数点演算に対して、ランタイム・ライブラリの呼び出し命令を生成するか、FPU（浮動小数点ユニット）の浮動小数点演算命令を生成するかを選択します。 ccrh コマンドの -Xfloat オプションに相当します。 なお、本プロパティは、 【共通オプション】タブの【出力ファイルの種類と場所】 カテゴリの 【CPU コアの指定】 プロパティで 【G3M 向けオブジェクト (-Xcpu=g3m)】 を選択した場合のみ表示します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	自動選択 (なし)	浮動小数点演算命令を生成します。
		ソフトウェアで行う (-Xfloat=soft)	浮動小数点演算に対して、ランタイム・ライブラリの呼び出し命令を生成します。
FPU で行う (-Xfloat=fpu)		浮動小数点演算に対して、FPU の浮動小数点演算命令を生成します。	

div / divu 除算命令を生成する

除算に対して、divq、および divqu 命令を生成する代わりに、div、および divu 命令を生成するかどうかを選択します。

divq、および divqu 命令は高速ですが、実行サイクル数がオペランドの値により変わります。

ccrh コマンドの [-Xdiv](#) オプションに相当します。

デフォルト	コンパイル・オプションの設定値	
変更方法	ドロップダウン・リストによる選択	
指定可能値	はい (-Xdiv)	除算に対して、div、および divu 命令を生成します。
	いいえ	除算に対して、divq、および divqu 命令を生成します。

除算時の OV フラグ・チェック・コードを生成する

除算命令の後に OV フラグのチェックを行い、OV フラグが1のときに FE レベル・ソフトウェア例外を発生させるコード（fetrp 命令）を生成するかどうかを選択します。

ccrh コマンドの [-Xcheck_div_ov](#) オプションに相当します。

デフォルト	コンパイル・オプションの設定値	
変更方法	ドロップダウン・リストによる選択	
指定可能値	はい (-Xcheck_div_ov)	除算時に OV フラグのチェックを行うコードを生成します。
	いいえ	除算時に OV フラグのチェックを行わないコードを生成します。

除算時例外の fetrap 命令ベクタ番号

OV フラグが1のときに生成する fetrap 命令のベクタ番号を指定します。

ccrh コマンドの [-Xcheck_div_ov](#) オプションに相当します。

なお、本プロパティは、[除算時の OV フラグ・チェック・コードを生成する] プロパティで [はい (-Xcheck_div_ov)] を選択した場合のみ表示します。

デフォルト	コンパイル・オプションの設定値	
変更方法	テキスト・ボックスによる直接入力	
指定可能値	1 ~ 15（10 進数）	

積和演算命令を生成する

単精度浮動小数点積和演算に対して、積和演算命令（fmaf.s、fmsf.s、fnmaf.s、fnmsf.s）を生成するかどうかを選択します。

ccrh コマンドの [-Xuse_fmaf](#) オプションに相当します。

デフォルト	コンパイル・オプションの設定値	
変更方法	ドロップダウン・リストによる選択	
指定可能値	はい (-Xuse_fmaf)	単精度浮動小数点積和演算に対して、積和演算命令を生成します。 実行速度は上がりますが、演算精度が変わります。
	いいえ	積和演算命令を生成しません。

浮動小数点比較で無効演算例外を発生させる	浮動小数点比較において、比較する値に非数が含まれている場合に無効演算例外を発生する比較条件を使用するコードを生成するかどうかを選択します。 ccrh コマンドの -Xunordered_cmpf オプションに相当します。			
	デフォルト	コンパイル・オプションの設定値		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (-Xunordered_cmpf)	浮動小数点比較において、比較する値に非数が含まれている場合に無効演算例外を発生する比較条件を使用するコードを生成します。	
		いいえ	浮動小数点比較において、無効演算例外の検出を行いません。	
32 ビット分岐命令を使用する	jarl、および jr 命令に対して、far jump 機能を使用するかどうかを選択します。 far jump 機能を使用することにより、jarl、および jr 命令を jarl32、および jr32 命令とみなしてアセンブルを行います。 ccrh コマンドの -Xasm_option=-Xasm_far_jump オプションに相当します。 なお、本プロパティは、 【共通オプション】タブの【ビルド方法】 カテゴリの「一括ビルドを行う」プロパティで「はい」を選択した場合のみ表示します。			
	デフォルト	コンパイル・オプションの設定値		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	はい (-Xasm_option=-Xasm_far_jump)	jarl、および jr 命令を jarl32、および jr32 命令とみなしてアセンブルを行います。	
		いいえ	jarl、および jr 命令としてアセンブルを行います。	
関数呼び出し命令を指定する	関数呼び出しの分岐に対して、生成する命令を選択します。 ccrh コマンドの -Xcall_jump オプションに相当します。			
	デフォルト	コンパイル・オプションの設定値		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	jarl32 および jr32 命令を生成 (-Xcall_jump=32)	関数呼び出しの分岐に対して、jarl32、および jr32 命令を生成します。	
		jarl および jr 命令を生成 (なし)	関数呼び出しの分岐に対して、jarl、および jr 命令を生成します。	
Far Jump ファイル名	Far Jump ファイル名を指定します。 Far Jump ファイルには、ファイルに記述した関数への分岐命令に対して、jarl32、および jr32 命令を使用したコードを出力します。 関数本体が jarl、および jr 命令では分岐できない範囲（± 2M バイト以上）にあり、ccrh コマンドがエラーを出力する場合、Far Jump ファイルを用いてコンパイルし直します。 なお、拡張子は“.fjp”としてください。 ccrh コマンドの -Xfar_jump オプションに相当します。			
	デフォルト	コンパイル・オプションの設定値		
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 Far Jump ファイルを指定 ダイアログ による編集		
	指定可能値	259 文字までの文字列		

生成するプログラムの種類	生成するプログラムの種類を選択します。 ccrh コマンドの -Xmulti_level オプションに相当します。 なお、本プロパティは、マルチコア用プロジェクトの場合のみ表示します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	シングルコア用のプログラムを生成する(なし)	シングルコア用プログラムを生成します。 プログラム中の #pragma pmodule 指定は無視します。
		マルチコア用のプログラムを生成する(-Xmulti_level=1)	マルチコア用プログラムを生成します。 プログラム中の #pragma pmodule 指定が有効となり、セクション名の末尾に PM 番号が付加されます。

(8) [出力ファイル]

出力ファイルに関する詳細情報の表示、および設定の変更を行います。

オブジェクト・ファイル名	コンパイル後に生成するオブジェクト・ファイルの名前を指定します。 “ <code>.obj</code> ” 以外の拡張子を指定することはできません。 拡張子を省略した場合は、“ <code>.obj</code> ” を自動的に付加します。 空欄の場合は、ソース・ファイル名の拡張子を“ <code>.obj</code> ” に置き換えたものとなります。 ccrh コマンドの <code>-o</code> オプションに相当します。		
	デフォルト	空欄	
	変更方法	テキスト・ボックスによる直接入力	
	指定可能値	259 文字までの文字列	
アセンブリ・ソース・ファイルを出力する	C ソースのコンパイル結果のアセンブリ・ソース・ファイルを出力するかどうかを選択します。 ccrh コマンドの <code>-Xasm_path</code> オプションに相当します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (<code>-Xasm_path</code>)	C ソースのコンパイル結果のアセンブリ・ソース・ファイルを出力します。
		いいえ	C ソースのコンパイル結果のアセンブリ・ソース・ファイルを出力しません。

アセンブリ・ソース・ファイル出力フォルダ	アセンブリ・ソース・ファイルの出力先フォルダを指定します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 アセンブリ・ソース・ファイルは、C ソース・ファイルの拡張子を“.asm”で置き換えたファイル名で出力します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの -Xasm_path オプションに相当します。 なお、本プロパティは、[アセンブリ・ソース・ファイルを出力する] プロパティで [はい (-Xasm_path)] を選択した場合のみ表示します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集	
	指定可能値	247 文字までの文字列	
プリプロセス処理したソースを出力する	ソース・ファイルに対して、プリプロセス処理を実行した結果をファイルに出力するかどうかを選択します。 ccrh コマンドの -P オプションに相当します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-P)	ソース・ファイルに対して、プリプロセス処理を実行した結果をファイルに出力します。
		いいえ	ソース・ファイルに対して、プリプロセス処理を実行した結果をファイルに出力しません。
プリプロセス処理したソース・ファイル出力フォルダ	プリプロセス処理したソース・ファイルの出力先フォルダを指定します。 ファイルは、ソース・ファイルの拡張子を“.i”で置き換えた名前でも出力します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの -Xprep_path オプションに相当します。 なお、本プロパティは、[プリプロセス処理したソースを出力する] プロパティで [はい (-P)] を選択した場合のみ表示します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集	
	指定可能値	247 文字までの文字列	

(9) [アセンブル・リスト]

アセンブル・リストに関する詳細情報の表示、および設定の変更を行います。

アセンブル・リスト・ ファイルを出力する	アセンブル・リスト・ファイルを出力するかどうかを選択します。 ccrh コマンドの -Xasm_option=-Xprn_path オプションに相当します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xasm_option=-Xprn_path) アセンブル・リスト・ファイルを出力します。 いいえ アセンブル・リスト・ファイルを出力しません。
アセンブル・リスト・ ファイル出力フォルダ	アセンブル・リスト・ファイルの出力先フォルダを指定します。 アセンブル・リスト・ファイルは、ソース・ファイル名の拡張子を“.prn”で置き換えた名前で出力します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの -Xasm_option=-Xprn_path オプションに相当します。 なお、本プロパティは、[アセンブル・リスト・ファイルを出力する] プロパティで [はい (-Xasm_option=-Xprn_path)] を選択した場合のみ表示します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列

(10) [MISRA-C:2004 ルール検査]

MISRA-C:2004 ルール検査に関する詳細情報の表示、および設定の変更を行います。

適用するルール	適用する MISRA-C:2004 ルールを選択します。 ccrh コマンドの -Xmisra2004 オプションに相当します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	すべてのルールを適用 (-Xmisra2004=all)	サポートしているすべてのルールをチェック対象とします。
		指定したルール番号を適用 (-Xmisra2004=apply)	サポートしているルールのうち、指定したルール番号をチェック対象とします。
		指定したルール番号を除外 (-Xmisra2004=ignore)	サポートしているルールのうち、指定したルール番号以外のルールをチェック対象とします。
		必須ルールを適用 (-Xmisra2004=required)	サポートしているルールのうち、ルールの分類が“required”になっているルールをチェック対象とします。
		必須ルールと指定したルールを適用 (-Xmisra2004=required_add)	サポートしているルールのうち、ルールの分類が“required”になっているルールと指定したルール番号をチェック対象とします。
		必須ルールから指定したルール番号を除外 (-Xmisra2004=required_remove)	サポートしているルールのうち、ルールの分類が“required”になっているルールから指定したルール番号を除いたルール番号をチェック対象とします。
		指定ファイルに記載されたルール番号を適用 (-misra2004=<ファイル名>)	サポートしているルールのうち、指定したファイル名に記載したルール番号をチェック対象とします。
適用ルールなし (なし)		MISRA-C:2004 ルールを適用しません。	
ルール番号記載ファイル	ルール番号記載ファイル（MISRA-C:2004 ルール・ファイル）を指定します。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 %ProjectName% : プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 ccrh コマンドの -Xmisra2004 オプションに相当します。 なお、本プロパティは、[適用するルール] プロパティで [指定ファイルに記載されたルール番号を適用 (-misra2004=<ファイル名>)] を選択した場合のみ表示します。		
	デフォルト	コンパイル・オプションの設定値	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 MISRA-C:2004 ルール・ファイルの指定 ダイアログによる編集	
	指定可能値	259 文字までの文字列	

ルール番号	チェック対象とするルール番号を指定します。 ルール番号は、必ず 10 進数で 1 つ以上指定してください。 ccrh コマンドの -Xmisra2004 オプションに相当します。 なお、本プロパティは、[適用するルール] プロパティで [指定したルール番号を適用 (-Xmisra2004=apply)] を選択した場合のみ表示します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 ルール番号の指定 ダイアログ による編集
	指定可能値	259 文字までの文字列
除外するルール番号	チェック対象から除外するルール番号を指定します。 ルール番号は、必ず 10 進数で 1 つ以上指定してください。 ccrh コマンドの -Xmisra2004 オプションに相当します。 なお、本プロパティは、[適用するルール] プロパティで [指定したルール番号を除外 (-Xmisra2004=ignore)] を選択した場合のみ表示します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 ルール番号の指定 ダイアログ による編集
	指定可能値	259 文字までの文字列
必須ルールの他にチェックするルール番号	必須ルールのほかにチェック対象とするルール番号を指定します。 ルール番号は、必ず 10 進数で 1 つ以上指定してください。 ccrh コマンドの -Xmisra2004 オプションに相当します。 なお、本プロパティは、[必須ルールと指定したルールを適用 (-Xmisra2004=required_add)] を選択した場合のみ表示します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 ルール番号の指定 ダイアログ による編集
	指定可能値	259 文字までの文字列
必須ルールから除外するルール番号	必須ルールのうち、チェック対象から除外するルール番号を指定します。 ルール番号は、必ず 10 進数で 1 つ以上指定してください。 ccrh コマンドの -Xmisra2004 オプションに相当します。 なお、本プロパティは、[適用するルール] プロパティで [必須ルールから指定したルール番号を除外 (-Xmisra2004=required_remove)] を選択した場合のみ表示します。	
	デフォルト	コンパイル・オプションの設定値
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 ルール番号の指定 ダイアログ による編集
	指定可能値	259 文字までの文字列

ルール・チェック対象外のファイル	MISRA-C2004 ルールのチェック対象外のファイルを指定します。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 %ProjectName% : プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 ccrh コマンドの <code>-Xignore_files_misra</code> オプションに相当します。 なお、本プロパティは、以下の場合のみ表示します。 - [適用するルール] プロパティで [すべてのルールを適用] を選択した場合 - [適用するルール] プロパティで [必須ルールを適用] を選択した場合 - [適用するルール] プロパティで [指定したルール番号を適用] を選択し、かつ [ルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [指定したルール番号を除外] を選択し、かつ [除外するルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [必須ルールと指定したルールを適用] を選択し、かつ [必須ルールの他にチェックするルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [必須ルールから指定したルール番号を除外] を選択し、かつ [必須ルールから除外するルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [指定ファイルに記載されたルール番号を適用] を選択し、かつ [ルール番号記載ファイル] プロパティにルール番号記載ファイルを指定した場合	
	デフォルト	コンパイル・オプションの設定値
	変更方法	[...] ボタンをクリックし、パス編集 ダイアログをオープン → [参照] ボタンをクリックし、対象外ファイルの追加 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	259 文字までの文字列

拡張キーワードや拡張仕様をメッセージ出力する	拡張キーワードや拡張仕様をメッセージ出力するかどうかを選択します。 ccrh コマンドの <code>-Xcheck_language_extention</code> オプションに相当します。 なお、本プロパティは、以下の場合のみ表示します。 - [適用するルール] プロパティで [すべてのルールを適用] を選択した場合 - [適用するルール] プロパティで [必須ルールを適用] を選択した場合 - [適用するルール] プロパティで [指定したルール番号を適用] を選択し、かつ [ルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [指定したルール番号を除外] を選択し、かつ [除外するルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [必須ルールと指定したルールを適用] を選択し、かつ [必須ルールの他にチェックするルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [必須ルールから指定したルール番号を除外] を選択し、かつ [必須ルールから除外するルール番号] プロパティにルール番号を指定した場合 - [適用するルール] プロパティで [指定ファイルに記載されたルール番号を適用] を選択し、かつ [ルール番号記載ファイル] プロパティにルール番号記載ファイルを指定した場合	
デフォルト	コンパイル・オプションの設定値	
変更方法	ドロップダウン・リストによる選択	
指定可能値	はい (-Xcheck_language_extention)	C 言語規格から独自に拡張した言語仕様のために MISRA-C:2004 ルール・チェックが部分抑止される場合にルール・チェックを有効にして、メッセージを出力します。
	いいえ	言語拡張により部分抑止される MISRA-C:2004 ルール・チェックを無効にします。

(11) [エラー出力]

エラー出力に関する詳細情報の表示、および設定の変更を行います。

エラー・メッセージ・ ファイルを出力する	エラー・メッセージ・ファイルを出力するかどうかを選択します。 ccrh コマンドの -Xerror_file オプションに相当します。 なお、本プロパティの選択にかかわらず、エラー・メッセージは 出力 パネル に表示します。				
	デフォルト	共通オプションの設定値			
	変更方法	ドロップダウン・リストによる選択			
	指定可能値	<table><tr><td>はい (-Xerror_file)</td><td>エラー・メッセージ・ファイルを出力します。</td></tr><tr><td>いいえ</td><td>エラー・メッセージ・ファイルを出力しません。</td></tr></table>	はい (-Xerror_file)	エラー・メッセージ・ファイルを出力します。	いいえ
はい (-Xerror_file)	エラー・メッセージ・ファイルを出力します。				
いいえ	エラー・メッセージ・ファイルを出力しません。				

エラー・メッセージ・ ファイル出力フォルダ	エラー・メッセージ・ファイルの出力先フォルダを指定します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの -Xerror_file オプションに相当します。 なお、本プロパティは、[エラー・メッセージ・ファイルを出力する] プロパティで [はい (-Xerror_file)] を選択した場合のみ表示します。	
	デフォルト	共通オプションの設定値
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列
エラー・メッセージ・ ファイル名	エラー・メッセージ・ファイルの名前を指定します。 拡張子は自由に指定可能です。 次のプレースホルダに対応しています。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %ProjectName% : プロジェクト名に置換します。 空欄の場合は、%ProjectName%.err を指定したものとみなします。 ccrh コマンドの -Xerror_file オプションに相当します。 なお、本プロパティは、[エラー・メッセージ・ファイルを出力する] プロパティで [はい (-Xerror_file)] を選択した場合のみ表示します。	
	デフォルト	共通オプションの設定値
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	259 文字までの文字列

(12) [警告メッセージ]

警告メッセージに関する詳細情報の表示、および設定の変更を行います。

表示させない警告メッ セージ	表示させない警告メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例：02042,02107）。 また、ハイフンを使用して、区間設定を行うこともできます（例：02222-02554,02699-02782）。 ccrh コマンドの -Xno_warning オプションに相当します。	
	デフォルト	共通オプションの設定値
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集
	指定可能値	2048 文字までの文字列

(13) [その他]

コンパイルに関するその他の詳細情報の表示、および設定の変更を行います。

コンパイル前に実行する コマンド	コンパイル処理前に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %CompiledFile% : コンパイル時の出力ファイルの絶対パスに置換します。 %InputFile% : コンパイル対象ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、コンパイル処理前に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。
デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 空欄 - 上記以外の場合 <i>コンパイル・オプションの設定値</i>
変更方法	[...] ボタンをクリックし、テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
指定可能値	1023 文字までの文字列 64 個まで指定可能です。

コンパイル後に実行する コマンド	コンパイル処理後に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %CompiledFile% : コンパイル時の出力ファイルの絶対パスに置換します。 %InputFile% : コンパイル対象ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、コンパイル処理後に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。						
	<table> <tr> <td data-bbox="536 1238 671 1435">デフォルト</td><td data-bbox="671 1238 1394 1435"> - [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 空欄 - 上記以外の場合 <i>コンパイル・オプションの設定値</i> </td></tr> <tr> <td data-bbox="536 1435 671 1525">変更方法</td><td data-bbox="671 1435 1394 1525">[...] ボタンをクリックし、テキスト編集 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能</td></tr> <tr> <td data-bbox="536 1525 671 1603">指定可能値</td><td data-bbox="671 1525 1394 1603">1023 文字までの文字列 64 個まで指定可能です。</td></tr> </table>	デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 空欄 - 上記以外の場合 <i>コンパイル・オプションの設定値</i>	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能	指定可能値	1023 文字までの文字列 64 個まで指定可能です。
デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 空欄 - 上記以外の場合 <i>コンパイル・オプションの設定値</i>						
変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能						
指定可能値	1023 文字までの文字列 64 個まで指定可能です。						
その他の追加オプション	その他に追加するコンパイル・オプションを入力します。 ここで設定したオプションは、コンパイル・オプション群の最後に付加します。 <table> <tr> <td data-bbox="536 1704 671 1906">デフォルト</td><td data-bbox="671 1704 1394 1906"> - [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 空欄 - 上記以外の場合 <i>コンパイル・オプションの設定値</i> </td></tr> <tr> <td data-bbox="536 1906 671 1995">変更方法</td><td data-bbox="671 1906 1394 1995">テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、文字列入力 ダイアログによる編集</td></tr> <tr> <td data-bbox="536 1995 671 2038">指定可能値</td><td data-bbox="671 1995 1394 2038">259 文字までの文字列</td></tr> </table>	デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 空欄 - 上記以外の場合 <i>コンパイル・オプションの設定値</i>	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集	指定可能値	259 文字までの文字列
デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 空欄 - 上記以外の場合 <i>コンパイル・オプションの設定値</i>						
変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集						
指定可能値	259 文字までの文字列						

【個別アセンブル・オプション】タブ

本タブでは、1つのアセンブリ・ソース・ファイルに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

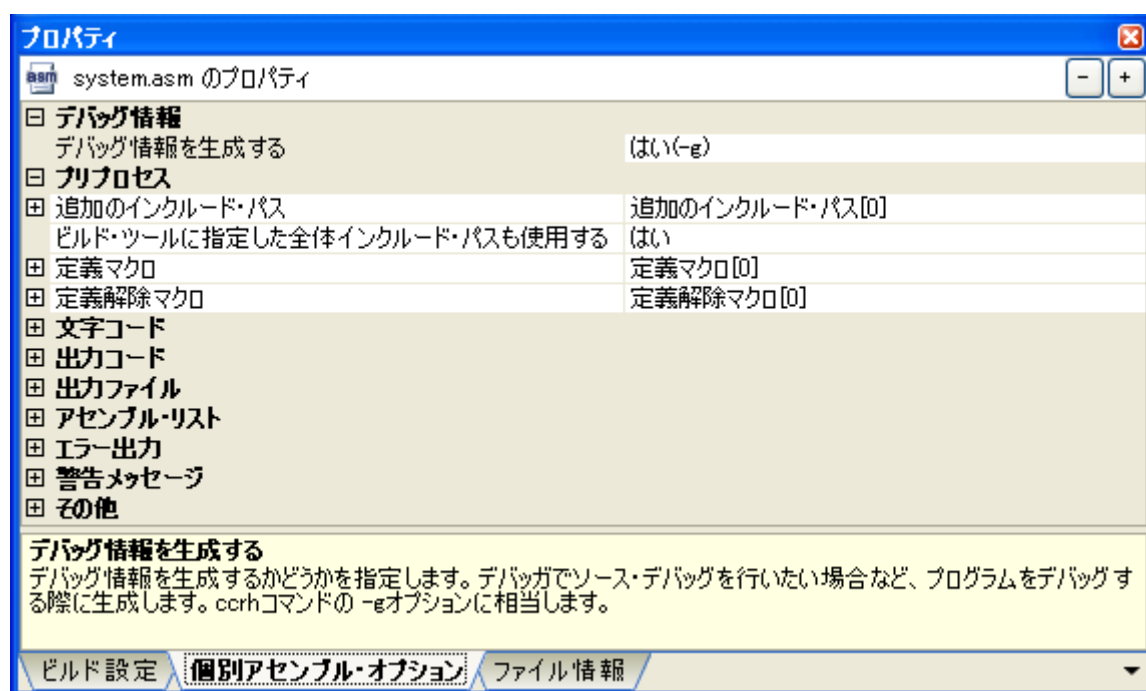
なお、本タブは、[【共通オプション】タブ](#)、[【コンパイル・オプション】タブ](#)、および[【アセンブル・オプション】タブ](#)の設定内容を継承します。

これらのタブと異なる値を設定した場合は、プロパティが太字表示となります。

- (1) [【デバッグ情報】](#)
- (2) [【プリプロセス】](#)
- (3) [【文字コード】](#)
- (4) [【出力コード】](#)
- (5) [【出力ファイル】](#)
- (6) [【アセンブル・リスト】](#)
- (7) [【エラー出力】](#)
- (8) [【警告メッセージ】](#)
- (9) [【その他】](#)

備考 本タブは、[【ビルド設定】タブ](#)の[【ビルド】](#)カテゴリの[【個別アセンブル・オプションを設定する】](#)プロパティで「はい」を選択した場合のみ表示します。

図 A—15 プロパティ パネル：【個別アセンブル・オプション】タブ



[各カテゴリの説明]

(1) [デバッグ情報]

デバッグ情報に関する詳細情報の表示、および設定の変更を行います。

デバッグ情報を生成する	デバッグ情報を生成するかどうかを選択します。 出力ファイル中にソース・デバッグ用の情報を出力することにより、デバッガでのソース・デバッグが可能となります。 ccrh コマンドの -g オプションに相当します。	
	デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 コンパイル・オプションの設定値 - 上記以外の場合 アセンブル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-g) デバッグ情報を生成します。 いいえ デバッグ情報を生成しません。

(2) [プリプロセス]

プリプロセスに関する詳細情報の表示、および設定の変更を行います。

追加のインクルード・パス	アセンブル時の追加のインクルード・パスを指定します。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %BuildModeName% : ビルド・モード名に置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 指定したインクルード・パスは、CC-RH の標準インクルード・ファイル・フォルダよりも優先して検索します。 パスはプロジェクト・フォルダを基点とします。 本プロパティを省略した場合は、CC-RH の標準インクルード・ファイル・フォルダのみ検索します。 ccrh コマンドの -I オプションに相当します。 指定したインクルード・パスはサブプロパティとして表示します。 インクルード・パスに大文字、小文字の区別はありません。	
	デフォルト	追加のインクルード・パス [定義数]
	変更方法	[...] ボタンをクリックし、 パス編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	259 文字までの文字列 256 個まで指定可能です。

ビルド・ツールに指定した全体インクルード・パスも使用する	使用するビルド・ツールの 【アセンブル・オプション】 タブの 【プリプロセス】 カテゴリの 【追加のインクルード・パス】 プロパティで指定したインクルード・パスも使用してアセンブルするかどうかを選択します。 なお、【共通オプション】 タブの 【ビルド方法】 カテゴリの 【一括ビルドを行う】 プロパティで 【はい】 を選択した場合は、【コンパイル・オプション】 タブの設定を使用します。 【一括ビルドを行う】 プロパティが 【はい】 の場合、インクルード・パスは、以下の順で追加します。 - 本タブの 【追加のインクルード・パス】 プロパティで指定したパス - 【コンパイル・オプション】 タブの 【追加のインクルード・パス】 プロパティで指定したパス - 【コンパイル・オプション】 タブの 【システム・インクルード・パス】 プロパティに表示しているパス 【一括ビルドを行う】 プロパティが 【いいえ】 の場合、インクルード・パスは、以下の順で追加します。 - 本タブの 【追加のインクルード・パス】 プロパティで指定したパス - 【アセンブル・オプション】 タブの 【追加のインクルード・パス】 プロパティで指定したパス - 【アセンブル・オプション】 タブの 【システム・インクルード・パス】 プロパティに表示しているパス ccrh コマンドの -I オプションに相当します。	
デフォルト	はい	
変更方法	ドロップダウン・リストによる選択	
指定可能値	はい	使用するビルド・ツールのプロパティで指定したインクルード・パスも使用してアセンブルします。
	いいえ	使用するビルド・ツールのプロパティで指定したインクルード・パスを使用しません。
定義マクロ	定義したいマクロ名を指定します。 「マクロ名 = 定義値」の形式で 1 行に 1 つずつ指定します。 「= 定義値」の部分は省略可能で、省略した場合、定義値を 1 とします。 ccrh コマンドの -D オプションに相当します。 指定したマクロはサブプロパティとして表示します。	
デフォルト	- 【共通オプション】 タブの 【ビルド方法】 カテゴリの 【一括ビルドを行う】 プロパティで 【はい】 を選択した場合 コンパイル・オプションの設定値 - 上記以外の場合 アセンブル・オプションの設定値	
変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能	
指定可能値	256 文字までの文字列 256 個まで指定可能です。	

定義解除マクロ	定義解除したいマクロ名を指定します。 「マクロ名」の形式で1行に1つつ指定します。 ccrh コマンドの -U オプションに相当します。 指定したマクロはサブプロパティとして表示します。	
	デフォルト	- [共通オプション] タブ の [ビルド方法] カテゴリの「一括ビルドを行う」プロパティで「はい」を選択した場合 コンパイル・オプションの設定値 - 上記以外の場合 アセンブル・オプションの設定値
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

(3) [文字コード]

文字コードに関する詳細情報の表示、および設定の変更を行います。

文字コード	ソース・ファイル中の日本語／中国語のコメント、文字列に対して、使用する文字コードを選択します。 ccrh コマンドの -Xcharacter_set オプションに相当します。		
	デフォルト	- [共通オプション] タブ の [ビルド方法] カテゴリの「一括ビルドを行う」プロパティで「はい」を選択した場合 コンパイル・オプションの設定値 - 上記以外の場合 アセンブル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	自動 (なし)	ソース・ファイル中の日本語の文字コードを SJIS と解釈します。
		SJIS(-Xcharacter_set=sjis)	ソース・ファイル中の日本語の文字コードを SJIS と解釈します。
		EUC(-Xcharacter_set=euc_jp)	ソース・ファイル中の日本語の文字コードを EUC と解釈します。
		UFT-8(-Xcharacter_set=utf8)	ソース・ファイル中の日本語の文字コードを UFT-8 と解釈します。
		Big5(-Xcharacter_set=big5)	ソース・ファイル中の中国語の文字コードを繁体字中国語と解釈します。
GB2312(-Xcharacter_set=gb2312)		ソース・ファイル中の中国語の文字コードを簡体字中国語と解釈します。	
	処理しない (-Xcharacter_set=none)	ソース・ファイル中の日本語／中国語の文字コードを解釈しません。	

(4) [出力コード]

出力コードに関する詳細情報の表示、および設定の変更を行います。

32 ビット分岐命令を使用する	jarl, および jr 命令に対して、far jump 機能を使用するかどうかを選択します。 far jump 機能を使用することにより、jarl, および jr 命令を jarl32, および jr32 命令とみなしてアセンブルを行います。 ccrh コマンドの <code>-Xasm_option=-Xasm_far_jump</code> オプションに相当します。	
	デフォルト	- 【共通オプション】 タブの 【ビルド方法】 カテゴリの 【一括ビルドを行う】 プロパティで 【はい】 を選択した場合 コンパイル・オプションの設定値 - 上記以外の場合 アセンブル・オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xasm_option=-Xasm_far_jump) jarl, および jr 命令を jarl32, および jr32 命令とみなしてアセンブルを行います。 いいえ jarl, および jr 命令としてアセンブルを行います。

(5) [出力ファイル]

出力ファイルに関する詳細情報の表示、および設定の変更を行います。

オブジェクト・ファイル名	アセンブル後に生成するオブジェクト・ファイルの名前を指定します。 “.obj” 以外の拡張子を指定することはできません。 拡張子を省略した場合は、“.obj” を自動的に付加します。 空欄の場合は、ソース・ファイル名の拡張子を “.obj” に置き換えたものとなります。 ccrh コマンドの <code>-o</code> オプションに相当します。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	259 文字までの文字列

(6) [アセンブル・リスト]

アセンブル・リストに関する詳細情報の表示、および設定の変更を行います。

アセンブル・リスト・ファイル を出力する	アセンブル・リスト・ファイルを出力するかどうかを選択します。 ccrh コマンドの -Xasm_option=-Xprn_path オプションに相当します。		
	デフォルト	- [共通オプション] タブ の [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 コンパイル・オプションの設定値 - 上記以外の場合 アセンブル・オプションの設定値	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい (-Xasm_option=-Xprn_path)	アセンブル・リスト・ファイルを出力します。
		いいえ	アセンブル・リスト・ファイルを出力しません。
アセンブル・リスト・ファイル出力フォルダ	アセンブル・リスト・ファイルの出力先フォルダを指定します。 アセンブル・リスト・ファイルは、ソース・ファイル名の拡張子を“.prn”で置き換えた名前で出力します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの -Xasm_option=-Xprn_path オプションに相当します。 なお、本プロパティは、 [アセンブル・リスト・ファイルを出力する] プロパティで [はい (-Xasm_option=-Xprn_path)] を選択した場合のみ表示します。		
	デフォルト	- [共通オプション] タブ の [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 コンパイル・オプションの設定値 - 上記以外の場合 アセンブル・オプションの設定値	
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタン をクリックし、 フォルダの参照 ダイアログ による編集	
	指定可能値	247 文字までの文字列	

(7) [エラー出力]

エラー出力に関する詳細情報の表示、および設定の変更を行います。

エラー・メッセージ・ファイルを出力する	エラー・メッセージ・ファイルを出力するかどうかを選択します。 ccrh コマンドの -Xerror_file オプションに相当します。 なお、本プロパティの選択にかかわらず、エラー・メッセージは 出力パネル に表示します。	
	デフォルト	共通オプションの設定値
	変更方法	ドロップダウン・リストによる選択
	指定可能値	はい (-Xerror_file) エラー・メッセージ・ファイルを出力します。
		いいえ エラー・メッセージ・ファイルを出力しません。
エラー・メッセージ・ファイル出力フォルダ	エラー・メッセージ・ファイルの出力先フォルダを指定します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 絶対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とした相対パスに変換します（ドライブが異なる場合を除く）。 次のプレースホルダに対応しています。 %BuildModeName% : ビルド・モード名に置換します。 空欄の場合は、プロジェクト・フォルダを指定したものとみなします。 ccrh コマンドの -Xerror_file オプションに相当します。 なお、本プロパティは、[エラー・メッセージ・ファイルを出力する] プロパティで [はい (-Xerror_file)] を選択した場合のみ表示します。	
	デフォルト	共通オプションの設定値
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	
	247 文字までの文字列	
エラー・メッセージ・ファイル名	エラー・メッセージ・ファイルの名前を指定します。 拡張子は自由に指定可能です。 次のプレースホルダに対応しています。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %ProjectName% : プロジェクト名に置換します。 空欄の場合は、%ProjectName%.err を指定したものとみなします。 ccrh コマンドの -Xerror_file オプションに相当します。 なお、本プロパティは、[エラー・メッセージ・ファイルを出力する] プロパティで [はい (-Xerror_file)] を選択した場合のみ表示します。	
	デフォルト	共通オプションの設定値
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	
	259 文字までの文字列	

(8) [警告メッセージ]

警告メッセージに関する詳細情報の表示、および設定の変更を行います。

表示させない警告メッセージ	表示させない警告メッセージの番号を指定します。 複数指定する場合は、メッセージ番号をカンマで区切って指定します（例：02042,02107）。 また、ハイフンを使用して、区間設定を行うこともできます（例：02222-02554,02699-02782）。 ccrh コマンドの -Xno_warning オプションに相当します。	
	デフォルト	共通オプションの設定値
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 文字列入力 ダイアログ による編集
	指定可能値	2048 文字までの文字列

(9) [その他]

アセンブルに関するその他の詳細情報の表示、および設定の変更を行います。

アセンブル前に実行する コマンド	アセンブル処理前に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %AssembledFile% : アセンブル時の出力ファイルの絶対パスに置換します。 %BuildModeName% : ビルド・モード名に置換します。 %InputFile% : アセンブル対象ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、アセンブル処理前に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。
デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 空欄 - 上記以外の場合 <i>アセンブル・オプションの設定値</i>
変更方法	[...] ボタンをクリックし、テキスト編集 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能
指定可能値	1023 文字までの文字列 64 個まで指定可能です。

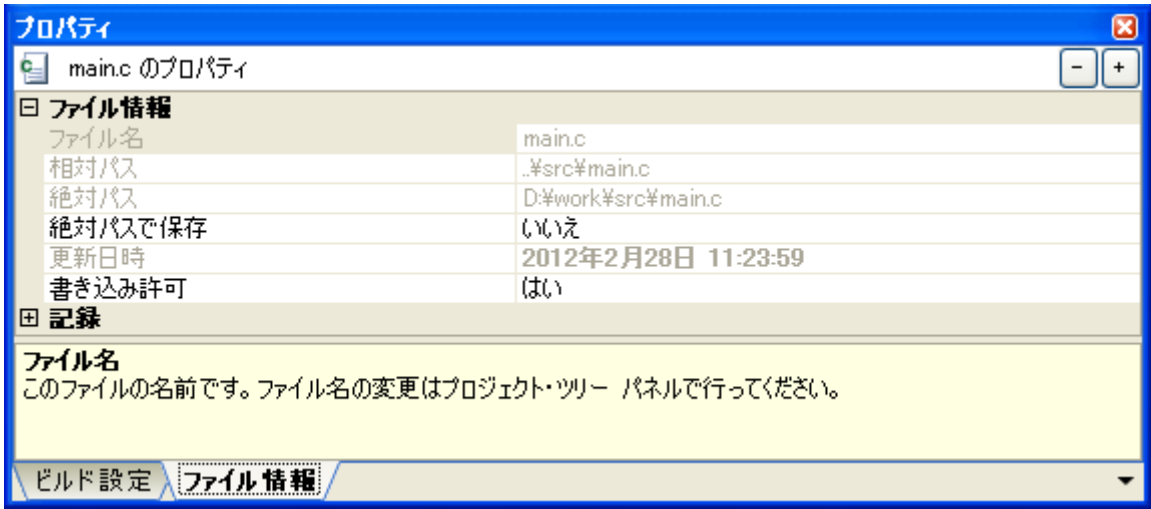
アセンブル後に実行するコマンド	アセンブル処理後に実行するコマンドを指定します。 バッチファイルを指定する場合は、call 命令を使用してください（例：call a.bat）。 次のプレースホルダに対応しています。 %ActiveProjectDir% : アクティブ・プロジェクト・フォルダの絶対パスに置換します。 %ActiveProjectName% : アクティブ・プロジェクト名に置換します。 %AssembledFile% : アセンブル時の出力ファイルの絶対パスに置換します。 %BuildModeName% : ビルド・モード名に置換します。 %InputFile% : アセンブル対象ファイルの絶対パスに置換します。 %MainProjectDir% : メイン・プロジェクト・フォルダの絶対パスに置換します。 %MainProjectName% : メイン・プロジェクト名に置換します。 %MicomToolPath% : 本製品のインストール・フォルダの絶対パスに置換します。 %Options% : ビルド実行時のコマンド・ライン・オプションに置換します。 %OutputDir% : 出力フォルダの絶対パスに置換します。 %OutputFile% : 出力ファイルの絶対パスに置換します。 %Program% : 実行中のプログラム名に置換します。 %ProjectDir% : プロジェクト・フォルダの絶対パスに置換します。 %ProjectName% : プロジェクト名に置換します。 %TempDir% : テンポラリ・フォルダの絶対パスに置換します。 %WinDir% : Windows システム・フォルダの絶対パスに置換します。 先頭行に“#!python”と記述すると、2 行目から最終行までの内容を Python コンソールのスクリプトと判断し、アセンブル処理後に Python コンソールで実行します。 なお、スクリプト中にはプレースホルダの記述も可能です。 指定したコマンドはサブプロパティとして表示します。						
	<table> <tr> <td data-bbox="536 1238 671 1435">デフォルト</td><td data-bbox="671 1238 1394 1435"> - [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 - 空欄 - 上記以外の場合 <i>アセンブル・オプションの設定値</i> </td></tr> <tr> <td data-bbox="536 1435 671 1525">変更方法</td><td data-bbox="671 1435 1394 1525"> [...] ボタンをクリックし、テキスト編集 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能 </td></tr> <tr> <td data-bbox="536 1525 671 1608">指定可能値</td><td data-bbox="671 1525 1394 1608"> 1023 文字までの文字列 64 個まで指定可能です。 </td></tr> </table>	デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 - 空欄 - 上記以外の場合 <i>アセンブル・オプションの設定値</i>	変更方法	[...] ボタンをクリックし、テキスト編集 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能	指定可能値	1023 文字までの文字列 64 個まで指定可能です。
デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 - 空欄 - 上記以外の場合 <i>アセンブル・オプションの設定値</i>						
変更方法	[...] ボタンをクリックし、テキスト編集 ダイアログによる編集 サブプロパティはテキスト・ボックスによる直接入力も可能						
指定可能値	1023 文字までの文字列 64 個まで指定可能です。						
その他の追加オプション	その他に追加するアセンブル・オプションを入力します。 ここで設定したオプションは、アセンブル・オプション群の最後に付加します。 <table> <tr> <td data-bbox="536 1709 671 1906">デフォルト</td><td data-bbox="671 1709 1394 1906"> - [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 - 空欄 - 上記以外の場合 <i>アセンブル・オプションの設定値</i> </td></tr> <tr> <td data-bbox="536 1906 671 1995">変更方法</td><td data-bbox="671 1906 1394 1995"> テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、文字列入力 ダイアログによる編集 </td></tr> <tr> <td data-bbox="536 1995 671 2038">指定可能値</td><td data-bbox="671 1995 1394 2038"> 259 文字までの文字列 </td></tr> </table>	デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 - 空欄 - 上記以外の場合 <i>アセンブル・オプションの設定値</i>	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、文字列入力 ダイアログによる編集	指定可能値	259 文字までの文字列
デフォルト	- [共通オプション] タブの [ビルド方法] カテゴリの [一括ビルドを行う] プロパティで [はい] を選択した場合 - 空欄 - 上記以外の場合 <i>アセンブル・オプションの設定値</i>						
変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、文字列入力 ダイアログによる編集						
指定可能値	259 文字までの文字列						

[ファイル情報] タブ

本タブでは、各ファイルに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

- (1) [ファイル情報]
- (2) [記録]

図 A—16 プロパティ パネル：[ファイル情報] タブ



[各カテゴリの説明]

- (1) [ファイル情報]

ファイルに関する詳細情報の表示、および設定の変更を行います。

ファイル名	ファイル名を表示します。 ファイル名の変更は、プロジェクト・ツリーで行ってください。	
	デフォルト	ファイル名
	変更方法	変更不可
相対パス	ファイルのプロジェクト・フォルダからの相対パス名を表示します。	
	デフォルト	ファイルのプロジェクト・フォルダからの相対パス名
	変更方法	変更不可
絶対パス	ファイルの絶対パス名を表示します。	
	デフォルト	ファイルの絶対パス名
	変更方法	変更不可

絶対パスで保存	ファイルの場所を絶対パスで保存するかどうかを選択します。 なお、本プロパティは、依存関係ファイルについては表示しません。		
	デフォルト	いいえ	
	変更方法	ドロップダウン・リストによる選択	
	指定可能値	はい	ファイルの場所を絶対パスで保存します。
		いいえ	ファイルの場所を相対パスで保存します。
更新日時	ファイルを最後に変更した日時を表示します。		
	デフォルト	ファイルの更新日時	
	変更方法	変更不可	
書き込み許可	ファイルに書き込みを許可するかどうかを選択します。 なお、本プロパティは、依存関係ファイルについては表示しません。		
	デフォルト	はい（ファイルに書き込みを許可している場合） いいえ（ファイルに書き込みを許可していない場合）	
		ドロップダウン・リストによる選択	
	指定可能値	はい	ファイルに書き込みを許可します。
		いいえ	ファイルに書き込みを許可しません。

(2) [記録]

記録に関する詳細情報の表示、および設定の変更を行います。

なお、本カテゴリは、依存関係ファイルについては表示しません。

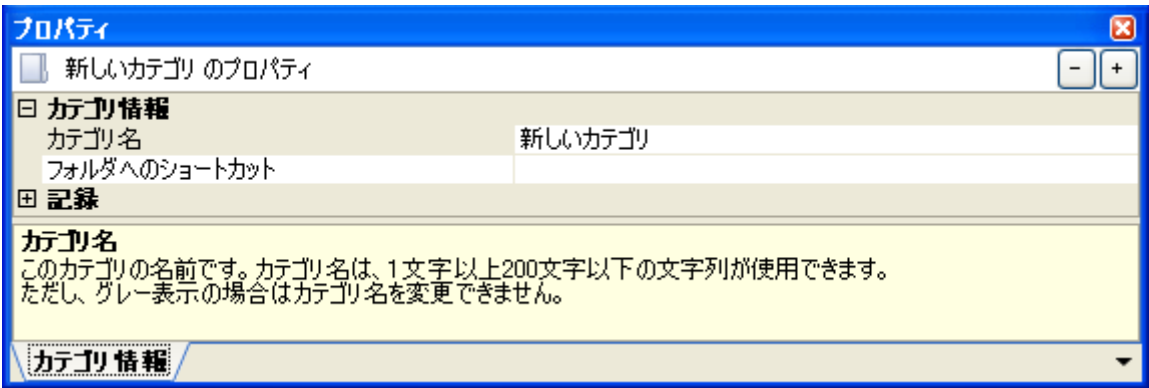
メモ	ファイルにメモを追加します。 1行に1項目ずつ指定します。 追加したメモはサブプロパティとして表示します。		
	デフォルト	メモ [項目数]	
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能	
	指定可能値	256 文字までの文字列 256 個まで指定可能です。	

[カテゴリ情報] タブ

本タブでは、カテゴリ・ノード（ユーザが追加したファイルのカテゴリ）、ファイル・ノード、ビルド・ツール生成ファイル・ノードに対して、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

- (1) [カテゴリ情報]
- (2) [記録]

図 A—17 プロパティ パネル：[カテゴリ情報] タブ



[各カテゴリの説明]

- (1) [カテゴリ情報]

カテゴリに関する詳細情報の表示、および設定の変更を行います。

カテゴリ名	ファイルを分類するためのカテゴリ名を指定します。 なお、本プロパティは、ファイル・ノード、ビルド・ツール生成ファイル・ノードについてはグレー表示となり、変更することはできません。	
	デフォルト	ファイルのカテゴリ名
	変更方法	テキスト・ボックスによる直接入力
	指定可能値	1 ～ 200 文字までの文字列
フォルダへのショート カット	フォルダへのショートカットを指定します。 相対パスで指定した場合は、メイン・プロジェクト、またはサブプロジェクトのフォルダを基点とします。 なお、本プロパティは、ファイル・ノード、ビルド・ツール生成ファイル・ノードについては表示されません。	
	デフォルト	空欄
	変更方法	テキスト・ボックスによる直接入力、または [...] ボタンをクリックし、 フォルダの参照 ダイアログ による編集
	指定可能値	247 文字までの文字列

(2) [記録]

記録に関する詳細情報の表示、および設定の変更を行います。

なお、本カテゴリは、ファイル・ノード、ビルド・ツール生成ファイル・ノードについては表示しません。

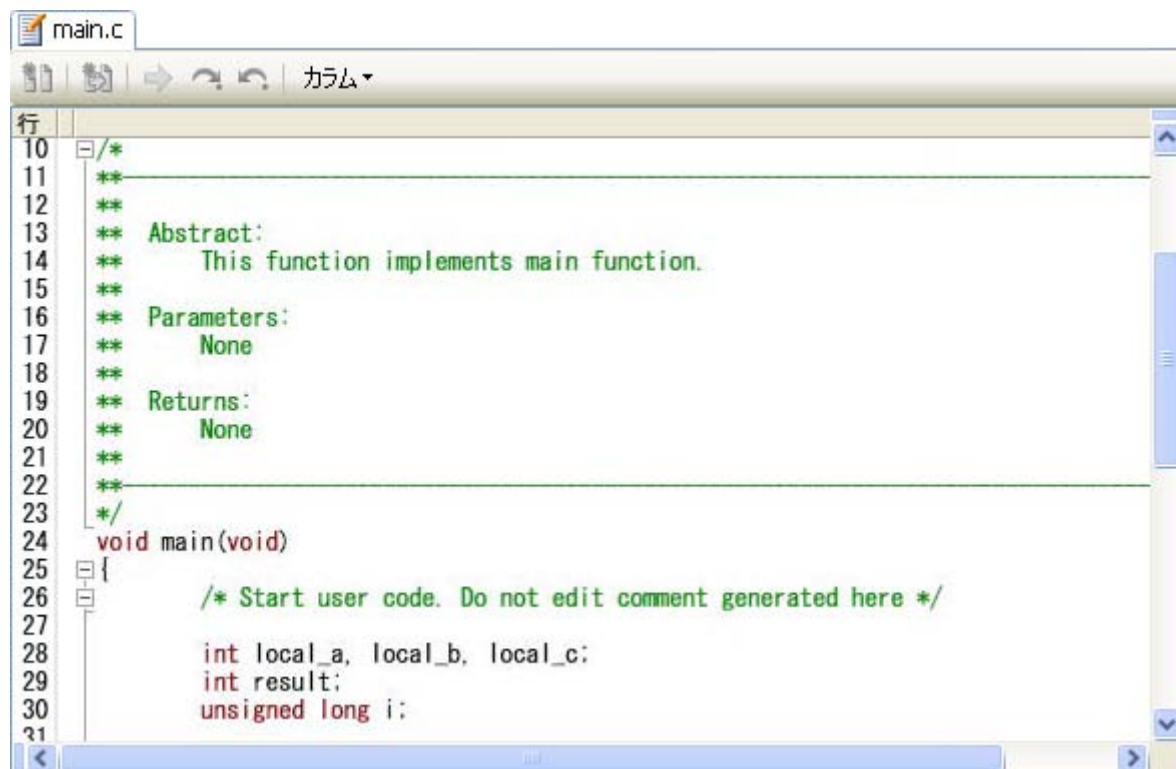
メモ	ファイルのカテゴリにメモを追加します。 1 行に 1 項目ずつ指定します。 追加したメモはサブプロパティとして表示します。	
	デフォルト	メモ [項目数]
	変更方法	[...] ボタンをクリックし、 テキスト編集 ダイアログ による編集 サブプロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

エディタ パネル

テキスト・ファイル／ソース・ファイルの表示／編集を行います。

本パネルについての詳細は、「CubeSuite+ 統合開発環境 ユーザーズマニュアル RH850 コーディング編」を参照してください。

図 A—18 エディタ パネル



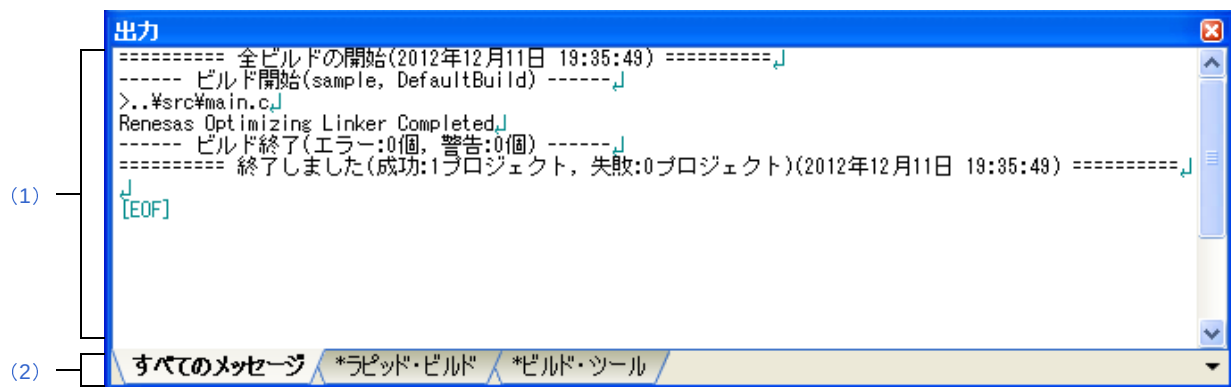
出力 パネル

ビルド・ツールから出力するメッセージの表示を行います。

メッセージは、出力元のツールごとに分類したタブ上でそれぞれ個別に表示します。

備考 ツールバーの , または [Ctrl] キーを押下しながらマウス・ホイールを前後方に動かすことにより、本パネルの表示を拡大／縮小することができます。

図 A—19 出力 パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [[ファイル] メニュー (出力 パネル専用部分)]
- [[編集] メニュー (出力 パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー → [出力] の選択

[各エリアの説明]

(1) メッセージ・エリア

各ツールから出力したメッセージを表示します。

ビルド結果の表示では、ビルドを行うごとに、以前のメッセージをクリアしたのち新しいメッセージを表示します ([すべてのメッセージ] タブを除く)。

備考 メッセージの最大表示行数は 500000 行です。

500001 行以上のメッセージを出力した場合は、古い行から削除します。

なお、メッセージの表示色は、出力メッセージの種別により、次のように異なります（表示の際の文字色／背景色は、[オプション ダイアログ](#)における「全般 - フォントと色」カテゴリ項目の設定に依存します）。

メッセージ種別	表示例（デフォルト）		説明
通常メッセージ	AaBbCc	文字色	何らかの情報を通知する際に表示します。
		背景色	
警告メッセージ	AaBbCc	文字色	操作に対して、何らかの警告を通知する際に表示します。
		背景色	
エラー・メッセージ	AaBbCc	文字色	致命的なエラー、または操作ミスにより実行が不可能な場合に表示します。
		背景色	

本エリアは、次の機能を備えています。

(a) タグ・ジャンプ

出力したメッセージをダブルクリック、またはメッセージにキャレットを合わせて [Enter] キーを押下することにより、[エディタ パネル](#)をオープンして該当ファイルの該当行番号を表示します。

これにより、ビルド時に出力したエラー・メッセージなどから、ソース・ファイルの該当するエラー行へジャンプすることができます。

(b) ヘルプの表示

警告メッセージ、またはエラー・メッセージを表示している行にキャレットがある状態で、コンテキスト・メニューの「メッセージに関するヘルプ」を選択するか、または [F1] キーを押下することにより、その行のメッセージに関するヘルプを表示します。

(c) ログの保存

「[ファイル] メニュー→「名前を付けて出力 - タブ名を保存 ...」」を選択することにより、[名前を付けて保存 ダイアログ](#)をオープンし、現在選択しているタブ上に表示している内容をテキスト・ファイル (*.txt) に保存することができます（非選択状態のタブ上のメッセージは保存の対象となりません）。

(2) タブ選択エリア

メッセージの出力元を示すタブを選択します。

表示するタブは次のとおりです。

タブ名	説明
すべてのメッセージ	すべてのメッセージを出力順に一括して表示します（ラビッド・ビルド実行時を除く）。
ラビッド・ビルド	ラビッド・ビルドの実行により、ビルド・ツールから出力したメッセージを表示します。
ビルド・ツール	ビルド、リビルド、バッチ・ビルドの実行により、ビルド・ツールから出力したメッセージを表示します。

注意 新たなメッセージを非選択状態のタブ上に出力しても、自動的なタブの表示切り替えは行いません。
この場合、タブ名の先頭に  マークが付加し、新たなメッセージを出力していることを示します。

[[ファイル] メニュー（出力 パネル専用部分）]

出力 パネル専用の [ファイル] メニューは次のとおりです（その他の項目は共通です）。

出力 - タブ名を保存	現在選択しているタブ上に表示している内容を、前回保存したテキスト・ファイル (*.txt) に保存します（「(c) ログの保存」参照）。 なお、起動後に初めて本項目を選択した場合は、[名前を付けてタブ名を保存 ...] の選択と同等の動作となります。
名前を付けて出力 - タブ名を保存 ...	現在選択しているタブ上に表示している内容を、指定したファイル (*.txt) に保存するために、名前を付けて保存 ダイアログをオープンします（「(c) ログの保存」参照）。

[[編集] メニュー（出力 パネル専用部分）]

出力 パネル専用の [編集] メニューは次のとおりです（その他の項目はすべて無効となります）。

コピー	選択している文字列をクリップ・ボードにコピーします。
すべて選択	本パネルに表示しているすべてのメッセージを選択状態にします。
検索 ...	検索・置換 ダイアログを [クイック検索] タブが選択状態でオープンします。
置換 ...	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。

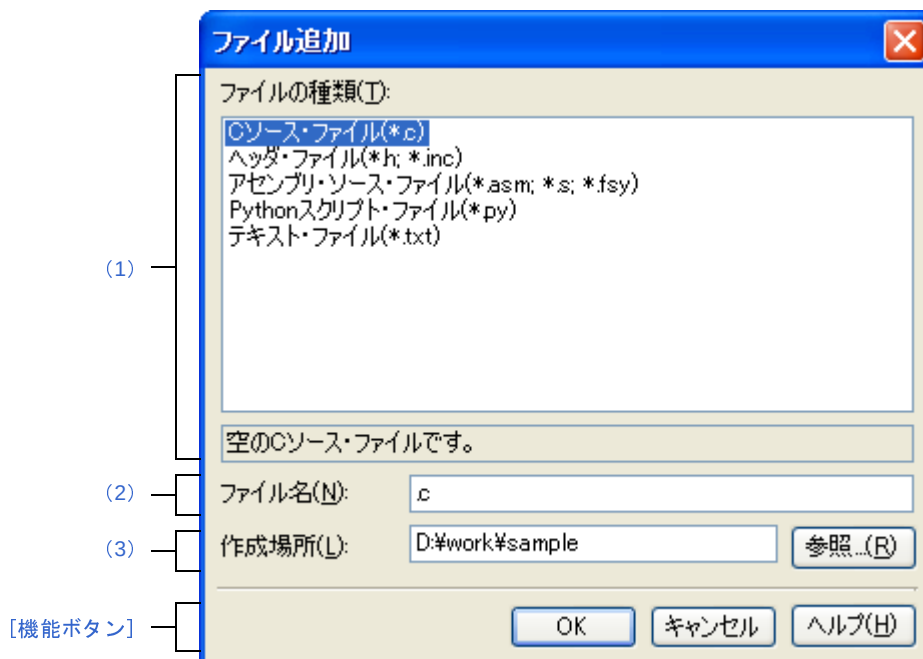
[コンテキスト・メニュー]

コピー	選択している文字列をクリップ・ボードにコピーします。
すべて選択	本パネルに表示しているすべてのメッセージを選択状態にします。
クリア	本パネルに表示しているすべてのメッセージを消去します。
タグ・ジャンプ	キャレット行にファイル名、行、桁の情報があれば、その位置へジャンプします。
メッセージに関するヘルプ	現在のキャレット位置のメッセージに関するヘルプを表示します。 ただし、警告メッセージ/エラー・メッセージのみが対象となります。

ファイル追加 ダイアログ

新規にファイルを作成し、プロジェクトへの追加を行います。

図 A—20 ファイル追加 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [ファイル] メニュー → [追加] → [新しいファイルを追加 ...] を選択
- プロジェクト・ツリーパネルにおいて、プロジェクト・ノード、サブプロジェクト・ノード、ファイル・ノード、カテゴリ・ノードのいずれかを選択したのち、コンテキスト・メニュー → [追加] → [新しいファイルを追加 ...] を選択

[各エリアの説明]

(1) [ファイルの種類] エリア

作成するファイルの種類を選択します。

ファイルの種類を選択すると、下部のボックスにその説明を表示します。

表示するファイルの種類を以下に示します。

- C ソース・ファイル (*.c)
- ヘッダ・ファイル (*.h; *.inc)
- アセンブリ・ソース・ファイル (*.asm; *.s; *.fsy)
- Python スクリプト・ファイル (*.py)
- テキスト・ファイル (*.txt)

(2) [ファイル名] エリア

作成するファイルの名前を直接入力します。

デフォルトでは、“.c” を表示します。

備考 拡張子を指定しなかった場合は、[ファイルの種類] エリアで選択した拡張子を付加します。

また、[ファイルの種類] エリアと異なる拡張子を指定した場合も、[ファイルの種類] エリアで選択した拡張子を付加します（例えば、ファイル名に“aaa.txt”，ファイルの種類に“C ソース・ファイル (*.c)”を指定した場合、ファイル名は“aaa.txt.c”となります）。

(3) [作成場所] エリア

ファイルの作成場所のパスをテキスト・ボックスに直接入力、または[参照 ...] ボタンから選択します。

デフォルトでは、プロジェクト・フォルダのパスを表示します。

ただし、カテゴリ・ノード（フォルダへのショートカットを設定し、そのフォルダが存在する場合のみ）のコンテキスト・メニューから本ダイアログをオープンした場合は、カテゴリに設定したフォルダのパスを表示します。

(a) ボタン

参照 ...	フォルダの参照 ダイアログをオープンします。 フォルダを選択すると、テキスト・ボックスにパスを追加します。
--------	--

備考 1. テキスト・ボックスが空欄の場合は、プロジェクト・フォルダを指定したものとみなします。

2. 相対パスで指定した場合は、プロジェクト・フォルダからの相対パスとみなします。

備考 [ファイル名] エリア、[作成場所] エリアで指定可能な文字数は、パス名とファイル名をあわせて 259 文字までです。入力内容が正しくない場合、以下のメッセージを [ファイル名] エリアにツールチップ表示します。

メッセージ	説明
パスを含むファイル名が長すぎます。259 文字以内にしてください。	パスを含むファイル名が 259 文字を越えています。
指定したパスに存在しないフォルダが含まれています。	パスに存在しないフォルダが含まれています。
ファイル名、もしくは、パス名が不正です。文字 (¥, /, :, *, ?, ", <, >,) は使用できません。	不正なパスを含むファイル名を指定しました。 ファイル名、およびフォルダ名に文字 (¥, /, :, *, ?, ", <, >,) は使用できません。

[機能ボタン]

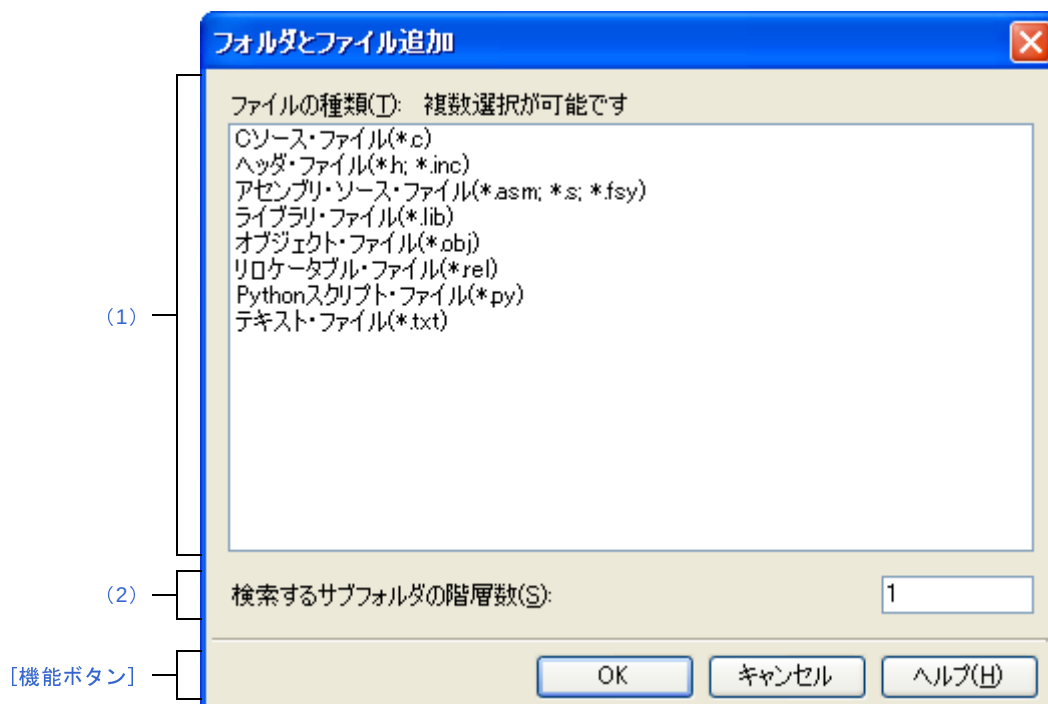
ボタン	機能
OK	入力したファイル名でファイルを作成して、プロジェクトに追加し、 エディタ パネル でオープンします。そののち、本ダイアログをクローズします。
キャンセル	ファイルの生成を行わずに、本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

フォルダとファイル追加 ダイアログ

既存のファイルとフォルダ構成のプロジェクトへの追加を行います。

フォルダはカテゴリとして追加します。

図 A—21 フォルダとファイル追加 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- エクスプローラなどからフォルダをドラッグし、プロジェクト・ツリー パネル上でドロップ

[各エリアの説明]

(1) [ファイルの種類] エリア

プロジェクトに追加するファイルの種類を選択します。

[Ctrl] キー+左クリック, または [Shift] キー+左クリックにより, 複数選択することができます。

何も選択しない場合は, すべての種類を選択したものとみなします。

表示するファイルの種類を以下に示します。

- C ソース・ファイル (*.c)
- ヘッダ・ファイル (*.h; *.inc)
- アセンブリ・ソース・ファイル (*.asm; *.s; *.fsy)
- ライブラリ・ファイル (*.lib)
- オブジェクト・ファイル (*.obj)
- リロケータブル・ファイル (*.rel)
- Python スクリプト・ファイル (*.py)
- テキスト・ファイル (*.txt)

(2) [検索するサブフォルダの階層数] エリア

プロジェクトに追加するサブフォルダの階層数を直接入力します。

デフォルトでは, “1” を表示します。

備考 入力可能な値は 10 までの 10 進数です。

入力内容が正しくない場合, 以下のメッセージをツールチップ表示します。

メッセージ	説明
0 未満もしくは 10 を越える値を指定できません。	サブフォルダの指定階層数が 0 未満, または 10 を越えています。
10 進数で指定してください。	10 進数以外の数値や文字列を指定しました。

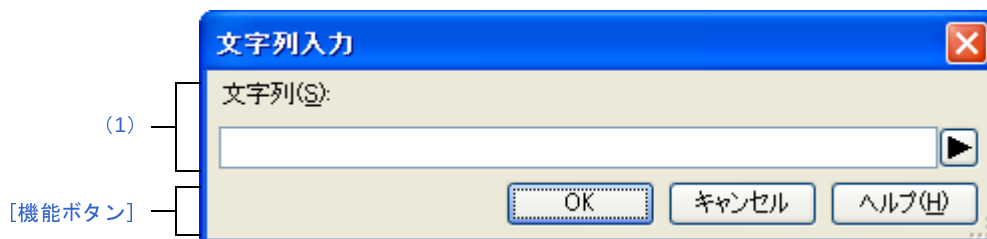
[機能ボタン]

ボタン	機能
OK	ドラッグ・アンド・ドロップしたフォルダとそのフォルダ下に存在するファイルをプロジェクトに追加します。そののち, 本ダイアログをクローズします。
キャンセル	フォルダとファイルの追加を行わずに, 本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

文字列入力 ダイアログ

1 行分の文字列の入力、編集を行います。

図 A—22 文字列入力 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロジェクト・ツリー パネルにおいて、ファイルを選択したのち、コンテキスト・メニュー→[拡張子の変更...]を選択してメッセージ ダイアログをオープン
→ダイアログ上で[はい] ボタンをクリック
- プロパティ パネルにおいて、以下のプロパティを選択したのち、[...] ボタンをクリック
 - [共通オプション] タブの[警告メッセージ] カテゴリの[表示させない警告メッセージ]、[その他] カテゴリの[ビルド・オプション一覧表示フォーマット]、[その他の追加オプション]
 - [コンパイル・オプション] タブの[その他] カテゴリの[その他の追加オプション]
 - [アセンブル・オプション] タブの[その他] カテゴリの[その他の追加オプション]
 - [リンク・オプション] タブの[出力コード] カテゴリの[実行開始アドレス]、[メッセージ] カテゴリの[抑止するインフォメーション・メッセージの番号]、[ワーニング・メッセージの番号]、[インフォメーション・メッセージの番号]、[インフォメーション、ワーニング・メッセージの番号]、[その他] カテゴリの[その他の追加オプション]
 - [ヘキサ出力オプション] タブの[その他] カテゴリの[その他の追加オプション]
 - [ライブラリ生成オプション] タブの[メッセージ] カテゴリの[抑止するインフォメーション・メッセージの番号]、[ワーニング・メッセージの番号]、[インフォメーション・メッセージの番号]、[インフォメーション、ワーニング・メッセージの番号]、[その他] カテゴリの[その他の追加オプション]
 - [個別コンパイル・オプション] タブの[警告メッセージ] カテゴリの[表示させない警告メッセージ]、[その他] カテゴリの[その他の追加オプション]
 - [個別アセンブル・オプション] タブの[警告メッセージ] カテゴリの[表示させない警告メッセージ]、[その他] カテゴリの[その他の追加オプション]

[各エリアの説明]

(1) [文字列] エリア

1 行分の文字列の入力を行います。

デフォルトでは、本ダイアログの呼び出し元の内容を反映します。

なお、改行することはできません。

備考 入力可能な文字数は、32767 文字までです。

入力内容が正しくない場合、以下のメッセージをツールチップ表示します。

メッセージ	説明
呼び出し元で指定している最大文字数文字を越える文字を指定できません。	入力した文字列の文字数が、呼び出し元で指定している最大文字数を越えています。

(a) ボタン

	本ダイアログの呼び出し元に指定可能なプレースホルダをポップアップ表示します (昇順)。 プレースホルダを選択すると、前後に“%”を付加して [文字列] に表示します。
---	--

注意 本ボタンは、本ダイアログの呼び出し元がプレースホルダに対応している場合のみ表示されます。

備考 指定可能なプレースホルダは、本ダイアログの呼び出し元によって異なります。

具体的なプレースホルダについては、呼び出し元の説明を参照してください。

[機能ボタン]

ボタン	機能
OK	入力した文字列を本ダイアログの呼び出し元に反映し、本ダイアログをクローズします。
キャンセル	入力した文字列を本ダイアログの呼び出し元に反映せずに、本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

テキスト編集 ダイアログ

複数行のテキストの入力、編集を行います。

図 A—23 テキスト編集 ダイアログ（呼び出し元がプレースホルダに対応している場合）

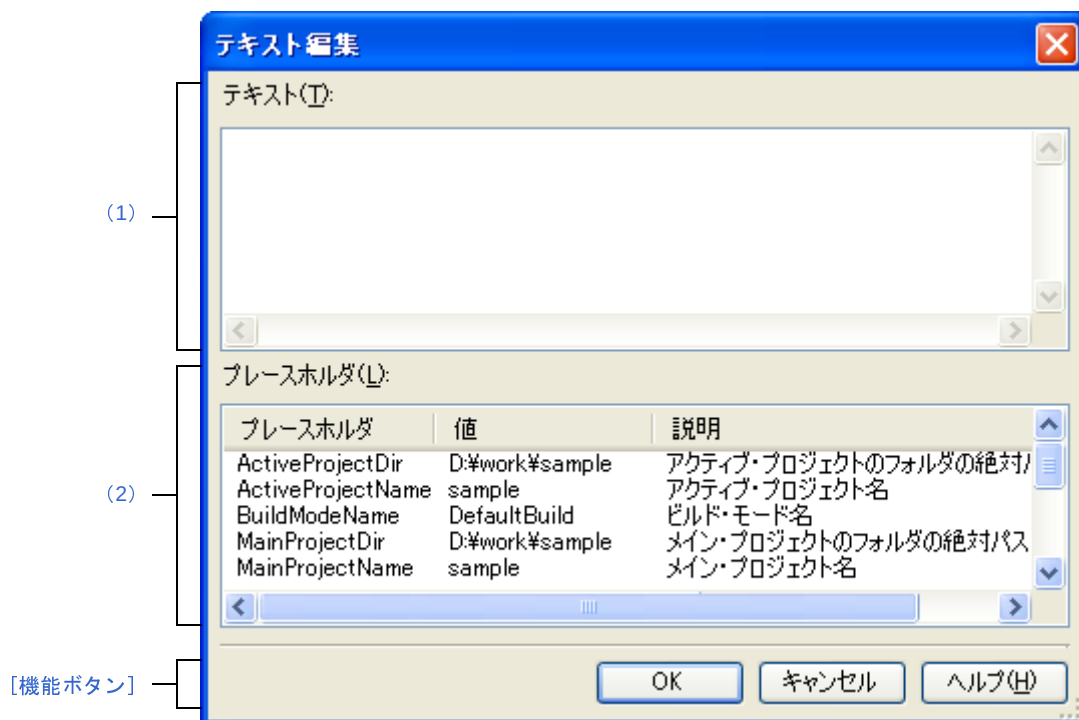
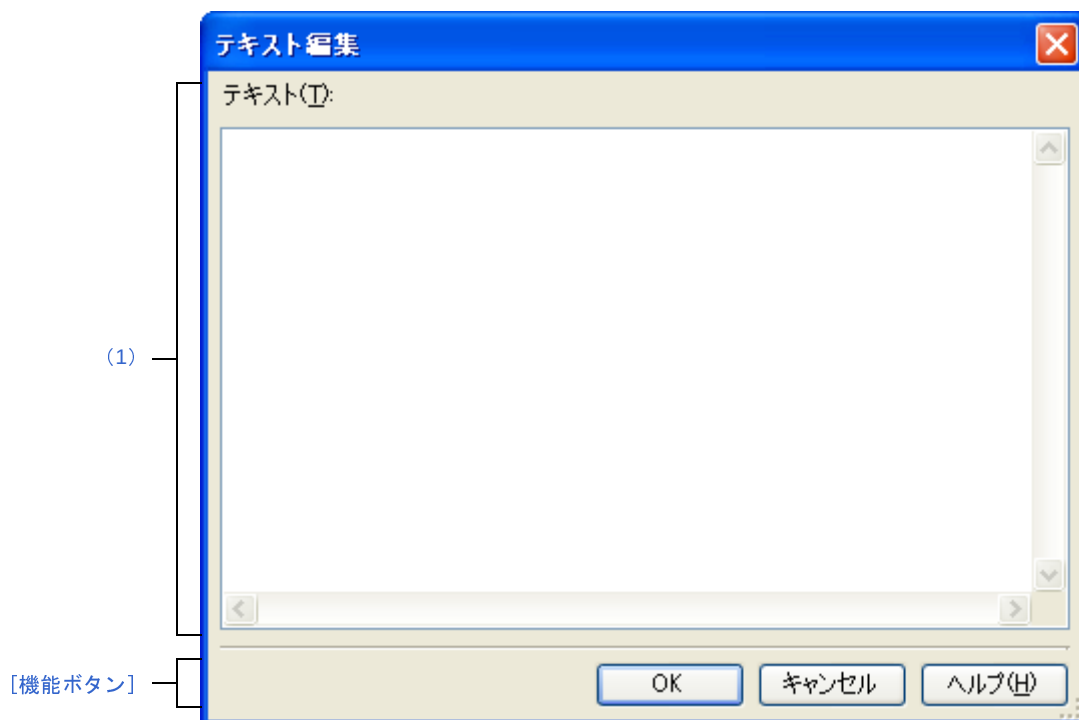


図 A—24 テキスト編集 ダイアログ（呼び出し元がプレースホルダに対応していない場合）



ここでは、次の項目について説明します。

- [\[オープン方法\]](#)
- [\[各エリアの説明\]](#)
- [\[機能ボタン\]](#)

[オープン方法]

- [プロパティ パネル](#)において、以下のプロパティを選択したのち、[\[...\]](#) ボタンをクリック
 - [\[共通オプション\] タブ](#)の [\[よく使うオプション \(コンパイル\)\]](#) カテゴリの [\[定義マクロ\]](#), [\[よく使うオプション \(アセンブル\)\]](#) カテゴリの [\[定義マクロ\]](#), [\[よく使うオプション \(リンク\)\]](#) カテゴリの [\[分割出力ファイル\]](#), [\[よく使うオプション \(ヘキサ出力\)\]](#) カテゴリの [\[分割出力ファイル\]](#), [\[記録\]](#) カテゴリの [\[メモ\]](#), [\[その他\]](#) カテゴリの [\[ビルド前に実行するコマンド\]](#), [\[ビルド後に実行するコマンド\]](#)
 - [\[コンパイル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[コンパイル単位の先頭にインクルードするファイル\]](#), [\[定義マクロ\]](#), [\[定義解除マクロ\]](#), [\[その他\]](#) カテゴリの [\[コンパイル前に実行するコマンド\]](#), [\[コンパイル後に実行するコマンド\]](#)
 - [\[アセンブル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[定義マクロ\]](#), [\[定義解除マクロ\]](#), [\[その他\]](#) カテゴリの [\[アセンブル前に実行するコマンド\]](#), [\[アセンブル後に実行するコマンド\]](#)
 - [\[リンク・オプション\] タブ](#)の [\[入力ファイル\]](#) カテゴリの [\[オブジェクト・ファイル\]](#), [\[バイナリ・ファイル\]](#), [\[シンボル定義\]](#), [\[出力ファイル\]](#) カテゴリの [\[分割出力ファイル\]](#), [\[セクション\]](#) カテゴリの [\[スタートアップの属するセクション\]](#), [\[外部定義シンボルをファイル出力するセクション\]](#), [\[セクション・アライメント\]](#), [\[ROM から RAM へマップするセクション\]](#), [\[ベリファイ\]](#) カテゴリの [\[メモリ種別のアドレス範囲\]](#), [\[その他\]](#) カテゴリの [\[リンク前に実行するコマンド\]](#), [\[リンク後に実行するコマンド\]](#)
 - [\[ヘキサ出力オプション\] タブ](#)の [\[出力ファイル\]](#) カテゴリの [\[分割出力ファイル\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の [\[入力ファイル\]](#) カテゴリの [\[オブジェクト・ファイル\]](#), [\[バイナリ・ファイル\]](#), [\[その他\]](#) カテゴリの [\[ライブラリ生成前に実行するコマンド\]](#), [\[ライブラリ生成後に実行するコマンド\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[コンパイル単位の先頭にインクルードするファイル\]](#), [\[定義マクロ\]](#), [\[定義解除マクロ\]](#), [\[その他\]](#) カテゴリの [\[コンパイル前に実行するコマンド\]](#), [\[コンパイル後に実行するコマンド\]](#)
 - [\[個別アセンブル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[定義マクロ\]](#), [\[定義解除マクロ\]](#), [\[その他\]](#) カテゴリの [\[アセンブル前に実行するコマンド\]](#), [\[アセンブル後に実行するコマンド\]](#)
 - [\[ファイル情報\] タブ](#)の [\[記録\]](#) カテゴリの [\[メモ\]](#)
 - [\[カテゴリ情報\] タブ](#)の [\[記録\]](#) カテゴリの [\[メモ\]](#)

[各エリアの説明]

(1) [テキスト]

複数行のテキストの編集を行います。

デフォルトでは、本ダイアログの呼び出し元の内容を反映します。

備考 入力可能な行数は 65535 行まで、文字数は 65535 文字までです。

入力内容が正しくない場合、以下のメッセージをツールチップ表示します。

メッセージ	説明
呼び出し元で指定している最大文字数文字を越える文字を指定できません。制限を越えた行頭のかっこ内に今の文字数を表示しました。	入力した文字列の文字数が、呼び出し元で指定している最大文字数を越えています。

(2) [プレースホルダ] エリア

本ダイアログの呼び出し元に指定可能なプレースホルダの一覧を表示します（昇順）。

行をダブルクリックすると、プレースホルダの前後に“%”を付加して[テキスト]に表示します。

(a) [プレースホルダ]

プレースホルダを表示します。

(b) [値]

プレースホルダの置換後の文字列を表示します。

(c) [説明]

プレースホルダの説明を表示します。

注意 本エリアは、本ダイアログの呼び出し元がプレースホルダに対応している場合のみ表示されます。

備考 指定可能なプレースホルダは、本ダイアログの呼び出し元によって異なります。

具体的なプレースホルダについては、呼び出し元の説明を参照してください。

[機能ボタン]

ボタン	機能
OK	入力したテキストを本ダイアログをオープンしたテキスト・ボックスに反映し、本ダイアログをクローズします。
キャンセル	入力したテキストを本ダイアログをオープンしたテキスト・ボックスに反映せずに、本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

パス編集 ダイアログ

パス、またはパスを含むファイル名の編集、追加を行います。

図 A—25 パス編集 ダイアログ（パスを編集する場合）

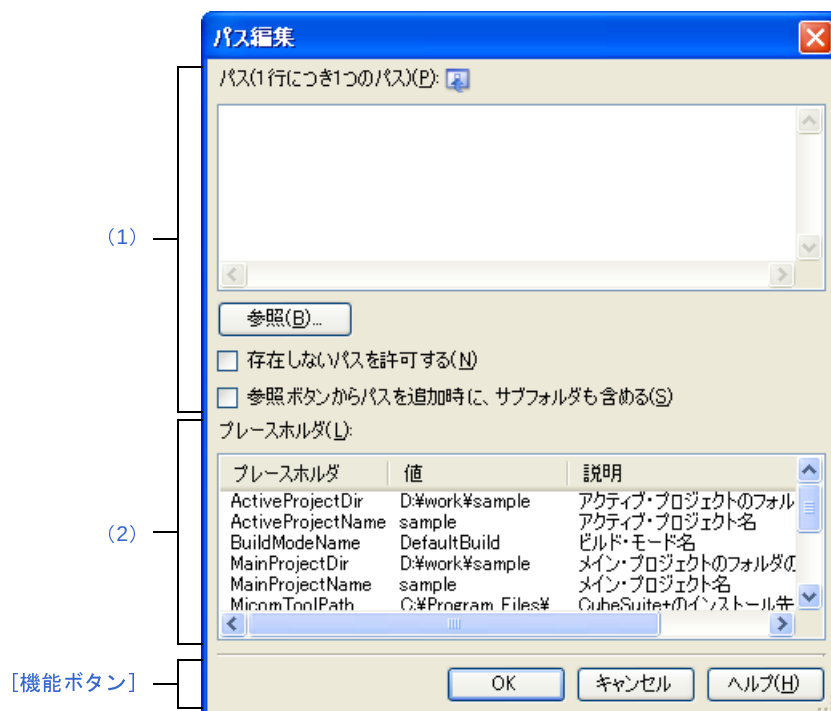
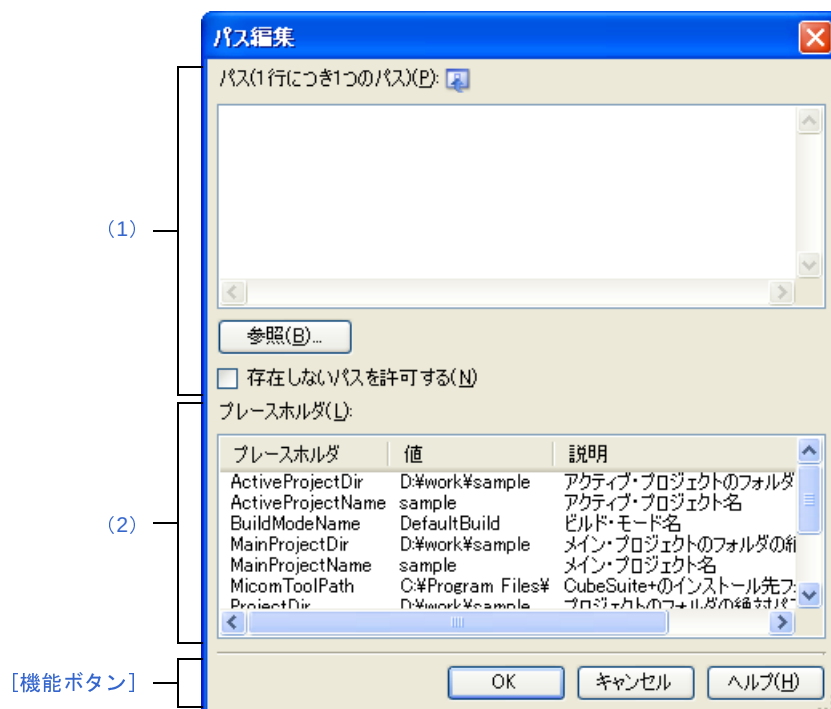


図 A—26 パス編集 ダイアログ（パスを含むファイル名を編集する場合）



ここでは、次の項目について説明します。

- [\[オープン方法\]](#)
- [\[各エリアの説明\]](#)
- [\[機能ボタン\]](#)

[オープン方法]

- [プロパティ パネル](#)において、以下のプロパティを選択したのち、[\[...\] ボタン](#)をクリック
 - [\[共通オプション\] タブ](#)の [\[よく使うオプション \(コンパイル\)\]](#) カテゴリの [\[追加のインクルード・パス\]](#), [\[よく使うオプション \(アセンブル\)\]](#) カテゴリの [\[追加のインクルード・パス\]](#), [\[よく使うオプション \(リンク\)\]](#) カテゴリの [\[使用するライブラリ・ファイル\]](#)
 - [\[コンパイル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[追加のインクルード・パス\]](#), [\[MISRA-C:2004 ルール検査\]](#) カテゴリの [\[ルール・チェック対象外のファイル\]](#)
 - [\[アセンブル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[追加のインクルード・パス\]](#)
 - [\[リンク・オプション\] タブ](#)の [\[ライブラリ\]](#) カテゴリの [\[使用するライブラリ・ファイル\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の [\[ライブラリ\]](#) カテゴリの [\[使用するライブラリ・ファイル\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[追加のインクルード・パス\]](#), [\[MISRA-C:2004 ルール検査\]](#) カテゴリの [\[ルール・チェック対象外のファイル\]](#)
 - [\[個別アセンブル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[追加のインクルード・パス\]](#)

[各エリアの説明]

(1) パス編集エリア

パス、またはパスを含むファイル名の編集、追加を行います。

(a) [パス (1 行につき 1 つのパス)]

直接入力により、パス、またはパスを含むファイル名の編集、追加を行います。

パス、またはパスを含むファイル名は複数行指定可能です。1 行につき 1 つのパス、またはパスを含むファイル名を指定してください。

デフォルトで、本ダイアログをオープンしたテキスト・ボックスの内容を反映します。

パスの追加は、以下の方法でも行うことができます。

- [\[参照 ...\]](#) ボタンをクリックし、[フォルダの参照 ダイアログ](#)によるフォルダの選択
- エクスプローラなどからフォルダをドラッグ・アンド・ドロップ

パスを含むファイル名の追加は、以下の方法でも行うことができます。

- [\[参照 ...\]](#) ボタンをクリックし、[Far Jump ファイルを指定 ダイアログ](#)によるファイルの選択
- [\[参照 ...\]](#) ボタンをクリックし、[対象外ファイルの追加 ダイアログ](#)によるファイルの選択
- エクスプローラなどからファイルをドラッグ・アンド・ドロップ

注意 絶対パスで非常に長いパスを相対パスで指定すると、[OK] ボタンのクリック時にエラーになる場合があります。その場合は、絶対パスで指定してください。

備考 入力可能な行数は 10000 行まで、文字数は Windows のパスの最大文字数までです。
入力内容が正しくない場合、以下のメッセージをツールチップ表示します。

メッセージ	説明
パスを指定してください。	空白文字だけの行になっています。
パスが長すぎます。呼び出し元で指定している最大文字数文字以下のパスを指定してください。	パスを含むファイル名が呼び出し元で指定している最大文字数を越えています。
指定したパスに存在しないフォルダが含まれています。	パスに存在しないフォルダが含まれています。
ファイル名、もしくは、パス名が不正です。文字 (¥, /, :, *, ?, ", <, >,) は使用できません。	不正なパスを含むファイル名を指定しました。 ファイル名、およびフォルダ名に文字 (¥, /, :, *, ?, ", <, >,) は使用できません。
呼び出し元で指定している最大パス数、またはファイル数行を越える行を指定できません。	入力したパス、またはファイルの総数が呼び出し元で指定している最大パス数、またはファイル数を越えています。

(b) ボタン

参照 ...	- パスを追加する場合 フォルダの参照 ダイアログをオープンします。 フォルダを選択すると、[パス (1 行につき 1 つのパス)] にパスを追加します。 - パスを含むファイル名を追加する場合 Far Jump ファイルを指定 ダイアログをオープンします。 ファイルを選択すると、[パス (1 行につき 1 つのパス)] にファイル名を追加します。
--------	--

(c) [存在しないパスを許可する]

このチェック・ボックスをチェックすると、[パス (1 行につき 1 つのパス)] に指定したパスの存在確認や文字列のチェックを行いません。

(d) [参照ボタンからパスを追加時に、サブフォルダも含める]

このチェック・ボックスをチェックしたのち、[参照 ...] ボタンからパスの指定を行うと、サブフォルダも含めて [パス (1 行につき 1 つのパス)] にパスを追加します (5 階層まで)。

注意 本項目は、パスを編集する場合のみ表示されます。

(2) [プレースホルダ] エリア

本ダイアログの呼び出し元に指定可能なプレースホルダの一覧を表示します (昇順)。

行をダブルクリックすると、プレースホルダの前後に “%” を付加してパス編集エリアに表示します。

(a) [プレースホルダ]

プレースホルダを表示します。

(b) [値]

プレースホルダの置換後の文字列を表示します。

(c) [説明]

プレースホルダの説明を表示します。

注意 本エリアは、本ダイアログの呼び出し元がプレースホルダに対応している場合のみ表示されます。

備考 指定可能なプレースホルダは、本ダイアログの呼び出し元によって異なります。

具体的なプレースホルダについては、呼び出し元の説明を参照してください。

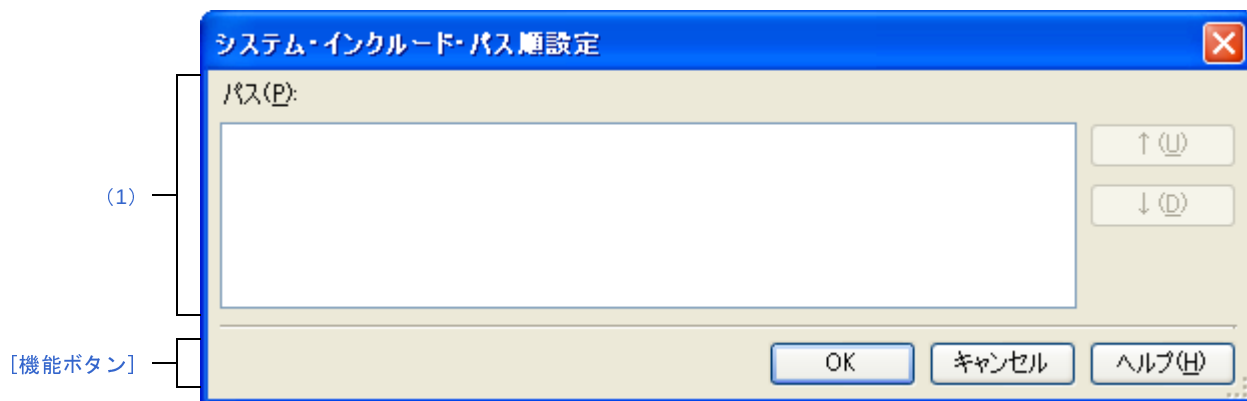
[機能ボタン]

ボタン	機能
OK	入力したパスを本ダイアログをオープンしたテキスト・ボックスに反映し、本ダイアログをクローズします。
キャンセル	入力したパスを本ダイアログをオープンしたテキスト・ボックスに反映せずに、本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

システム・インクルード・パス順設定 ダイアログ

コンパイラに対して指定するシステム・インクルード・パスの参照、および指定順の設定を行います。

図 A—27 システム・インクルード・パス順設定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルにおいて、以下のプロパティを選択したのち、[...] ボタンをクリック
 - [共通オプション] タブの [よく使うオプション (コンパイル)] カテゴリの [システム・インクルード・パス]
 - [よく使うオプション (アセンブル)] カテゴリの [システム・インクルード・パス]
- [コンパイル・オプション] タブの [プリプロセス] カテゴリの [システム・インクルード・パス]
- [アセンブル・オプション] タブの [プリプロセス] カテゴリの [システム・インクルード・パス]

[各エリアの説明]

(1) パス一覧表示エリア

コンパイラに対して指定するシステム・インクルード・パスの一覧を表示します。

(a) [パス]

システム・インクルード・パス名の一覧を、コンパイラへの指定順に表示します。

デフォルトでは、プロジェクトに登録している順番となります。

パスの表示順を変更することにより、コンパイラへの指定順を設定することができます。

表示順の変更は、[↑]、および [↓] ボタン、またはパス名のドラッグ・アンド・ドロップにより行います。

- 備考 1. パス名にマウス・カーソルをあわせると、そのパスを絶対パスでポップアップ表示します。
2. 新規に追加したシステム・インクルード・パスは、一覧の最後のパスの次に追加します。
3. パス名をドラッグ・アンド・ドロップする際、連続して並んでいるパス名のみ複数選択することができます。

(b) ボタン

↑	選択しているパスを上へ移動します。
↓	選択しているパスを下へ移動します。

備考 上記のボタンは、パスを選択していない場合は無効となります。

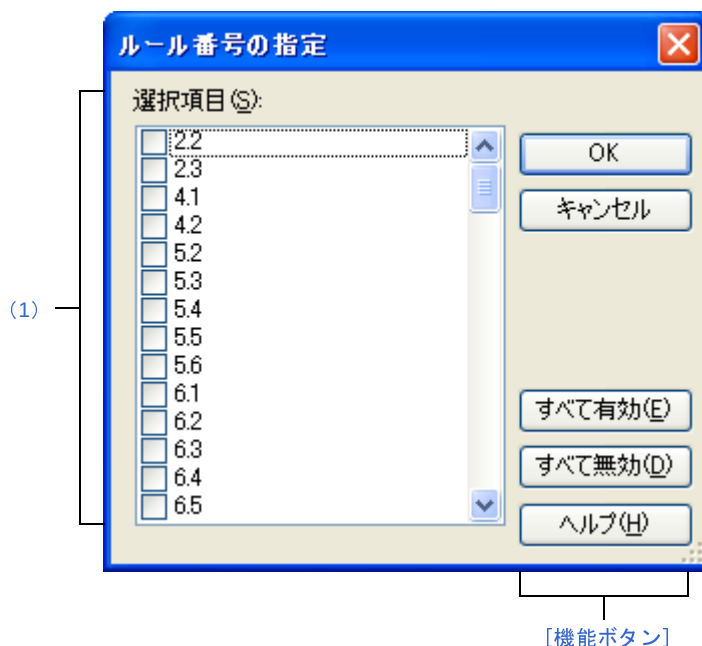
[機能ボタン]

ボタン	機能
OK	コンパイラへのパスの指定順を パス一覧表示エリア の表示順に設定し、本ダイアログをクローズします。
キャンセル	パスの指定順の設定をキャンセルし、本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

ルール番号の指定 ダイアログ

本ダイアログの呼び出し元に設定する MISRA-C:2004 ルール番号の選択を行います。

図 A—28 ルール番号の指定 ダイアログ



ここでは、次の項目について説明します。

- [\[オープン方法\]](#)
- [\[各エリアの説明\]](#)
- [\[機能ボタン\]](#)

[オープン方法]

- プロパティ パネルにおいて、以下のプロパティを選択したのち、[...] ボタンをクリック
- [\[コンパイル・オプション\]](#) タブの [MISRA-C:2004 ルール検査] カテゴリの [ルール番号], [除外するルール番号], [必須ルールの他にチェックするルール番号], [必須ルールから除外するルール番号]
- [\[個別コンパイル・オプション\]](#) タブの [MISRA-C:2004 ルール検査] カテゴリの [ルール番号], [除外するルール番号], [必須ルールの他にチェックするルール番号], [必須ルールから除外するルール番号]

[各エリアの説明]

(1) [選択項目]

本ダイアログの呼び出し元に指定可能な MISRA-C:2004 ルール番号の一覧を表示します（昇順）。

設定するルール番号をチェック・ボックスにより選択します。

備考 本ダイアログの呼び出し元において、すでにルール番号を設定していた場合は、該当するルール番号のチェック・ボックスはデフォルトでチェック状態となります。

[機能ボタン]

ボタン	機能
OK	本ダイアログをクローズし、選択したルール番号を呼び出し元に設定します。
キャンセル	ルール番号の選択をキャンセルし、本ダイアログをクローズします。
すべて有効	[選択項目]において、すべてのルール番号を選択します。
すべて無効	[選択項目]において、すべてのルール番号の選択を解除します。
ヘルプ	本ダイアログのヘルプを表示します。

セクション設定 ダイアログ

セクションの追加，変更，削除を行います。

図 A—29 セクション設定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルにおいて、以下のプロパティを選択したのち、[...] ボタンをクリック
- [リンク・オプション] タブの [セクション] カテゴリの [セクションの開始アドレス]

[各エリアの説明]

(1) アドレス - セクション・エリア

現在設定しているセクション配置の一覧を表示します。

(a) [アドレス]

セクションの開始アドレスを表示します。

(b) [セクション]

セクション名を表示します。

(c) [オーバーレイ n]

オーバーレイ配置されたセクション名を表示します (n : 1 で始まる数字)。

(d) ボタン

追加 ...	- 本エリアでアドレスを選択している場合 セクションのアドレス ダイアログをオープンします。 ダイアログ上で指定したアドレスを本エリアのアドレス昇順となる位置に挿入します (セクションは空欄)。 - 本エリアでセクションを選択している場合 セクション追加 ダイアログをオープンします。 ダイアログ上で指定したセクションを本エリアに追加します。 指定したセクションが属するセクション・グループ (アドレスと属するセクション) の該当列に空欄がない場合は、そのセクション・グループの最下部に新しいセクション行を追加します。 空欄がある場合は、そこにセクションを設定します。
変更 ...	- 本エリアでアドレスを選択している場合 セクションのアドレス ダイアログをオープンします。 ダイアログ上で指定したアドレスに応じて、セクション・グループを本エリアのアドレス昇順となる位置に移動します。 - 本エリアでセクションを選択している場合 セクション編集 ダイアログをオープンします。 本エリアで選択しているセクション名をダイアログ上で指定したセクション名に置き換えます。 なお、本ボタンは選択しているセルが空欄の場合は無効となります。
複数割り付け ...	オーバーレイ配置セクションの追加 ダイアログをオープンします。 本エリアに [オーバーレイ n] 列を追加し、ダイアログ上で指定したセクションを選択しているセクション・グループの該当列に設定します。
削除	- 本エリアでアドレスを選択している場合 削除するセクション ダイアログをオープンします。 ダイアログ上で選択したセクションを本エリアから削除します。 セクション・グループに属するセクションがなくなる場合は、そのセクション・グループを削除します。 - 本エリアでセクションを選択している場合 選択しているセクションを本エリアから削除します。 セクション・グループに属するセクションがなくなる場合は、そのセクション・グループを削除します。 [オーバーレイ n] 列に属するセクション名がなくなる場合は、その列を削除します。 なお、本ボタンは選択しているセルが空欄の場合は無効となります。
↑	選択しているセクションを上へ移動します。 なお、本ボタンはアドレスを選択している場合は無効となります。

↓	選択しているセクションを下へ移動します。 なお、本ボタンはアドレスを選択している場合は無効となります。
インポート ...	インポートするファイルを選択 ダイアログをオープンします。 ダイアログ上で指定したファイルからセクション設定を取得し、本エリアに反映します。
エクスポート ...	エクスポートするファイルを選択 ダイアログをオープンします。 本エリアの内容をダイアログ上で指定したファイルに出力します。

[機能ボタン]

ボタン	機能
OK	設定したセクションを本ダイアログをオープンしたテキスト・ボックスに反映し、本ダイアログをクローズします。
キャンセル	設定を無効とし、本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

セクション追加 ダイアログ

セクション編集 ダイアログ

オーバーレイ配置セクションの追加 ダイアログ

セクションの追加，編集，複数割り付けを行う際のセクション名の設定を行います。

図 A—30 セクション追加 ダイアログ

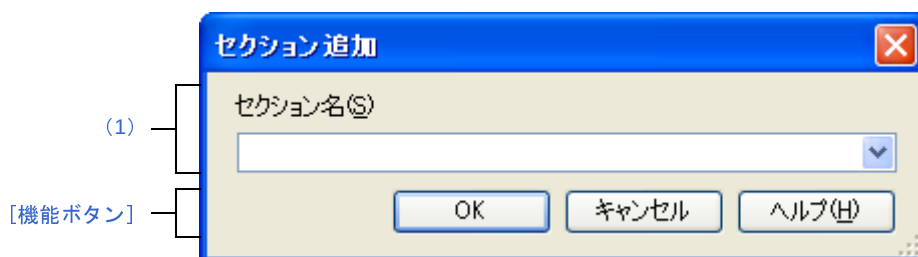


図 A—31 セクション編集 ダイアログ

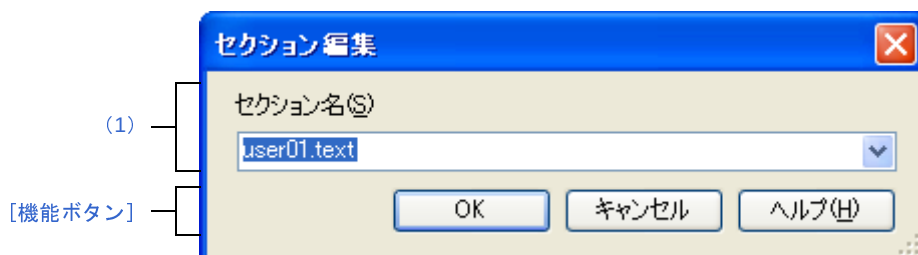
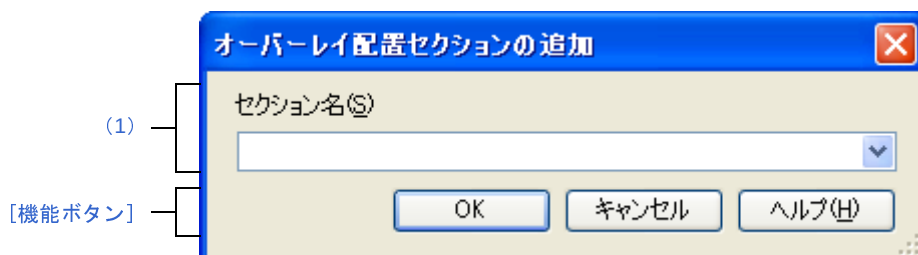


図 A—32 オーバーレイ配置セクションの追加 ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- セクション追加 ダイアログ
 - [セクション設定 ダイアログ](#) において、アドレス - セクション・エリア内のセクションを選択したのち、[追加 ...] ボタンをクリック
- セクション編集 ダイアログ
 - [セクション設定 ダイアログ](#) において、アドレス - セクション・エリア内のセクションを選択したのち、[変更 ...] ボタンをクリック
- オーバーレイ配置セクションの追加 ダイアログ
 - [セクション設定 ダイアログ](#) において、[複数割り付け ...] ボタンをクリック

[各エリアの説明]

(1) [セクション名]

セクション名を指定します。

セクション名をテキスト・ボックスに直接入力、またはドロップダウン・リストにより選択します。

使用可能な文字は、英大文字 (A ~ Z)、英小文字 (a ~ z)、数字 (0 ~ 9)、アンダスコア (_)、ドル記号 (\$)、ドット (.), チルダ (~) です。

また、ワイルドカード (*) も使用可能です。

なお、セクション名の先頭文字に数字 (0 ~ 9) を使用することはできません。

ドロップダウン・リストには、以下の予約セクションが設定されています。

.bss, .const, .data, .text

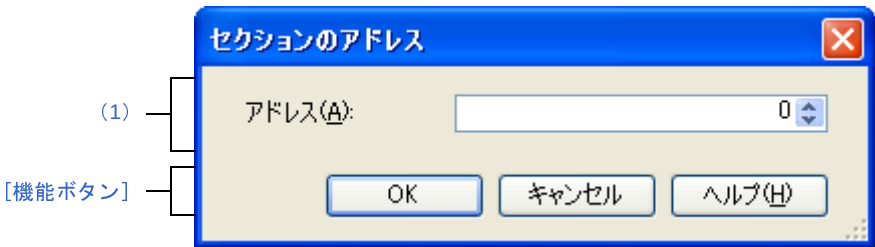
[機能ボタン]

ボタン	機能
OK	- セクション追加 ダイアログの場合 本ダイアログをクローズし、指定したセクションをセクション設定 ダイアログのアドレス - セクション・エリアに追加します。 指定したセクションが属するセクション・グループ (アドレスと属するセクション) の該当列に空欄がない場合は、そのセクション・グループの最下部に新しいセクション行を追加します。空欄がある場合は、そこにセクションを設定します。 - セクション編集 ダイアログの場合 本ダイアログをクローズし、セクション設定 ダイアログのアドレス - セクション・エリアで選択しているセクション名を指定したセクション名に置き換えます。 - オーバーレイ配置セクションの追加 ダイアログの場合 本ダイアログをクローズし、セクション設定 ダイアログのアドレス - セクション・エリアに [オーバーレイ <i>n</i>] 列 (<i>n</i>: 1 で始まる数字) を追加します。 指定したセクションを選択しているセクション・グループの該当列に設定します。
キャンセル	設定を無効とし、本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

セクションのアドレス ダイアログ

セクションの追加，または変更時のアドレスの設定を行います。

図 A—33 セクションのアドレス ダイアログ



ここでは，次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [セクション設定 ダイアログ](#)において，アドレス - セクション・エリア内のアドレスを選択したのち，[追加 ...]，または[変更 ...] ボタンをクリック

[各エリアの説明]

(1) [アドレス]

セクションの開始アドレスを指定します。
アドレスをテキスト・ボックスに直接入力，または  ボタンにより選択します。
指定可能な値の範囲は，0 ～ FFFFFFFF（16 進数）です（デフォルト：0）。

[機能ボタン]

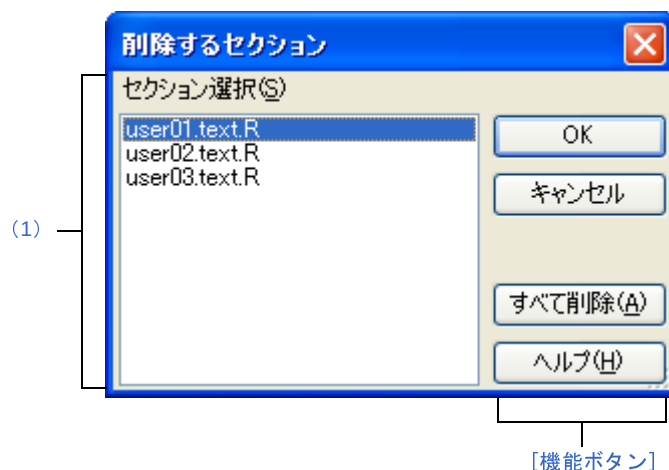
ボタン	機能
OK	- セクション設定 ダイアログの [追加 ...] ボタンからオープンした場合 本ダイアログをクローズし，指定したアドレスをセクション設定 ダイアログのアドレス - セクション・エリアの適切な位置に挿入します（セクションは空欄）。 - セクション設定 ダイアログの [変更 ...] ボタンからオープンした場合 本ダイアログをクローズし，指定したアドレスに応じて，セクション・グループ（アドレスと属するセクション）をセクション設定 ダイアログのアドレス - セクション・エリアの適切な位置に移動します。
キャンセル	設定を無効とし，本ダイアログをクローズします。

ボタン	機能
ヘルプ	本ダイアログのヘルプを表示します。

削除するセクション ダイアログ

セクションの削除を行います。

図 A—34 削除するセクション ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [セクション設定 ダイアログ](#)において、アドレス - セクション・エリア内のアドレスを選択したのち、[削除 ...] ボタンをクリック

[各エリアの説明]

(1) [セクション選択]

[セクション設定 ダイアログ](#)で選択しているアドレスに属するすべてのセクション名を表示します。

セクション名をクリックすることにより、削除するセクションを選択します。

[Ctrl] キー+左クリック、または [Shift] キー+左クリックにより、複数選択することができます。

[機能ボタン]

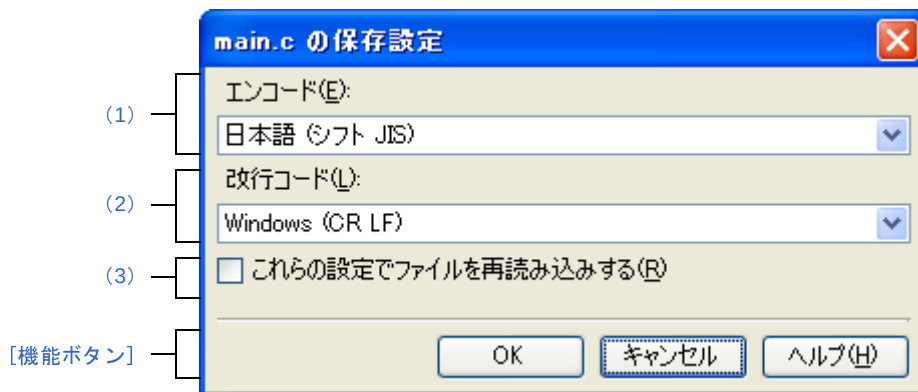
ボタン	機能
OK	本ダイアログをクローズし、選択したセクションを セクション設定 ダイアログ のアドレス - セクション・エリアから削除します。 セクション・グループ（アドレスと属するセクション）に属するセクションがなくなる場合は、そのセクション・グループを削除します。 アドレス - セクション・エリアの「オーバーレイ <i>n</i> 」列に属するセクション名がなくなる場合は、その列を削除します。
キャンセル	設定を無効とし、本ダイアログをクローズします。
すべて削除	本ダイアログをクローズし、すべてのセクション（ セクション設定 ダイアログ のアドレス - セクション・エリアで選択しているセクション・グループ）を削除します。
ヘルプ	本ダイアログのヘルプを表示します。

ファイルの保存設定 ダイアログ

エディタ パネルで編集中のファイルのエンコードと改行コードの設定を行います。

備考 タイトルバーには、設定対象ファイルの名前を表示します。

図 A—35 ファイルの保存設定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- エディタ パネルにフォーカスがある状態で、[ファイル] メニュー→ [ファイル名の保存設定 ...] を選択

[各エリアの説明]

(1) [エンコード]

設定するエンコードをドロップダウン・リストにより選択します。

ドロップダウン・リストの項目は、以下の順番で表示します。

ただし、同じエンコード名、および現在の OS が対応していないエンコード名は表示しません。

- 現在のファイルのエンコード名 (デフォルト)
- 現在の OS の既定のエンコード名
- 最近使用した エンコード名 (最大4 件)
- 現在のロケールでよく使用されているエンコード名

(例：ロケールが日本の場合)

- 日本語 (シフト JIS)
- 日本語 (JIS 1 バイト カタカナ可 - SO/SI)
- 日本語 (EUC)

- Unicode (UTF-8)

- 現在の OS が対応する上記以外のエンコード名（アルファベット順）

(2) [改行コード]

設定する改行コードをドロップダウン・リストにより選択します。

以下の項目を選択することができます。

- 現在の改行コードを維持
- Windows (CR LF)
- Macintosh (CR)
- Unix (LF)

デフォルトでは、現在の改行コードを選択します。

(3) [これらの設定でファイルを再読み込みする]

☒	[OK] ボタンをクリックした際に、指定したエンコード、および改行コードでファイルの再読み込みを行います。
☐	[OK] ボタンをクリックした際に、ファイルの再読み込みを行いません（デフォルト）。

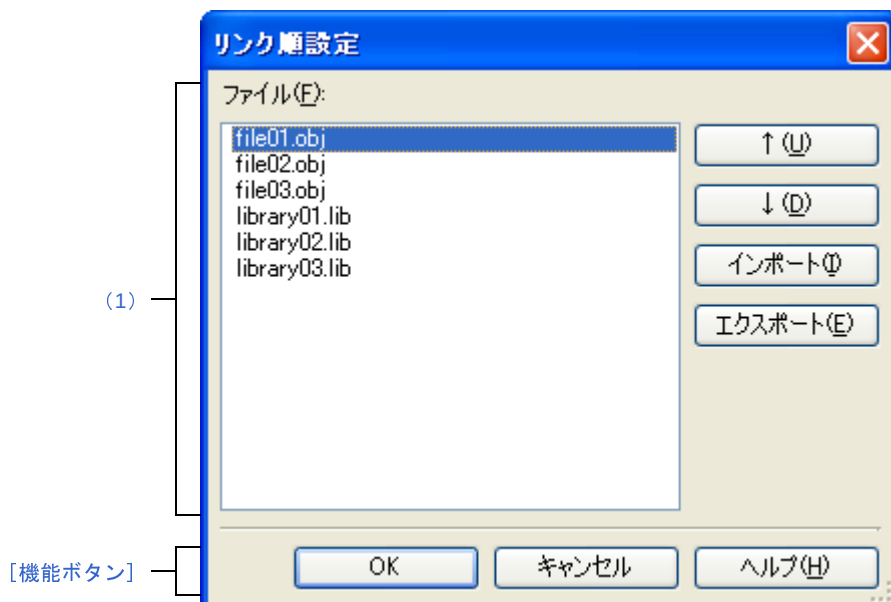
[機能ボタン]

ボタン	機能
OK	指定したエンコード、および改行コードを対象ファイルに設定し、本ダイアログをクローズします。 [これらの設定でファイルを再読み込みする] をチェックした場合、指定したエンコード、および改行コードを対象ファイルに設定し、ファイルを読み込み直したのち、このダイアログをクローズします。
キャンセル	設定を無効とし、本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

リンク順設定 ダイアログ

リンクカに入力するオブジェクト・ファイル、およびライブラリ・ファイルの参照、およびリンク順の設定を行います。

図 A—36 リンク順設定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロジェクト・ツリー パネルにおいて、ビルド・ツール・ノードを選択したのち、コンテキスト・メニュー→[リンク順を設定する ...] を選択

[各エリアの説明]

(1) [ファイル] エリア

以下のファイルのファイル名一覧を、リンクカへの入力順に表示します。

- 選択しているメイン・プロジェクト、またはサブプロジェクトに追加しているソース・ファイルから生成するオブジェクト・ファイル
- 選択しているメイン・プロジェクト、またはサブプロジェクトのプロジェクト・ツリーに直接追加したオブジェクト・ファイル
- 選択しているメイン・プロジェクト、またはサブプロジェクトのプロジェクト・ツリーに直接追加したライブラリ・ファイル

デフォルトでは、プロジェクトに追加している順番となります。

ファイルの表示順を変更することにより、リンカへのファイルの入力順を設定することができます。

表示順の変更は、[↑]、および[↓] ボタン、またはファイル名のドラッグ・アンド・ドロップにより行います。

- 備考 1.** ファイル名にマウス・カーソルをあわせると、そのファイルの存在する場所がプロジェクト・ファイルと同一のドライブの場合は相対パスで、異なるドライブの場合は絶対パスでポップアップ表示します。
- 2.** 新規に追加したソース・ファイルから生成するオブジェクト・ファイル、および新規に追加したオブジェクト・ファイルは、一覧の最後のオブジェクト・ファイルの次に追加します。
新規に追加したライブラリ・ファイルは、一覧の最後に追加します。
- 3.** ファイルをドラッグ・アンド・ドロップする際、連続して並んでいるファイル名のみ複数選択することができます。

(a) ボタン

↑	選択しているファイルを上へ移動します。 なお、本ボタンは、ファイルを選択していない場合は無効となります。
↓	選択しているファイルを下へ移動します。 なお、本ボタンは、ファイルを選択していない場合は無効となります。
インポート	インポートするファイルを選択 ダイアログをオープンします。 選択したリンク順指定ファイルからファイル名の記述順を取得し、[ファイル] に反映します。 なお、本ボタンは、[ファイル] に何も表示されていない場合は無効となります。
エクスポート	エクスポートするファイルを選択 ダイアログをオープンします。 [ファイル] に表示しているファイル名一覧を、指定したリンク順指定ファイルに出力します。 なお、本ボタンは、[ファイル] に何も表示されていない場合は無効となります。

備考 リンク順指定ファイルの利用方法については、「[2.11.2 ファイルのリンク順を設定する](#)」を参照してください。

[機能ボタン]

ボタン	機能
OK	リンカへのファイルの入力順を「ファイル」の表示順に設定し、本ダイアログをクローズします。
キャンセル	リンク順の設定をキャンセルし、本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

ビルド・モード設定 ダイアログ

ビルド・モードの追加と削除，および現在のビルド・モードの一括設定を行います。

図 A—37 ビルド・モード設定 ダイアログ



ここでは，次の項目について説明します。

- [\[オープン方法\]](#)
- [\[各エリアの説明\]](#)
- [\[機能ボタン\]](#)

[オープン方法]

- [ビルド] メニュー→ [ビルド・モードの設定 ...] を選択

[各エリアの説明]

(1) [変更後のビルド・モード] エリア

[\[ビルド・モードの一覧\]](#) エリアで選択しているビルド・モードを表示します。

(a) ボタン

すべてに適用	表示しているビルド・モードを現在開いているプロジェクトのメイン・プロジェクト，およびすべてのサブプロジェクトに設定します。
--------	---

(2) [ビルド・モードの一覧] エリア

現在開いているプロジェクト（メイン・プロジェクト，およびサブプロジェクト）に存在するすべてのビルド・モードを一覧表示します。

デフォルトでは、すべてのプロジェクトの現在のビルド・モードが一致している場合は、そのビルド・モードを選択します。

一致していない場合は、“DefaultBuild”を選択します。

一部のメイン・プロジェクト、およびサブプロジェクトのみに存在するビルド・モードには、“*”を付加します。

なお、ビルド・モードには、あらかじめ“DefaultBuild”が用意されており、常に先頭に表示します。

(a) ボタン

複製 ...	選択しているビルド・モードを複製します。 文字列入力 ダイアログがオープンし、入力した名前でビルド・モードを複製し、現在開いているプロジェクトのメイン・プロジェクト、およびすべてのサブプロジェクトに追加します。 なお、“*”を付加しているビルド・モードを複製する場合、そのビルド・モードがメイン・プロジェクト、およびサブプロジェクトに存在しなければ、DefaultBuildを複製します。 登録可能なビルド・モード数は、20 個までです。
削除	選択しているビルド・モードを削除します。 ただし、DefaultBuild を削除することはできません。
名前の変更 ...	選択しているビルド・モードの名前を変更します。 文字列入力 ダイアログがオープンし、入力した名前でビルド・モードの名前を変更します。

注意 ビルド・モードを複製、およびビルド・モードの名前を変更する場合、すでに存在するビルド・モードと同名の名前を使用することはできません。

備考 1. ビルド・モード名として指定可能な文字数は 127 文字までです。

入力内容が正しくない場合、以下のメッセージをツールチップ表示します。

メッセージ	説明
同名のビルド・モードがすでに存在します。	同名のビルド・モードがすでに存在します。
127 文字を越える文字を指定できません。	長い名前（128 文字以上）のビルド・モードを指定しました。
ビルド・モード名が不正です。文字 (¥, /, :, *, ?, ", <, >,) は使用できません。	不正なビルド・モード名を指定しました。 ビルド・モード名のフォルダを作成するため、文字 (¥, /, :, *, ?, ", <, >,) は使用できません。

2. 登録可能なビルド・モード数は、20 個までです。

入力内容が正しくない場合、以下のメッセージをツールチップ表示します。

メッセージ	説明
1 つのプロジェクト／サブプロジェクトに設定できるビルド・モード数は、20 個までです。	登録するビルド・モード数が 20 個を越えました。

[機能ボタン]

ボタン	機能
閉じる	本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

バッチ・ビルド ダイアログ

プロジェクト（メイン・プロジェクト、およびサブプロジェクト）が持つビルド・モードを一括して、ビルド、リビルド、クリーンを行います。

備考 バッチ・ビルド順は、プロジェクトのビルド順に従い、サブプロジェクト、メイン・プロジェクトの順となります。

1つのメイン・プロジェクト、またはサブプロジェクトについて複数のビルド・モードを選択した場合は、そのサブプロジェクトで選択しているすべてのビルド・モードでビルドを行ったのち、次のサブプロジェクト、またはメイン・プロジェクトのビルドを行います。

図 A—38 バッチ・ビルド ダイアログ



ここでは、次の項目について説明します。

- [\[オープン方法\]](#)
- [\[各エリアの説明\]](#)
- [\[機能ボタン\]](#)

[オープン方法]

- [ビルド] メニュー→ [バッチ・ビルド ...] を選択

[各エリアの説明]

(1) [ビルド・モード一覧] エリア

現在開いているプロジェクトが持つメイン・プロジェクト、およびサブプロジェクトの名前と、それらが持つビルド・モード、定義マクロの組み合わせの一覧を表示します。

(a) [プロジェクト]

現在開いているプロジェクトが持つメイン・プロジェクト、およびサブプロジェクトを表示します。

ビルドを行うメイン・プロジェクト、およびサブプロジェクトとビルド・モードの組み合わせをチェック・ボックスにより選択します。

プロジェクトを作成後、最初に本ダイアログをオープンした場合は、すべてのチェック・ボックスをチェックしません。2回目以降は前回のチェック状態を保持します。

(b) [ビルド・モード]

メイン・プロジェクト、およびサブプロジェクトが持つビルド・モードを表示します。

(c) [定義マクロ]

メイン・プロジェクト、およびサブプロジェクトとそのビルド・モードの組み合わせに対して、[プロパティ](#) パネルの [\[コンパイル・オプション\]](#) タブ、および [\[アセンブル・オプション\]](#) タブで設定している定義マクロを“|”で区切って表示します。

なお、コンパイル・オプションの定義マクロ、アセンブル・オプションの定義マクロの順で表示し、コンパイル・オプションの定義マクロとアセンブル・オプションの定義マクロの間は“,”で区切って表示します。

備考 1. 以下のいずれかの方法により、行を複数選択することができます。

- [Ctrl] キーを押下しながらマウスで左クリック
- [Shift] キーを押下しながらマウスで左クリック
- マウスをドラッグして範囲選択

また、選択状態にてスペース・キーを押下することにより、チェック・ボックスのチェック状態を切り替えることができます。

2. 各列のヘッダをクリックすることにより、その列をソートすることができます（デフォルト：昇順）。

同じヘッダを再度クリックすると、逆方向にソートします。

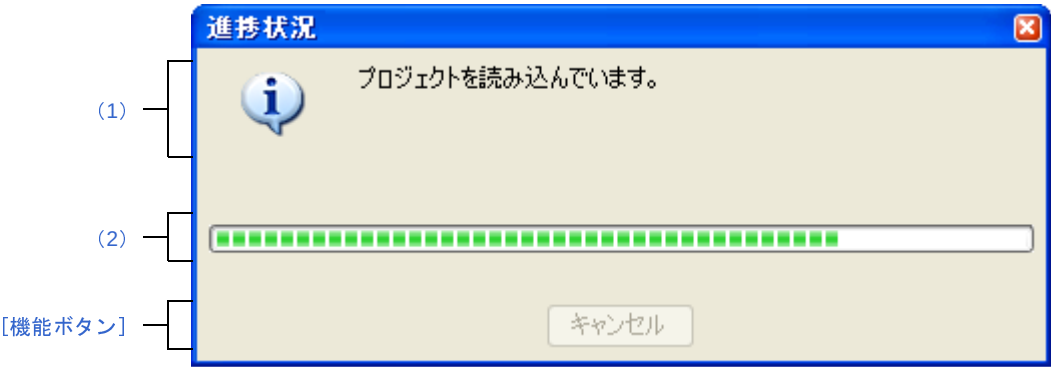
[機能ボタン]

ボタン	機能
ビルド	本ダイアログをクローズし、選択しているプロジェクトをそのビルド・モードでビルドします。 ビルドの実行結果は、 出力パネル に表示します。 ビルド完了後、ビルド・モードは本ダイアログをオープンする前の設定に戻ります。 なお、本ボタンは、プロジェクトを選択していない場合は無効となります。
リビルド	本ダイアログをクローズし、選択しているプロジェクトをそのビルド・モードでリビルドします。 リビルドの実行結果は、 出力パネル に表示します。 リビルド完了後、ビルド・モードは本ダイアログをオープンする前の設定に戻ります。 なお、本ボタンは、プロジェクトを選択していない場合は無効となります。
クリーン	本ダイアログをクローズし、選択しているプロジェクトのそのビルド・モードでビルドしたファイルを削除します。 クリーンの実行結果は、 出力パネル に表示します。 クリーン完了後、ビルド・モードは本ダイアログをオープンする前の設定に戻ります。 なお、本ボタンは、プロジェクトを選択していない場合は無効となります。
閉じる	本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

処理中表示 ダイアログ

時間を要する処理を行っている際に、その進捗状況の表示を行います。
本ダイアログは、実行中の処理が完了した場合、自動的にクローズします。

図 A—39 処理中表示 ダイアログ



ここでは、次の項目について説明します。

- [\[オープン方法\]](#)
- [\[各エリアの説明\]](#)
- [\[機能ボタン\]](#)

[オープン方法]

- 時間を要する処理において、メッセージが発生した際に自動的に表示

[各エリアの説明]

(1) メッセージ表示エリア

処理中に発生したメッセージを表示します（編集不可）。

(2) プログレスバー

現在実行中の処理の進捗状況をバーの長さで表示します。

なお、進捗率が 100% に達した場合（右端までバーの長さが達した場合）、本ダイアログは自動的にクローズします。

[機能ボタン]

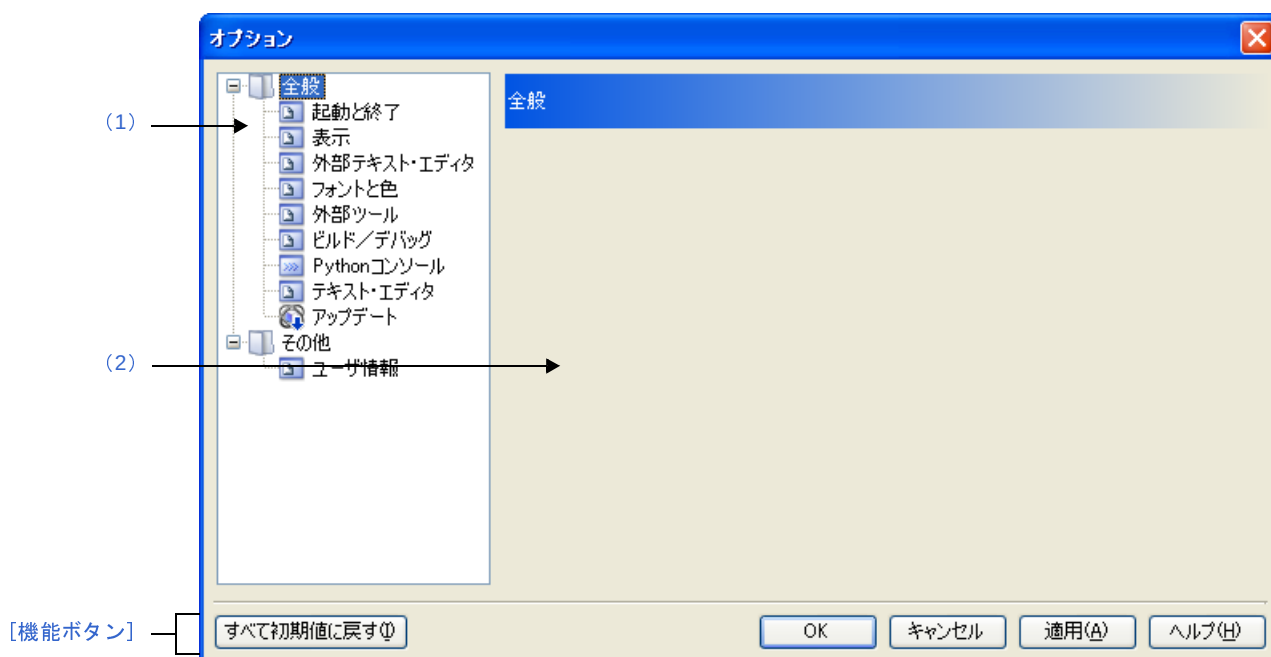
ボタン	機能
キャンセル	現在実行中の処理を中断し、本ダイアログをクローズします。 ただし、実行中の処理の中断が不可能な場合、本ボタンは無効となります。

オプション ダイアログ

CubeSuite+ の各種環境設定を行います。

本ダイアログでの設定は、使用中のユーザの設定として保存します。

図 A—40 オプション ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [ツール] メニュー→ [オプション ...] を選択

[各エリアの説明]

(1) カテゴリ選択エリア

設定したい項目を次のカテゴリから選択します。

カテゴリ	設定内容
[全般 - 起動と終了] カテゴリ	起動、または終了時に関連した設定を行います。
[全般 - 表示] カテゴリ	表示に関連した設定を行います。
[全般 - 外部テキスト・エディタ] カテゴリ	外部テキスト・エディタに関連した設定を行います。
[全般 - フォントと色] カテゴリ	各パネルで表示するフォントと色に関連した設定を行います。
[全般 - 外部ツール] カテゴリ	外部ツールを起動する際の設定を行います。
[全般 - ビルド／デバッグ] カテゴリ	ビルド、またはデバッグに関連した設定を行います。
[全般 - Python コンソール] カテゴリ	Python コンソールに関連した設定を行います。
[全般 - テキスト・エディタ] カテゴリ	テキスト・エディタに関連した設定を行います。
[全般 - アップデート] カテゴリ	アップデートに関連した設定を行います。
[その他 - ユーザ情報] カテゴリ	ユーザ情報に関連した設定を行います。

備考 [全般 - ビルド／デバッグ] 以外のカテゴリについては、「CubeSuite+ 統合開発環境 ユーザーズマニュアル 起動編」を参照してください。

(2) 設定エリア

選択したカテゴリに対して、各種オプションを設定するエリアです。

各カテゴリの設定方法についての詳細は、該当するカテゴリの項を参照してください。

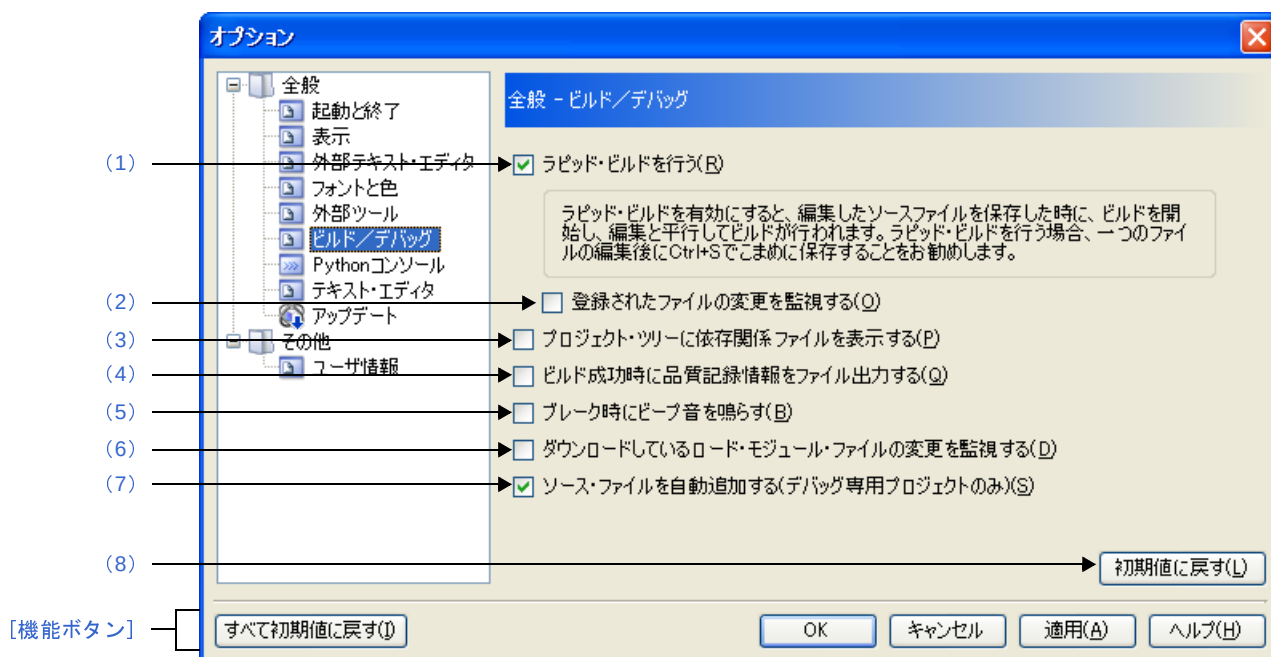
[機能ボタン]

ボタン	機能
すべて初期値に戻す	本ダイアログのすべての設定項目をデフォルトの状態に戻します。 ただし、[全般 - 外部ツール] カテゴリでは、新規登録した内容の削除は行いません。
OK	変更した設定内容を適用し、本ダイアログをクローズします。
キャンセル	変更した設定内容を無効とし、本ダイアログをクローズします。
適用	変更した設定内容を適用します（本ダイアログをクローズしません）。
ヘルプ	本ダイアログのヘルプを表示します。

[全般 - ビルド／デバッグ] カテゴリ

全般に関わる設定のうち、ビルド、またはデバッグに関連した設定を行います。

図 A—41 オプション ダイアログ ([全般 - ビルド／デバッグ] カテゴリ)



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [ツール] メニュー→ [オプション ...] を選択

[各エリアの説明]

- (1) [ラピッド・ビルドを行う]

☒	ラピッド・ビルド機能 ^注 を有効にします（デフォルト）。
☐	ラピッド・ビルド機能を使用しません。

注 編集したソース・ファイルの保存時に、ビルドを自動で開始する機能です。

本機能を有効にすることにより、ソース・ファイルの編集と同時にビルドを行うことができます。

なお、本機能を使用する場合、ソース・ファイル編集後、こまめに上書き保存することを推奨します。

(2) [登録されたファイルの変更を監視する]

☒	プロジェクトに登録したソース・ファイルの変更を監視し、外部テキスト・エディタなどで編集／保存したときに、ラピッド・ビルドを開始します。
☐	プロジェクトに登録したソース・ファイルの変更を監視し、外部テキスト・エディタなどで編集／保存したときに、ラピッド・ビルドを開始しません（デフォルト）。

備考 [ラピッド・ビルドを行う] チェック・ボックスにチェックが付いている場合のみ有効です。

注意 1. 本項目をチェックし、かつ、ラピッド・ビルドの対象となったファイルをビルド前に実行するコマンド、ビルド後に実行するコマンドなどで自動で編集／上書きするように登録した場合、ラピッド・ビルドが終了しなくなります。

ラピッド・ビルドが終了しなくなった場合は、本項目のチェックを外して、ラピッド・ビルドを停止してください。

2. 本項目をチェックし、かつ、プロジェクトに登録されたソース・ファイルで存在しないファイル（プロジェクト・ツリーでグレー表示されたファイル）がある場合、エクスプローラなどでファイルを再登録しても、監視状態にはなりません。

監視状態にするためには、プロジェクト・ファイルを読み込み直すか、または本項目のチェックを一旦外してダイアログを閉じた後、再度本項目をチェックしてください。

(3) [プロジェクト・ツリーに依存関係ファイルを表示する]

☒	ソース・ファイルが依存しているファイル群をプロジェクト・ツリーに表示します。
☐	ソース・ファイルが依存しているファイル群をプロジェクト・ツリーに表示しません（デフォルト）。

(4) [ビルド成功時に品質記録情報をファイル出力する]

☒	ビルド成功時に品質記録情報ファイルを出力します。
☐	ビルド成功時に品質記録情報ファイルを出力しません（デフォルト）。

備考 1. 品質記録情報ファイルは、ラピッド・ビルドを行う場合、デバッグ専用プロジェクトをビルドする場合、ファイル単位でコンパイル／アセンブルする場合は出力しません。

2. 品質記録情報ファイルには、以下の情報を出力します。

- ファイルの作成日時
- ビルド結果のログ
- ビルド中に使用したコマンド・ファイルの情報
- 本製品の詳細バージョンや現在のプロジェクトの情報

3. 品質記録情報ファイルは、各プロジェクトのプロジェクト・フォルダに“品質記録情報 (プロジェクト名, ビルド・モード名).txt” というファイル名で出力します。

同名のファイルが存在する場合は上書きします。

また、プロジェクト・ツリーのビルド・ツール生成ファイル・ノードにも表示します。

(5) [ブレーク時にビーブ音を鳴らす]

☒	プログラムの実行が、ブレーク・イベント（ハードウェア・ブレーク／ソフトウェア・ブレーク）により停止した際、ビーブ音を鳴らします。
☐	プログラムの実行が、ブレーク・イベント（ハードウェア・ブレーク／ソフトウェア・ブレーク）により停止した際、ビーブ音を鳴らしません（デフォルト）。

(6) [ダウンロードしているロード・モジュール・ファイルの変更を監視する]

☒	デバッグ・ツールにダウンロードしているロード・モジュール・ファイルの変更を監視し、変更があった場合は、ダウンロードの実行を確認するメッセージダイアログを表示します。
☐	デバッグ・ツールにダウンロードしているロード・モジュール・ファイルの変更の監視を行いません（デフォルト）。

(7) [ソース・ファイルを自動追加する (デバッグ専用プロジェクトのみ)]

☒	デバッグ専用プロジェクトにおいて、デバッグ・ツールにロード・モジュール・ファイルをダウンロードする際、プロジェクト・ツリーにソース・ファイルを自動追加します（デフォルト）。
☐	デバッグ専用プロジェクトにおいて、デバッグ・ツールにロード・モジュール・ファイルをダウンロードする際、プロジェクト・ツリーへのソース・ファイルの自動追加を行いません。

注意 本機能は、プロジェクト・ツリーのダウンロード・ファイル・ノードにロード・モジュール・ファイルを追加した場合のみ有効となります。

デバッグ・ツールのプロパティ パネルの [ダウンロード・ファイル設定] タブにてロード・モジュール・ファイルを追加した場合は、プロジェクト・ツリーにソース・ファイルは追加されません。

(8) ボタン・エリア

初期値に戻す	現在表示している項目をすべてデフォルトに戻します。
--------	---------------------------

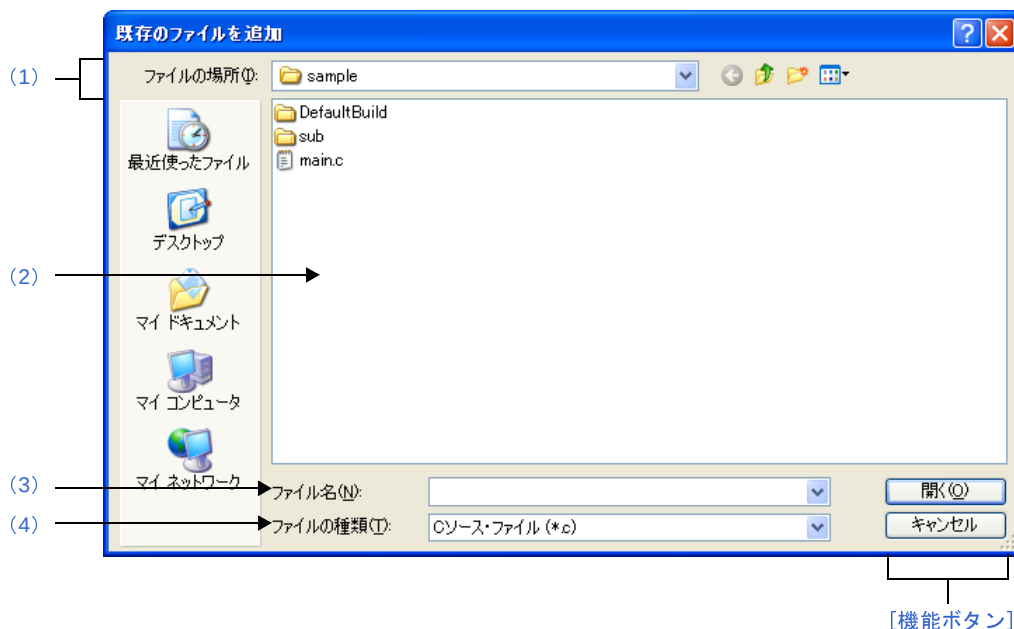
[機能ボタン]

ボタン	機能
すべて初期値に戻す	本ダイアログのすべての設定項目をデフォルトの状態に戻します。 ただし、[全般 - 外部ツール] カテゴリでは、新規登録した内容の削除は行いません。
OK	変更した設定内容を適用し、本ダイアログをクローズします。
キャンセル	変更した設定内容を無効とし、本ダイアログをクローズします。
適用	変更した設定内容を適用します（本ダイアログをクローズしません）。
ヘルプ	本ダイアログのヘルプを表示します。

既存のファイルを追加 ダイアログ

プロジェクトに追加する既存のファイルの選択を行います。

図 A—42 既存のファイルを追加 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [ファイル] メニュー → [追加] → [既存のファイルを追加 ...] を選択
- プロジェクト・ツリー パネルにおいて、プロジェクト・ノード、サブプロジェクト・ノード、ファイル・ノード、ファイルのいずれかを選択したのち、コンテキスト・メニュー → [追加] → [既存のファイルを追加 ...] を選択

[各エリアの説明]

(1) [ファイルの場所] エリア

プロジェクトに追加するファイルが存在するフォルダを選択します。

デフォルトでは、プロジェクト・フォルダを選択します。

(2) ファイルの一覧エリア

[ファイルの場所]、および [ファイルの種類] で選択した条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

プロジェクトに追加するファイルのファイル名を指定します。

(4) [ファイルの種類] エリア

プロジェクトに追加するファイルのファイルの種類（ファイル・タイプ）を選択します。

C ソース・ファイル (*.c)	C ソース・ファイル（デフォルト）
ヘッダ・ファイル (*.h; *.inc)	ヘッダ・ファイル
アセンブリ・ソース・ファイル (*.asm; *.s; *.fsy)	アセンブリ・ソース・ファイル
ライブラリ・ファイル (*.lib)	ライブラリ・ファイル
オブジェクト・ファイル (*.obj)	オブジェクト・ファイル
リロケータブル・ファイル (*.rel)	リロケータブル・ファイル
テキスト・ファイル (*.txt)	テキスト形式
すべてのファイル (*.*)	すべての形式

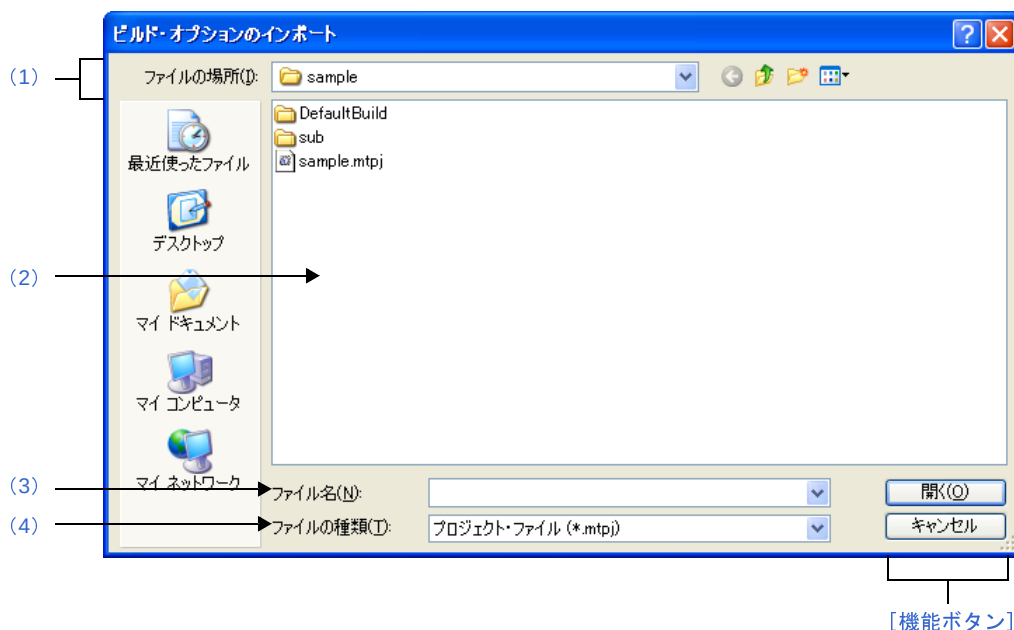
[機能ボタン]

ボタン	機能
開く	指定したファイルをプロジェクトに追加します。
キャンセル	本ダイアログをクローズします。

ビルド・オプションのインポート ダイアログ

ビルド・オプションのインポート対象となるプロジェクト・ファイルの選択を行います。

図 A—43 ビルド・オプションのインポート ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロジェクト・ツリー パネルにおいて、ビルド・ツール・ノードを選択したのち、コンテキスト・メニュー→[ビルド・オプションのインポート ...] を選択

[各エリアの説明]

(1) [ファイルの場所] エリア

ビルド・オプションのインポート対象となるプロジェクト・ファイルが存在するフォルダを選択します。
デフォルトでは、現在のプロジェクト・フォルダを選択します。

(2) ファイルの一覧エリア

[ファイルの場所]、および [ファイルの種類] で選択した条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

プロジェクト・ファイルのファイル名を指定します。

(4) [ファイルの種類] エリア

プロジェクト・ファイルの種類（ファイル・タイプ）を選択します。

プロジェクト・ファイル (*.mtpj)	プロジェクト・ファイル
サブプロジェクト・ファイル (*.mtsp)	サブプロジェクト・ファイル

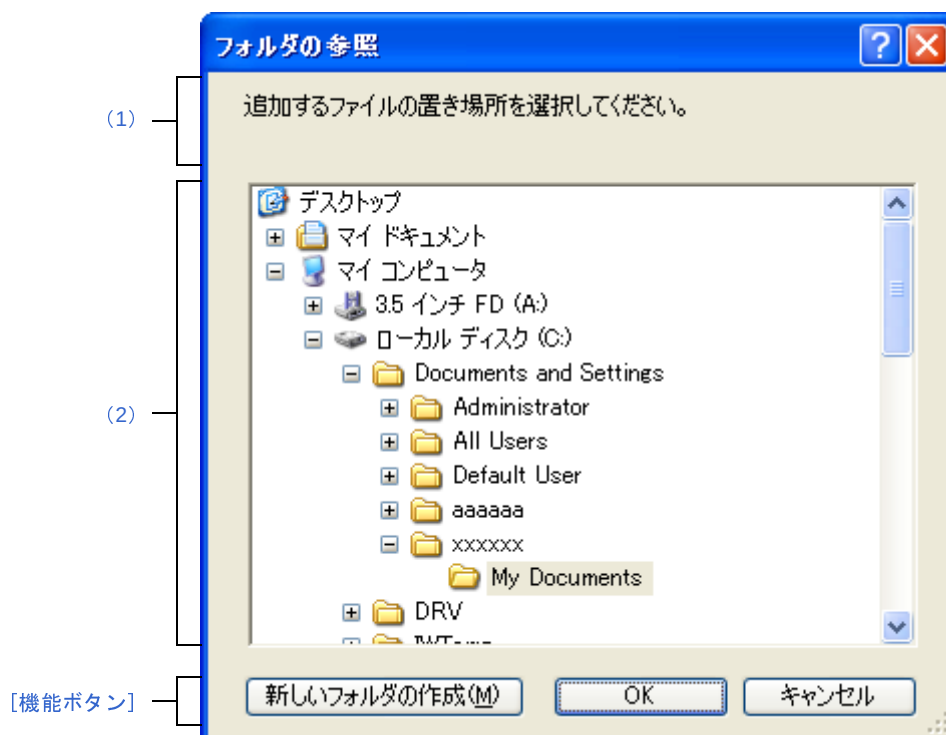
[機能ボタン]

ボタン	機能
開く	指定したプロジェクト・ファイルのビルド・オプションを現在のプロジェクトにインポートします。
キャンセル	本ダイアログをクローズします。

フォルダの参照 ダイアログ

本ダイアログの呼び出し元に設定するフォルダの選択を行います。

図 A—44 フォルダの参照 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- **ファイル追加 ダイアログ**において、[作成場所] エリア内の [参照 ...] ボタンをクリック
- **パス編集 ダイアログ**において、パス編集エリア内の [参照 ...] ボタンをクリック
- **プロパティ パネル**において、以下のプロパティを選択したのち、[...] ボタンをクリック
 - **[共通オプション] タブ**の [出力ファイルの種類と場所] カテゴリの [中間ファイル出力フォルダ], [よく使うオプション (リンク)] カテゴリの [出力フォルダ], [よく使うオプション (ヘキサ出力)] カテゴリの [出力フォルダ], [よく使うオプション (ライブラリ)] カテゴリの [出力フォルダ], [エラー出力] カテゴリの [エラー・メッセージ・ファイル出力フォルダ]
 - **[コンパイル・オプション] タブ**の [出力ファイル] カテゴリの [アセンブリ・ソース・ファイル出力フォルダ], [アセンブル・リスト] カテゴリの [アセンブル・リスト・ファイル出力フォルダ]
 - **[アセンブル・オプション] タブ**の [アセンブル・リスト] カテゴリの [アセンブル・リスト・ファイル出力フォルダ]

- **[リンク・オプション] タブ**の[出力ファイル] カテゴリの[出力フォルダ]
- **[ヘキサ出力オプション] タブ**の[出力ファイル] カテゴリの[出力フォルダ], [結合ヘキサ・ファイルの出力フォルダ]
- **[ライブラリ生成オプション] タブ**の[出力ファイル] カテゴリの[出力フォルダ]
- **[個別コンパイル・オプション] タブ**の[出力ファイル] カテゴリの[アセンブリ・ソース・ファイル出力フォルダ], [アセンブル・リスト] カテゴリの[アセンブル・リスト・ファイル出力フォルダ], [エラー出力] カテゴリの[エラー・メッセージ・ファイル出力フォルダ]
- **[個別アセンブル・オプション] タブ**の[アセンブル・リスト] カテゴリの[アセンブル・リスト・ファイル出力フォルダ], [エラー出力] カテゴリの[エラー・メッセージ・ファイル出力フォルダ]

[各エリアの説明]

(1) メッセージ・エリア

本ダイアログで選択するフォルダに関するメッセージを表示します。

(2) フォルダの場所エリア

本ダイアログの呼び出し元に設定するフォルダを選択します。

なお、デフォルトで選択されるフォルダは、呼び出し元によって異なります。

(a) ファイル追加 ダイアログの場合

呼び出し元に設定しているフォルダが選択されます。

呼び出し元が空欄、または存在しないパスを設定している場合は、プロジェクト・フォルダが選択されます。

(b) パス編集 ダイアログ、およびプロパティ パネルの場合

プロジェクト・フォルダが選択されます。

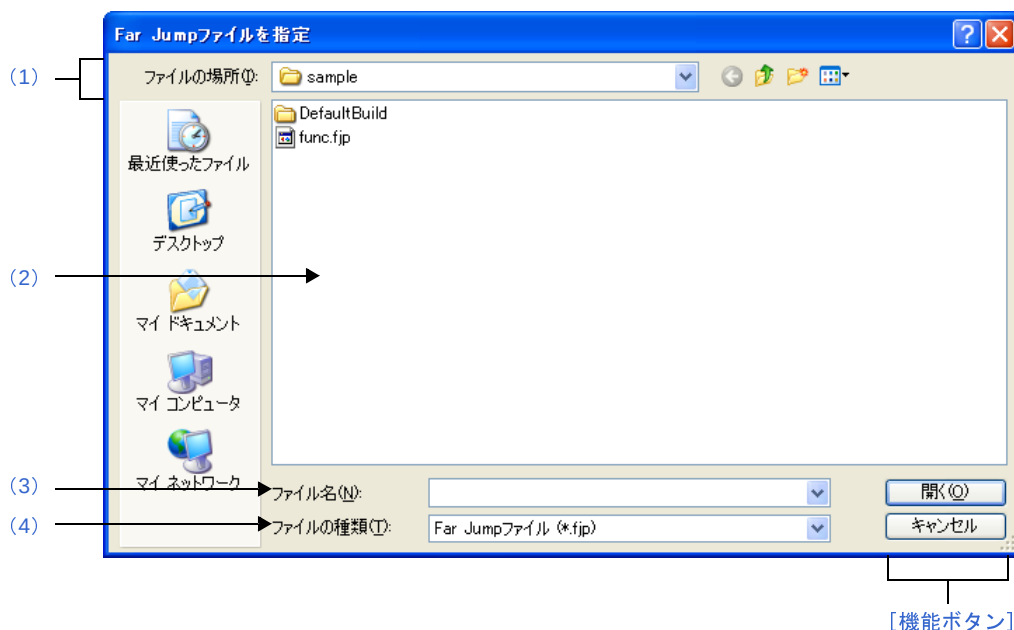
[機能ボタン]

ボタン	機能
新しいフォルダの作成	選択したフォルダの直下に新しいフォルダを作成します。 フォルダ名は、デフォルトで“新しいフォルダ”となります。
OK	指定したフォルダのパスを本ダイアログの呼び出し元に設定します。
キャンセル	本ダイアログをクローズします。

Far Jump ファイルを指定 ダイアログ

本ダイアログの呼び出し元に設定する Far Jump ファイルの選択を行います。

図 A—45 Far Jump ファイルを指定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルにおいて、以下のプロパティを選択したのち、[...] ボタンをクリック
- [コンパイル・オプション] タブの [出力コード] カテゴリの [Far Jump ファイル名]
- [個別コンパイル・オプション] タブの [出力コード] カテゴリの [Far Jump ファイル名]

[各エリアの説明]

(1) [ファイルの場所] エリア

本ダイアログの呼び出し元に設定するファイルが存在するフォルダを選択します。
デフォルトでは、プロジェクト・フォルダを選択します。

(2) ファイルの一覧エリア

[ファイルの場所]、および [ファイルの種類] で選択した条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

本ダイアログの呼び出し元に設定するファイルの名前を指定します。

(4) [ファイルの種類] エリア

本ダイアログの呼び出し元に設定するファイルの種類（ファイル・タイプ）を選択します。

Far Jump ファイル (*.fjp)	Far Jump ファイル
-----------------------	---------------

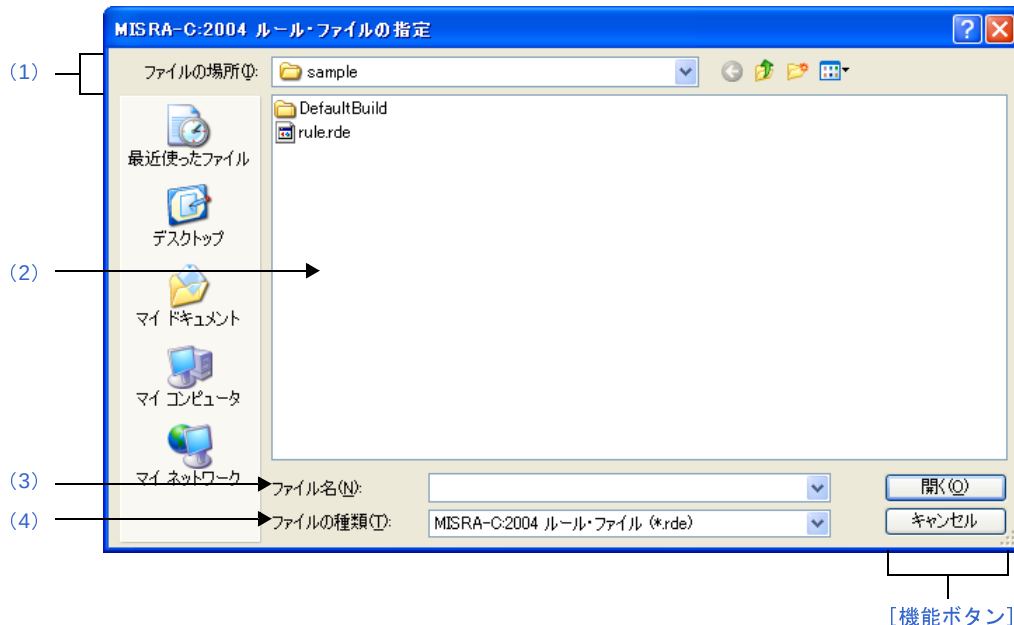
[機能ボタン]

ボタン	機能
開く	本ダイアログの呼び出し元に指定したファイルを設定します。
キャンセル	本ダイアログをクローズします。

MISRA-C:2004 ルール・ファイルの指定 ダイアログ

本ダイアログの呼び出し元に設定する MISRA-C:2004 ルール・ファイルの選択を行います。

図 A—46 MISRA-C:2004 ルール・ファイルの指定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルにおいて、以下のプロパティを選択したのち、[...] ボタンをクリック
- [コンパイル・オプション] タブの [MISRA-C:2004 ルール検査] カテゴリの [ルール番号記載ファイル]
- [個別コンパイル・オプション] タブの [MISRA-C:2004 ルール検査] カテゴリの [ルール番号記載ファイル]

[各エリアの説明]

(1) [ファイルの場所] エリア

本ダイアログの呼び出し元に設定するファイルが存在するフォルダを選択します。
デフォルトでは、プロジェクト・フォルダが選択されます。

(2) ファイルの一覧エリア

[ファイルの場所]、および [ファイルの種類] で選択された条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

本ダイアログの呼び出し元に設定するファイルの名前を指定します。

(4) [ファイルの種類] エリア

本ダイアログの呼び出し元に設定するファイルの種類（ファイル・タイプ）を選択します。

MISRA-C:2004 ルール・ファイル (*.rde)	MISRA-C:2004 ルール・ファイル（デフォルト）
すべてのファイル (*.*)	すべての形式

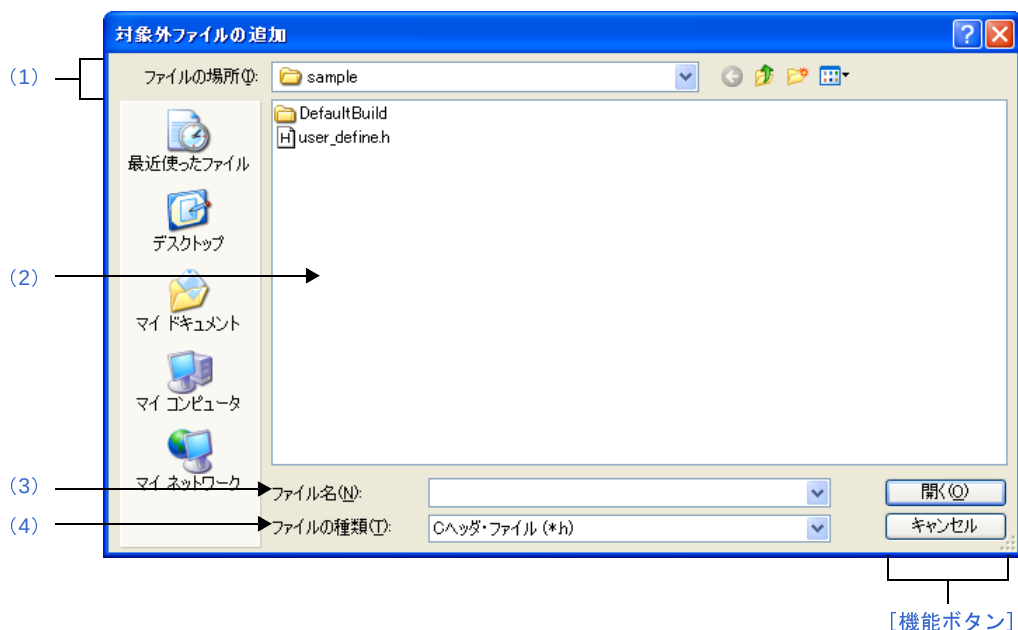
[機能ボタン]

ボタン	機能
開く	本ダイアログの呼び出し元に指定したファイルを設定します。
キャンセル	本ダイアログをクローズします。

対象外ファイルの追加 ダイアログ

本ダイアログの呼び出し元に設定する MISRA-C:2004 ルールのチェック対象外のファイルの選択を行います。

図 A—47 対象外ファイルの追加 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルにおいて、以下のプロパティを選択したのち、[...] ボタンをクリックして [パス編集 ダイアログ](#) をオープン
 - [コンパイル・オプション] タブの [MISRA-C:2004 ルール検査] カテゴリの [ルール・チェック対象外のファイル]
 - [個別コンパイル・オプション] タブの [MISRA-C:2004 ルール検査] カテゴリの [ルール・チェック対象外のファイル]
- ダイアログ上で [参照 ...] ボタンをクリック

[各エリアの説明]

(1) [ファイルの場所] エリア

本ダイアログの呼び出し元に設定するファイルが存在するフォルダを選択します。

デフォルトでは、プロジェクト・フォルダが選択されます。

(2) ファイルの一覧エリア

[ファイルの場所]、および [ファイルの種類] で選択された条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

本ダイアログの呼び出し元に設定するファイルの名前を指定します。

(4) [ファイルの種類] エリア

本ダイアログの呼び出し元に設定するファイルの種類（ファイル・タイプ）を選択します。

C ヘッダ・ファイル (*.h)	C ヘッダ・ファイル（デフォルト）
C ソース・ファイル (*.c)	C ソース・ファイル
すべてのファイル (*.*)	すべての形式

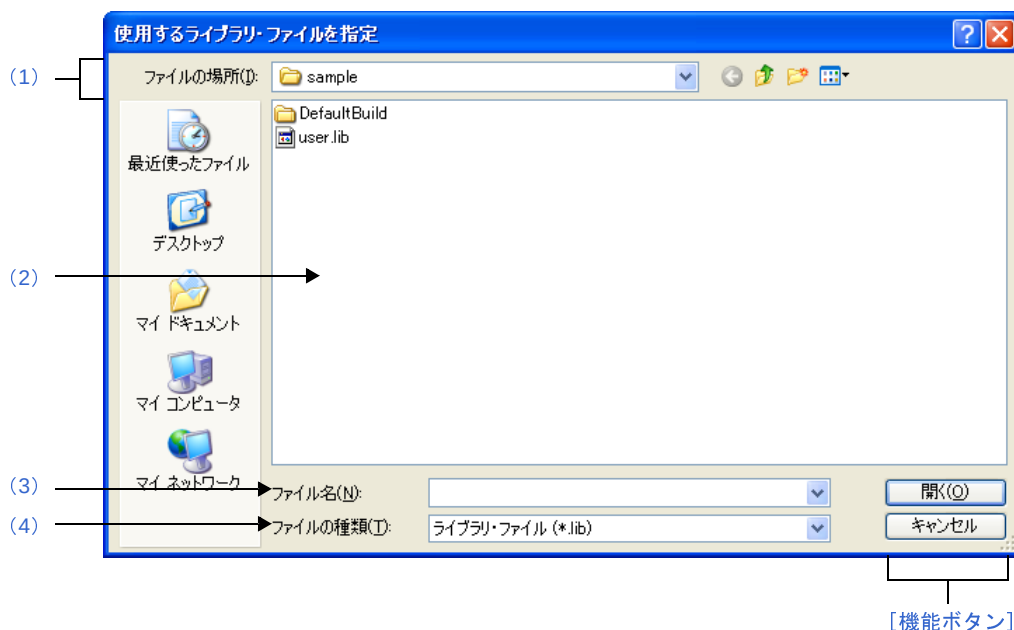
[機能ボタン]

ボタン	機能
開く	本ダイアログの呼び出し元に指定したファイルを設定します。
キャンセル	本ダイアログをクローズします。

使用するライブラリ・ファイルを指定 ダイアログ

本ダイアログの呼び出し元に設定するライブラリ・ファイルの選択を行います。

図 A—48 使用するライブラリ・ファイルを指定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルにおいて、以下のプロパティを選択したのち、[...] ボタンをクリックして [パス編集 ダイアログ](#) をオープン
 - [共通オプション] タブの [よく使うオプション (リンク)] カテゴリの [使用するライブラリ・ファイル]
 - [リンク・オプション] タブの [ライブラリ] カテゴリの [使用するライブラリ・ファイル]
 - [ライブラリ生成オプション] タブの [ライブラリ] カテゴリの [使用するライブラリ・ファイル]
- ダイアログ上で [参照 ...] ボタンをクリック

[各エリアの説明]

(1) [ファイルの場所] エリア

本ダイアログの呼び出し元に設定するファイルが存在するフォルダを選択します。

デフォルトでは、プロジェクト・フォルダが選択されます。

(2) ファイルの一覧エリア

[ファイルの場所]、および [ファイルの種類] で選択された条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

本ダイアログの呼び出し元に設定するファイルの名前を指定します。

(4) [ファイルの種類] エリア

本ダイアログの呼び出し元に設定するファイルの種類（ファイル・タイプ）を選択します。

ライブラリ・ファイル (*.lib)	ライブラリ・ファイル
--------------------	------------

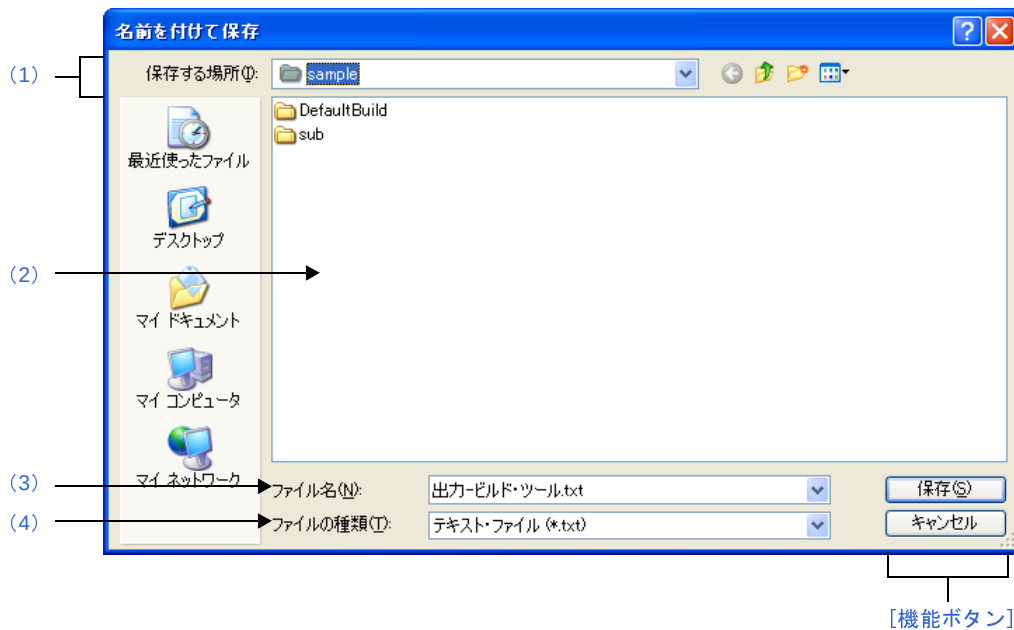
[機能ボタン]

ボタン	機能
開く	本ダイアログの呼び出し元に指定したファイルを設定します。
キャンセル	本ダイアログをクローズします。

名前を付けて保存 ダイアログ

編集集中のファイル、または各パネルの内容に名前を付けてファイルに保存します。

図 A—49 名前を付けて保存 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- エディタ パネルにフォーカスがある状態で、[ファイル] メニュー→ [名前を付けてファイル名を保存 ...] を選択
- 出力 パネルにフォーカスがある状態で、[ファイル] メニュー→ [名前を付けてタブ名を保存 ...] を選択

[各エリアの説明]

(1) [保存する場所] エリア

パネルに表示している内容をファイルに保存するためのフォルダを選択します。

デフォルトでは、以下のフォルダを選択します。

(a) エディタ パネルの場合

現在編集しているファイルが存在しているフォルダ

(b) 出力パネルの場合

初めて保存する場合はプロジェクト・フォルダ、2回目以降は前回選択したフォルダ

(2) ファイルの一覧エリア

〔保存する場所〕エリア、および〔ファイルの種類〕エリアで選択した条件に合致するファイルの一覧を表示します。

(3) 〔ファイル名〕エリア

保存する際のファイル名を指定します。

(4) 〔ファイルの種類〕エリア

(a) エディタパネルの場合

編集集中のファイルの種類に依存して、次のファイルの種類（ファイル・タイプ）を表示します。

備考 以下の表示文字列はプロジェクト・ツリーに登録しているファイルのみが対象です。

C ソース・ファイル (*.c)	C ソース・ファイル
ヘッダ・ファイル (*.h; *.inc)	ヘッダ・ファイル
アセンブリ・ソース・ファイル (*.asm; *.s)	アセンブリ・ソース・ファイル
リンク順指定ファイル (*.mtls)	リンク順指定ファイル
リンク・マップ・ファイル (*.map)	リンク・マップ・ファイル
ライブラリ・リスト・ファイル (*.lbp)	ライブラリ・リスト・ファイル
インテル拡張ヘキサ・ファイル (*.hex)	インテル拡張ヘキサ・ファイル
モトローラ・Sタイプ・ファイル (*.mot)	モトローラ・Sタイプ・ファイル
テキスト・ファイル (*.txt)	テキスト形式

(b) 出力パネルの場合

次のファイルの種類（ファイル・タイプ）を表示します。

テキスト・ファイル (*.txt)	テキスト形式
-------------------	--------

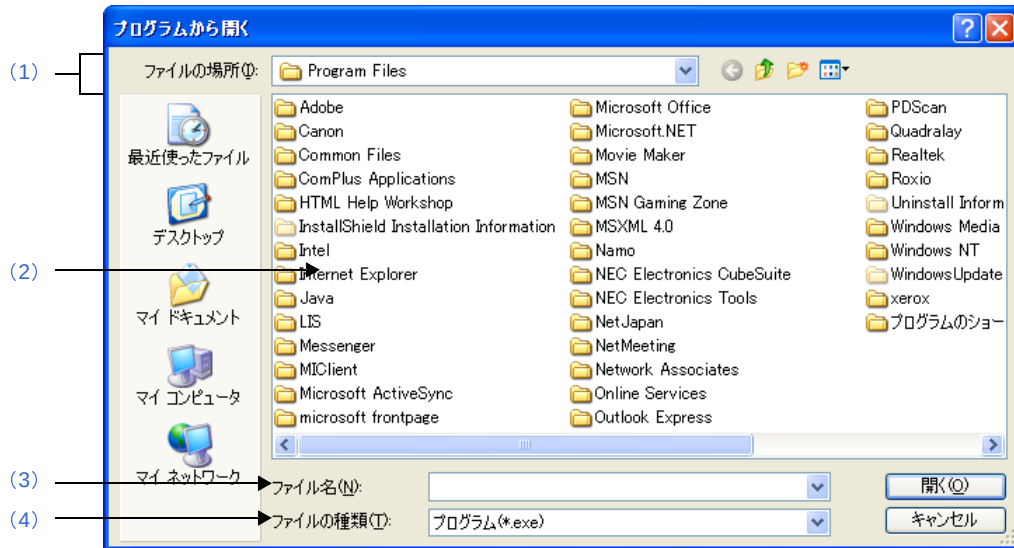
[機能ボタン]

ボタン	機能
保存	指定したファイル名でファイルを保存します。
キャンセル	本ダイアログをクローズします。

プログラムから開く ダイアログ

プロジェクト・ツリー上で選択しているファイルを開くアプリケーションの選択を行います。

図 A—50 プログラムから開く ダイアログ



[機能ボタン]

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロジェクト・ツリーパネルにおいて、ファイルを選択したのち、コンテキスト・メニュー→[アプリケーションを指定して開く...]を選択

[各エリアの説明]

(1) [ファイルの場所] エリア

ファイルを開くアプリケーションが存在するフォルダを選択します。

デフォルトでは、プログラム・フォルダ（Windows XP の場合は “C:\Program Files”）を選択します。

(2) ファイルの一覧エリア

[ファイルの場所]、および [ファイルの種類] で選択した条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

ファイルを開くアプリケーションの実行ファイル名を指定します。

(4) [ファイルの種類] エリア

ファイルを開くアプリケーションの実行ファイルの種類（ファイル・タイプ）を選択します。

プログラム (*.exe)	実行形式（デフォルト）
すべてのファイル (*.*)	すべての形式

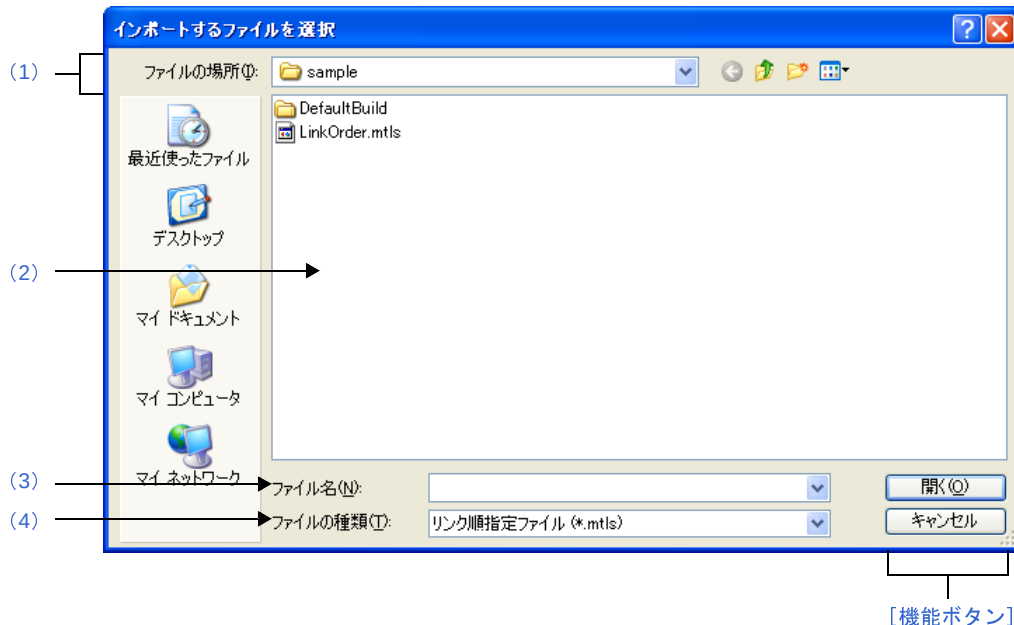
[機能ボタン]

ボタン	機能
開く	指定したアプリケーションでファイルをオープンします。
キャンセル	本ダイアログをクローズします。

インポートするファイルを選択 ダイアログ

本ダイアログの呼び出し元にインポートするファイルの選択を行います。

図 A—51 インポートするファイルを選択 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [リンク順設定 ダイアログ](#)において、[インポート] ボタンをクリック
- [セクション設定 ダイアログ](#)において、[インポート ...] ボタンをクリック

[各エリアの説明]

(1) [ファイルの場所] エリア

本ダイアログの呼び出し元にインポートするファイルが存在するフォルダを選択します。

デフォルトでは、初めて選択する場合はプロジェクト・フォルダ、2 回目以降は前回選択したフォルダを選択します。

(2) ファイルの一覧エリア

[ファイルの場所]、および [ファイルの種類] で選択した条件に合致するファイルの一覧を表示します。

(3) [ファイル名] エリア

本ダイアログの呼び出し元にインポートするファイルの名前を指定します。

(4) [ファイルの種類] エリア

本ダイアログの呼び出し元にインポートするファイルの種類（ファイル・タイプ）を選択します。

(a) リンク順設定 ダイアログの場合

リンク順指定ファイル (*.mtls)	リンク順指定ファイル
---------------------	------------

(b) セクション設定 ダイアログの場合

セクション情報ファイル (*.hsi)	セクション情報ファイル
すべてのファイル (*.*)	すべての形式

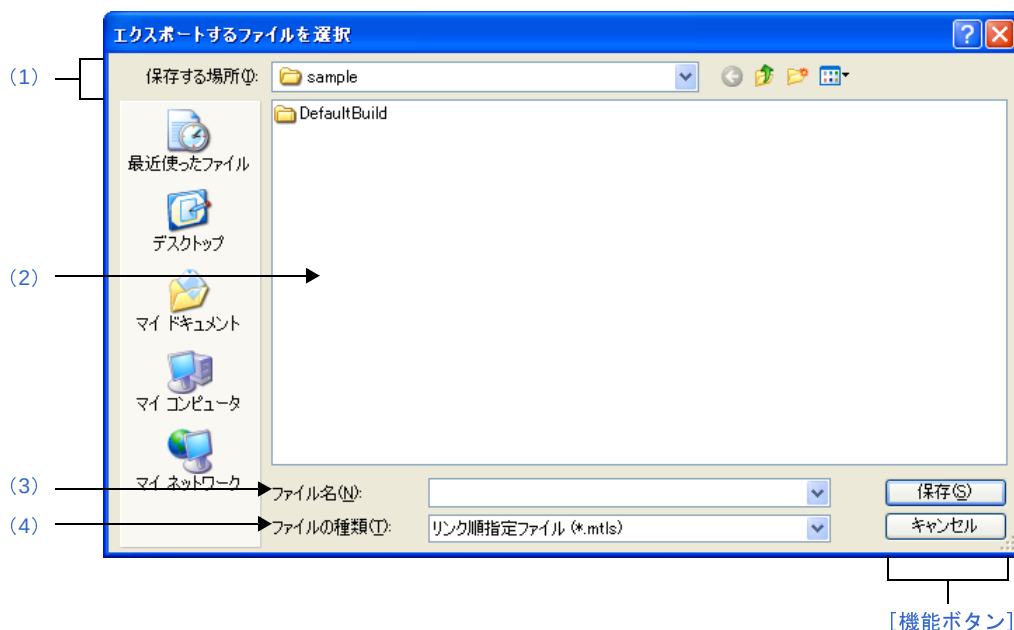
[機能ボタン]

ボタン	機能
開く	本ダイアログの呼び出し元に指定したファイルをインポートします。
キャンセル	本ダイアログをクローズします。

エクスポートするファイルを選択 ダイアログ

本ダイアログの呼び出し元の情報をファイルに出力します。

図 A—52 エクスポートするファイルを選択 ダイアログ



ここでは、次の項目について説明します。

- [\[オープン方法\]](#)
- [\[各エリアの説明\]](#)
- [\[機能ボタン\]](#)

[オープン方法]

- [リンク順設定 ダイアログ](#)において、[エクスポート] ボタンをクリック
- [セクション設定 ダイアログ](#)において、[エクスポート ...] ボタンをクリック

[各エリアの説明]

(1) [保存する場所] エリア

本ダイアログの呼び出し元の情報をファイルに出力するためのフォルダを選択します。

デフォルトでは、初めて選択する場合はプロジェクト・フォルダ、2回目以降は前回選択したフォルダを選択します。

(2) ファイルの一覧エリア

〔保存する場所〕エリア、および〔ファイルの種類〕で選択した条件に合致するファイルの一覧を表示します。

(3) 〔ファイル名〕エリア

出力する際のファイル名を指定します。

(4) 〔ファイルの種類〕エリア

次のファイルの種類（ファイル・タイプ）を表示します。

(a) リンク順設定 ダイアログの場合

リンク順指定ファイル (*.mtls)	リンク順指定ファイル
---------------------	------------

(b) セクション設定 ダイアログの場合

セクション情報ファイル (*.hsi)	セクション情報ファイル
すべてのファイル (*.*)	すべての形式

[機能ボタン]

ボタン	機能
保存	指定したファイル名でファイルを出力します。
キャンセル	本ダイアログをクローズします。

付録B コマンド・リファレンス

ここでは、ビルド・ツール（CC-RH）に含まれる各コマンドの仕様についての詳細を説明します。

B.1 概 要

CC-RH は、C 言語やアセンブリ言語で記述したソース・プログラムから、ターゲット・システムで実行可能なファイルを生成します。

以下のコマンドで構成されており、1つのコンパイル・ドライバ（ccrh）がコンパイルからリンクまでの全フェーズを制御します。

ccrh : コンパイル・ドライバの起動コマンド

asrh : アセンブラの起動コマンド

rlink : 最適化リンカの起動コマンド

各コマンドの処理について説明します。

(1) コンパイラ（ccrh）

C ソース・プログラムに対して、プリプロセス指令の処理、コメント処理、最適化を行い、アセンブリ・ソース・プログラムを生成します。

(2) アセンブラ（asrh）

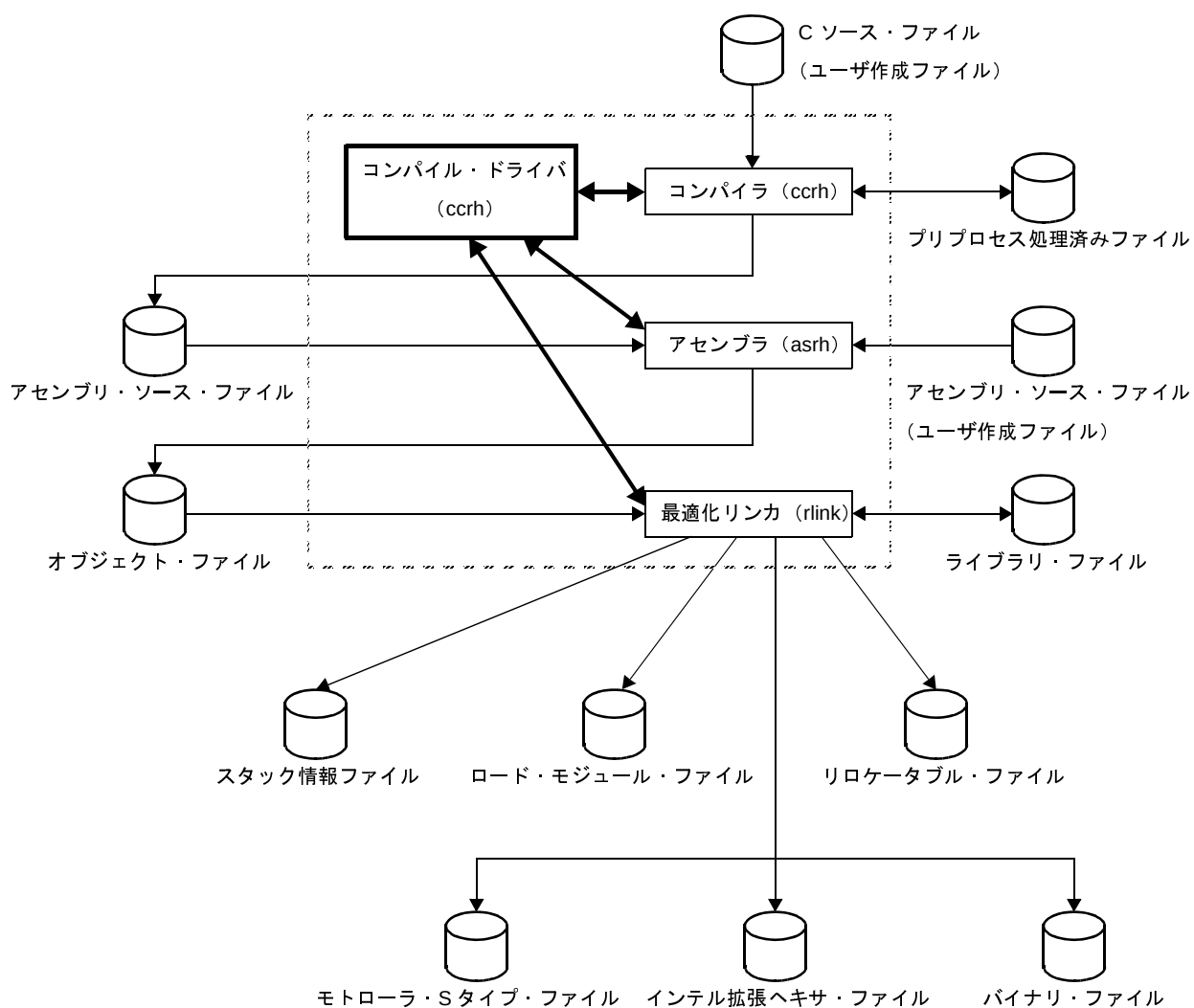
アセンブリ・ソース・プログラムを機械語命令に変換して、再配置可能なオブジェクト・ファイルを生成します。

(3) 最適化リンカ（rlink）

オブジェクト・ファイル、ライブラリ・ファイルをリンクし、ターゲット・システムで実行可能なオブジェクト・ファイル（ロード・モジュール・ファイル）を生成します。

また、ROM 化処理、リロケータブル・ファイル結合時の最適化、ライブラリ・ファイルの作成や編集、インテル拡張ヘキサ・ファイルやモトローラ・S タイプ・ファイルへの変換を行います。

図 B—1 ccrh における処理の流れ



B.2 入出力ファイル

コンパイル・ドライバである ccrh コマンドの入出力ファイルを以下に示します。

表 B—1 ccrh コマンドの入出力ファイル

ファイル種別	拡張子	入出力	説明
C ソース・ファイル	.c	入力	C 言語で記述したソース・ファイル ユーザ作成ファイルです。
プリプロセス処理済みファイル	.i ^{注1}	出力	入力ファイルに対してプリプロセス処理を実行した結果を出力したファイル ASCII イメージ・ファイルです。 -P オプション指定時に出力します。
アセンブリ・ソース・ファイル	.asm ^{注1}	出力	コンパイルにより C ソースから生成したアセンブリ言語ファイル -S オプション指定時に出力します。
	.asm .s	入力	アセンブリ言語で記述したソース・ファイル ユーザ作成ファイルです。
ヘッダ・ファイル	任意	入力	ソース・ファイルで参照するファイル C 言語、またはアセンブリ言語で記述したファイルです。 ユーザ作成ファイルです。 拡張子は任意ですが、以下を推奨します。 - #include 指令 : .h - \$include 制御命令 : .inc
オブジェクト・ファイル	.obj ^{注1}	入出力	機械語情報と機械語の配置アドレスに関する再配置情報、およびシンボル情報を含んだ ELF 形式ファイル
アセンブル・リスト・ファイル ^{注2}	.prn ^{注1}	出力	アセンブル結果の情報を保持したリスト・ファイル -Xasm_option=-Xprn_path オプション指定時に出力します。
ライブラリ・ファイル	.lib ^{注1}	入出力	複数のオブジェクト・ファイルが登録された ELF 形式ファイル -Xlk_option=-form=library オプション指定時に出力します。
ロード・モジュール・ファイル	.abs ^{注1}	入出力	リンク結果のオブジェクト・コードの ELF 形式ファイル ヘキサ・ファイルを出力する際の入力ファイルとなります。 -Xlk_option=-form=absolute オプション指定時に出力します。 ただし、-Xlk_option オプションで -form オプションが指定されていない場合は、上記オプションが指定されたものと解釈して処理します。

ファイル種別	拡張子	入出力	説明
リロケータブル・ファイル	.rel ^{注1}	出力	リロケータブルなオブジェクト・ファイル -Xlk_option=-form=relocate オプション指定時に出力します。
インテル拡張ヘキサ・ファイル ^{注2}	.hex ^{注1}	入出力	ロード・モジュール・ファイルをインテル拡張ヘキサ・フォーマットに変換したファイル -Xlk_option=-form=hexadecimal オプション指定時に出力します。
モトローラ・Sタイプ・ファイル ^{注2}	.mot ^{注1}	入出力	ロード・モジュール・ファイルをモトローラ・Sタイプに変換したファイル -Xlk_option=-form=stype オプション指定時に出力します。
バイナリ・ファイル	.bin ^{注1}	出力	ロード・モジュール・ファイルをバイナリ・フォーマットに変換したファイル -Xlk_option=-form=binary オプション指定時に出力します。
シンボル・アドレス・ファイル	.fsy	入出力	外部定義シンボルをアセンブラ制御命令で記述したアセンブリ・ソース・ファイル -Xlk_option=-fsymbol オプション指定時に出力します。
リンク・マップ・ファイル ^{注2}	.map ^{注1}	出力	リンク結果の情報を持つリスト・ファイル -Xlk_option=-list オプション指定時に出力します。
ライブラリ・リスト・ファイル ^{注2}	.lbp ^{注1}	出力	ライブラリ作成結果の情報を持つリスト・ファイル -Xlk_option=-list オプション指定時に出力します。
スタック情報ファイル	.sni	出力	スタック使用量の情報を持つファイル -Xlk_option=-stack オプション指定時に出力します。
外部シンボル割り付け情報ファイル	.bls ^{注1}	出力	コンパイラが外部変数アクセス最適化で使用する外部変数割り付け情報ファイル -Omap オプション指定時に出力します。
静的解析情報ファイル	任意	入出力	本製品が利用する情報を持つファイル 拡張子は任意ですが、“ .cref ”を推奨します。 -Xcref オプション指定時に出力します。
エラー・メッセージ・ファイル	任意	出力	エラー・メッセージを内容とするファイル 拡張子は任意ですが、“ .err ”を推奨します。 -Xerror_file オプション指定時に出力します。
サブコマンド・ファイル	任意	入力	実行プログラムのパラメータを内容とするファイル ユーザ作成ファイルです。

注1. オプション指定により拡張子の変更が可能です。

2. 各ファイルについての詳細は、「[第3章 ビルドの出力リスト](#)」を参照してください。

B.3 環境変数

ここでは、環境変数について説明します。

最適化リンカの環境変数とコマンド・ラインでの指定例を以下に示します。

- HLNK_LIBRARY1, HLNK_LIBRARY2, HLNK_LIBRARY3

最適化リンカが使用するデフォルト・ライブラリ名を指定します。

ライブラリは、-library オプションで指定したものを優先してリンクします。

その後、未解決のシンボルがある場合、HLNK_LIBRARY1, HLNK_LIBRARY2, HLNK_LIBRARY3 の順に、デフォルト・ライブラリを検索します。

例

```
>set HLNK_LIBRARY1=usr1.lib
>set HLNK_LIBRARY2=usr2.lib
>set HLNK_LIBRARY3=usr3.lib
```

- HLNK_TMP

最適化リンカがテンポラリ・ファイルを作成するフォルダを指定します。

この環境変数の指定がない場合は、カレント・フォルダに作成します。

例

```
>set HLNK_TMP=D:¥workspace¥tmp
```

- HLNK_DIR

最適化リンカの入力ファイル格納フォルダを指定します。

-input オプション、-library オプションで指定したファイルの検索順序は、カレント・フォルダ、HLNK_DIR 指定フォルダとなります。

ただし、ワイルドカードで指定したファイルは、カレント・フォルダ内だけを検索します。

例

```
>set HLNK_DIR=D:¥workspace¥obj1;D:¥workspace¥obj2
```


B.4 操作方法

ここでは、各コマンドの操作方法について説明します。

- コマンド・ラインでの操作方法
- サブコマンド・ファイルの使用方法
- CubeSuite+ でのオプション設定

B.4.1 コマンド・ラインでの操作方法

コンパイル・ドライバである ccrh コマンドの起動により、コンパイル、アセンブル、リンクなどを行うことができます。

また、アセンブラ (asrh)、最適化リンカ (rlink) は単体起動することも可能です。

(1) 指定形式

各コマンドは、コマンド・ラインで以下のように入力します。

```
>ccrh[ Δ option]... Δ file[ Δ file| Δ option]...
```

```
>asrh[ Δ option]... Δ file[ Δ file| Δ option]...
```

```
>rlink[{ Δ file| Δ option}...]
```

option : オプション名
file : ファイル名
[] : [] 内は省略可能
... : 直前の [] 内のパターンの繰り返しが可能
{ } : | で区切られた項目を選択
Δ : 1 個以上の空白

コマンドを入力する際の注意事項を以下に示します。

- オプションの指定形式は、使用するコマンドによって異なります。

各コマンドのオプションに対する注意事項については、「[B.5.1 コンパイル・オプション](#)」, 「[B.5.2 アセンブル・オプション](#)」, 「[B.5.3 リンク・オプション](#)」を参照してください。

- ファイル名は、Windows で認められるものであれば指定可能です。

ただし、“@” はサブコマンド・ファイル指定と判断するため、“@” をファイル名の先頭文字として使用することはできません。

また、“-” はオプション指定と判断するため、“-” もファイル名の先頭文字として使用することはできません。

さらに、“(”, “)” はリンク・オプションの一部と判断するため、“(”, “)” をファイル名に使用することはできません。

- 指定可能なファイル名の長さは、Windows に依存します (259 文字まで)。
- ファイル名のアルファベットは、大文字／小文字を区別しません。

- 入力として複数のファイルが指定可能です。

異なる種類のファイル（C ソース・ファイルとアセンブリ・ソース・ファイル、またはオブジェクト・ファイルなど）を混在して指定することも可能です。

ただし、拡張子を除いたソース・ファイル名が同じファイルを複数指定することはできません（異なるフォルダにある場合も含みます）。

複数のファイルを指定した場合は、1つのファイルでエラーとなっても、残りのファイルの処理が継続可能であれば、処理を継続します。

なお、生成したオブジェクト・ファイルはリンク後も削除しません。

(2) 操作例

コマンド・ラインでの操作例を以下に示します。

備考 各オプションについての詳細は、「[B.5 オプション](#)」を参照してください。

(a) コンパイル、アセンブル、リンクを1コマンドで行う場合

C ソース・ファイル file1.c を ccrh でコンパイルして、アセンブリ・ソース・ファイル file1.asm を生成します。

次に、アセンブリ・ソース・ファイル file1.asm と file2.asm を asrh でアセンブルして、オブジェクト・ファイル file1.obj, file2.obj を生成します。

また、アセンブル・リスト・ファイル sample.prn も出力します。

最後に、オブジェクト・ファイル file1.obj, file2.obj, file3.obj を rlink でリンクして、リンク・マップ・ファイル sample.map, およびロード・モジュール・ファイル sample.abs を生成します。

```
>ccrh file1.c file2.asm file3.obj -Xasm_option=-Xprn_path=sample.prn -Xlk_option=-list -osample.abs -Xcommon=v850e3v5
```

備考 ccrh に対して、asrh のみに指定可能なオプションを指定する場合は -Xasm_option オプション、rlink のみに指定可能なオプションを指定する場合は -Xlk_option オプションを使用します。

(b) コンパイル、アセンブルを1コマンドで行い、リンクは単独で行う場合

C ソース・ファイル file1.c を ccrh でコンパイルして、アセンブリ・ソース・ファイル file1.asm を生成します。

次に、アセンブリ・ソース・ファイル file1.asm と file2.asm を asrh でアセンブルして、オブジェクト・ファイル file1.obj, file2.obj を生成します。

また、アセンブル・リスト・ファイル sample.prn も出力します。

```
>ccrh -c file1.c file2.asm -Xasm_option=-Xprn_path=sample.prn -Xcommon=v850e3v5
```

備考 ccrh に対して、asrh のみに指定可能なオプションを指定する場合は -Xasm_option オプションを使用します。

オブジェクト・ファイル file1.obj, file2.obj, file3.obj を rlink でリンクして、リンク・マップ・ファイル sample.map, およびロード・モジュール・ファイル sample.abs を生成します。

```
>rlink file1.obj file2.obj file3.obj -output=sample.abs -list
```

(c) コンパイル, アセンブル, リンクともに単独で行う場合

C ソース・ファイル file1.c を ccrh でコンパイルして、アセンブリ・ソース・ファイル file1.asm を生成します。

```
>ccrh -S file1.c -Xcommon=v850e3v5
```

アセンブリ・ソース・ファイル file1.asm と file2.asm を asrh でアセンブルして、オブジェクト・ファイル file1.obj, file2.obj を生成します。

また、アセンブル・リスト・ファイル sample.prn も出力します。

```
>asrh -c file1.asm file2.asm -Xprn_path=sample.prn -Xcommon=v850e3v5
```

オブジェクト・ファイル file1.obj, file2.obj, file3.obj を rlink でリンクして、リンク・マップ・ファイル sample.map, およびロード・モジュール・ファイル sample.abs を生成します。

```
>rlink file1.obj file2.obj file3.obj -output=sample.abs -list
```

B. 4.2 サブコマンド・ファイルの使用方法

サブコマンド・ファイルとは、コマンドに対して指定するオプションやファイル名を記述したファイルです。

コマンドは、サブコマンド・ファイルの内容をコマンド・ラインの引数のように扱います。

サブコマンド・ファイルは、コマンド・ラインで引数を指定しきれない場合や、コマンドを実行するたびに同じオプションを繰り返し指定するような場合に使用します。

(1) コンパイラ, アセンブラでサブコマンド・ファイルを使用する場合

(a) サブコマンド・ファイルの記述に関する注意事項

- 指定する引数は、複数行に分けて記述することができます。
ただし、オプションやファイル名の途中で改行することはできません。
- サブコマンド・ファイルをネストすることはできません。
- サブコマンド・ファイルの文字コードは、-Xcharacter_set オプションで指定することはできません。
サブコマンド・ファイル中に ASCII 文字以外を使用する場合は、BOM 付き UTF-8、または SJIS のファイルにしてください。
- 以下の文字は、特殊文字として扱います。
これらの特殊文字自体は、ccrh コマンドへのコマンド・ラインの中に含まずに削除します。

" (ダブルクォーテーション)	次のダブルクォーテーションまでの文字列を連続した文字列として扱います。
# (シャープ)	行頭に指定した場合は、行末までをコメントとして扱います。

^ (ハット)	直後の文字を特殊文字として扱いません。
---------	---------------------

(b) サブコマンド・ファイルの指定例

サブコマンド・ファイル sub.txt をエディタで作成します。

```
-Xcommon=v850e3v5
-c
-Dtest
-Idir
-Osize
```

コマンド・ラインにおいて、サブコマンド・ファイル指定オプション @ により sub.txt を指定します。

```
>ccrh @sub.txt -ofile.obj file.c
```

コマンド・ラインは以下のように展開されます。

```
>ccrh -Xcommon=v850e3v5 -c -Dtest -Idir -Osize -ofile.obj file.c
```

(2) 最適化リンクでサブコマンド・ファイルを使用する場合**(a) サブコマンド・ファイルの記述に関する注意事項**

- オプションは、先頭のハイフン (-) を省略して指定することができます。
 - オプションとパラメータの区切りは、“=” の代わりに空白を使用することも可能です。
 - 1 行につき 1 つのオプションを指定します。
 - 1 行に記述できない場合は、“&” を使用して複数行に記述することができます。
 - サブコマンド・ファイル内に -subcommand オプションを指定することはできません。
 - 以下の文字は、特殊文字として扱います。
- これらの特殊文字自体は、rlink コマンドへのコマンド・ラインの中に含まずに削除します。

& (アンド)	次の行を継続行として扱います。
;(セミコロン)	行末までをコメントとして扱います。

(b) サブコマンド・ファイルの指定例

サブコマンド・ファイル sub.txt をエディタで作成します。

```
input file2.obj file3.obj      ; ここはコメントです。
library lib1.lib, &             ; “&” は継続行を示します。
lib2.lib
```

コマンド・ラインにおいて、サブコマンド・ファイル指定オプション -subcommand により sub.txt を指定します。

```
>rlink file1.obj -subcommand=sub.txt file4.obj
```

コマンド・ラインは以下のように展開されます。

```
>rlink file1.obj file2.obj file3.obj -library=lib1.lib,lib2.lib file4.obj
```

B. 4.3 CubeSuite+ でのオプション設定

CubeSuite+ でオプションを設定する方法について説明します。

CubeSuite+ のプロジェクト・ツリーパネルにおいて、ビルド・ツール・ノードを選択したのち、[表示] メニュー→[プロパティ] を選択すると、プロパティパネルがオープンします。

- アプリケーション用のプロジェクトの場合

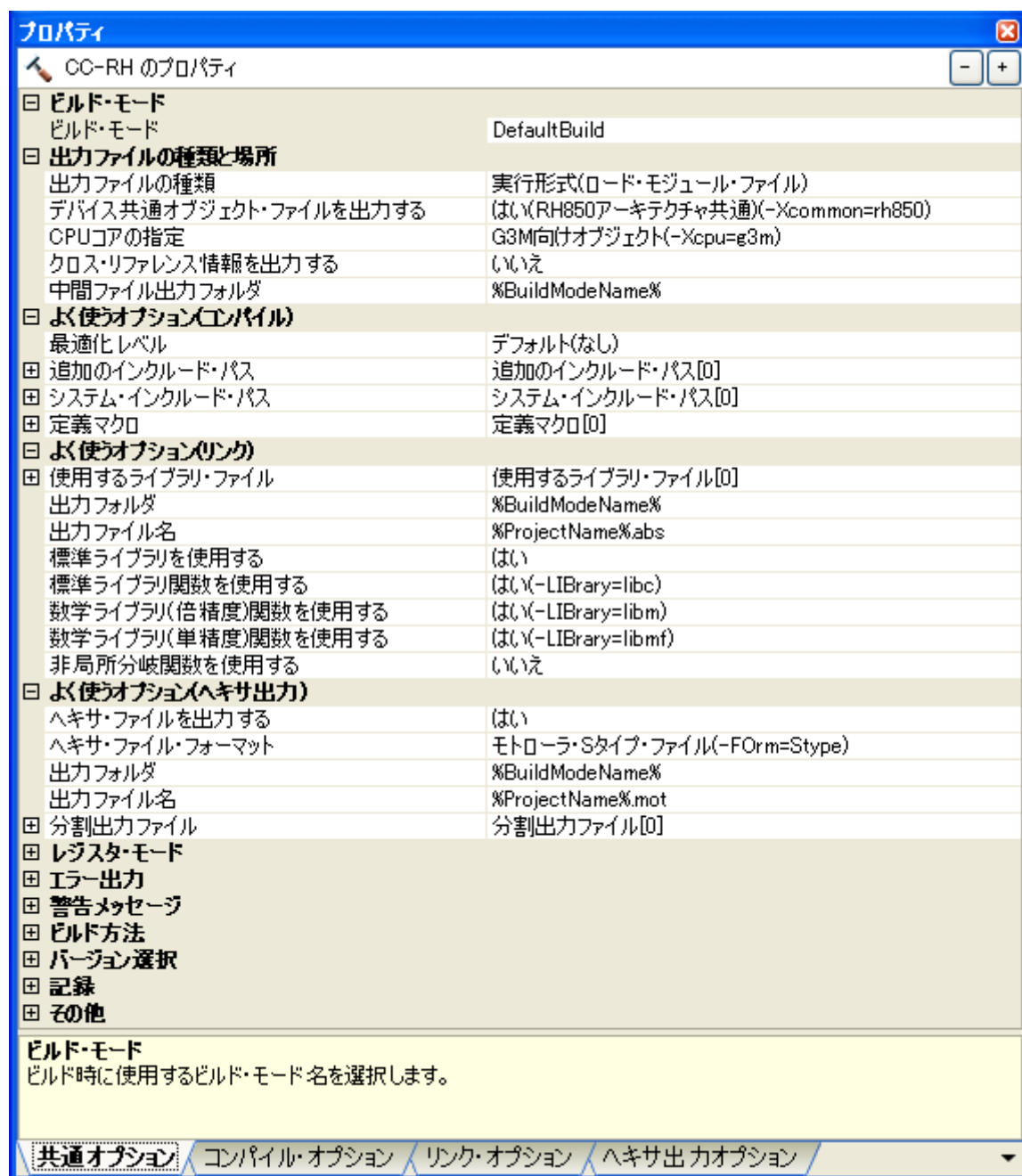
[共通オプション] タブ／[コンパイル・オプション] タブ／[アセンブル・オプション] タブ／[リンク・オプション] タブ／[ヘキサ出力オプション] タブを選択します。

- ライブラリ用のプロジェクトの場合

[共通オプション] タブ／[コンパイル・オプション] タブ／[アセンブル・オプション] タブ／[ライブラリ生成オプション] タブを選択します。

タブ上で各プロパティを設定することにより、対応する ccrh オプションを設定することができます。

図 B—2 プロパティ パネル



B.5 オプション

ここでは、ccrh のオプションについて、各フェーズごとに説明します。

コンパイル・フェーズ → 「[B.5.1 コンパイル・オプション](#)」参照

アセンブル・フェーズ → 「[B.5.2 アセンブル・オプション](#)」参照

リンク・フェーズ → 「[B.5.3 リンク・オプション](#)」参照

B.5.1 コンパイル・オプション

ここでは、コンパイル・フェーズのオプションについて説明します。

オプションに関する注意事項を以下に示します。

- オプションの大文字／小文字は区別します。
- パラメータとして数値を指定する場合は、10 進数、または“0x”（“0X”）で始まる 16 進数での指定が可能です。
- 16 進数のアルファベットは、大文字／小文字を区別しません。
- パラメータとしてファイル名を指定する場合は、パス付き（絶対パス、または相対パス）での指定が可能です。
- パスなし、および相対パスで指定する場合は、カレント・フォルダを基準とします。
- パラメータ中に空白を含める場合（パス名など）は、そのパラメータ全体をダブルクォーテーション（"）で囲ってください。

オプションの分類と説明を以下に示します。

表 B—2 コンパイル・オプション

分類	オプション	説明
バージョン／ヘルプ表示指定	-V	ccrh のバージョン情報を表示します。
	-h	ccrh のオプションの説明を表示します。
出力ファイル指定	-O	出力ファイル名を指定します。
	-Xobj_path	コンパイル途中に生成されるオブジェクト・ファイルを保存するフォルダを指定します。
	-Xasm_path	コンパイル途中に生成されるアセンブリ・ソース・ファイルを保存するフォルダを指定します。
	-Xprep_path	プリプロセス処理済みファイルを保存するフォルダを指定します。
ソース・デバッグ制御	-g	ソース・デバッグ用の情報を出力します。
デバイス指定	-Xcommon	デバイス共通のオブジェクト・ファイルを生成することを指定します。
	-Xcpu	指定したコア向けのオブジェクトを生成することを指定します。
処理中断指定	-P	入力ファイルに対してプリプロセス処理のみ実行します。
	-S	アセンブル以降の処理を実行しません。
	-C	リンク以降の処理を実行しません。

分類	オプション	説明
プリプロセッサ制御	-D	プリプロセッサ・マクロ、およびアセンブラ・シンボルを定義します。
	-U	-D オプションによるプリプロセッサ・マクロ、およびアセンブラ・シンボルの定義を解除します。
	-I	インクルード・ファイルを検索するフォルダを指定します。
	-Xpreinclude	コンパイル単位の先頭にインクルードするファイルを指定します。
	-Xpreprocess	プリプロセス結果の出力を制御します。
C 言語制御	-Xansi	C ソース・プログラムを ANSI 規格に厳密にあわせて処理します。
	-Xenum_type	列挙型に対して、どの整数型として扱うかを指定します。
	-Xvolatile	外部変数の volatile 化を指定します。
	-Xcheck	C ソース・ファイルの互換性をチェックします。
	-Xmisra2004	MISRA-C:2004 ルールによるソース・チェックを行います。
	-Xignore_files_misra	MISRA-C:2004 ルールによるソース・チェックの対象外のファイルを指定します。
	-Xcheck_language_extention	言語拡張により部分抑止される MISRA-C:2004 ルールのソース・チェックを有効にします。
日本語／中国語文字列制御	-Xcharacter_set	日本語／中国語の文字コードを指定します。
最適化指定	-O	最適化のレベル、または各最適化項目の詳細を指定します。
	-Xintermodule	大域最適化を行います。
	-Xinline_strcpy	標準ライブラリ関数 strcpy, strcmp, memcpy, memset の呼び出しをインライン展開します。
	-Xmerge_string	文字列定数をマージします。
	-Xalias	ポインタ指示先の型を考慮した最適化の指定を行います。
	-Xmerge_files	複数の C ソース・ファイルをマージしてコンパイルを行います。
	-Xwhole_program	コンパイル対象ファイルがプログラム全体であることを仮定して最適化を行います。

分類	オプション	説明
生成コード制御	-Xpack	構造体パッキングを行います。
	-Xbit_order	ビット・フィールドのメンバの並び順を指定します。
	-Xpass_source	出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力します。
	-Xswitch	switch 文のコード出力方式を指定します。
	-Xreg_mode	レジスタ・モードを指定します。
	-Xreserve_r2	r2 レジスタを予約します。
	-Xep	ep レジスタの扱い方を指定します。
	-Xfloat	浮動小数点演算命令の生成を制御します。
	-Xcall_jump	関数呼び出しの分岐命令の生成を制御します。
	-Xfar_jump	far jump の出力を制御します。
	-Xdiv	除算に対して、div、および divu 命令を生成します。
	-Xcheck_div_ov	除算時に OV フラグのチェックを行います。
	-Xuse_fmaf	積和演算命令を生成します。
	-Xunordered_cmpf	浮動小数点比較において、無効演算例外の検出を行います。
	-Xmulti_level	マルチコア用プログラムの生成を指定します。
	-Xpatch	パッチを適用します。
情報ファイル出力制御	-Xcref	静的解析情報ファイルを出力します。
エラー出力制御	-Xerror_file	エラー・メッセージをファイルに出力します。
警告メッセージ出力制御	-Xno_warning	指定した警告メッセージの出力を抑止します。
フェーズ個別オプション指定	-Xasm_option	アセンブル・オプションを指定します。
	-Xlk_option	リンク・オプションを指定します。
サブコマンド・ファイル指定	@	サブコマンド・ファイルを指定します。

バージョン／ヘルプ表示指定

バージョン／ヘルプ表示指定オプションには、次のものがあります。

-V

-h

-V

ccrh のバージョン情報を表示します。

[指定形式]

```
-V
```

- 省略時解釈

ccrh のバージョン情報を表示せずに、コンパイルを行います。

[詳細説明]

- ccrh のバージョン情報を標準エラー出力に出力します。

コンパイルは行いません。

[使用例]

- ccrh のバージョン情報を標準エラー出力に出力します。

```
>ccrh -V -Xcommon=v850e3v5
```

-h

ccrh のオプションの説明を表示します。

[指定形式]

```
-h
```

- 省略時解釈

ccrh のオプションの説明を表示しません。

[詳細説明]

- ccrh のオプションの説明を標準エラー出力に出力します。
コンパイルは行いません。

[使用例]

- ccrh のオプションの説明を標準エラー出力に出力します。

```
>ccrh -h -Xcommon=v850e3v5
```

出力ファイル指定

出力ファイル指定オプションには、次のものがあります。

- o
- Xobj_path
- Xasm_path
- Xprep_path

-o

出力ファイル名を指定します。

[指定形式]

`-ofile`

- 省略時解釈

出力ファイル名は、指定オプションにより異なります。

なお、ファイルの出力先はカレント・フォルダとなります。

--P オプションを指定している場合

出力ファイル名は、入力ファイル名の拡張子を“.i”に置き換えたものとなります。

--S オプションを指定している場合

出力アセンブリ・ソース・ファイル名は、ソース・ファイル名の拡張子を“.asm”に置き換えたものとなります。

--c オプションを指定している場合

出力オブジェクト・ファイル名は、ソース・ファイル名の拡張子を“.obj”に置き換えたものとなります。

- 上記以外の場合

出力ロード・モジュール・ファイル名は、最初に入力したファイル名の拡張子を“.abs”に置き換えたものとなります。

[詳細説明]

- 出力ファイル名を *file* に指定します。

- *file* がすでに存在する場合は、そのファイルを上書きします。

- -P/-S/-c オプションと同時に指定することにより処理を中断した場合にも、本オプションは有効となります。

--P オプションと同時に指定した場合

file には、入力ファイルに対してプリプロセス処理を行った結果のファイル名を指定したものとみなします。

--S オプションと同時に指定した場合

file には、アセンブリ・ソース・ファイル名を指定したものとみなします。

--c オプションと同時に指定した場合

file には、オブジェクト・ファイル名を指定したものとみなします。

- 上記以外の場合
 - file* には、ロード・モジュール・ファイル名を指定したものとみなします。
- 出力ファイルが複数の場合は、エラーとなります。
- *file* を省略した場合は、エラーとなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[個別コンパイル・オプション\] タブ](#)の [出力ファイル] カテゴリの [\[オブジェクト・ファイル名\]](#)

[使用例]

- ロード・モジュール・ファイルをファイル名 sample.abs で出力します。

```
>ccrh -osample.abs -Xcommon=v850e3v5 main.c
```

-Xobj_path

コンパイル途中に生成されるオブジェクト・ファイルを保存するフォルダを指定します。

[指定形式]

```
-Xobj_path [=path]
```

- 省略時解釈

カレント・フォルダに、ソース・ファイル名の拡張子を“.obj”で置き換えたファイル名でオブジェクト・ファイルを保存します。

[詳細説明]

- コンパイル途中に生成されるオブジェクト・ファイルを保存するフォルダを *path* に指定します。
- *path* に存在するフォルダ名を指定した場合は、フォルダ *path* に、ソース・ファイル名の拡張子を“.obj”で置き換えたファイル名でオブジェクト・ファイルを保存します。
存在しないフォルダ名を指定した場合は、エラーとなります。
- *path* には存在するファイル名を指定することも可能です。
出力するオブジェクト・ファイルが1つの場合は、*path* というファイル名で保存します。
出力するオブジェクト・ファイルが複数の場合は、エラーとなります。
存在しないファイル名を指定した場合は、エラーとなります。
- *=path* を省略した場合は、カレント・フォルダに、ソース・ファイル名の拡張子を“.obj”で置き換えたファイル名でオブジェクト・ファイルを保存します。
- ソース・ファイルとして同じ名前のファイル（異なるフォルダにある場合を含む）を複数指定した場合は、警告を出力して、最後に指定したソース・ファイルに対するオブジェクト・ファイルのみを保存します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\] タブ](#)の「出力ファイルの種類と場所」カテゴリの「[中間ファイル出力フォルダ](#)」

[使用例]

- コンパイル途中に生成されるオブジェクト・ファイルをフォルダ D:\sample に保存します。

```
>ccrh -Xobj_path=D:\sample -Xcommon=v850e3v5 main.c
```

-Xasm_path

コンパイル途中に生成されるアセンブリ・ソース・ファイルを保存するフォルダを指定します。

[指定形式]

```
-Xasm_path [=path]
```

- 省略時解釈

アセンブリ・ソース・ファイルを出しません（-S オプション指定時を除く）。

[詳細説明]

- コンパイル途中に生成されるアセンブリ・ソース・ファイルを保存するフォルダを *path* に指定します。
- *path* に存在するフォルダ名を指定した場合は、フォルダ *path* に、C ソース・ファイル名の拡張子を “.asm” で置き換えたファイル名でアセンブリ・ソース・ファイルを保存します。
存在しないフォルダ名を指定した場合は、エラーとなります。
- *path* には存在するファイル名を指定することも可能です。
出力するアセンブリ・ソース・ファイルが1つの場合は、*path* というファイル名で保存します。
出力するアセンブリ・ソース・ファイルが複数の場合は、エラーとなります。
存在しないファイル名を指定した場合は、エラーとなります。
- *=path* を省略した場合は、カレント・フォルダに、C ソース・ファイル名の拡張子を “.asm” で置き換えたファイル名でアセンブリ・ソース・ファイルを保存します。
- ソース・ファイルとして同じ名前のファイル（異なるフォルダにある場合を含む）を複数指定した場合は、警告を出力して、最後に指定したソース・ファイルに対するアセンブリ・ソース・ファイルのみを保存します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [\[出力ファイル\]](#) カテゴリの [\[アセンブリ・ソース・ファイルを出力する\]](#), [\[アセンブリ・ソース・ファイル出力フォルダ\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [\[出力ファイル\]](#) カテゴリの [\[アセンブリ・ソース・ファイルを出力する\]](#), [\[アセンブリ・ソース・ファイル出力フォルダ\]](#)

[使用例]

- コンパイル途中に生成されるアセンブリ・ソース・ファイルをフォルダ D:\sample に保存します。

```
>ccrh -Xasm_path=D:\sample -Xcommon=v850e3v5 main.c
```

-Xprep_path

プリプロセス処理済みファイルを保存するフォルダを指定します。

[指定形式]

```
-Xprep_path[=path]
```

- 省略時解釈

プリプロセス処理済みファイルを出しません（-P オプション指定時を除く）。

[詳細説明]

- プリプロセス処理済みファイルを保存するフォルダを *path* に指定します。
- *path* に存在するフォルダ名を指定した場合は、フォルダ *path* に、C ソース・ファイル名の拡張子を “.i” で置き換えたファイル名でプリプロセス処理済みファイルを保存します。
存在しないフォルダ名を指定した場合は、エラーとなります。
- *path* には存在するファイル名を指定することも可能です。
出力するプリプロセス処理済みファイルが1つの場合は、*path* というファイル名で保存します。
出力するプリプロセス処理済みファイルが複数の場合は、エラーとなります。
存在しないファイル名を指定した場合は、エラーとなります。
- *=path* を省略した場合は、カレント・フォルダに、C ソース・ファイル名の拡張子を “.i” で置き換えたファイル名でプリプロセス処理済みファイルを保存します。
- ソース・ファイルとして同じ名前のファイル（異なるフォルダにある場合を含む）を複数指定した場合は、警告を出力して、最後に指定したソース・ファイルに対するプリプロセス処理済みファイルのみを保存します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[出力ファイル] カテゴリの[\[プリプロセス処理したソース・ファイル出力フォルダ\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[出力ファイル] カテゴリの[\[プリプロセス処理したソース・ファイル出力フォルダ\]](#)

[使用例]

- プリプロセス処理済みファイルをフォルダ D:\sample に保存します。

```
>ccrh -Xprep_path=D:\sample -Xcommon=v850e3v5 main.c
```


ソース・デバッグ制御

ソース・デバッグ制御オプションには、次のものがあります。

- [-g](#)

-g

ソース・デバッグ用の情報を出力します。

[指定形式]

-g

- 省略時解釈

ソース・デバッグ用の情報を出力しません。

[詳細説明]

- ソース・デバッグ用の情報を出力ファイル中に出力します。
- 本オプションを指定することにより、ソース・デバッグが可能となります。
- 最適化オプションと同時に指定した場合は、デバッグのしやすさに影響があります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [\[デバッグ情報\]](#) カテゴリの [\[デバッグ情報を生成する\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [\[デバッグ情報\]](#) カテゴリの [\[デバッグ情報を生成する\]](#)

[使用例]

- ソース・デバッグ用の情報を出力ファイル中に出力します。

```
>ccrh -g -Xcommon=v850e3v5 main.c
```

デバイス指定

デバイス指定オプションには、次のものがあります。

- Xcommon
- Xcpu

-Xcommon

デバイス共通のオブジェクト・ファイルを生成することを指定します。

[指定形式]

```
-Xcommon=series
```

- 省略時解釈

エラーとなります。

ただし、-V/-h/-P オプションを指定している場合は、エラーなりません。

[詳細説明]

- デバイス共通のオブジェクト・ファイルを生成することを指定します。
- 本オプションを指定した場合、ターゲットの命令セット・アーキテクチャの命令のみを使用し、また、命令セット・アーキテクチャに対応した共通のマジック・ナンバ *series* をオブジェクト・ファイルに埋め込みます。
- *series* に指定可能なものを以下に示します。
これ以外のものを指定した場合は、エラーとなります。

v850e3v5	命令セット・アーキテクチャが V850E3V5 である品種をターゲット・デバイスとして指定したもののリンクが可能です。
rh850	“v850e3v5” を指定した場合と同じです。

- *series* を省略した場合は、エラーとなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\] タブ](#)の[\[出力ファイルの種類と場所\]](#) カテゴリの[\[デバイス共通オブジェクト・ファイルを出力する\]](#)

[使用例]

- 生成するオブジェクト・ファイルに、命令セット・アーキテクチャが V850E3V5 である品種に共通のマジック・ナンバを埋め込みます。

```
>ccrh -Xcommon=v850e3v5 -c main.c
```

-Xcpu

指定したコア向けのオブジェクトを生成することを指定します。

[指定形式]

```
-Xcpu=core
```

- 省略時解釈

-Xcommon オプションを指定した場合、G3M 向けオブジェクトを生成します。

[詳細説明]

- コア *core* 向けのオブジェクトを生成することを指定します。

- *core* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

g3m	G3M 向けオブジェクトを生成します。
g3k	G3K 向けオブジェクトを生成します。

- *core* を省略した場合は、エラーとなります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[共通オプション\] タブ](#)の [\[出力ファイルの種類と場所\]](#) カテゴリの [\[CPU コアの指定\]](#)

[使用例]

- G3M 向けオブジェクトを生成することを指定します。

```
>ccrh -Xcpu=g3m -c -Xcommon=v850e3v5 main.c
```

処理中断指定

処理中断指定オプションには、次のものがあります。

- P
- S
- C

-P

入力ファイルに対してプリプロセス処理のみ実行します。

[指定形式]

-P

- 省略時解釈

プリプロセス処理以降も処理を継続します。
ファイルは出力しません。

[詳細説明]

- 入力ファイルに対してプリプロセス処理のみ実行して、結果をファイルに出力します。
- 出力ファイル名は、入力ファイル名の拡張子を“.i”に置き換えたものになります。
- -o オプションと同時に指定することにより、出力ファイル名を指定することができます。
- -Xpreprocess オプションを指定することにより、出力ファイルの内容を制御することができます。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[出力ファイル] カテゴリの [\[プリプロセス処理したソースを出力する\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[出力ファイル] カテゴリの [\[プリプロセス処理したソースを出力する\]](#)

[使用例]

- 入力ファイルに対してプリプロセス処理のみ実行して、結果をファイル main.i に出力します。

```
>ccrh -P -Xcommon=v850e3v5 main.c
```

-S

アセンブル以降の処理を実行しません。

[指定形式]

```
-S
```

- 省略時解釈

アセンブル以降も処理を継続します。

[詳細説明]

- アセンブル以降の処理を実行しません。
- ソース・ファイル名の拡張子を“.asm”で置き換えた名前でアセンブリ・ソース・ファイルを出力します。
- -o オプションと同時に指定することにより、出力ファイル名を指定することができます。

[使用例]

- アセンブル以降の処理を実行せず、アセンブリ・ソース・ファイル main.asm を出力します。

```
>ccrh -S -Xcommon=v850e3v5 main.c
```

-C

リンク以降の処理を実行しません。

[指定形式]

```
-c
```

- 省略時解釈

リンク以降も処理を継続します。

[詳細説明]

- リンク以降の処理を実行しません。
- ソース・ファイル名の拡張子を“.obj”で置き換えた名前でオブジェクト・ファイルを出力します。
- -o オプションと同時に指定することにより、出力ファイル名を指定することができます。

[使用例]

- リンク以降の処理を実行せず、オブジェクト・ファイル main.obj を出力します。

```
>ccrh -c -Xcommon=v850e3v5 main.c
```

プリプロセッサ制御

プリプロセッサ制御オプションには、次のものがあります。

- D
- U
- I
- Xpreinclude
- Xpreprocess

-D

プリプロセッサ・マクロ、およびアセンブラ・シンボルを定義します。

[指定形式]

```
-Dname [=def] [name [=def]] ...
```

- 省略時解釈
なし

[詳細説明]

- プリプロセッサ・マクロ、およびアセンブラのユーザ定義シンボルとして *name* を定義します。
- ソース・プログラムの前に、`#define name def`、および `.SET name def`（アセンブリ・ソース・プログラムの場合のみ）を記述するのと同様です。
- *name* が、アセンブラ・シンボルには使用可能であるがプリプロセッサ・マクロには使用できない文字（@, ., ~）を含む場合、警告を出力してアセンブラ・シンボルとしてのみ定義します。
- 本オプションで C 言語の既定マクロ `__LINE__`、`__FILE__`、`__DATE__`、`__TIME__`、`__CCRH__` を再定義することはできません（`-D__CCRH__[=1]`、および `-D__CCRH[=1]` を除く）。
入力ファイルが C ソース・ファイルの場合にこれらを再定義すると、エラーとなります。
- *name* を省略した場合は、エラーとなります。
- `=def` を省略した場合、*def* は 1 とみなします。
- 本オプションは、複数指定が可能です。
- 同じプリプロセッサ・マクロ、およびアセンブラ・シンボルに対して、本オプションと `-U` オプションを同時に指定した場合は、あとから指定したものが有効となります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\] タブ](#) の [\[よく使うオプション \(コンパイル\)\]](#) カテゴリの [\[定義マクロ\]](#)
 - [\[コンパイル・オプション\] タブ](#) の [\[プリプロセス\]](#) カテゴリの [\[定義マクロ\]](#)
 - [\[個別コンパイル・オプション\] タブ](#) の [\[プリプロセス\]](#) カテゴリの [\[定義マクロ\]](#)

【使用例】

- プリプロセッサ・マクロとして sample=256 を定義します。

```
>ccrh -Dsample=256 -Xcommon=v850e3v5 main.c
```


-U

-D オプションによるプリプロセッサ・マクロ、およびアセンブラ・シンボルの定義を解除します。

[指定形式]

```
-Uname[, name]...
```

- 省略時解釈
なし

[詳細説明]

- D オプションによるプリプロセッサ・マクロ、およびアセンブラのユーザ定義シンボル *name* の定義を解除します。
- ソース・プログラムの前に、`#undef name` を記述するのと同様です。
- *name* を省略した場合は、エラーとなります。
- 本オプションでは、`#define name def`、および `.SET name def`（アセンブリ・ソース・プログラムの場合のみ）の記述による定義は解除できません。
- 本オプションにより、C 言語の既定定義マクロを解除することもできますが、`__LINE__`、`__FILE__`、`__DATE__`、`__TIME__`、`__CCRH__`、`__CCRH` を解除することはできません。
入力ファイルが C ソース・ファイルの場合に *name* にこれらを指定すると、エラーとなります。
- 本オプションは、複数指定が可能です。
- 同じプリプロセッサ・マクロ、およびアセンブラ・シンボルに対して、本オプションと -D オプションを同時に指定した場合は、あとから指定したものが有効となります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [プリプロセッサ] カテゴリの [\[定義解除マクロ\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [プリプロセッサ] カテゴリの [\[定義解除マクロ\]](#)

[使用例]

- D オプションによるプリプロセッサ・マクロ `test` の定義を解除します。

```
>ccrh -Utest -Xcommon=v850e3v5 main.c
```

-I

インクルード・ファイルを検索するフォルダを指定します。

[指定形式]

```
-Ipath[,path]...
```

- 省略時解釈

インクルード・ファイルを 標準インクルード・ファイル・フォルダからのみ検索します。

[詳細説明]

- プリプロセッサ指令 `#include`、およびアセンブラ制御命令 `$INCLUDE/$BINCLUDE` で読み込むインクルード・ファイルを検索するフォルダを *path* に指定します。
- インクルード・ファイルの検索は、以下の順番で行います。

(1) `#include` の場合

- (a) ソース・ファイルのあるフォルダ（ファイルを ” ” で指定した場合）
- (b) `-I` オプションで指定したフォルダ
- (c) 標準インクルード・ファイル・フォルダ^注

注 本製品のインストール・フォルダ¥ CubeSuite+ ¥ ccrh ¥ Vx.xx ¥ inc

(2) `$INCLUDE/$BINCLUDE` の場合

- (a) `-I` オプションで指定したフォルダ
- (b) ソース・ファイルのあるフォルダ
- (c) 元の C ソース・ファイルのあるフォルダ
- (d) カレント・フォルダ

- *path* が存在しない場合は、警告を出力します。
- *path* を省略した場合は、エラーとなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[共通オプション\] タブ](#)の [\[よく使うオプション \(コンパイル\)\]](#) カテゴリの [\[追加のインクルード・パス\]](#), [\[システム・インクルード・パス\]](#)
- [\[コンパイル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[追加のインクルード・パス\]](#), [\[システム・インクルード・パス\]](#)
- [\[個別コンパイル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[追加のインクルード・パス\]](#), [\[ビルド・ツールに指定した全体インクルード・パスも使用する\]](#)

[使用例]

- インクルード・ファイルをカレント・フォルダ, フォルダ D:¥include, 標準フォルダの順で検索します。

```
>ccrh -ID:¥include -Xcommon=v850e3v5 main.c
```

-Xpreinclude

コンパイル単位の先頭にインクルードするファイルを指定します。

[指定形式]

```
-Xpreinclude=file[,file]
```

- 省略時解釈

コンパイル単位の先頭にインクルードするファイルはないものとみなします。

[詳細説明]

- コンパイル単位の先頭にインクルードするファイルを *file* に指定します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[コンパイル単位の先頭にインクルードするファイル\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[コンパイル単位の先頭にインクルードするファイル\]](#)

[使用例]

- コンパイル単位の先頭にファイル sample.h をインクルードします。

```
>ccrh main.c -Xpreinclude=sample.h -Xcommon=v850e3v5
```

-Xpreprocess

プリプロセス結果の出力を制御します。

[指定形式]

```
-Xpreprocess=string[,string]
```

- 省略時解釈

プリプロセス結果のファイルに、C ソースのコメント、および行番号情報を出力しません。

[詳細説明]

- プリプロセス結果のファイルに、C ソースのコメント、および行番号情報を出力します。
- 本オプションは、-P オプション指定時のみ有効です。
- P オプションを指定しない場合は、本オプションを無視します。
- *string* に指定可能なものを以下に示します。
- これ以外のものを指定した場合は、エラーとなります。

comment	C ソースのコメントを出力します。
line	行番号情報 ^注 を出力します。

注 行番号情報のフォーマットを以下に示します。

```
#line 行番号 "ファイル名"
```

- 行番号は 10 進数で、最大値は unsigned int の最大数となります。
- ファイル名はフルパスで、パス中の¥は¥¥に、" は¥" に変換します。
- 印字可能文字（スペースを含む）でない場合は、¥3桁の8進数("¥¥%03o")で出力します。
- 改行文字は¥nに変換します。
- 入力ソース・ファイル中に前処理指令 # 数値 "文字列"、または #line 数値 "文字列" を記述している場合は、数値を行番号、文字列をファイル名として適用します。
- *string* を省略した場合は、エラーとなります。
- OS 標準の文字コードで出力します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [コンパイル・オプション] タブの [プリプロセス] カテゴリの [プリプロセス結果に C ソース・コメントを出力する]、[プリプロセス結果に行番号情報を出力する]
 - [個別コンパイル・オプション] タブの [プリプロセス] カテゴリの [プリプロセス結果に C ソース・コメントを出力する]、[プリプロセス結果に行番号情報を出力する]

【使用例】

- プリプロセス結果のファイルに、C ソースのコメント、および行番号情報を出力します。

```
>ccrh -Xpreprocess=comment,line -P -Xcommon=v850e3v5 main.c
```

以下は上記の例と同等です。

```
>ccrh -Xpreprocess=comment -Xpreprocess=line -P -Xcommon=v850e3v5 main.c
```

C 言語制御

C 言語制御オプションには、次のものがあります。

- Xansi
- Xenum_type
- Xvolatile
- Xcheck
- Xmisra2004
- Xignore_files_misra
- Xcheck_language_extention

-Xansi

C ソース・プログラムを ANSI 規格に厳密にあわせて処理します。

[指定形式]

-Xansi

- 省略時解釈

従来の C 言語の仕様との両立性を持たせ、警告を出力して処理を続行します。

[詳細説明]

- C ソース・プログラムを ANSI 規格^注に厳密にあわせて処理し、規格に反する記述に対してエラーや警告を出力します。
- 本オプション指定時は、マクロ名 `__STDC__` を、値が 1 のマクロとして定義します。
- 言語仕様に厳密なコンパイル時の処理は、以下のようになります。
 - ビット・フィールド
ビット・フィールドに int 型以外の型を指定した場合は、エラーとなります。
本オプションを指定しない場合は、int 型以外の型の指定を許可します（警告は出力しません）。
 - # 行番号
エラーとなります。
本オプションを指定しない場合は、“#line 行番号”と同様に扱います。
 - #pragma inline された関数の引数
指定した関数の呼び出しと定義の間で、戻り値の型や引数の型が異なるが型変換が可能である場合は、エラーとなります。
本オプションを指定しない場合は、戻り値の型は呼び出し側の型に、引数は関数定義での型に変換して、インライン展開を行います。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [\[C 言語\]](#) カテゴリの [\[ANSI 規格に厳密に合わせてコンパイルする\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [\[C 言語\]](#) カテゴリの [\[ANSI 規格に厳密に合わせてコンパイルする\]](#)

注 ISO/IE C9899:1990 (C90) で規程されている規格のことです。

ccrh は、ISO/IE C9899:1999 (C99) で追加された仕様の一部も受容しますが、本オプションを指定した場合は、規格に反する記述はエラーとなります。

【使用例】

- C ソース・プログラムを ANSI 規格に厳密にあわせて処理し、規格に反する記述に対してエラーや警告を出力します。

```
>ccrh -Xansi -Xcommon=v850e3v5 main.c
```


-Xenum_type

列挙型に対して、どの整数型として扱うかを指定します。

[指定形式]

```
-Xenum_type=string
```

- 省略時解釈

列挙型に対して、signed int として扱います。

[詳細説明]

- 列挙型に対して、どの整数型として扱うかを指定します。

- *string* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

auto	各列挙型について、その型のすべての列挙子の値を表現可能な最小の整数型として扱います。
------	--

- *string* を省略した場合は、エラーとなります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[コンパイル・オプション\]](#) タブの [C 言語] カテゴリの [\[enum の型\]](#)

[使用例]

- 各列挙型について、その型のすべての列挙子の値を表現可能な最小の整数型として扱います。

```
>ccrh -Xenum_type=auto -Xcommon=v850e3v5 main.c
```

-Xvolatile

外部変数の volatile 化を指定します。

[指定形式]

```
-Xvolatile
```

- 省略時解釈

volatile 修飾のある変数のみを volatile 宣言したものとして扱います。

[詳細説明]

- すべての外部変数を volatile 宣言したものとして扱います。
外部変数のアクセス回数、アクセス順序は C ソース・ファイルで記述したとおりになります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [C 言語] カテゴリの [\[外部変数を volatile 化する\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [C 言語] カテゴリの [\[外部変数を volatile 化する\]](#)

[使用例]

- すべての外部変数を volatile 宣言したものとして扱います。

```
>ccrh -Xvolatile -Xcommon=v850e3v5 main.c
```

-Xcheck

C ソース・ファイルの互換性をチェックします。

[指定形式]

<code>-Xcheck=comp</code>

- 省略時解釈

C ソース・ファイルの互換性のチェックを行いません。

[詳細説明]

- *comp* に指定したコンパイラ用にコーディングした C ソース・ファイルを本コンパイラでコンパイルする際、互換性に影響するオプション指定、およびソース記述のチェックを行い、影響のある部分について警告、またはエラーを出力します。

- *comp* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

shc	SHC 用にコーディングした C ソース・ファイルをチェックします。
-----	------------------------------------

- *comp* を省略した場合は、エラーとなります。

- 主なチェック項目を以下に示します。

- オプション : *-Xbit_order=pos*

言語仕様で規定されていない実装依存の内容がコンパイラ間で異なります。

メッセージに出力されたオプションの選択を確認してください。

- 拡張機能 : *#pragma section, #pragma entry, #pragma stacksize, #pragma address, #pragma global_register*

プログラムの動作に影響を及ぼす可能性がある拡張仕様です。

メッセージに出力された拡張仕様の記述を確認してください。

- *volatile* 修飾した変数

読み出しや書き込みのサイズがコンパイラ間で異なる場合があります。

volatile 修飾したビット・フィールドは、本コンパイラでは宣言型より小さいサイズでアクセスすることがありますが、SH コンパイラでは宣言型のサイズどおりにアクセスします。

- 2 項演算の汎整数拡張

unsigned int 型と *long* 型をオペランドとする 2 項演算（加減乗除や比較など）の結果が SH コンパイラと異なる場合があります。

SH コンパイラでは *-strict_ansi* オプションを指定しない場合 *signed long* 型で演算します。

本コンパイラではオペランドを *unsigned int* 型に変換してから演算します。

- *signed long* 型を超える整数定数の型

SH コンパイラでは *unsigned long* 型で表現可能な範囲の値を *signed long long* 型とします。

本コンパイラでは unsigned long 型で表現可能な範囲の値は unsigned long 型とします。

- ビット・フィールドの割り付け

SH コンパイラではビット・フィールドの型が直前のビット・フィールドの型と異なる場合はビットを連続して配置しません。

本コンパイラでは -Xpack オプションの指定に応じて連続して配置する場合があります。

- 構造体、およびビット・フィールド・メンバの割り付けについては、メッセージを出力しません。

割り付けを意識した宣言を行っている場合は、「CubeSuite+ 統合開発環境 ユーザーズマニュアル RH850 コーディング編」を参照してください。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[コンパイル・オプション\] タブ](#)の [\[C 言語\]](#) カテゴリの [\[C プログラムの互換性をチェックする\]](#)
- [\[個別コンパイル・オプション\] タブ](#)の [\[C 言語\]](#) カテゴリの [\[C プログラムの互換性をチェックする\]](#)

[使用例]

- SH コンパイラ用にコーディングした C ソース・ファイルの互換性をチェックします。

```
>ccrh -Xcheck=shc -Xcommon=v850e3v5 main.c
```

-Xmisra2004

MISRA-C:2004 ルールによるソース・チェックを行います。

[指定形式]

```
-Xmisra2004=string
```

- 省略時解釈

MISRA-C:2004 ルールによるソース・チェックを行いません。

[詳細説明]

- MISRA-C:2004 ルールによるソース・チェックを行います。

指定したチェック項目 *string* に該当した場合、メッセージを出力します。

- *string* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

チェック項目 (<i>string</i>)	パラメータ (<i>value</i>)	説明
all	なし	サポートしているすべてのルールをチェック対象とします。
apply	<i>num</i> [, <i>num</i>]...	サポートしているルールのうち、 <i>num</i> で指定した番号のルールをチェック対象とします。
ignore	<i>num</i> [, <i>num</i>]...	サポートしているルールのうち、 <i>num</i> で指定した番号以外のルールをチェック対象とします。
required	なし	サポートしているルールのうち、ルールの分類が“required”になっているルールをチェック対象とします。
required_add	<i>num</i> [, <i>num</i>]...	サポートしているルールのうち、ルールの分類が“required”になっているルールと <i>num</i> で指定した番号のルールをチェック対象とします。
required_remove	<i>num</i> [, <i>num</i>]...	サポートしているルールのうち、ルールの分類が“required”になっているルールから <i>num</i> で指定した番号を除いたルールをチェック対象とします。
<i>file</i>		サポートしているルールのうち、指定したファイル <i>file</i> に記載した番号のルールをチェック対象とします。 ファイル内では、1 行につき 1 ルール番号を指定します。

- *num* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

2.2 2.3

4.1 4.2

5.2 5.3 5.4 5.5 5.6

6.1 6.2 6.3 6.4 6.5

7.1
8.1 8.2 8.3 8.5 8.6 8.7 8.11 8.12
9.1 9.2 9.3
10.1 10.2 10.3 10.4 10.5 10.6
11.1 11.2 11.3 11.4 11.5
12.1 12.3 12.4 12.5 12.6 12.7 12.8 12.9 12.10 12.11 12.12 12.13
13.1 13.2 13.3 13.4
14.2 14.3 14.4 14.5 14.6 14.7 14.8 14.9 14.10
15.1 15.3 15.4 15.5
16.1 16.3 16.5 16.6 16.9
17.5
18.1 18.4
19.3 19.6 19.7 19.8 19.11 19.13 19.14 19.15
20.4 20.5 20.6 20.7 20.8 20.9 20.10 20.11 20.12

- *string* を省略した場合は、エラーとなります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- **[コンパイル・オプション]** タブの [MISRA-C:2004 ルール検査] カテゴリの **[適用するルール]**, **[ルール番号記載ファイル]**, **[ルール番号]**, **[除外するルール番号]**, **[必須ルールの他にチェックするルール番号]**, **[必須ルールから除外するルール番号]**
- **[個別コンパイル・オプション]** タブの [MISRA-C:2004 ルール検査] カテゴリの **[適用するルール]**, **[ルール番号記載ファイル]**, **[ルール番号]**, **[除外するルール番号]**, **[必須ルールの他にチェックするルール番号]**, **[必須ルールから除外するルール番号]**

[使用例]

- MISRA-C:2004 ルール番号 5.2, 5.3, 5.4 のルールをチェック対象としたソース・チェックを行います。

```
>ccrh -Xmisra2004=apply5.2,5.3,5.4 -Xcommon=v850e3v5 main.c
```

-Xignore_files_misra

MISRA-C:2004 ルールによるソース・チェックの対象外のファイルを指定します。

[指定形式]

```
-Xignore_files_misra=file[,file]...
```

- 省略時解釈

すべての C ソース・ファイルをチェック対象とします。

[詳細説明]

- *file* で指定したファイルを MISRA-C:2004 ルールによるソース・チェックの対象外とします。
- 本オプションは、-Xmisra2004 オプション指定時のみ有効です。
 - Xmisra2004 オプションを指定しない場合は、本オプションを無視します（警告は出力しません）。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[MISRA-C:2004 ルール検査] カテゴリの[\[ルール・チェック対象外のファイル\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[MISRA-C:2004 ルール検査] カテゴリの[\[ルール・チェック対象外のファイル\]](#)

[使用例]

- sample.c を MISRA-C:2004 ルールによるソース・チェックの対象外とします。

```
>ccrh -Xmisra2004=all -Xignore_files_misra=sample.c -Xcommon=v850e3v5 main.c
```

-Xcheck_language_extention

言語拡張により部分抑止される MISRA-C:2004 ルールのソース・チェックを有効にします。

[指定形式]

```
-Xcheck_language_extention
```

- 省略時解釈

言語拡張により部分抑止される MISRA-C:2004 ルールのソース・チェックを無効にします。

[詳細説明]

- C 言語規格から独自に拡張した言語仕様のために MISRA-C:2004 ルールのソース・チェックが抑止される以下の場合について、ソース・チェックを有効にします。
 - プロトタイプ宣言がなく（ルール 8.1）、該当関数に #pragma interrupt 指定がある場合
- 本オプションは、-Xmisra2004 オプション指定時のみ有効です。
 - Xmisra2004 オプションを指定しない場合は、本オプションを無視します（警告は出力しません）。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [\[MISRA-C:2004 ルール検査\]](#) カテゴリの [\[拡張キーワードや拡張仕様をメッセージ出力する\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [\[MISRA-C:2004 ルール検査\]](#) カテゴリの [\[拡張キーワードや拡張仕様をメッセージ出力する\]](#)

[使用例]

- 言語拡張により部分抑止される MISRA-C:2004 ルールのソース・チェックを有効にします。

```
>ccrh -Xmisra2004=all -Xcheck_language_extention -Xcommon=v850e3v5 main.c
```


日本語／中国語文字列制御

日本語／中国語文字列制御オプションには、次のものがあります。

- [-Xcharacter_set](#)

-Xcharacter_set

日本語／中国語の文字コードを指定します。

[指定形式]

```
-Xcharacter_set=code
```

- 省略時解釈

日本語の文字コードを SJIS として扱います。

[詳細説明]

- ソース・ファイル中の日本語／中国語のコメント、文字列に対して、使用する文字コードを指定します。

- *code* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

なお、ソース・ファイル中で使用している文字コードと異なるものを指定した場合、動作は保証されません。

none	日本語／中国語の文字コードを解釈しません
euc_jp	EUC（日本語）
sjis	SJIS
utf8	UTF-8
big5	繁体字中国語
gb2312	簡体字中国語

- *code* を省略した場合は、エラーとなります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[コンパイル・オプション\] タブ](#)の「文字コード」カテゴリの「[文字コード](#)」
- [\[個別コンパイル・オプション\] タブ](#)の「文字コード」カテゴリの「[文字コード](#)」

[使用例]

- ソース・ファイル中の日本語のコメント、文字列に対して、使用する文字コードに EUC を指定します。

```
>ccrh -Xcharacter_set=euc_jp -Xcommon=v850e3v5 main.c
```

最適化指定

最適化指定オプションには、次のものがあります。

- O
- Xintermodule
- Xinline_strcpy
- Xmerge_string
- Xalias
- Xmerge_files
- Xwhole_program

-O

最適化のレベル、または各最適化項目の詳細を指定します。

[指定形式]

```
-O[level]  
-O[item[=value][,item[=value]]...]
```

- 省略時解釈
デバッグに影響しない最適化のみを行います（-Odefault オプションの指定と同じです）。

[詳細説明]

- 最適化のレベル、または各最適化項目の詳細を指定します。
- level に指定可能なものを以下に示します。
これ以外のものを指定した場合は、エラーとなります。

nothing	デバッグ優先の最適化 デバッグのしやすさを重視し、デフォルトで実行する最適化を含むすべての最適化を抑止します。
default	デフォルト デバッグに影響しない範囲の最適化（式の最適化、およびレジスタ割り付けなど）を行います。
size	オブジェクト・サイズ優先の最適化 ROM/RAM 容量の削減を重視して、一般的なプログラムに対して有効な最大限の最適化を行います。
speed	実行速度優先の最適化 実行速度の短縮を重視して、一般的なプログラムに対して有効な最大限の最適化を行います。

- level を省略した場合は、size を指定したものとみなします。
- item、および value に指定可能なものを以下に示します。
これ以外のものを指定した場合は、エラーとなります。

最適化項目 (item)	パラメータ (value)	説明
unroll	0 ~ 4294967295 (整数値)	ループ展開 0, または 1 を指定した場合は、展開を抑止します。 2 以上を指定した場合は、実行回数が N 回 (N は定数) のループと、<i>value</i> 回展開されたコードを含むループに変換します。 ただし、展開後のコード・サイズが大きいか、またはループの実行回数が少ない場合は、展開数が少なくなったり、展開しない場合があります。 また、内側にループを含むような複雑な構造のループは、展開しない場合があります。 <i>value</i> を省略した場合は、4 を指定したものとみなします。 本項目は、-Ospeed オプションを指定した場合は、-Ounroll=4 オプションを指定したものとみなします。
inline	0 ~ 3 (整数値)	関数のインライン展開 <i>value</i> は、展開のレベルを表します。 0 : #pragma inline 指定した関数を含めて、すべてのインライン展開を抑止します。 1 : #pragma inline 指定した関数のみ展開します。 2 : 自動的に展開対象の関数を判別して展開します。 3 : コード・サイズがなるべく増加しない範囲で、自動的に展開対象の関数を判別して展開します。 ただし、1 ~ 3 を指定した場合でも、関数の内容やコンパイル状況により、#pragma inline 指定した関数が展開されない場合があります。 <i>value</i> を省略した場合は、2 を指定したものとみなします。 本項目は、-Osize, または -Ospeed オプションを指定した場合に有効です (-Osize オプションを指定した場合は -Oinline=3, -Ospeed オプションを指定した場合は -Oinline=2 オプションを指定したものとみなします)。 -Osize, -Ospeed, -Oinline オプションのいずれも指定していない場合は、-Oinline=1 オプション指定したものとみなします。 -Onothing オプションを指定した場合は、-Oinline=0 オプションを指定したものとみなします。
inline_size	0 ~ 65535 (整数値)	インライン展開サイズ コード・サイズが何%増加するまでインライン展開を行うかを指定します。 <i>value</i> を省略した場合は、100 を指定したものとみなします。 本項目は、-Oinline=2 オプションを指定した場合に有効です (-Ospeed オプションによる指定も含みます)。
delete_static_func	on, または off	未使用 static 関数の削除 <i>value</i> を省略した場合は、on を指定したものとみなします。 本項目は、-Onothing オプション指定時以外に有効です。
pipeline	on, または off	パイプライン最適化 <i>value</i> を省略した場合は、on を指定したものとみなします。 本項目は、-Ospeed オプション指定時は、常に有効です。

最適化項目 (<i>item</i>)	パラメータ (<i>value</i>)	説明
tail_call	on, または off	関数末尾の関数呼び出しの jr 化 on を指定した場合、関数の末尾が関数呼び出しであり、かつ一定の条件を満たす場合に、その呼び出しに対して jarl 命令ではなく jr 命令を生成して lp の退避／復帰コードを削除し、コード・サイズを削減します。 ただし、一部のデバッグ機能を使用することができなくなります。 value を省略した場合は、on を指定したものとみなします。 本項目は、-Ospeed, または -Osize オプションを指定した場合に有効です。
map	ファイル名	外部変数アクセス最適化 最適化リンクが生成する外部シンボル割り付け情報を元にベースアドレスを設定し、外部変数、または静的変数のアクセスをベース・アドレス相対で行うコードを生成します。 また、スタートアップ・ルーチンでシンボル “__gp_data” を定義して gp に値を設定するコードを記述した場合、外部シンボル割り付け情報を元に、可能であれば変数のアクセスを gp 相対で行うコードを生成します。 ファイル名には、最適化リンクが生成した外部シンボル割り付け情報ファイルを指定します。 ファイル名を省略した場合は、一度リンク処理まで実行し、外部シンボル割り付け情報ファイルを作成したのち、再度コンパイルからリンクまでの処理を行います。
smap	なし	コンパイル単位内で定義された外部変数に対する外部変数アクセス最適化 コンパイル対象ファイル内で定義された外部変数、または静的変数についてベース・アドレスを設定し、アクセスをベース・アドレス相対で行うコードを生成します。

- *item* を省略した場合、本オプションは -Osize を指定したものとみなします。
- 同じ *item* について、本オプションを複数指定した場合は、あとから指定したものが有効となります。
- -Oitem を指定しなかった場合の最適化項目は、-Olevel の指定によって以下のように解釈します。

最適化項目 (<i>item</i>)	最適化レベル (<i>level</i>)			
	nothing	default	size	speed
unroll	1	1	1	4
inline	0	1	3	2
inline_size	—	—	—	100
delete_static_func	off	on	on	on
pipeline	off	off	off	on
tail_call	off	off	on	on
map	—	—	—	—
smap	—	—	—	—

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\]](#) タブの [\[よく使うオプション \(コンパイル\)\]](#) カテゴリの [\[最適化レベル\]](#)
 - [\[コンパイル・オプション\]](#) タブの [\[最適化\]](#) カテゴリの [\[最適化レベル\]](#), [\[最適化 \(詳細\)\]](#) カテゴリの [\[ループ展開最大数\]](#), [\[未使用 static 関数の削除を行う\]](#), [\[関数のインライン展開を行う\]](#), [\[インライン展開サイズの最大増加率\]](#), [\[パイプライン最適化を行う\]](#), [\[関数末尾の関数呼び出しに jr 命令を使用する\]](#), [\[外部変数アクセス最適化を行う\]](#)
 - [\[個別コンパイル・オプション\]](#) タブの [\[最適化\]](#) カテゴリの [\[最適化レベル\]](#), [\[最適化 \(詳細\)\]](#) カテゴリの [\[ループ展開最大数\]](#), [\[未使用 static 関数の削除を行う\]](#), [\[関数のインライン展開を行う\]](#), [\[インライン展開サイズの最大増加率\]](#), [\[パイプライン最適化を行う\]](#), [\[関数末尾の関数呼び出しに jr 命令を使用する\]](#), [\[外部変数アクセス最適化を行う\]](#)

【使用例】

- オブジェクト・サイズ優先の最適化を行います。

```
>ccrh -Osize -Xcommon=v850e3v5 main.c
```

-Xintermodule

大域最適化を行います。

[指定形式]

-Xintermodule

- 省略時解釈

大域最適化を行いません。

[詳細説明]

- 大域最適化を行います。

- 主な最適化内容を以下に示します。

- 手続き間別名解析を利用した最適化

出力コード例を以下に示します。

[C ソース]

```
extern int x[2];
static int func1(int *a, int *b) {
    *a=0;
    *b=1;
    return *a;
}
int func2() {
    return func1(&x[0], &x[1]);
}
```

[出力アセンブラ・ソース]

```
_func1.1:
    .stack  _func1.1 = 0
    mov     #_x, r2
    st.w    r0, 0x00000000[r2]
    mov     0x00000001, r5
    st.w    r5, 0x00000004[r2]
    mov     0x00000000, r10      ; a と b の指す先が違うため 0 を直接代入
    jmp     [r31]
_func2:
    .stack  _func2 = 0
    mov     #_x, r6
    addi    0x00000004, r6, r7
    br9     _func1.1
```

- パラメータ, 戻り値の定数伝播

出力コード例を以下に示します。

```
[C ソース]
static int func(int x, int y, int z) {
    return x-y+z;
}
int func2() {
    return func(3,4,5);
}

[ 出力アセンブラ・ソース ]
_func.1:
    .stack   _func.1 = 0
    mov     0x00000004, r10      ; 4 (=3-4+5) を直接代入
    jmp     [r31]
_func2:
    .stack   _func2 = 0
    mov     0x00000005, r8
    mov     0x00000004, r7
    mov     0x00000003, r6
    br9     _func.1
```

- 本オプションは, CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[最適化 (詳細)] カテゴリの [\[大域最適化を行う\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[最適化 (詳細)] カテゴリの [\[大域最適化を行う\]](#)

[使用例]

- ソース・ファイル main.c, sub.c の大域最適化を行います。

```
>ccrh -Xintermodule -Xwhole_program -Xcommon=v850e3v5 main.c sub.c
```

-Xinline_strcpy

標準ライブラリ関数 strcpy, strcmp, memcpy, memset の呼び出しをインライン展開します。

[指定形式]

```
-Xinline_strcpy
```

- 省略時解釈

標準ライブラリ関数 strcpy, strcmp, memcpy, memset のインライン展開を行いません。

[詳細説明]

- 標準ライブラリ関数 strcpy, strcmp, memcpy, memset の呼び出しをインライン展開します。
- 本オプションは、-Xpack オプションと同時に指定することはできません。
- strcpy については、第二引数が文字列の場合にのみ、インライン展開を行います。
- 本オプションを指定した場合、配列、および文字列は自動的に 4 バイト境界に配置されます。
- 生成されるプログラムの実行速度は高速になりますが、コード・サイズは増大します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[\[最適化 \(詳細\)\]](#)カテゴリの[\[strcpy / strcmp / memcpy / memset の展開を行う\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[\[最適化 \(詳細\)\]](#)カテゴリの[\[strcpy / strcmp / memcpy / memset の展開を行う\]](#)

[使用例]

- 標準ライブラリ関数 strcpy, strcmp, memcpy, memset の呼び出しをインライン展開します。

```
>ccrh -Xinline_strcpy -Xcommon=v850e3v5 main.c
```


-Xmerge_string

文字列定数をマージします。

[指定形式]

```
-Xmerge_string
```

- 省略時解釈

ソース・ファイル内で同じ文字列定数が複数存在する場合、それぞれを別々の領域に割り付けます。

[詳細説明]

- ソース・ファイル内で同じ文字列定数が複数存在する場合、これらをまとめて1つの領域に割り付けます。
- #pragma section sconst (const) の指定に依らず、同じ文字列定数を同一領域に割り付けます。
ただし、異なるセクションを指定した場合、文字列定数の割り付け先のセクションはソース中での出現順序に依存します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[最適化 (詳細)] カテゴリの[\[文字列定数のマージを行う\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[最適化 (詳細)] カテゴリの[\[文字列定数のマージを行う\]](#)

[使用例]

- ソース・ファイル内で同じ文字列定数が複数存在する場合、これらをまとめて1つの領域に割り付けます。

```
>ccrh -Xmerge_string -Xcommon=v850e3v5 main.c
```

-Xalias

ポインタ指示先の型を考慮した最適化の指定を行います。

[指定形式]

```
-Xalias=value
```

- 省略時解釈

ANSI 規格に基づくポインタ指示先の型を考慮した最適化を行いません。

[詳細説明]

- ANSI 規格に基づくポインタ指示先の型を考慮した最適化を行うかどうかを指定します。

- *value* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

ansi	ANSI 規格に基づき、ポインタ指示先の型を考慮した最適化を行います。
noansi	ANSI 規格に基づくポインタ指示先の型を考慮した最適化を行いません。

- *value* を省略した場合は、エラーとなります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[コンパイル・オプション\] タブ](#)の [\[最適化 \(詳細\)\]](#) カテゴリの [\[ポインタ指示先の型を考慮した最適化を行う\]](#)
- [\[個別コンパイル・オプション\] タブ](#)の [\[最適化 \(詳細\)\]](#) カテゴリの [\[ポインタ指示先の型を考慮した最適化を行う\]](#)

[使用例]

- ANSI 規格に基づき、ポインタ指示先の型を考慮した最適化を行います。

```
>ccrh -Xalias=ansi -Xcommon=v850e3v5 main.c
```

-Xmerge_files

複数の C ソース・ファイルをマージしてコンパイルを行います。

[指定形式]

```
-Xmerge_files
```

- 省略時解釈

マージせずに入力ファイル単位でコンパイルを行います。

[詳細説明]

- 複数の C ソース・ファイルをマージしてコンパイルを行い、1 ファイルで出力します。
- 出力ファイル名は、-o オプションを指定した場合は指定したファイル名となり、-o オプションを指定しなかった場合は最初に指定した C ソース・ファイル名に対して -o オプションの省略時解釈に従ったファイル名となります。
- 本オプションは、入力 C ソース・ファイルが 1 ファイルの場合、および -P オプションと同時に指定した場合は無効となります。
- 本オプションは、-S、または -c オプションと同時に指定した場合は、2 番目以降に指定した C ソース・ファイル名について -o オプションの省略時解釈に従ったファイル名の空ファイルを出力します。
- -Oinline オプションと同時に指定した場合は、ファイル間インライン展開を行います。
- 本オプションを指定して生成したオブジェクト・ファイルをリンクする際にリンク・オプション -delete、-rename、-replace のいずれかを同時に指定した場合、動作は保証されせん。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[最適化 (詳細)] カテゴリの [\[大域最適化を行う\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[最適化 (詳細)] カテゴリの [\[大域最適化を行う\]](#)

[使用例]

- main.c, sub.c をマージしてコンパイルを行い、1 ファイルで出力します。

```
>ccrh -Xmerge_files -Xwhole_program -Xcommon=v850e3v5 main.c sub.c
```

-Xwhole_program

コンパイル対象ファイルがプログラム全体であることを仮定して最適化を行います。

[指定形式]

```
-Xwhole_program
```

- 省略時解釈

コンパイル対象ファイルがプログラム全体であることを仮定しません。

[詳細説明]

- コンパイル対象ファイルがプログラム全体であることを仮定して最適化を行います。
- 以下の条件を満たすことを前提にコンパイルを行い、条件を満たさなかった場合の動作は保証されません。
 - コンパイル対象ファイル内で定義した extern 変数の値、およびアドレスが、コンパイル対象ファイル以外で変更、および参照されない。
 - コンパイル対象ファイル内からコンパイル対象ファイル以外で定義した関数を呼び出したとしても、呼び出された関数からコンパイル対象ファイル内の関数が呼ばれることがない。
- 本オプションを指定した場合、-Xintermodule オプションを指定したものとみなしてコンパイルを行います。
また、入力 C ソース・ファイルが複数の場合、-Xmerge_files オプションを指定したものとみなしてコンパイルを行います。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[最適化 (詳細)] カテゴリの [\[大域最適化を行う\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[最適化 (詳細)] カテゴリの [\[大域最適化を行う\]](#)

[使用例]

- コンパイル対象ファイルがプログラム全体であることを仮定して最適化を行います。

```
>ccrh -Xwhole_program -Xcommon=v850e3v5 main.c
```

生成コード制御

生成コード制御オプションには、次のものがあります。

- Xpack
- Xbit_order
- Xpass_source
- Xswitch
- Xreg_mode
- Xreserve_r2
- Xep
- Xfloat
- Xcall_jump
- Xfar_jump
- Xdiv
- Xcheck_div_ov
- Xuse_fmaf
- Xunordered_cmpf
- Xmulti_level
- Xpatch

-Xpack

構造体パッキングを行います。

[指定形式]

`-Xpack=num`

- 省略時解釈

構造体パッキングを行いません。

[詳細説明]

- 構造体パッキングを行います。
- 本オプションを指定した場合、構造体のメンバをその型でアライメントせず、指定した *num* バイトのアライメントに詰めてコードを生成します。
- *num* には、1, 2, 4 のいずれかを指定することができます。
これ以外のものを指定した場合は、エラーとなります。
- *num* を省略した場合は、エラーとなります。
- 本オプションは、-Xinline_strcpy オプションと同時に指定することはできません。

- C ソース中に #pragma 指令で構造体パッキングを指定している場合に本オプションを指定すると、最初の #pragma 指令が出現するまでは、オプションの指定値をすべての構造体に適用します。
それ以降は、#pragma 指令の値を適用します。
ただし、#pragma 指令の出現後でも、指定がデフォルトになった部分（#pragma 指令でパッキング値を指定しない場合）は、オプションの指定値を適用します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [\[出力コード\]](#) カテゴリの [\[構造体パッキングを行う\]](#)

[使用例]

- 構造体のメンバを 1 バイトのアライメントに詰めてコードを生成します。

```
>ccrh -Xpack=1 -Xcommon=v850e3v5 main.c
```

-Xbit_order

ビット・フィールドのメンバの並び順を指定します。

[指定形式]

```
-Xbit_order=pos
```

- 省略時解釈

ビット・フィールドのメンバを下位ビットから割り付けます。

[詳細説明]

- ビット・フィールドのメンバの並び順を指定します。
- *pos* に指定可能なものを以下に示します。
これ以外のものを指定した場合は、エラーとなります。

left	上位ビットからメンバを割り付けます。
right	下位ビットからメンバを割り付けます。

- *pos* を省略した場合は、エラーとなります。
- C ソース中に `#pragma` 指令でビット・フィールドのメンバの並び順を指定している場合、最初の `#pragma bit_order` 指令が出現するまでは、オプション指定値がすべてのビット・フィールドのメンバに適用されます。それ以降は、`#pragma` 指令の値が適用されます。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [【コンパイル・オプション】](#) タブの【出力コード】カテゴリの [【ビット・フィールド・メンバの並び順】](#)

[使用例]

- ビット・フィールドのメンバを上位ビットから割り付けます。

```
>ccrh -Xbit_order=left -Xcommon=v850e3v5 main.c
```

-Xpass_source

出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力します。

[指定形式]

```
-Xpass_source
```

- 省略時解釈

出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力しません。

[詳細説明]

- 出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力します。
- 出力するコメントは、あくまで参考であり、厳密にはコードと対応していない場合もあります。
また、実行文以外の行にはコメントとして出力されないものもあります（型宣言やラベルなど）。
たとえば、グローバル変数とローカル変数、関数宣言などのコメントの出力位置がずれることがあります。
また、最適化オプションを指定した場合などにより、コードが削除され、コメントのみが残ることもあります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [出力コード] カテゴリの [\[アセンブリ・ソース・ファイルにコメントを出力する\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [出力コード] カテゴリの [\[アセンブリ・ソース・ファイルにコメントを出力する\]](#)

[使用例]

- 出力するアセンブリ・ソース・ファイル中に C ソース・プログラムをコメントとして出力します。

```
>ccrh -Xpass_source -S -Xcommon=v850e3v5 main.c
```


-Xswitch

switch 文のコード出力形式を指定します。

[指定形式]

```
-Xswitch=type
```

- 省略時解釈

コンパイラが switch 文ごとに最適な出力形式を自動的に選択します。

[詳細説明]

- switch 文のコード出力形式を指定します。
- type に指定可能なものを以下に示します。
これ以外のものを指定した場合は、エラーとなります。

ifelse	case 文の並びに沿った if-else 文と同じ形式で出力します。 頻度が多い順に case 文を書いているときやラベル数が少ないときに、本項目を指定します。 上から順に比較するので、頻繁に合致する case 文を先に記述すると余計な比較が減り、実行速度向上につながります。
binary	バイナリ・サーチ形式で出力します。 バイナリ・サーチ・アルゴリズムに用いて合致する case 文を探します。 ラベル数が多いときに本項目を選択すると、どの case 文も同じくらいの速さで見つけることができます。
table	テーブル・ジャンプ形式で出力します。 case 文の値を基にインデックス化したテーブルを参照し、switch 文の値により case ラベルを選択して処理を行います。 どの case 文にも同じくらい速く分岐します。 ただし、case 値が連続していないときは無駄な領域ができます。 また、case ラベルの最大値と最小値の差が 8192 を超える場合は、本オプションを無視して、switch 文ごとに最適な出力形式を自動的に選択します。

- type を省略した場合は、エラーとなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の「[出力コード] カテゴリ」の「[switch 文の出力コードの選択](#)」
 - [\[個別コンパイル・オプション\] タブ](#)の「[出力コード] カテゴリ」の「[switch 文の出力コードの選択](#)」

[使用例]

- switch 文のコードに対して、バイナリ・サーチ形式で出力します。

```
>ccrh -Xswitch=binary -Xcommon=v850e3v5 main.c
```

-Xreg_mode

レジスタ・モードを指定します。

[指定形式]

```
-Xreg_mode=mode
```

- 省略時解釈

32 レジスタ・モードのオブジェクト・ファイルを生成します。

[詳細説明]

- 指定したレジスタ・モードのオブジェクト・ファイルを生成します。
- ccrh が使用するレジスタを 32 本（32 レジスタ・モード）、22 本（22 レジスタ・モード、または common レジスタ・モード）のいずれかに制限し、オブジェクト・ファイル内にレジスタ・モードを示すマジック・ナンバを埋め込みます。
- common レジスタ・モードは、レジスタ・モードに依存しないオブジェクト・ファイルを生成するために使用します。
- *mode* に指定可能なものを以下に示します。
これ以外のものを指定した場合は、エラーとなります。

レジスタ・モード (<i>mode</i>)	作業用レジスタ	レジスタ変数用レジスタ
common	r10 ~ r14	r25 ~ r29
22	r10 ~ r14	r25 ~ r29
32	r10 ~ r19	r20 ~ r29

- *mode* を省略した場合は、エラーとなります。
- C ソース・ファイルに対して、使用可能なレジスタのみを用いたコードを生成します。
- アセンブリ・ソース・ファイルに対して、使用可能なレジスタ以外のレジスタが使用されている場合は、エラーとなります。
- 本オプションは、すべてのソース・ファイルに対して同じレジスタ・モードを指定する必要があります。
ソース・ファイルごとに指定を変えることはできません。
レジスタ・モードが異なるオブジェクト・ファイルが混在している場合は、リンク時にエラーとなります。
- 本オプションを指定することにより、ソフトウェア・レジスタ・バンク機能のレジスタ・モードを切り替えることができます。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\]](#) タブの [\[レジスタ・モード\]](#) カテゴリの [\[レジスタ・モード\]](#)

【使用例】

- 22 レジスタ・モードのオブジェクト・ファイルを生成します。

```
>ccrh -Xreg_mode=22 -Xcommon=v850e3v5 main.c
```

-Xreserve_r2

r2 レジスタを予約します。

【指定形式】

```
-Xreserve_r2
```

- 省略時解釈

r2 レジスタの予約を行わずに、コンパイラで使します。

【詳細説明】

- r2 レジスタを予約し、コンパイラでは r2 レジスタを使用しないコードを生成します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [【共通オプション】](#) タブの [\[レジスタ・モード\]](#) カテゴリの [\[r2 レジスタを予約する\]](#)

【使用例】

- r2 レジスタを予約し、コンパイラでは r2 レジスタを使用しないコードを生成します。

```
>ccrh -Xreserve_r2 -Xcommon=v850e3v5 main.c
```

-Xep

ep レジスタの扱い方を指定します。

[指定形式]

```
-Xep=mode
```

- 省略時解釈

ep レジスタを関数呼び出し前後で値を保証するレジスタとして扱います。

[詳細説明]

- ep レジスタの扱い方を指定します。

- *mode* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

fix	ep レジスタの値をプロジェクト全体で固定します。 プロジェクト内の #pragma section 指令において、以下の属性指定文字を使用した場合は、本パラメータを指定してください。 ep_auto, ep_disp4, ep_disp5, ep_disp7, ep_disp8, ep_disp16, ep_disp23
callee	ep レジスタを関数呼び出し前後で値を保証するレジスタとして扱います。 -Omap, または -Osmap オプションを指定した場合は、本パラメータを指定してください。

- *mode* を省略した場合は、エラーとなります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[共通オプション\] タブ](#)の [レジスタ・モード] カテゴリの [\[ep レジスタの扱い\]](#)

[使用例]

- ep レジスタの値をプロジェクト全体で固定します。

```
>ccrh -Xep=fix -Xcommon=v850e3v5 main.c
```

-Xfloat

浮動小数点演算命令の生成を制御します。

[指定形式]

```
-Xfloat=type
```

- 省略時解釈

-Xcpu=g3m オプションを指定した場合、および -Xcpu オプションを指定しなかった場合は、浮動小数点演算命令を生成します。

-Xcpu=g3k オプションを指定した場合は、ランタイム関数の呼び出し命令を生成します。

[詳細説明]

- 浮動小数点演算命令の生成を制御します。

- type に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

soft	浮動小数点演算に対して、ランタイム・ライブラリの呼び出し命令を生成します。 また、FPU 数学対応ライブラリはリンクしません。
fpu	浮動小数点演算に対して、FPU（浮動小数点ユニット）の浮動小数点演算命令を生成します。 また、FPU 数学対応ライブラリをリンクします。

- type を省略した場合は、エラーとなります。

- 本オプションは、-Xcpu=g3k オプションと同時に指定した場合は無効となり、常にランタイム関数の呼び出し命令を生成します。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[コンパイル・オプション\] タブ](#)の[出力コード] カテゴリの[\[浮動小数点演算方法\]](#)
- [\[個別コンパイル・オプション\] タブ](#)の[出力コード] カテゴリの[\[浮動小数点演算方法\]](#)

[使用例]

- 浮動小数点演算に対して、ランタイム・ライブラリの呼び出し命令を生成します。

```
>ccrh -Xfloat=soft -Xcommon=v850e3v5 main.c
```

-Xcall_jump

関数呼び出しの分岐命令の生成を制御します。

[指定形式]

```
-Xcall_jump=num
```

- 省略時解釈

関数呼び出しの分岐に対して、jarl、および jr 命令を生成します。

[詳細説明]

- 関数呼び出しの分岐命令の生成を制御します。

- *num* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

22	関数呼び出しの分岐に対して、jarl、および jr 命令を生成します。
32	関数呼び出しの分岐に対して、jarl32、および jr32 命令を生成します。

- *num* を省略した場合は、エラーとなります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[コンパイル・オプション\] タブ](#)の[出力コード] カテゴリの[\[関数呼び出し命令を指定する\]](#)
- [\[個別コンパイル・オプション\] タブ](#)の[出力コード] カテゴリの[\[関数呼び出し命令を指定する\]](#)

[使用例]

- 関数呼び出しの分岐に対して、jarl32、および jr32 命令を生成します。

```
>ccrh -Xcall_jump=32 -Xcommon=v850e3v5 main.c
```

-Xfar_jump

far jump の出力を制御します。

[指定形式]

```
-Xfar_jump=file
```

- 省略時解釈

-Xcall_jump オプションに従った命令を生成します。

[詳細説明]

- C ソース・ファイルについて、far jump 呼び出し関数一覧ファイル *file* 中で指定した関数（C 言語で記述した割り込み関数を含む）への分岐に対して、jarl32、および jr32 命令を生成します。
- *file* の推奨拡張子は、“.fjp” です。
- *file* が存在しない場合は、エラーとなります。
- *file* を省略した場合は、エラーとなります。
- -Xcall_jump=22 オプションを指定した際、関数本体が jarl、および jr 命令では分岐できない範囲（± 2M バイト以上）にあり、リンク時にエラーとなる場合、本オプションを使用してコンパイルし直します。
- 本オプションを複数指定した場合は、最後に指定したものが有効となります。
- 出力コード例を以下に示します。

- C ソース

```
far_func(); /* デフォルトでは jarl _far_func, lp を出力する */
```

- 出力アセンブリ・ソース

```
jarl32 _far_func, lp
```

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [出力コード] カテゴリの [\[Far Jump ファイル名\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [出力コード] カテゴリの [\[Far Jump ファイル名\]](#)

備考 far jump 呼び出し関数一覧ファイルの書式に関する注意事項を以下に示します。

- 1 行に 1 個の関数名を記述します。
複数の関数名を記述した場合は、最初の名前のみを有効とします。
- 記述する関数名は、C 言語の関数名の先頭に “_” を付けた名前（アセンブリ・ソースでのラベル名）とします。
ただし、関数名の代わりに、以下の形式で指定することもできます。

形式	意味
{all function}	すべての関数呼び出しを対象とします。
{all interrupt}	すべての割り込み関数呼び出しを対象とします。

- 関数だけでなく、ランタイム・ルーチンも指定可能です。
その場合は、関数名としてランタイム・ルーチン名をそのまま記述してください。
- 関数名の前後に、空白やタブを挿入することが可能です。
- 使用可能な文字は、ASCII 文字のみです。
なお、行頭の空白を除いたあと、次の空白、または行末までの非空白文字列を関数名とし、空白から行末までは無視します。
- コメントを挿入することはできません。
- 1 行に記述可能な文字数は、1023 文字までです（空白やタブも含みます）。

関数の指定例を以下に示します。

```
_func_led  
_func_beep  
_func_motor  
:  
_func_switch
```

[使用例]

- func.fjp 中で指定した関数への分岐に対して、jmp 命令を使用したコードを生成します。

```
>ccrh -Xfar_jump=func.fjp -Xcommon=v850e3v5 main.c
```

-Xdiv

除算に対して、div、および divu 命令を生成します。

[指定形式]

```
-Xdiv
```

- 省略時解釈

除算に対して、divq、および divqu 命令を生成します。

[詳細説明]

- 除算に対して、divq、および divqu 命令を生成する代わりに、div、および divu 命令を生成します。
- divq、および divqu 命令は高速ですが、実行サイクル数がオペランドの値により変わります。
そのため、リアルタイム性の保証などのために、常に実行サイクル数が一定であることが必要な場合に、本オプションを指定します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[出力コード] カテゴリの [\[div / divu 除算命令を生成する\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[出力コード] カテゴリの [\[div / divu 除算命令を生成する\]](#)

[使用例]

- 除算に対して、div、および divu 命令を生成します。

```
>ccrh -Xdiv -Xcommon=v850e3v5 main.c
```

-Xcheck_div_ov

除算時に OV フラグのチェックを行います。

[指定形式]

```
-Xcheck_div_ov=num
```

- 省略時解釈

除算時に OV フラグのチェックを行わないコードを生成します。

[詳細説明]

- 除算命令の後に OV フラグのチェックを行い、OV フラグが 1 のときに FE レベル・ソフトウェア例外を発生させるコード（fetrp 命令）を生成します。
- *num* に指定可能な値は、1 ～ 15（fetrp 命令のオペランドで指定可能な値）です。
これ以外のものを指定した場合は、エラーとなります。
- *num* を省略した場合は、エラーとなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[\[出力コード\]](#)カテゴリの[\[除算時の OV フラグ・チェック・コードを生成する\]](#)、[\[除算時例外の fetrap 命令ベクタ番号\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[\[出力コード\]](#)カテゴリの[\[除算時の OV フラグ・チェック・コードを生成する\]](#)、[\[除算時例外の fetrap 命令ベクタ番号\]](#)

[使用例]

- 除算時に OV フラグのチェックを行います。

```
>ccrh -Xcheck_div_ov=1 -Xcommon=v850e3v5 main.c
```

-Xuse_fmaf

積和演算命令を生成します。

[指定形式]

```
-Xuse_fmaf
```

- 省略時解釈

積和演算命令を生成しません。

[詳細説明]

- 単精度浮動小数点積和演算に対して、積和演算命令（fmaf.s, fmsf.s, fnmaf.s, fnmsf.s）を生成します。
- 本オプションを指定すると、実行速度は上がりますが、演算精度が変わります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[出力コード] カテゴリの[\[積和演算命令を生成する\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[出力コード] カテゴリの[\[積和演算命令を生成する\]](#)

[使用例]

- 単精度浮動小数点積和演算に対して、積和演算命令を生成します。

```
>ccrh -Xuse_fmaf -Xcommon=v850e3v5 main.c
```

-Xunordered_cmpf

浮動小数点比較において、無効演算例外の検出を行います。

[指定形式]

```
-Xunordered_cmpf
```

- 省略時解釈

浮動小数点比較において、無効演算例外の検出を行いません。

[詳細説明]

- 浮動小数点比較において、比較する値に非数が含まれている場合に無効演算例外を発生する比較条件を使用するコードを生成します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の[\[出力コード\]](#)カテゴリの[\[浮動小数点比較で無効演算例外を発生させる\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[\[出力コード\]](#)カテゴリの[\[浮動小数点比較で無効演算例外を発生させる\]](#)

[使用例]

- 浮動小数点比較において、無効演算例外の検出を行います。

```
>ccrh -Xunordered_cmpf -Xcommon=v850e3v5 main.c
```

-Xmulti_level

マルチコア用プログラムの生成を指定します。

[指定形式]

```
-Xmulti_level=level
```

- 省略時解釈

シングルコア用プログラムを生成します。

[詳細説明]

- 指定したコア用のプログラムを生成します。
- *level* に指定可能なものを以下に示します。
これ以外のものを指定した場合は、エラーとなります。

0	シングルコア用プログラムを生成します。 プログラム中の #pragma pmodule 指定は無視します。
1	マルチコア用プログラムを生成します。 プログラム中の #pragma pmodule 指定は有効となります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [\[出力コード\]](#) カテゴリの [\[生成するプログラムの種類\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [\[出力コード\]](#) カテゴリの [\[生成するプログラムの種類\]](#)

[使用例]

- マルチコア用プログラムを生成します。

```
>ccrh -Xmulti_level=1 -Xcommon=v850e3v5 file1.c file2.c
```

-Xpatch

パッチを適用します。

[指定形式]

```
-Xpatch=string
```

- 省略時解釈

パッチを適用しません。

[詳細説明]

- *string* に指定可能なものは、“dw_access” です。

これ以外のものを指定した場合は、エラーとなります。

- -Xpatch=dw_access オプションを指定した場合、ld.dw/st.dw 命令の生成を抑止したコードを生成します。

[使用例]

- ld.dw/st.dw 命令の生成を抑止したコードを生成します。

```
>ccrh -Xpatch=dw_access -Xcommon=v850e3v5 main.c
```

情報ファイル出力制御

情報ファイル出力制御オプションには、次のものがあります。

- [-Xcref](#)

-Xcref

静的解析情報ファイルを出力します。

[指定形式]

```
-Xcref=path
```

- 省略時解釈

静的解析情報ファイルを出力しません。

[詳細説明]

- コンパイル途中に生成される静的解析情報ファイルの保存先を *path* に指定します。
- *path* に存在するフォルダ名を指定した場合は、フォルダ *path* に、C ソース・ファイル名の拡張子を “.cref” で置き換えたファイル名で静的解析情報ファイルを保存します。
- *path* に存在するファイル名を指定した場合、および存在しないフォルダ名、またはファイル名を指定した場合は、出力する静的解析情報ファイルが1つの場合は、*path* というファイル名で保存します。
出力する静的解析情報ファイルが複数の場合は、エラーとなります。
- *=path* を省略した場合は、エラーとなります。
- ソース・ファイルとして同じ名前のファイル（異なるフォルダにある場合を含む）を複数指定した場合は、警告を出力して、最後に指定したソース・ファイルに対する静的解析情報ファイルのみを保存します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\]](#) タブの [出力ファイルの種類と場所] カテゴリの [\[クロス・リファレンス情報を出力する\]](#)

[使用例]

- 静的解析情報ファイルをファイル名 info.cref で出力します。

```
>ccrh -Xcref=info.cref -Xcommon=v850e3v5 main.c
```


エラー出力制御

エラー出力制御オプションには、次のものがあります。

- [-Xerror_file](#)

-Xerror_file

エラー・メッセージをファイルに出力します。

[指定形式]

```
-Xerror_file=file
```

- 省略時解釈

エラー・メッセージを標準エラー出力のみに出力します。

[詳細説明]

- エラー・メッセージを標準エラー出力、およびファイル *file* に出力します。
- *file* がすでに存在する場合は、そのファイルを上書きします。
- *file* を省略した場合は、エラーとなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\]](#) タブの [エラー出力] カテゴリの [\[エラー・メッセージ・ファイルを出力する\]](#), [\[エラー・メッセージ・ファイル出力フォルダ\]](#), [\[エラー・メッセージ・ファイル名\]](#)
 - [\[個別コンパイル・オプション\]](#) タブの [エラー出力] カテゴリの [\[エラー・メッセージ・ファイルを出力する\]](#), [\[エラー・メッセージ・ファイル出力フォルダ\]](#), [\[エラー・メッセージ・ファイル名\]](#)

[使用例]

- エラー・メッセージを標準エラー出力、およびファイル *err* に出力します。

```
>ccrh -Xerror_file=err -Xcommon=v850e3v5 main.c
```

警告メッセージ出力制御

警告メッセージ出力制御オプションには、次のものがあります。

- [-Xno_warning](#)

-Xno_warning

指定した警告メッセージの出力を抑止します。

[指定形式]

```
-Xno_warning=num[,num] ...  
-Xno_warning=num1-num2
```

- 省略時解釈

すべての警告メッセージを出力します。

[詳細説明]

- 指定した警告メッセージの出力を抑止します。
- *num*, *num1*, *num2* には、メッセージ番号を指定します。
存在しないメッセージ番号を指定した場合は、無視します。
- *num*, または *num1*, および *num2* を省略した場合は、エラーとなります。
- *num1-num2* の形式で指定すると、その範囲に含まれるメッセージ番号を指定したものとみなします。
- 本オプションで指定するメッセージ番号は、W に続く 7 桁の数字のうち、下位 5 桁です。
メッセージ番号については、「CubeSuite+ 統合開発環境 ユーザーズマニュアル メッセージ編」を参照してください。
- 本オプションで制御することができるのは、コンパイラ、アセンブラの警告メッセージです（最適化リンカの警告メッセージは対象外です）。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\] タブ](#)の[警告メッセージ] カテゴリの[\[表示させない警告メッセージ\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の[警告メッセージ] カテゴリの[\[表示させない警告メッセージ\]](#)

[使用例]

- 警告メッセージ W0511146, W0511147 の出力を抑止します。

```
>ccrh -Xno_warning=11146,11147 -Xcommon=v850e3v5 main.c
```

フェーズ個別オプション指定

フェーズ個別オプション指定オプションには、次のものがあります。

- Xasm_option
- Xlk_option

-Xasm_option

アセンブル・オプションを指定します。

[指定形式]

```
-Xasm_option=arg
```

- 省略時解釈

指定したオプションはすべて ccrh のドライバが解釈します。

[詳細説明]

- arg をアセンブル・オプションとして、アセンブラに渡します。
- ccrh のドライバがアセンブリ・ソース・ファイルとして認識しない識別子を持つファイルをアセンブラに渡す際には、本オプションを使用します。
- arg が存在しないアセンブル・オプションである場合は、エラーとなります。
- arg を省略した場合は、エラーとなります。

[使用例]

- Xasm_far_jump オプションをアセンブラに渡します。

```
>ccrh -Xasm_option=-Xasm_far_jump -Xcommon=v850e3v5 main.c
```

上記の例は、以下と同じ意味となります。

```
>ccrh -S -Xcommon=v850e3v5 main.c
>asrh -Xasm_far_jump -Xcommon=v850e3v5 main.asm
```

-Xlk_option

リンク・オプションを指定します。

[指定形式]

```
-Xlk_option=arg
```

- 省略時解釈

指定したオプションはすべて ccrh のドライバが解釈します。

[詳細説明]

- *arg* をリンク・オプションとして、最適化リンカに渡します。
- ccrh のドライバが最適化リンカへの入力として認識しない識別子を持つファイルを最適化リンカに渡す際には、本オプションを使用します。
- *arg* が存在しないリンク・オプションである場合は、エラーとなります。
- *arg* を省略した場合は、エラーとなります。

[使用例]

- -form=relocate オプションを最適化リンカに渡します。

```
>ccrh -Xlk_option=-form=relocate -Xcommon=v850e3v5 main.c
```

上記の例は、以下と同じ意味となります。

```
>ccrh -c -Xcommon=v850e3v5 main.c  
>rlink -form=relocate main.obj
```

サブコマンド・ファイル指定

サブコマンド・ファイル指定オプションには、次のものがあります。

- @

@

サブコマンド・ファイルを指定します。

[指定形式]

```
@file
```

- 省略時解釈

コマンド・ラインで指定したオプション、およびファイル名のみを認識します。

[詳細説明]

- *file* をサブコマンド・ファイルとして扱います。
- *file* が存在しない場合は、エラーとなります。
- *file* を省略した場合は、エラーとなります。
- サブコマンド・ファイルについての詳細は、「[B. 4. 2 サブコマンド・ファイルの使用方法](#)」を参照してください。

[使用例]

- command.txt をサブコマンド・ファイルとして扱います。

```
>ccrh @command.txt -Xcommon=v850e3v5
```

B. 5.2 アセンブル・オプション

ここでは、アセンブル・フェーズのオプションについて説明します。

オプションに関する注意事項を以下に示します。

- オプションの大文字／小文字は区別します。
- パラメータとして数値を指定する場合は、10 進数、または“0x”（“0X”）で始まる 16 進数での指定が可能です。
- 16 進数のアルファベットは、大文字／小文字を区別しません。
- パラメータとしてファイル名を指定する場合は、パス付き（絶対パス、または相対パス）での指定が可能です。
- パスなし、および相対パスで指定する場合は、カレント・フォルダを基準とします。
- パラメータ中に空白を含める場合（パス名など）は、そのパラメータ全体をダブルクォーテーション（"）で囲んでください。
- ccrh コマンドに対して、-Xprn_path、または-Xasm_far_jump オプションを指定する場合は、-Xasm_option オプションを使用する必要があります。

オプションの分類と説明を以下に示します。

表 B—3 アセンブル・オプション

分類	オプション	説明
バージョン／ヘルプ表示指定	-V	asrh のバージョン情報を表示します。
	-h	asrh のオプションの説明を表示します。
出力ファイル指定	-o	出力ファイル名を指定します。
	-Xobj_path	アセンブル途中に生成されるオブジェクト・ファイルの保存先を指定します。
	-Xprn_path	アセンブル・リスト・ファイルの保存先を指定します。
ソース・デバッグ制御	-g	ソース・デバッグ用の情報を出力します。
デバイス指定	-Xcommon	デバイス共通のオブジェクト・ファイルを生成することを指定します。
	-Xcpu	指定したコア向けのオブジェクトを生成することを指定します。
プリプロセッサ制御	-D	アセンブラ・シンボルを定義します。
	-U	-D オプションによるアセンブラ・シンボルの定義を解除します。
	-I	インクルード・ファイルを検索するフォルダを指定します。
日本語／中国語文字列制御	-Xcharacter_set	日本語／中国語の文字コードを指定します。
生成コード制御	-Xreg_mode	レジスタ・モードを指定します。
	-Xreserve_r2	r2 レジスタを予約します。
	-Xep	ep レジスタの扱い方を指定します。

分類	オプション	説明
アセンブラ制御指定	-Xasm_far_jump	アセンブリ・ソース・ファイルに対して、far jump の出力を制御します。
エラー出力制御	-Xerror_file	エラー・メッセージをファイルに出力します。
警告メッセージ出力制御	-Xno_warning	指定した警告メッセージの出力を抑止します。
サブコマンド・ファイル指定	@	サブコマンド・ファイルを指定します。

バージョン／ヘルプ表示指定

バージョン／ヘルプ表示指定オプションには、次のものがあります。

-V

-h

-V

asrh のバージョン情報を表示します。

[指定形式]

```
-V
```

- 省略時解釈

asrh のバージョン情報を表示せずに、アセンブルを行います。

[詳細説明]

- asrh のバージョン情報を標準エラー出力に出力します。

アセンブルは行いません。

[使用例]

- asrh のバージョン情報を標準エラー出力に出力します。

```
>asrh -V -Xcommon=v850e3v5
```


-h

asrh のオプションの説明を表示します。

[指定形式]

```
-h
```

- 省略時解釈

asrh のオプションの説明を表示しません。

[詳細説明]

- asrh のオプションの説明を標準エラー出力に出力します。

アセンブルは行いません。

[使用例]

- asrh のオプションの説明を標準エラー出力に出力します。

```
>asrh -h -Xcommon=v850e3v5
```

出力ファイル指定

出力ファイル指定オプションには、次のものがあります。

- o
- Xobj_path
- Xprn_path

-o

出力ファイル名を指定します。

[指定形式]

```
-ofile
```

- 省略時解釈

カレント・フォルダにファイルを出力します。

出力オブジェクト・ファイル名は、ソース・ファイル名の拡張子を“.obj”に置き換えたものとなります。

[詳細説明]

- 出力オブジェクト・ファイル名を *file* に指定します。
- *file* がすでに存在する場合は、そのファイルを上書きします。
- 本オプションを指定しても、フェイタル・エラーがある場合には、オブジェクト・ファイルの出力は行いません。
- 出力ファイルが複数の場合は、エラーとなります。
- *file* を省略した場合は、エラーとなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[個別アセンブル・オプション\]](#) タブの [出力ファイル] カテゴリの [\[オブジェクト・ファイル名\]](#)

[使用例]

- オブジェクト・ファイルをファイル名 sample.obj で出力します。

```
>asrh -osample.obj -Xcommon=v850e3v5 main.asm
```

-Xobj_path

アセンブル途中に生成されるオブジェクト・ファイルを保存するフォルダを指定します。

[指定形式]

```
-Xobj_path[=path]
```

- 省略時解釈

カレント・フォルダに、ソース・ファイル名の拡張子を“.obj”で置き換えたファイル名でオブジェクト・ファイルを保存します。

[詳細説明]

- アセンブル途中に生成されるオブジェクト・ファイルを保存するフォルダを *path* に指定します。
- *path* に存在するフォルダ名を指定した場合は、フォルダ *path* に、ソース・ファイル名の拡張子を“.obj”で置き換えたファイル名でオブジェクト・ファイルを保存します。
存在しないフォルダ名を指定した場合は、エラーとなります。
- *path* には存在するファイル名を指定することも可能です。
出力するオブジェクト・ファイルが1つの場合は、*path* というファイル名で保存します。
出力するオブジェクト・ファイルが複数の場合は、エラーとなります。
存在しないファイル名を指定した場合は、エラーとなります。
- *=path* を省略した場合は、カレント・フォルダに、ソース・ファイル名の拡張子を“.obj”で置き換えたファイル名でオブジェクト・ファイルを保存します。
- ソース・ファイルとして同じ名前のファイル（異なるフォルダにある場合を含む）を複数指定した場合は、警告を出力して、最後に指定したソース・ファイルに対するオブジェクト・ファイルのみを保存します。

[使用例]

- アセンブル途中に生成されるオブジェクト・ファイルをフォルダ D:\sample に保存します。

```
>asrh -Xobj_path=D:\sample -Xcommon=v850e3v5 main.asm
```

-Xprn_path

アセンブル・リスト・ファイルを保存するフォルダを指定します。

[指定形式]

```
-Xprn_path[=path]
```

- 省略時解釈

アセンブル・リスト・ファイルを出しません。

[詳細説明]

- アセンブル時に出力するアセンブル・リスト・ファイルを保存するフォルダを *path* に指定します。
- *path* に存在するフォルダ名を指定した場合は、フォルダ *path* に、ソース・ファイル名の拡張子を “.prn” で置き換えたファイル名でアセンブル・リスト・ファイルを保存します。
存在しないフォルダ名を指定した場合は、エラーとなります。
- *path* には存在するファイル名を指定することも可能です。
- *path* というファイル名でアセンブル・リスト・ファイルを保存します。
存在しないファイル名を指定した場合は、エラーとなります。
- *=path* を省略した場合は、カレント・フォルダに、ソース・ファイル名の拡張子を “.prn” で置き換えたファイル名でアセンブル・リスト・ファイルを保存します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [\[アセンブル・リスト\]](#) カテゴリの [\[アセンブル・リスト・ファイル出力する\]](#)、[\[アセンブル・リスト・ファイル出力フォルダ\]](#)
 - [\[アセンブル・オプション\] タブ](#)の [\[アセンブル・リスト\]](#) カテゴリの [\[アセンブル・リスト・ファイル出力する\]](#)、[\[アセンブル・リスト・ファイル出力フォルダ\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [\[アセンブル・リスト\]](#) カテゴリの [\[アセンブル・リスト・ファイル出力する\]](#)、[\[アセンブル・リスト・ファイル出力フォルダ\]](#)
 - [\[個別アセンブル・オプション\] タブ](#)の [\[アセンブル・リスト\]](#) カテゴリの [\[アセンブル・リスト・ファイル出力する\]](#)、[\[アセンブル・リスト・ファイル出力フォルダ\]](#)

[使用例]

- アセンブル時に出力するアセンブル・リスト・ファイルをフォルダ D:\sample に保存します。

```
>asrh -Xprn_path=D:\sample -Xcommon=v850e3v5 main.asm
```

ソース・デバッグ制御

ソース・デバッグ制御オプションには、次のものがあります。

- [-g](#)

-g

ソース・デバッグ用の情報を出力します。

[指定形式]

```
-g
```

- 省略時解釈

ソース・デバッグ用の情報を出力しません。

[詳細説明]

- ソース・デバッグ用の情報を出力ファイル中に出力します。
- 本オプションを指定することにより、ソース・デバッグが可能となります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[アセンブル・オプション\] タブ](#)の [デバッグ情報] カテゴリの [\[デバッグ情報を生成する\]](#)
 - [\[個別アセンブル・オプション\] タブ](#)の [デバッグ情報] カテゴリの [\[デバッグ情報を生成する\]](#)

[使用例]

- ソース・デバッグ用の情報を出力ファイル中に出力します。

```
>asrh -g -Xcommon=v850e3v5 main.asm
```

デバイス指定

デバイス指定オプションには、次のものがあります。

- Xcommon
- Xcpu

-Xcommon

デバイス共通のオブジェクト・ファイルを生成することを指定します。

[指定形式]

```
-Xcommon=series
```

- 省略時解釈

エラーとなります。

ただし、-V/-h オプションを指定している場合は、エラーとなりません。

[詳細説明]

- デバイス共通のオブジェクト・ファイルを生成することを指定します。
- 本オプションを指定した場合、ターゲットの命令セット・アーキテクチャの命令のみを使用し、また、命令セット・アーキテクチャに対応した共通のマジック・ナンバ *series* をオブジェクト・ファイルに埋め込みます。
- *series* に指定可能なものを以下に示します。
これ以外のものを指定した場合は、エラーとなります。

v850e3v5	命令セット・アーキテクチャが V850E3V5 である品種をターゲット・デバイスとして指定したもののリンクが可能です。
rh850	“v850e3v5” を指定した場合と同じです。

- *series* を省略した場合は、エラーとなります。

[使用例]

- 生成するオブジェクト・ファイルに、命令セット・アーキテクチャが V850E3V5 である品種に共通のマジック・ナンバを埋め込みます。

```
>asrh -Xcommon=v850e3v5 -c main.asm
```

-Xcpu

指定したコア向けのオブジェクトを生成することを指定します。

[指定形式]

```
-Xcpu=core
```

- 省略時解釈

-Xcommon オプションを指定した場合、G3M 向けオブジェクトを生成します。

[詳細説明]

- コア *core* 向けのオブジェクトを生成することを指定します。

- *core* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

g3m	G3M 向けオブジェクトを生成します。
g3k	G3K 向けオブジェクトを生成します。

- *core* を省略した場合は、エラーとなります。

[使用例]

- G3M 向けオブジェクトを生成することを指定します。

```
>asrh -Xcpu=g3m -c -Xcommon=v850e3v5 main.asm
```

プリプロセッサ制御

プリプロセッサ制御オプションには、次のものがあります。

- D
- U
- I

-D

アセンブラ・シンボルを定義します。

[指定形式]

```
-Dname [=def] [name [=def]] ...
```

- 省略時解釈
なし

[詳細説明]

- アセンブラ・シンボルとして *name* を定義します。
- アセンブリ・ソース・プログラムの前に、*name* .SET *def* を記述するのと同様です。
- *name* が、アセンブラ・シンボルには使用可能であるがプリプロセッサ・マクロには使用できない文字（@, ., ~）を含む場合、警告を出力してアセンブラ・シンボルとしてのみ定義します。
- *name* を省略した場合は、エラーとなります。
- *=def* を省略した場合は、*def* は1とみなします。
- 本オプションは、複数指定が可能です。
- 同じアセンブラ・シンボルに対して、本オプションと-Uオプションを同時に指定した場合は、あとから指定したものが有効となります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\] タブ](#)の [\[よく使うオプション \(アセンブル\)\]](#) カテゴリの [\[定義マクロ\]](#)
 - [\[アセンブル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[定義マクロ\]](#)
 - [\[個別アセンブル・オプション\] タブ](#)の [\[プリプロセス\]](#) カテゴリの [\[定義マクロ\]](#)

[使用例]

- アセンブラ・シンボルとして sample=256 を定義します。

```
>asrh -Dsample=256 -Xcommon=v850e3v5 main.asm
```


-U

-D オプションによるアセンブラ・シンボルの定義を解除します。

[指定形式]

```
-Uname[, name]...
```

- 省略時解釈
なし

[詳細説明]

- -D オプションによるアセンブラ・シンボル *name* の定義を解除します。
- *name* を省略した場合は、エラーとなります。
- 本オプションでは、*name* .SET *def* の記述による定義は解除できません。
- 本オプションは、複数指定が可能です。
- 同じアセンブラ・シンボルに対して、本オプションと -D オプションを同時に指定した場合は、あとから指定したものが有効となります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[アセンブル・オプション\] タブ](#)の [プリプロセス] カテゴリの [\[定義解除マクロ\]](#)
 - [\[個別アセンブル・オプション\] タブ](#)の [プリプロセス] カテゴリの [\[定義解除マクロ\]](#)

[使用例]

- -D オプションによるアセンブラ・シンボル *test* の定義を解除します。

```
>asrh -Utest -Xcommon=v850e3v5 main.asm
```

-I

インクルード・ファイルを検索するフォルダを指定します。

[指定形式]

```
-Ipath[,path]...
```

- 省略時解釈

インクルード・ファイルを 標準インクルード・ファイル・フォルダからのみ検索します。

[詳細説明]

- アセンブラ制御命令 \$INCLUDE/\$BINCLUDE で読み込むインクルード・ファイルを検索するフォルダを *path* に指定します。

インクルード・ファイルの検索は、以下の順番で行います。

- (1) -I オプションで指定したフォルダ
- (2) ソース・ファイルのあるフォルダ
- (3) 元の C ソース・ファイルのあるフォルダ
- (4) カレント・フォルダ

- *path* が存在しない場合は、警告を出力します。

- *path* を省略した場合は、エラーとなります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[共通オプション\]](#) タブの [\[よく使うオプション \(アセンブル\)\]](#) カテゴリの [\[追加のインクルード・パス\]](#), [\[システム・インクルード・パス\]](#)
- [\[アセンブル・オプション\]](#) タブの [\[プリプロセス\]](#) カテゴリの [\[追加のインクルード・パス\]](#), [\[システム・インクルード・パス\]](#)
- [\[個別アセンブル・オプション\]](#) タブの [\[プリプロセス\]](#) カテゴリの [\[追加のインクルード・パス\]](#), [\[ビルド・ツールに指定した全体インクルード・パスも使用する\]](#)

[使用例]

- インクルード・ファイルをフォルダ D:\include, D:\src, カレント・フォルダの順で検索します。

```
>asrh -ID:\include -Xcommon=v850e3v5 D:\src\main.asm
```

日本語／中国語文字列制御

日本語／中国語文字列制御オプションには、次のものがあります。

- [-Xcharacter_set](#)

-Xcharacter_set

日本語／中国語の文字コードを指定します。

[指定形式]

```
-Xcharacter_set=code
```

- 省略時解釈

日本語の文字コードを SJIS として扱います。

[詳細説明]

- ソース・ファイル中の日本語／中国語のコメント、文字列に対して、使用する文字コードを指定します。

- *code* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

なお、ソース・ファイル中で使用している文字コードと異なるものを指定した場合、動作は保証されません。

none	日本語／中国語の文字コードを解釈しません
euc_jp	EUC（日本語）
sjis	SJIS
utf8	UTF-8
big5	繁体字中国語
gb2312	簡体字中国語

- *code* を省略した場合は、エラーとなります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[アセンブル・オプション\] タブ](#)の「文字コード」カテゴリの「[文字コード](#)」
- [\[個別アセンブル・オプション\] タブ](#)の「文字コード」カテゴリの「[文字コード](#)」

[使用例]

- ソース・ファイル中の日本語のコメント、文字列に対して、使用する文字コードに EUC を指定します。

```
>asrh -Xcharacter_set=euc_jp -Xcommon=v850e3v5 main.asm
```

生成コード制御

生成コード制御オプションには、次のものがあります。

- Xreg_mode
- Xreserve_r2
- Xep

-Xreg_mode

レジスタ・モードを指定します。

[指定形式]

```
-Xreg_mode=mode
```

- 省略時解釈

32 レジスタ・モードのオブジェクト・ファイルを生成します。

[詳細説明]

- 指定したレジスタ・モードのオブジェクト・ファイルを生成します。
- ccrh が使用するレジスタを 32 本（32 レジスタ・モード）、22 本（22 レジスタ・モード、または common レジスタ・モード）のいずれかに制限し、オブジェクト・ファイル内にレジスタ・モードを示すマジック・ナンバを埋め込みます。
- common レジスタ・モードは、レジスタ・モードに依存しないオブジェクト・ファイルを生成するために使用します。
- *mode* に指定可能なものを以下に示します。
これ以外のものを指定した場合は、エラーとなります。

レジスタ・モード (<i>mode</i>)	作業用レジスタ	レジスタ変数用レジスタ
common	r10 ~ r14	r25 ~ r29
22	r10 ~ r14	r25 ~ r29
32	r10 ~ r19	r20 ~ r29

- *mode* を省略した場合は、エラーとなります。
- アセンブリ・ソース・ファイルに対して、使用可能なレジスタ以外のレジスタが使用されている場合は、エラーとなります。
- 本オプションは、すべてのソース・ファイルに対して同じレジスタ・モードを指定する必要があります。
ソース・ファイルごとに指定を変えることはできません。
レジスタ・モードが異なるオブジェクト・ファイルが混在している場合は、リンク時にエラーとなります。

- 本オプションを指定することにより、ソフトウェア・レジスタ・バンク機能のレジスタ・モードを切り替えることができます。

【使用例】

- 22 レジスタ・モードのオブジェクト・ファイルを生成します。

```
>asrh -Xreg_mode=22 -Xcommon=v850e3v5 main.asm
```

-Xreserve_r2

r2 レジスタを予約します。

[指定形式]

```
-Xreserve_r2
```

- 省略時解釈

r2 レジスタの予約を行わずに、コンパイラで使します。

[詳細説明]

- r2 レジスタを予約し、コンパイラでは r2 レジスタを使用しないコードを生成します。

[使用例]

- r2 レジスタを予約し、コンパイラでは r2 レジスタを使用しないコードを生成します。

```
>asrh -Xreserve_r2 -Xcommon=v850e3v5 main.asm
```

-Xep

ep レジスタの扱い方を指定します。

[指定形式]

```
-Xep=mode
```

- 省略時解釈
ep レジスタを関数呼び出し前後で値を保証するレジスタとして扱います。

[詳細説明]

- ep レジスタの扱い方を指定します。
- mode に指定可能なものを以下に示します。
これ以外のものを指定した場合は、エラーとなります。

fix	ep レジスタの値をプロジェクト全体で固定します。 プロジェクト内の #pragma section 指令において、以下の属性指定文字を使用した場合は、本パラメータを指定してください。 ep_auto, ep_disp4, ep_disp5, ep_disp7, ep_disp8, ep_disp16, ep_disp23
callee	ep レジスタを関数呼び出し前後で値を保証するレジスタとして扱います。 -Omap, または -Osmap オプションを指定した場合は、本パラメータを指定してください。

- mode を省略した場合は、エラーとなります。

[使用例]

- ep レジスタの値をプロジェクト全体で固定します。

```
>asrh -Xep=fix -Xcommon=v850e3v5 main.asm
```

アセンブラ制御指定

アセンブラ制御指定オプションには、次のものがあります。

-Xasm_far_jump

-Xasm_far_jump

アセンブリ・ソース・ファイルに対して、far jump の出力を制御します。

[指定形式]

-Xasm_far_jump

- 省略時解釈

jarl, または jr 命令としてアセンブルを行います。

[詳細説明]

- アセンブリ・ソース・ファイルに対して、ソース中に記述されたすべての jarl, および jr 命令を jarl32, および jr32 命令とみなしてアセンブルを行います。
- 個別の命令ごとに制御したい場合は、ソース中で jarl22/jarl32, jr22/jr32 のように明記します。
- 本オプションは、jump 命令には影響しません。
- 本オプションは、C ソース・ファイルに対しては無効です。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[コンパイル・オプション\] タブ](#)の [\[出力コード\]](#) カテゴリの [\[32 ビット分岐命令を使用する\]](#)
 - [\[アセンブル・オプション\] タブ](#)の [\[出力コード\]](#) カテゴリの [\[32 ビット分岐命令を使用する\]](#)
 - [\[個別コンパイル・オプション\] タブ](#)の [\[出力コード\]](#) カテゴリの [\[32 ビット分岐命令を使用する\]](#)
 - [\[個別アセンブル・オプション\] タブ](#)の [\[出力コード\]](#) カテゴリの [\[32 ビット分岐命令を使用する\]](#)

[使用例]

- アセンブリ・ソース中に記述されたすべての jarl, および jr 命令を jarl32, および jr32 命令とみなしてアセンブルを行います。

```
>asrh -Xasm_far_jump -Xcommon=v850e3v5 main.asm
```


エラー出力制御

エラー出力制御オプションには、次のものがあります。

- [-Xerror_file](#)

-Xerror_file

エラー・メッセージをファイルに出力します。

[指定形式]

```
-Xerror_file=file
```

- 省略時解釈

エラー・メッセージを標準エラー出力のみに出力します。

[詳細説明]

- エラー・メッセージを標準エラー出力、およびファイル *file* に出力します。
- *file* がすでに存在する場合は、そのファイルを上書きします。
- *file* を省略した場合は、エラーとなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[個別アセンブル・オプション\]](#) タブの [エラー出力] カテゴリの [\[エラー・メッセージ・ファイルを出力する\]](#), [\[エラー・メッセージ・ファイル出力フォルダ\]](#), [\[エラー・メッセージ・ファイル名\]](#)

[使用例]

- エラー・メッセージを標準エラー出力、およびファイル *err* に出力します。

```
>asrh -Xerror_file=err -Xcommon=v850e3v5 main.asm
```

警告メッセージ出力制御

警告メッセージ出力制御オプションには、次のものがあります。

- [-Xno_warning](#)

-Xno_warning

指定した警告メッセージの出力を抑止します。

[指定形式]

```
-Xno_warning=num[,num] ...  
-Xno_warning=num1-num2
```

- 省略時解釈

すべての警告メッセージを出力します。

[詳細説明]

- 指定した警告メッセージの出力を抑止します。
- *num*, *num1*, *num2* には、メッセージ番号を指定します。
存在しないメッセージ番号を指定した場合は、無視します。
- *num*, または *num1*, および *num2* を省略した場合は、エラーとなります。
- *num1-num2* の形式で指定すると、その範囲に含まれるメッセージ番号を指定したものとみなします。
- 本オプションで指定するメッセージ番号は、W に続く 7 桁の数字のうち、下位 5 桁です。
メッセージ番号については、「CubeSuite+ 統合開発環境 ユーザーズマニュアル メッセージ編」を参照してください。
- 本オプションで制御することができるのは、コンパイラ、アセンブラの警告メッセージです（最適化リンカの警告メッセージは対象外です）。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[個別アセンブル・オプション\]](#) タブの [警告メッセージ] カテゴリの [\[表示させない警告メッセージ\]](#)

[使用例]

- 警告メッセージ W0550002, W0550003 の出力を抑止します。

```
>asrh -Xno_warning=50002,50003 -Xcommon=v850e3v5 main.asm
```

サブコマンド・ファイル指定

サブコマンド・ファイル指定オプションには、次のものがあります。

- @

@

サブコマンド・ファイルを指定します。

[指定形式]

```
@file
```

- 省略時解釈

コマンド・ラインで指定したオプション、およびファイル名のみを認識します。

[詳細説明]

- *file* をサブコマンド・ファイルとして扱います。
- *file* が存在しない場合は、エラーとなります。
- *file* を省略した場合は、エラーとなります。
- サブコマンド・ファイルについての詳細は、「[B. 4. 2 サブコマンド・ファイルの使用方法](#)」を参照してください。

[使用例]

- command.txt をサブコマンド・ファイルとして扱います。

```
>asrh @command.txt -Xcommon=v850e3v5
```

B.5.3 リンク・オプション

ここでは、リンク・フェーズのオプションについて説明します。

オプションに関する注意事項を以下に示します。

- オプションの大文字／小文字は区別しません。
- オプション、およびパラメータの大文字は、短縮形の指定が可能であることを表しています。
大文字以降の文字の指定は任意です。

例 -FOrm=Absolute の場合、例えば以下のように指定することも可能です。

```
-fo=a
-fo=abs
-for=absolu
```

- パラメータとしてファイル名を指定する場合、“(”、および “)” を使用することはできません。
- ccrh コマンドに対して、リンク・オプションを指定する場合は、-Xlk_option オプションを使用する必要があります。

オプションの分類と説明を以下に示します。

表 B—4 リンク・オプション

分類	オプション	説明
入力制御	-Input	入力ファイルを指定します。
	-LIBrary	入カライブラリ・ファイルを指定します。
	-Binary	入カバイナリ・ファイルを指定します。
	-DEFine	未定義シンボルを強制定義します。
	-ENTry	実行開始アドレスを指定します。

分類	オプション	説明
出力制御	-FOrM	出力形式を指定します。
	-DEBug	出力ファイル中にデバッグ情報を出力します。
	-NODeBug	デバッグ情報を出力しません。
	-RECOrd	出力するデータ・レコードのサイズを指定します。
	-ROm	ROM から RAM へマップするセクションを指定します。
	-OUtput	出力ファイルを指定します。
	-MAp	外部シンボル割り付け情報ファイルを出力します。
	-SPace	出力範囲のメモリの空き領域を充てんします。
	-Message	インフォメーション・メッセージを出力します。
	-NOMessage	インフォメーション・メッセージの出力を抑止します。
	-MSg_unused	参照されない外部定義シンボルをユーザに通知します。
	-BYte_count	データ・レコードのバイト数の最大値を指定します。
	-PADDING	セクションの終端にデータを埋め込みます。
	-OVERRUN_FETCH	オーバーラン・フェッチに伴う空き領域の読み出しを回避します。
リスト出力	-LISt	リスト・ファイルを出力します。
	-SHow	リスト・ファイルへの出力情報を指定します。
セクション指定	-STARt	セクションの開始アドレスを指定します。
	-FSymbol	外部定義シンボルをシンボル・アドレス・ファイルに出力します。
	-ALIGNED_SECTION	セクションのアライメント数を 16 バイトに変更します。
ベリファイ指定	-CPu	セクションの割り付けアドレスの整合性をチェックします。
サブコマンド・ファイル指定	-SUBcommand	オプションをサブコマンド・ファイルで指定します。

分類	オプション	説明
その他	-S9	S9 レコードを終端に出力します。
	-STAck	スタック情報ファイルを出力します。
	-COmpress	デバッグ情報を圧縮します。
	-NOCOpmpress	デバッグ情報を圧縮しません。
	-MEMory	リンク時に使用するメモリ量を指定します。
	-REName	外部シンボル名、セクション名を変更します。
	-DELeTe	外部シンボル名、またはライブラリ・モジュールを削除します。
	-REPlace	ライブラリ・モジュールを置換します。
	-EXTract	ライブラリ・モジュールを抽出します。
	-STRip	ロード・モジュール・ファイル、ライブラリ・ファイルのデバッグ情報を削除します。
	-CHange_message	インフォメーション、ワーニング、エラーのメッセージ種別を変更します。
	-Hide	出力ファイル内のローカル・シンボル名情報を消去します。
	-Total_size	リンク後の合計セクション・サイズを標準エラー出力に表示します。
	-LOgo	コピーライトを出力します。
	-NOLOgo	コピーライトの出力を抑止します。
	-END	本オプションより前に指定したオプション列を実行します。
	-EXIt	オプション指定の終了を指定します。

入力制御

入力制御オプションには、次のものがあります。

- Input
- Library
- Binary
- DEFine
- ENTry

-Input

入力ファイルを指定します。

[指定形式]

```
-Input=file[(module[,module]...)] [{, | Δ }file[(module[,module]...)]...
```

- 省略時解釈
なし

[詳細説明]

- 入力ファイル *file* を指定します。
複数指定する場合は、カンマ (,), または空白文字で区切ります。
- ワイルドカード (*, ?) も使用可能です。
ワイルドカードで指定した文字列は、アルファベット順に展開します。
数字と英文字では数字を先に、英大文字と英小文字では英大文字を先に展開します。
- 入力ファイルとして指定できるのは、コンパイラ、またはアセンブラが出力したオブジェクト・ファイル、最適化リンカが出力したリロケートブル・ファイル、ロード・モジュール・ファイル、インテル拡張ヘキサ・ファイル、およびモトローラ・Sタイプ・ファイルです。
また、“*library(module)*” の形式で、ライブラリ内モジュールを指定することもできます。
モジュール名は拡張子なしで指定します。
- 入力ファイル名に拡張子の指定がないとき、モジュール名の指定がない場合は “.obj”, モジュール名の指定がある場合は “.lib” を指定したものとみなします。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\]](#) タブの [入力ファイル] カテゴリの [\[オブジェクト・ファイル\]](#)
 - [\[ライブラリ生成オプション\]](#) タブの [入力ファイル] カテゴリの [\[オブジェクト・ファイル\]](#)

[注意]

- 本オプションは、サブコマンド・ファイル内のみで使用することができます。
- 本オプションをコマンド・ライン上で指定した場合は、エラーとなります。
- コマンド・ライン上で入力ファイルを指定する場合は、-input オプションなしで指定してください。

[使用例]

- a.obj と lib1.lib 内のモジュール e を入力します。

<コマンド・ライン>

```
>rlink -subcommand=sub.txt
```

<サブコマンド・ファイル sub.txt >

```
-input=a.obj lib1(e)
```

- “c” で始まり、拡張子が “.obj” であるファイルをすべて入力します。

<コマンド・ライン>

```
>rlink -subcommand=sub.txt
```

<サブコマンド・ファイル sub.txt >

```
-input=c*.obj
```

[備考]

- 本オプションは、-form=object、および -extract オプションを指定した場合は無効となります。
 - 入力ファイルにインテル拡張ヘキサ・ファイルを指定した場合は -form=hexadecimal オプション、モトローラ・S タイプ・ファイルを指定した場合は -form=stype オプションのみを指定することができます。
- 出力ファイル名を指定していない場合は、“先頭入力ファイル名_combine.拡張子” となります（入力ファイルが a.mot の場合、出力ファイルは a_combine.mot となります）。

-LIBrary

入カライブラリ・ファイルを指定します。

[指定形式]

```
-LIBrary=file[,file]...
```

- 省略時解釈
なし

[詳細説明]

- 入カライブラリ・ファイル *file* を指定します。
複数指定する場合は、カンマ (,) で区切ります。
- ワイルドカード (*, ?) も使用可能です。
ワイルドカードで指定した文字列は、アルファベット順に展開します。
数字と英文字では数字を先に、英大文字と英小文字では英大文字を先に展開します。
- 入カファイル名に拡張子の指定がない場合は “.lib” を指定したものとみなします。
- `-form=library`, または `-extract` オプションと同時に指定した場合は、指定したライブラリ・ファイルを編集対象ライブラリとして入力します。
それ以外の場合は、入カファイルとして指定されたファイル間でのリンク処理後に、未定義シンボルをライブラリ・ファイルから検索します。
- ライブラリ・ファイル内シンボルの検索は、以下の順序で行います。
 - 本オプションで指定したユーザ・ライブラリ・ファイル（指定順）
 - 本オプションで指定したシステム・ライブラリ・ファイル（指定順）
 - デフォルト・ライブラリ（環境変数 `HLNK_LIBRARY1`, `HLNK_LIBRARY2`, `HLNK_LIBRARY3`^注の順）

注 環境変数についての詳細は、「[B.3 環境変数](#)」を参照してください。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\] タブ](#)の [\[よく使うオプション \(リンク\)\]](#) カテゴリの [\[使用するライブラリ・ファイル\]](#), [\[標準ライブラリを使用する\]](#), [\[標準ライブラリ関数を使用する\]](#), [\[数学ライブラリ \(倍精度\) 関数を使用する\]](#), [\[数学ライブラリ \(単精度\) 関数を使用する\]](#), [\[非局所分岐関数を使用する\]](#), [\[よく使うオプション \(ライブラリ生成\)\]](#) カテゴリの [\[標準ライブラリを使用する\]](#), [\[標準ライブラリ関数を使用する\]](#), [\[数学ライブラリ \(倍精度\) 関数を使用する\]](#), [\[数学ライブラリ \(単精度\) 関数を使用する\]](#), [\[非局所分岐関数を使用する\]](#)
 - [\[リンク・オプション\] タブ](#)の [\[ライブラリ\]](#) カテゴリの [\[使用するライブラリ・ファイル\]](#), [\[システム・ライブラリ・ファイル\]](#), [\[標準ライブラリを使用する\]](#), [\[標準ライブラリ関数を使用する\]](#), [\[数学ライブラリ \(倍精度\) 関数を使用する\]](#), [\[数学ライブラリ \(単精度\) 関数を使用する\]](#), [\[非局所分岐関数を使用する\]](#)

- [ライブラリ生成オプション] タブの [ライブラリ] カテゴリの [使用するライブラリ・ファイル], [システム・ライブラリ・ファイル], [標準ライブラリを使用する], [標準ライブラリ関数を使用する], [数学ライブラリ (倍精度) 関数を使用する], [数学ライブラリ (単精度) 関数を使用する], [非局所分岐関数を使用する]

[使用例]

- a.lib と b.lib を入力します。

```
rlink main.obj -library=a.lib,b
```

- “c” で始まり、拡張子が “.lib” であるファイルをすべて入力します。

```
rlink main.obj -library=c*.lib
```

-Binary

入力バイナリ・ファイルを指定します。

[指定形式]

```
-Binary=file(section[:alignment] [/attribute] [,symbol]) [,file(section[:alignment] [/attribute] [,symbol])] ...
```

- 省略時解釈
なし

[詳細説明]

- 入力バイナリ・ファイル *file* を指定します。
複数指定する場合は、カンマ (,) で区切ります。
- 入力ファイル名に拡張子の指定がない場合は “.bin” を指定したものとみなします。
- 入力したバイナリ・データは、指定したセクション *section* のデータとして配置します。
セクションのアドレスは -start オプションで指定します。
section を省略した場合は、エラーとなります。
- シンボル *symbol* を指定すると、定義シンボルとしてリンクすることもできます。
C プログラムで参照している変数名の場合、プログラム中での参照名の先頭に “_” を付加します。
- 本オプションで指定したセクションには、セクション属性、アライメント数の指定が可能です。
- セクション属性 *attribute* に指定可能なものは、CODE、または DATA です。
attribute を省略した場合は、デフォルトとして、書き込み、読み取り、実行すべての属性が有効になります。
- アライメント数 *alignment* に指定可能な値は 2 の累乗 (1, 2, 4, 8, 16, 32) です。
それ以外の値を指定することはできません。
alignment を省略した場合は、デフォルトとして、1 が有効となります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#) の [入力ファイル] カテゴリの [\[バイナリ・ファイル\]](#)
 - [\[ライブラリ生成オプション\] タブ](#) の [入力ファイル] カテゴリの [\[バイナリ・ファイル\]](#)

[使用例]

- b.bin を D1bin セクションとして、0x200 番地から配置します。
- c.bin を D2bin セクション (アライメント数 4) として、D1bin の後に配置します。
- c.bin データを定義シンボル *_datab* としてリンクします。

```
>rlink a.obj -start=D*/200 -binary=b.bin(D1bin),c.bin(D2bin:4,_datab)
```

【備考】

- 本オプションは、-form={object|library} オプション、または -strip オプションを指定した場合は無効となります。
- 入力オブジェクト・ファイルを指定していない場合、本オプションは指定することができません。

-DEFine

未定義シンボルを強制定義します。

[指定形式]

```
-DEFine=symbol1={symbol2|value}[,symbol1={symbol2|value}]...
```

- 省略時解釈
なし

[詳細説明]

- 未定義シンボル *symbol1* を外部定義シンボル *symbol2*, または数値 *value* で強制定義します。
- *value* は 16 進数で指定します。
先頭が A ~ F の場合は、先にシンボルを検索し、該当するシンボルがなければ数値と解釈します。
先頭が 0 の場合は、常に数値と解釈します。
- シンボル名が C 変数名の場合は、プログラム中での定義名の先頭に “_” を付加します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\]](#) タブの [入力ファイル] カテゴリの [\[シンボル定義\]](#)

[使用例]

- *_sym1* を外部定義シンボル *data* と同値として定義します。

```
>rlink -define=_sym1=data a.obj b.obj
```

- *_sym2* を 0x4000 として定義します。

```
>rlink -define=_sym2=4000 a.obj b.obj
```

[備考]

- 本オプションは、`-form={object|relocate|library}` オプションを指定した場合は無効となります。

-ENTry

実行開始アドレスを指定します。

[指定形式]

```
-ENTry={symbol|address}
```

- 省略時解釈
なし

[詳細説明]

- 実行開始アドレスを外部定義シンボル *symbol*、またはアドレス *address* で指定します。
- *address* は 16 進数で指定します。
先頭が A ~ F の場合は、先に定義シンボルを検索し、該当するシンボルがなければアドレスと解釈します。
先頭が 0 の場合は、常にアドレスと解釈します。
- シンボル名が C 変数名の場合は、プログラム中での定義名の先頭に “_” を付加します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の [\[出力コード\]](#) カテゴリの [\[実行開始アドレスを指定する\]](#)、[\[実行開始アドレス\]](#)

[使用例]

- C の main 関数を実行開始アドレスとして指定します。

```
>rlink -entry=_main a.obj b.obj
```

- 0x100 を実行開始アドレスとして指定します。

```
>rlink -entry=100 a.obj b.obj
```

[備考]

- 本オプションは、`-form={object|relocate|library}` オプション、または `-strip` オプションを指定した場合は無効となります。

出力制御

出力制御オプションには、次のものがあります。

- -Form
- -DEBug
- -NODEBug
- -RECord
- -ROm
- -OUtput
- -MAp
- -SPace
- -Message
- -NOMessage
- -MSg_unused
- -BYte_count
- -PADDING
- -OVERRUN_FETCH

-Form

出力形式を指定します。

[指定形式]

```
-Form={format}
```

- 省略時解釈

ロード・モジュール・ファイルを出力します（-form=absolute オプションの指定と同じです）。

[詳細説明]

- 出力形式 *format* を指定します。
- *format* に指定可能なものを以下に示します。

Absolute	ロード・モジュール・ファイルを出力します。
Relocate	リロケータブル・ファイルを出力します。
Object	オブジェクト・ファイルを出力します。 -extract オプションでライブラリから1個のモジュールをオブジェクト・ファイルとして取り出すときに使用します。
Library[={S U}]	ライブラリ・ファイルを出力します。 library=s を指定した場合は、出力ファイルをシステム・ライブラリ・ファイルとします。 library=u を指定した場合は、出力ファイルをユーザ・ライブラリ・ファイルとします。 library のみを指定した場合は、library=u を指定したものとみなします。
Hexadecimal	インテル拡張ヘキサ・ファイルを出力します。 詳細については、「 3.5 インテル拡張ヘキサ・ファイル 」を参照してください。
Stype	モトローラ・Sタイプ・ファイルを出力します。 詳細については、「 3.6 モトローラ・Sタイプ・ファイル 」を参照してください。
Binary	バイナリ・ファイルを出力します。

- *format* を省略した場合は、Absolute を指定したものとみなします。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\] タブ](#)の [\[よく使うオプション（ヘキサ出力）\]](#) カテゴリの [\[ヘキサ・ファイルを出力する\]](#), [\[ヘキサ・ファイル・フォーマット\]](#), [\[よく使うオプション（ライブラリ生成）\]](#) カテゴリの [\[出力ファイル・フォーマット\]](#)
 - [\[ヘキサ出力オプション\] タブ](#)の [\[出力ファイル\]](#) カテゴリの [\[ヘキサ・ファイルを出力する\]](#), [\[オブジェクト結合機能を使用する\]](#), [\[ヘキサ・フォーマット\]](#) カテゴリの [\[ヘキサ・ファイル・フォーマット\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の [\[出力ファイル\]](#) カテゴリの [\[出力ファイル・フォーマット\]](#)

[備考]

- 出力形式と入力ファイル、他のオプションとの関係を以下に示します。

表 B—5 出力形式と入力ファイル、他のオプションとの関係

出力形式	指定オプション	入力可能なファイル形式	指定可能なオプション ^{注1}
Absolute	-strip あり	ロード・モジュール・ファイル	-input, -output
	上記以外	オブジェクト・ファイル リロケータブル・ファイル バイナリ・ファイル ライブラリ・ファイル	-input, -library, -binary, -debug, -nodebug, -cpu, -start, -rom, -entry, -output, -map, -hide, -compress, -rename, -delete, -define, -fsymbol, -stack, -memory, -msg_unused, -show={symbol reference reference}, -aligned_section, -overrun_fetch
Relocate	-extract あり	ライブラリ・ファイル	-library, -output
	上記以外	オブジェクト・ファイル リロケータブル・ファイル バイナリ・ファイル ライブラリ・ファイル	-input, -library, -debug, -nodebug, -output, -hide, -rename, -delete, -show={symbol reference}
Object	-extract あり	ライブラリ・ファイル	-library, -output
Hexadecimal Stype Binary		オブジェクト・ファイル リロケータブル・ファイル バイナリ・ファイル ライブラリ・ファイル	-input, -library, -binary, -cpu, -start, -rom, -entry, -output, -map, -space, -rename, -delete, -define, -fsymbol, -stack, -record, -s9 ^{注2} , -byte_count ^{注3} , -memory, -msg_unused, -show={symbol reference reference}, -aligned_section, -overrun_fetch ^{注4}
		ロード・モジュール・ファイル	-input, -output, -record, -s9 ^{注2} , -byte_count ^{注3} , -show={symbol reference reference}
		インテル拡張ヘキサ・ファイル ^{注5}	-input, -output
		モトローラ・Sタイプ・ファイル ^{注5}	-input, -output, -s9 ^{注2}
Library	-strip あり	ライブラリ・ファイル	-library, -output, -memory ^{注6} , -show={symbol section}
	-extract あり	ライブラリ・ファイル	-library, -output
	上記以外	オブジェクト・ファイル リロケータブル・ファイル	-input, -library, -output, -hide, -rename, -delete, -replace, -memory ^{注6} , -show={symbol section}

注1. 以下のオプションは、常に指定可能です。

-message, -nomessage, -change_message, -logo, -nologo, -form, -list, -subcommand

2. -s9 オプションは、-form=stype オプションを指定した場合のみ指定可能です。

3. -byte_count オプションは、-form=hexadecimal オプションを指定した場合のみ指定可能です。

4. -overrun_fetch オプションは、-form=hexadecimal、または -form=stype オプションを指定した場合のみ指定可能です。

5. 入力ファイルにインテル拡張ヘキサ・ファイルを指定した場合は -form=hexadecimal オプション、モトローラ・S タイプ・ファイルを指定した場合は -form=stype オプションのみを指定することができます。
6. -memory オプションは、-hide オプションを指定した場合は指定することはできません。

【使用例】

- a.obj, b.obj からリロケートブル・ファイル c.rel を出力します。

```
>rlink a.obj b.obj -form=relocate -output=c.rel
```

- lib.lib からモジュール a を取り出し、オブジェクト・ファイルとして出力します。

```
>rlink -library=lib.lib -extract=a -form=object
```

- lib.lib からモジュール a を取り出し、ライブラリ・ファイル exta.lib を出力します。

```
>rlink -library=lib.lib -extract=a -form=library -output=exta
```

- lib.lib からモジュール a を取り出し、リロケートブル・ファイル a.rel を出力します。

```
>rlink -library=lib.lib -extract=a -form=relocate
```

-DEBug

出力ファイル中にデバッグ情報を出力します。

[指定形式]

```
-DEBug
```

- 省略時解釈

出力ファイル中にデバッグ情報を出力します（-debug オプションの指定と同じです）。

[詳細説明]

- 出力ファイル中にデバッグ情報を出力します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の [デバッグ情報] カテゴリの [\[デバッグ情報を出力する\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の [デバッグ情報] カテゴリの [\[デバッグ情報を出力する\]](#)

[使用例]

- 出力ファイル中にデバッグ情報を出力します。

```
>rlink a.obj b.obj -debug -output=c.abs
```

[備考]

- 本オプションは、-form={object|library|hexadecimal|stype|binary}、-strip オプション、または -extract オプションを指定した場合は無効となります。
- -form=absolute オプション、および -output オプションで出力ファイル名を複数した場合は、デバッグ情報を出力しません。

-NODEBug

デバッグ情報を出力しません。

[指定形式]

```
-NODEBug
```

- 省略時解釈

出力ファイル中にデバッグ情報を出力します（-debug オプションの指定と同じです）。

[詳細説明]

- デバッグ情報を出力しません。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の「デバッグ情報」カテゴリの「[デバッグ情報を出力する](#)」
 - [\[ライブラリ生成オプション\] タブ](#)の「デバッグ情報」カテゴリの「[デバッグ情報を出力する](#)」

[使用例]

- デバッグ情報を出力しません。

```
>rlink a.obj b.obj -nodebug -output=c.abs
```

[備考]

- 本オプションは、-form={object|library|hexadecimal|stype|binary}、-strip オプション、または -extract オプションを指定した場合は無効となります。

-RECORD

出力するデータ・レコードのサイズを指定します。

[指定形式]

```
-RECORD={record}
```

- 省略時解釈

それぞれのアドレスにあわせて、混在したデータ・レコードを出力します。

[詳細説明]

- アドレス範囲に関係なく、一定のデータ・レコード *record* で出力します。
- *record* に指定可能なものを以下に示します。

H16	ヘキサ・レコード
H20	拡張ヘキサ・レコード
H32	32 ビット・ヘキサ・レコード
S1	S1 レコード
S2	S2 レコード
S3	S3 レコード

- 指定したデータ・レコードより大きいアドレスが存在した場合に、アドレスに合わせてデータ・レコードを選択します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[ヘキサ出力オプション\]](#) タブの [\[ヘキサ・フォーマット\]](#) カテゴリの [\[レコード・サイズを統一する\]](#), [\[レコード・サイズを統一する\]](#)

[使用例]

- アドレス範囲に関係なく、32 ビット・ヘキサ・レコードで出力します。

```
>rlink a.obj b.obj -record=H32 -form=hexadecimal -output=c.abs
```

[備考]

- 本オプションは、`-form={hexadecimal|stype}` オプションを指定していない場合は無効となります。
- `-form=hexadecimal` オプション指定時に `-record={S1|S2|S3}` オプションを指定した場合、および、`-form=stype` オプション指定時に `-record={H16|H20|H32}` オプションを指定した場合は、エラーとなります。

-ROm

ROM から RAM ヘマップするセクションを指定します。

[指定形式]

```
-ROm=ROMsection=RAMsection[,ROMsection=RAMsection]...
```

- 省略時解釈
なし

[詳細説明]

- 初期化データ領域の ROM 用, RAM 用領域を確保し, ROM セクション内定義シンボルを RAM セクション内アドレスでリロケーションします。
- ROM セクション *ROMsection* には, 初期値のあるリロケータブル・セクションを指定します。
- RAM セクション *RAMsection* には, 存在しないセクション, またはサイズ 0 のリロケータブル・セクションを指定します。
- 本オプションは, CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\]](#) タブの [セクション] カテゴリの [\[ROM から RAM ヘマップするセクション\]](#)

[使用例]

- .data セクションと同サイズの .data.R セクションを確保し, .data セクション内定義シンボルを .data.R セクション上のアドレスでリロケーションします。

```
>rlink a.obj b.obj -rom=.data=.data.R -start=.data/100,.data.R/8000
```

[備考]

- 本オプションは, -form={object|relocate|library} オプション, または -strip オプションを指定した場合は無効となります。

-Output

出力ファイルを指定します。

[指定形式]

```
-Output=file[={address1-address2|section[:section]...}] [,file[={address1-address2|section[:section]...}]]...
```

- 省略時解釈

出力ファイル名を「先頭入力ファイル名・デフォルト拡張子」とします。

デフォルト拡張子を以下に示します。

-form=absolute オプション指定時	: abs
-form=relocate オプション指定時	: rel
-form=object オプション指定時	: obj
-form=library オプション指定時	: lib
-form=hexadecimal オプション指定時	: hex
-form=stype オプション指定時	: mot
-form=binary オプション指定時	: bin

[詳細説明]

- 出力ファイル *file* を指定します。
- *address1*, *address2* には、出力範囲の先頭アドレス、終了アドレスを 16 進数で指定します。
出力範囲に “-” を指定した場合は常にアドレスと解釈します。
- *section* には、出力するセクションを指定します。
複数指定する場合は、コロン (:) で区切ります。
- *-form={absolute|hexadecimal|stype|binary}* オプションと同時に指定した場合は、複数のファイルを指定することができます。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[共通オプション\] タブ](#)の [\[よく使うオプション \(リンク\)\]](#) カテゴリの [\[出力フォルダ\]](#), [\[出力ファイル名\]](#), [\[よく使うオプション \(ヘキサ出力\)\]](#) カテゴリの [\[出力フォルダ\]](#), [\[出力ファイル名\]](#), [\[分割出力ファイル\]](#), [\[よく使うオプション \(ライブラリ生成\)\]](#) カテゴリの [\[出力フォルダ\]](#), [\[出力ファイル名\]](#)
 - [\[リンク・オプション\] タブ](#)の [\[出力ファイル\]](#) カテゴリの [\[出力フォルダ\]](#), [\[出力ファイル名\]](#)
 - [\[ヘキサ出力オプション\] タブ](#)の [\[出力ファイル\]](#) カテゴリの [\[出力フォルダ\]](#), [\[出力ファイル名\]](#), [\[分割出力ファイル\]](#), [\[結合ヘキサ・ファイルの出力フォルダ\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の [\[出力ファイル\]](#) カテゴリの [\[出力フォルダ\]](#), [\[出力ファイル名\]](#)

[使用例]

- 0 ~ 0xffff 間を file1.abs に、0x10000 ~ 0x1ffff 間を file2.abs に出力します。

```
>rlink a.obj b.obj -output=file1.abs=0-ffff,file2.abs=10000-1ffff
```

- sec1, sec2 セクションを file1.abs に、sec3 セクションを file2.abs に出力します。

```
>rlink a.obj b.obj -output=file1.abs=sec1:sec2,file2.abs=sec3
```

[備考]

- 入力ファイルがインテル拡張・ヘキサ・ファイル、またはモトローラ・S タイプ・ファイルの場合、本オプションで複数の出力ファイルを指定することはできません。

本オプションを省略した場合、出力ファイル名は“先頭入力ファイル名_combine.拡張子”となります（入力ファイルが a.mot の場合、出力ファイルは a_combine.mot となります）。

-MAp

外部変数割り付け情報ファイルを出力します。

[指定形式]

```
-MAp [=file]
```

- 省略時解釈
なし

[詳細説明]

- コンパイラが外部変数アクセス最適化で使用する外部変数割り付け情報ファイル *file* を出力します。
- ファイル名の指定を省略した場合は、-output オプションで指定したファイル名、または“先頭入力ファイル名.bls”というファイル名となります。
- 外部変数割り付け情報ファイル作成時の変数宣言順と、再コンパイル後のオブジェクトを読み込んだ時の変数宣言順が異なる場合は、エラーとなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の[最適化 (詳細)] カテゴリの[\[外部シンボル割り付け情報ファイルを出力する\]](#)

[使用例]

- 外部変数割り付け情報ファイル file.bls を出力します。

```
>rlink a.obj b.obj -output=c.abs -map=file.bls
```

[備考]

- 本オプションは、-form={absolute|hexadecimal|stype|binary} オプションを指定した場合のみ有効となります。

-Space

出力範囲のメモリの空き領域を充てんします。

[指定形式]

```
-Space [=data]
```

- 省略時解釈
なし

[詳細説明]

- 出力範囲のメモリの空き領域を、ユーザが指定するデータ *data* で充てんします。
- *data* に指定可能なものを以下に示します。

数値	16 進数の数値
Random	乱数

- 空き領域の充てん方法は、-output オプションで指定する出力範囲によって、以下のように異なります。
 - 出力範囲としてセクションを指定した場合
 - 指定したセクション間に空き領域が存在した場合に、指定データを出力します。
 - 出力範囲として先頭アドレス、終了アドレスを指定した場合
 - 指定した範囲内に空き領域が存在した場合に、指定データを出力します。
- 出力データ・サイズは、1, 2, 4 バイト単位で有効となり、本オプションで指定する 16 進数の数値で決まります。
 - 3 バイト・データを指定した場合は、上位桁を 0 拡張し、4 バイトのデータとして扱います。
 - 奇数桁データを指定した場合は、上位桁に 0 拡張して、偶数桁入力として扱います。
- 空き領域のサイズが出力データ・サイズの倍数でない場合は、出力できるだけ出力し、警告を出力します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[ヘキサ出力オプション\] タブ](#)の [\[ヘキサ・フォーマット\]](#) カテゴリの [\[出力範囲のメモリの空き領域をデータで充てんする\]](#)、[\[空き領域出力データ\]](#)

[使用例]

- 100H 番地から 2FFH 番地の範囲内で、メモリの空き領域に対して ffH で充てんします。

```
>rlink a.obj b.obj -form=hexadecimal -output=file1=100-2ff -start=P1/100,P2/200 -space=ff
```

【備考】

- 本オプションにてデータの指定を省略した場合は、空き領域への充てんを行いません。
- 本オプションは、-form={binary|stype|hexadecimal} オプションを指定した場合のみ有効となります。
- 本オプションは、-output オプションにて出力範囲を指定しなかった場合は無効となります。

-Message

インフォメーション・メッセージを出力します。

[指定形式]

```
-Message
```

- 省略時解釈

インフォメーション・メッセージの出力を抑止します（-nomessage オプションの指定と同じです）。

[詳細説明]

- インフォメーション・メッセージを出力します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の [メッセージ] カテゴリの [\[インフォメーション・メッセージ出力を有効にする\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の [メッセージ] カテゴリの [\[インフォメーション・メッセージ出力を有効にする\]](#)

[使用例]

- インフォメーション・メッセージを出力します。

```
>rlink a.obj b.obj -message
```

-NOMessage

インフォメーション・メッセージの出力を抑止します。

[指定形式]

```
-NOMessage [=num [-num] [, num [-num]] ...]
```

- 省略時解釈

インフォメーション・メッセージの出力を抑止します。

[詳細説明]

- インフォメーション・メッセージの出力を抑止します。
- メッセージ番号 *num* を指定すると、指定した番号のメッセージの出力を抑止します。
また、ハイフン (-) を使用して、メッセージ番号の範囲を指定することもできます。
- *num* には、コンポーネント番号 (05)、発生フェーズ (6) に続けて出力される 4 桁の数値 (M0560004 の場合は 0004) を指定します。
4 桁の数値の先頭が 0 で始まる場合は、0 を省略することができます (M0560004 の場合は 4)。
- ワーニング、またはエラー種別のメッセージ番号を指定した場合は、-change_message オプションでインフォメーション種別に変更したと仮定し、メッセージの出力を抑止します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブの \[メッセージ\] カテゴリの \[インフォメーション・メッセージ出力を有効にする\]](#), [\[抑止するインフォメーション・メッセージの番号\]](#)
 - [\[ライブラリ生成オプション\] タブの \[メッセージ\] カテゴリの \[インフォメーション・メッセージ出力を有効にする\]](#), [\[抑止するインフォメーション・メッセージの番号\]](#)

[使用例]

- M0560004, M0560100 ~ M0560103, および M0560500 のメッセージの出力を抑止します。

```
>rlink a.obj b.obj -nomessage=4,100-103,500
```

-MSg_unused

参照されない外部定義シンボルをユーザに通知します。

[指定形式]

```
-MSg_unused
```

- 省略時解釈
なし

[詳細説明]

- リンク処理の中で一度も参照されることのなかった外部定義シンボルを、メッセージ出力によってユーザに通知します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\]](#) タブの [メッセージ] カテゴリの [\[参照されない定義シンボルを通知する\]](#)

[使用例]

- 参照されない外部定義シンボルをユーザに通知します。

```
>rlink a.obj b.obj -message -msg_unused
```

[備考]

- 本オプションは、入力ファイルがロード・モジュール・ファイルの場合は無効となります。
- 本オプションは、-message オプションと同時に指定する必要があります。
- コンパイル時にインライン展開された関数に対して、メッセージ出力する場合があります。
その場合は、関数定義に static 宣言を追加することにより、メッセージ出力を抑止することができます。
- 同一ファイル内の定数シンボルへの参照がある場合は、参照関係の解析が正しく行うことができず、メッセージ出力により通知される情報が不正確となります。

-BYte_count

データ・レコードのバイト数の最大値を指定します。

[指定形式]

```
-BYte_count=num
```

- 省略時解釈

バイト数の最大値を FF として、インテル拡張ヘキサ・ファイルを生成します。

[詳細説明]

- インテル拡張ヘキサ・ファイルを生成する際に、データ・レコードのバイト数の最大値 *num* を指定します。
- *num* には、1 バイトの 16 進数 (01 ~ FF) を指定します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[ヘキサ出力オプション\]](#) タブの [\[ヘキサ・フォーマット\]](#) カテゴリの [\[データ・レコードのバイト数を指定する\]](#), [\[データ・レコードのバイト数最大値\]](#)

[使用例]

- データ・レコードのバイト数の最大値として 10 を指定します。

```
>rlink a.obj b.obj -form=hexadecimal -byte_count=10
```

[備考]

- 本オプションは、生成ファイルがインテル拡張ヘキサ・ファイルでない場合 (-form=hexadecimal オプション未指定時) は無効となります。

-PADDING

セクションの終端にデータを埋め込みます。

[指定形式]

```
-PADDING
```

- 省略時解釈
なし

[詳細説明]

- セクション・サイズが、セクションのアライメントの倍数となるように、セクションの終端にデータを埋め込みます。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\]](#) タブの [出力コード] カテゴリの [\[セクション終端にパディング・データを埋め込む\]](#)

[使用例]

- 以下の場合、P セクションに 2 バイトのパディング・データを埋め込み、サイズを 0x08 バイトにしてリンク処理を行います。

P セクションのアライメント : 4 バイト
P セクションのサイズ : 0x06 バイト
C セクションのアライメント : 1 バイト
C セクションのサイズ : 0x03 バイト

```
>rlink a.obj b.obj -start=P,C/0 -padding
```

- 以下の場合、P セクションに 2 バイトのパディング・データを埋め込み、サイズを 0x08 バイトにしてリンク処理を行うと、C セクションと重複してしまうため、エラーを出力します。

P セクションのアライメント : 4 バイト
P セクションのサイズ : 0x06 バイト
C セクションのアライメント : 1 バイト
C セクションのサイズ : 0x03 バイト

```
>rlink a.obj b.obj -start=P/0,C/7 -padding
```


【備考】

- 生成するパディング・データの値は 0x00 です。
- 絶対アドレス・セクションにはパディングを行わないため、絶対アドレス・セクションのサイズはユーザにて調整してください。

-OVERRUN_FETCH

オーバーラン・フェッチに伴う空き領域の読み出しを回避します。

[指定形式]

```
-OVERRUN_FETCH
```

- 省略時解釈
なし

[詳細説明]

- セクション間、および終端コード・セクションの直後に 128 バイト以上の空き領域が存在する場合、空き領域に対して 128 バイトの NOP 命令を挿入します。

NOP 命令の挿入は -start オプションの配置に従って挿入し、挿入する直前のセクション名に対して以下のセクション名で出力します。

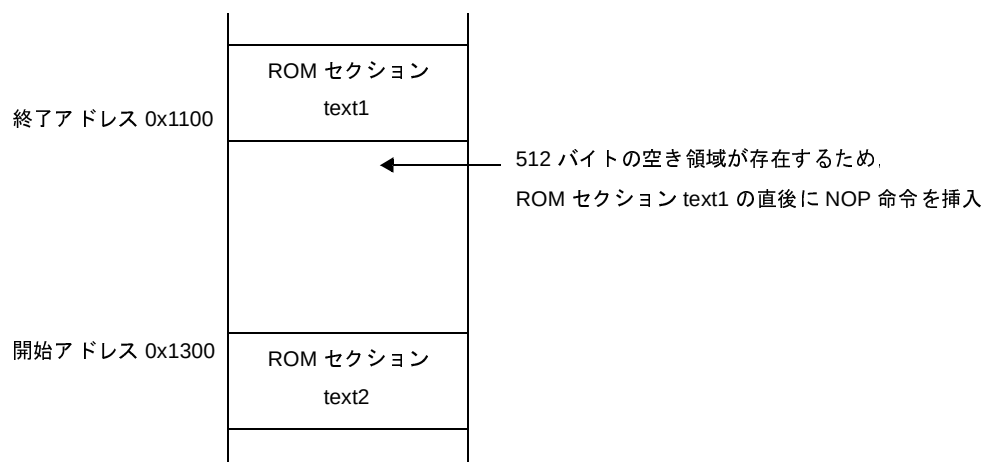
```
$sss_fetch??
```

sss : 挿入する直前のセクション名

?? : 01 ~ 99

(1) セクション間への挿入

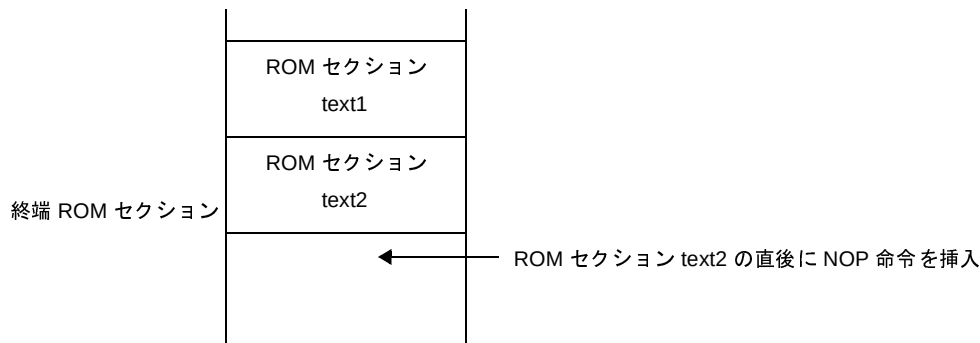
- ROM セクション（実行可能なプログラム・セクション、またはプログラム・セクション以外の ROM 領域配置セクション）と ROM セクション、および ROM セクションと RAM セクション（RAM 領域配置セクション）の間に 128 バイト以上の空き領域が存在する場合に NOP 命令を挿入します。



- ROM セクションと RAM セクションは、ROM セクションより大きいアドレスに RAM セクションが配置された場合に限り、NOP 命令を挿入します（ROM セクションより小さいアドレスに RAM セクションが配置された場合は NOP 命令を挿入しません）。

(2) 終端セクション直後への挿入

- 終端 ROM セクションの直後に NOP 命令を挿入します。



- -cpu オプションでアドレス範囲を指定する際、終端セクションの終端アドレスとして指定した終了アドレスの間に 128 バイトの空き領域がない場合、NOP 命令は挿入されません。
- 終端セクションの終端アドレスと 8M バイト境界のアドレスとの間に 128 バイトの空き領域がない場合、NOP 命令は挿入されません。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\]](#) タブの [出力コード] カテゴリの [\[オーバーラン・フェッチを回避する\]](#)

[使用例]

- オーバーラン・フェッチに伴う空き領域の読み出しを回避します。

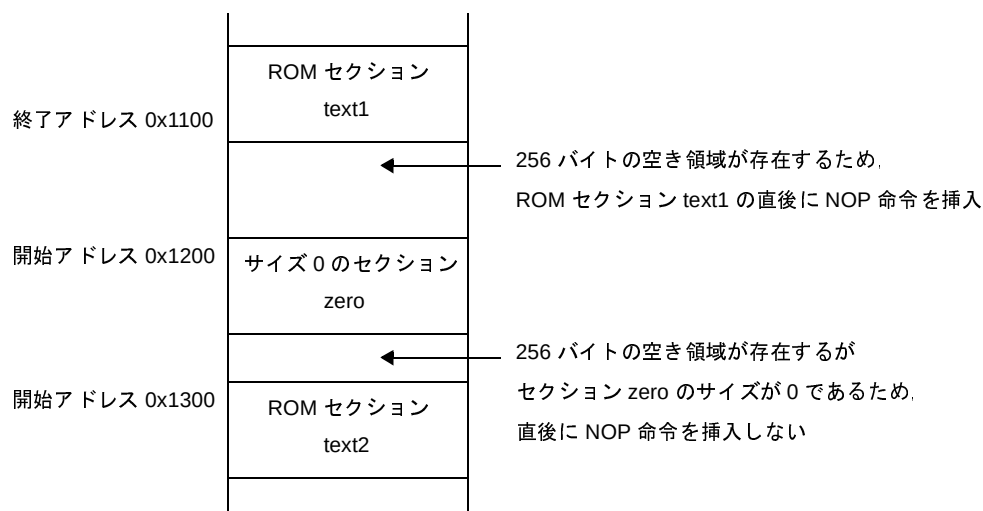
```
>rlink a.obj b.obj -overrun_fetch
```

[備考]

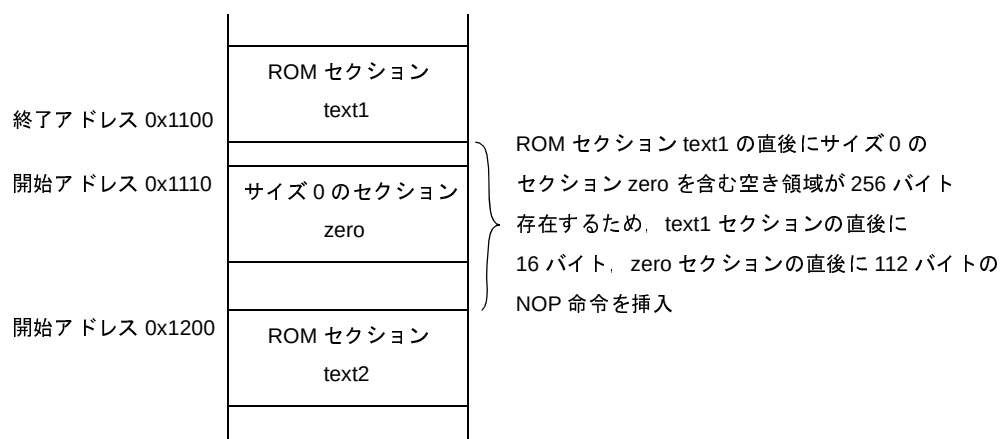
- -cpu オプションでアドレス範囲を指定していない場合、かつ終端セクションが ROM セクションの場合、終端 ROM セクションの直後に常に NOP 命令を挿入します。
- 本オプションは、入力ファイルがロード・モジュール・ファイルの場合は無効となります。
- 本オプションは、-form={object|relocate|library} オプションを指定した場合は無効となります。
- -start オプションでオーバーレイ配置指定を行った場合、オーバーレイ配置指定を含む start 列には NOP 命令の挿入を行いません。

例 1. -start=A,B,E/400,C,D:F:G/8000 オプションを指定した場合、オーバーレイ配置指定を含むセクション C, D, F, および G に対するセクション間、および終端コード・セクションに対して、NOP 命令の挿入を行いません。

2. サイズ0のセクションの直後に128バイト以上の空き領域が存在する場合、NOP命令の挿入を行います。



3. ROMセクションの直後に配置されたサイズ0のセクションを含む128バイト以上の空き領域が存在する場合、NOP命令を挿入します。



リスト出力

リスト出力オプションには、次のものがあります。

- [-LISt](#)
- [-SHow](#)

-LISt

リスト・ファイルを出力します。

[指定形式]

```
-LISt [=file]
```

- 省略時解釈
なし

[詳細説明]

- リスト・ファイル *file* を出力します。
- ファイル名の指定を省略した場合は、以下のリスト・ファイルを出力します。

指定オプション	ファイルの種類	ファイル名
-form=library オプション、または -extract オプション	ライブラリ・リスト・ファイル	出力ファイル名 ^注 .lbp
上記以外	リンク・マップ・ファイル	出力ファイル名 ^注 .map

注 出力ファイルが複数存在する場合は、先頭出力ファイル名となります。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の [\[リスト\]](#) カテゴリの [\[リンク・マップ・ファイルを出力する\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の [\[リスト\]](#) カテゴリの [\[リンク・マップ・ファイルを出力する\]](#)

[使用例]

- リンク・マップ・ファイルを file.map に出力します。

```
>rlink a.obj b.obj -list=file.map
```

-Show

リスト・ファイルへの出力情報を指定します。

[指定形式]

```
-Show[=info[,info]...]
```

- 省略時解釈
なし

[詳細説明]

- リスト・ファイルへの出力情報 *info* を指定します。
- *info* に指定可能なものを以下に示します。

出力情報 (<i>info</i>)	-form=library オプション指定時	-form=library オプション以外指定時
SYmbol	モジュール内シンボル名を出力	シンボル・アドレス、サイズ、種別、最適化内容を出力
Reference	指定不可	シンボル・アドレス、サイズ、種別、最適化内容、シンボルの参照回数を出力
SEction	モジュール内セクション名を出力	指定不可
Xreference	指定不可	クロス・リファレンス情報を出力
Total_size	指定不可	ROM 配置対象、RAM 配置対象ごとに、セクションの合計サイズを出力
ALL	モジュール内シンボル名、セクション名を出力	-form=relocate オプション指定時 -show=symbol,xreference,total_size オプション指定時と同じ情報を出力 -form=absolute オプション指定時 -show=symbol,reference,xreference,total_size 指定時と同じ情報を出力 -form=hexadecimal/stype/binary オプション指定時 -show=symbol,reference,xreference,total_size 指定時と同じ情報を出力 -form=object オプション指定時 指定不可

備考 出力情報の詳細については、「[3.2 リンク・マップ・ファイル](#)」,「[3.4 ライブラリ・リスト・ファイル](#)」を参照してください。

- 出力情報の指定を省略した場合については、[備考]を参照してください。
- -extract オプションを指定した場合、出力情報 (*info*) を指定することはできません。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [リンク・オプション] タブの [リスト] カテゴリの [リンク・マップ・ファイルを出力する], [シンボル情報を出力する], [シンボルの参照回数を出力する], [クロス・リファレンス情報を出力する], [セクションの合計サイズを出力する]
 - [ライブラリ生成オプション] タブの [リスト] カテゴリの [リンク・マップ・ファイルを出力する], [シンボル情報を出力する], [モジュール内セクション一覧を出力する], [クロス・リファレンス情報を出力する], [セクションの合計サイズを出力する]

【使用例】

- リンク・マップ・ファイルにシンボル・アドレス、サイズ、種別、最適化内容、シンボルの参照回数を出力します。

```
>rlink a.obj b.obj -list -show=symbol,reference
```

【備考】

- -form オプションと -show, または -show=all オプションの指定により、出力情報 *info* が有効／無効になる組み合わせを以下に示します。

		Symbol	Reference	Section	Xreference	Total_size
-form=absolute	-show のみ	有効	有効	無効	無効	無効
	-show=all	有効	有効	無効	有効	有効
-form=library	-show のみ	有効	無効	有効	無効	無効
	-show=all	有効	無効	有効	無効	無効
-form=relocate	-show のみ	有効	無効	無効	無効	無効
	-show=all	有効	無効	無効	有効 ^{注1}	有効
-form=object	-show のみ	有効	有効	無効	無効	無効
	-show=all	無効	無効	無効	無効	無効
-form=hexadecimal ^{注2} /type ^{注3} binary	-show のみ	有効	有効	無効	無効	無効
	-show=all	有効	有効	無効	有効	有効 ^{注1}

注1. 入力ファイルがロード・モジュール・ファイルの場合は無効となります。

2. 入力ファイルがインテル拡張ヘキサ・ファイルの場合は、-show オプションを指定することはできません。

3. 入力ファイルがモトローラ・S タイプ・ファイルの場合は、-show オプションを指定することはできません。

なお、クロス・リファレンス情報の出力については以下の制限があります。

- 入力ファイルがロード・モジュール・ファイルの場合、参照側アドレスの情報は出力されません。
- 同一ファイル内の定数シンボルへの参照についての情報は出力されません。
- コンパイル時に最適化が有効で、直下の関数を呼び出す場合についての情報は出力されません。
- -show=total_size オプションで表示する情報は、-total_size オプションにより表示する情報と同じです。

セクション指定

セクション指定オプションには、次のものがあります。

- START
- FSymbol
- ALIGNED_SECTION

-START

セクションの開始アドレスを指定します。

[指定形式]

```
-START=[ ( ) section [ { : | , } section ] ... [ ( ) [ { : | , } section ] ... [/address] [ , [ ( ) section [ { : | , } section ] ... [ ( ) [ { : | , } section ] ... [/address] ] ...
```

- 省略時解釈

絶対アドレス・セクションをアドレスの小さい順に配置し、絶対アドレス・セクションの末尾から指定された入力ファイル順に出現する相対アドレス・セクションを配置します。

[詳細説明]

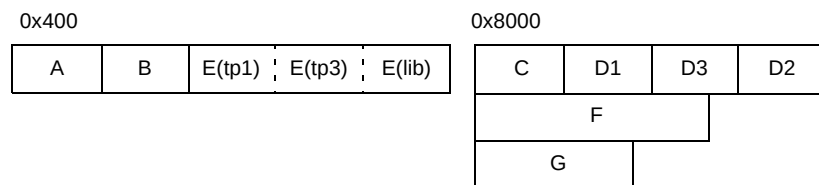
- セクション *section* の開始アドレス *address* を指定します。
address は 16 進数で指定します。
- *section* にはワイルドカード (*) も使用可能です。
ワイルドカードで指定したセクションは、入力順に展開します。
- セクションをコロン (:) で区切ることで、複数のセクション（カンマ (,) で区切って指定）を同一アドレスに割り付ける（セクション・オーバーレイ配置）ことが可能です。
同一アドレスに割り付け指定したセクション間は、指定順に割り付けます。
また、丸っこ “()” で囲むことで、オーバーレイ配置する対象セクションを変更することができます。
- 同一セクション内のオブジェクトは、入力ファイルの指定順、入力ライブラリの指定順に割り付けます。
- アドレスの指定を省略した場合は、0 番地から割り付けます。
- -start オプションで指定していないセクションは、最終割り付けアドレスに続いて割り付けます。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\]](#) タブの [セクション] カテゴリの [\[セクションの開始アドレス\]](#)

[使用例]

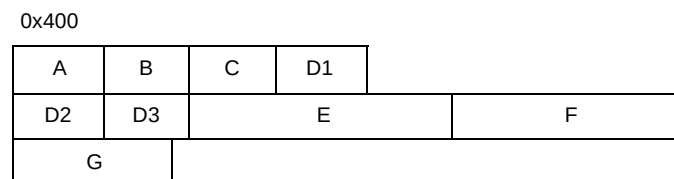
- 以下の順番でオブジェクトを入力する場合のセクション配置を示します（カッコ内は各オブジェクトが持つセクションです）。

```
tp1.obj(A,D1,E)
tp2.obj(B,D3,F)
tp3.obj(C,D2,E,G)
lib.lib(E)
```

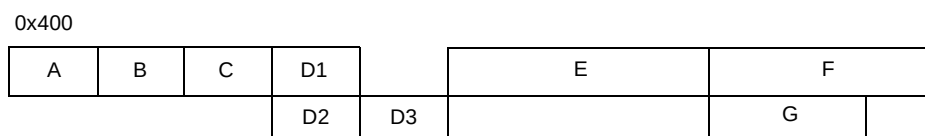
- -start=A,B,E/400,C,D*:F:G/8000 オプションを指定した場合



- “:” で区切った C, F, G セクションは、同一アドレスに割り付きます。
 - ワイルドカードで記述したセクション（ここでは D で始まる名前前のセクション）は、入力した順番で割り付きます。
 - 同名セクション内（ここでは E セクション）は、入力したオブジェクトから順番に割り付きます。
 - ライブラリ入力による同名セクション（ここでは E セクション）は、入力オブジェクトの次に割り付きます。
- -start=A,B,C,D1:D2,D3,E,F:G/400 オプションを指定した場合



- “:” で区切った直後のセクション（この例の場合は A, D2, G）を先頭として、それぞれの先頭が同一アドレスに割り付きます。
- -start=A,B,C,(D1:D2,D3),E,(F:G)/400 オプションを指定した場合



- “()” で同一アドレス配置を囲んだ場合，“()” の直前のセクション（この例の場合は C, E）の直後を先頭として，“()” 内の同一アドレス配置が行われます。

- “()” の直後のセクション（この例の場合はE）は，“()” 内の最後尾のセクションの直後に続けて配置されます。

【備考】

- 本オプションは、-form={object|relocate|library} オプション、または -strip オプションを指定した場合は無効となります。
- “()” は、ネストして記述することはできません。
- “()” 内では、1 つ以上のコロン “:” の記述が必要です。
“:” を記述しない場合，“()” を記述することはできません。
- “()” を記述した場合，“()” の外にコロン “:” を記述することはできません。

-FSymbol

外部定義シンボルをシンボル・アドレス・ファイルに出力します。

[指定形式]

```
-FSymbol=section[,section]...
```

- 省略時解釈
なし

[詳細説明]

- セクション *section* 内の外部定義シンボルをアセンブラ制御命令形式でファイル（シンボル・アドレス・ファイル）に出力します。
ファイル名は“出力ファイル名.fsy”となります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の [セクション] カテゴリの [\[スタートアップの属するセクション\]](#), [\[外部定義シンボルをファイル出力するセクション\]](#)

[使用例]

- セクション sct2, sct3 内の外部定義シンボルをシンボル・アドレス・ファイル test.fsy に出力します。

```
>rlink a.obj b.obj -fsymbol=sct2,sct3 -output=test.abs
```

シンボル・アドレス・ファイル test.fsy の出力例を以下に示します。

```
;RENESAS OPTIMIZING LINKER GENERATED FILE xxxx.xx.xx
;fsymbol = sct2,sct3

;SECTION NAME = sct2
    .public _f
_f .equ 0x0
    .public _g
_g .equ 0x16
;SECTION NAME = sct3
    .public _main
_main .equ 0x20
```

【備考】

- 本オプションは、-form={object|relocate|library} オプション、または -strip オプションを指定した場合は無効となります。

-ALIGNED_SECTION

セクションのアライメント数を 16 バイト に変更します。

[指定形式]

```
-ALIGNED_SECTION=section[,section]...
```

- 省略時解釈
なし

[詳細説明]

- セクション *section* のアライメント数を 16 バイト に変更します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\]](#) タブの [セクション] カテゴリの [\[セクション・アライメント\]](#)

[使用例]

- A セクションのアライメント数を 16 バイト に変更します。

```
>rlink a.obj b.obj -aligned_section=A
```

[備考]

- 本オプションは、-form={object|relocate|library} オプション、および -extract、-strip オプションを指定した場合は無効となります。

ベリファイ指定

ベリファイ指定オプションには、次のものがあります。

- **-CPu**

-CPu

セクションの割り付けアドレスの整合性をチェックします。

[指定形式]

```
-CPu=type=address1-address2[,type=address1-address2]...
```

- 省略時解釈

なし

[詳細説明]

- セクションの割り付けアドレスの整合性をチェックします。

メモリ種別 *type* のセクションの割り付けアドレスに対して、指定したアドレス範囲に収まらない場合は、エラーを出力します。

- *type* に指定可能なものを以下に示します。

これ以外のものを指定した場合は、エラーとなります。

ROm	セクションの割り付け領域を ROM とします。
RAm	セクションの割り付け領域を RAM とします。
FIX	セクションの割り付け領域をアドレス固定の領域（I/O エリアなど）とします。 アドレス範囲が ROM、および RAM と重複した場合は、“FIX” を有効とします。

- *address1*, *address2* には、整合性をチェックするアドレス範囲の先頭アドレス、終了アドレスを 16 進数で指定します。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[リンク・オプション\] タブ](#) の [\[ベリファイ\]](#) カテゴリの [\[セクションの割り付けアドレスをチェックする\]](#), [\[メモリ種別のアドレス範囲\]](#)

[使用例]

- .text セクションが 0x100 ~ 0x1FF, .bss セクションが 0x200 ~ 0x2FF の範囲に収まる場合は正常終了します。
収まらない場合はエラーを出力します。

```
>rlink a.obj b.obj -start=.text/100,.bss/200 -cpu=ROM=100-1FF,RAM=200-2FF
```

【備考】

- 本オプションは、-form={object|relocate|library} オプション、または -strip オプションを指定した場合は無効となります。

サブコマンド・ファイル指定

サブコマンド・ファイル指定オプションには、次のものがあります。

- [-SUBcommand](#)

-SUBcommand

オプションをサブコマンド・ファイルで指定します。

[指定形式]

```
-SUBcommand=file
```

- 省略時解釈

なし

[詳細説明]

- オプションをサブコマンド・ファイル *file* で指定します。
- サブコマンド・ファイルで指定したオプション内容を、コマンド・ライン上の本オプション指定位置に展開し、実行します。
- サブコマンド・ファイルについての詳細は、「[B. 4. 2 サブコマンド・ファイルの使用方法](#)」を参照してください。

[使用例]

- サブコマンド・ファイル test.sub を以下の内容で作成します。

```
input file2.obj file3.obj      ; ここはコメントです。
library lib1.lib, &             ; “&” は継続行を示します。
lib2.lib
```

コマンド・ライン上でサブコマンド・ファイル test.sub を指定します。

```
>rlink file1.obj -subcommand=test.sub file4.obj
```

コマンド・ラインは以下のように展開され、ファイルの入力順序は、file1.obj, file2.obj, file3.obj, file4.obj となります。

```
>rlink file1.obj file2.obj file3.obj -library=lib1.lib,lib2.lib file4.obj
```


その他

その他のオプションには、次のものがあります。

- S9
- STACK
- COmpress
- NOCOmpress
- MEMory
- REName
- DElete
- REPlace
- EXTract
- STRip
- CHange_message
- Hide
- Total_size
- LOgo
- NOLOgo
- END
- EXIt

-S9

S9 レコードを終端に出力します。

[指定形式]

-S9

- 省略時解釈
なし

[詳細説明]

- エントリ・アドレスが 0x10000 を超える場合でも、S9 レコードを終端に出力します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[ヘキサ出力オプション\]](#) タブの [\[ヘキサ・フォーマット\]](#) カテゴリの [\[S9 レコードを終端に出力する\]](#)

【使用例】

- エントリ・アドレスが 0x10000 を超える場合でも、S9 レコードを終端に出力します。

```
>rlink a.obj b.obj -form=stype -s9
```

【備考】

- 本オプションは、-form=stype オプションを指定していない場合は無効となります。

-STACK

スタック情報ファイルを出力します。

[指定形式]

```
-STACK
```

- 省略時解釈
なし

[詳細説明]

- スタック情報ファイルを出力します。
- ファイル名は、“出力ファイル名.sni” となります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\]](#) タブの [\[その他\]](#) カテゴリの [\[スタック情報ファイルを出力する\]](#)

[使用例]

- スタック情報ファイル c.sni を出力します。

```
>rlink a.obj b.obj -stack
```

[備考]

- 本オプションは、-form={object|relocate|library} オプション、および -strip オプションを指定した場合は無効となります。

-CCompress

デバッグ情報を圧縮します。

[指定形式]

```
-CCompress
```

- 省略時解釈

デバッグ情報を圧縮しません（-nocompress オプションの指定と同じです）。

[詳細説明]

- デバッグ情報を圧縮します。
- デバッグ情報を圧縮すると、デバッグのロード速度が速くなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\]](#) タブの [デバッグ情報] カテゴリの [\[デバッグ情報を圧縮する\]](#)

[使用例]

- デバッグ情報を圧縮します。

```
>rlink a.obj b.obj -compress
```

[備考]

- 本オプションは、-form={object|relocate|library|hexadecimal|stype|binary} オプション、または -strip オプションを指定した場合は無効となります。

-NOCompress

デバッグ情報を圧縮しません。

【指定形式】

```
-NOCompress
```

- 省略時解釈

デバッグ情報を圧縮しません。

【詳細説明】

- デバッグ情報を圧縮しません。
- 本オプションを指定すると、-compress オプションを指定した場合に比べてリンク時間が短くなります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の [\[デバッグ情報\]](#) カテゴリの [\[デバッグ情報を圧縮する\]](#)

【使用例】

- デバッグ情報を圧縮しません。

```
>rlink a.obj b.obj -nocompress
```

-MEMory

リンク時に使用するメモリ量を指定します。

[指定形式]

```
-MEMory= [occupancy]
```

- 省略時解釈

従来通りの処理を行います（-memory=high オプションの指定と同じです）。

[詳細説明]

- リンク時に使用するメモリ量 *occupancy* を指定します。

- *occupancy* に指定可能なものを以下に示します。

High	従来通りの処理を行います。
Low	リンク時に必要な情報のロードを細かく行うことにより、使用するメモリ量の削減を行います。 ファイル・アクセスの頻度が増えるため、メモリ使用量が実装メモリを超えない状況では High を指定した場合よりも処理が遅くなります。

- *occupancy* を省略した場合は、High を指定したものとみなします。

- 大規模なプロジェクトをリンクした際、最適化リンクのメモリ使用量が稼働マシンの実装メモリ量を超えてしまい、動作が遅くなっているような場合は、*occupancy* に Low を指定してください。

- 本オプションは、CubeSuite+ の以下のプロパティに相当します。

- [\[リンク・オプション\] タブ](#)の[\[その他\]](#)カテゴリの[\[メモリ使用量を削減する\]](#)
- [\[ライブラリ生成オプション\] タブ](#)の[\[その他\]](#)カテゴリの[\[メモリ使用量を削減する\]](#)

[使用例]

- 使用するメモリ量の削減を行います。

```
>rlink a.obj b.obj -memory=low
```

[備考]

- 以下の場合、-memory=low オプションの指定は無効となります。

- -form={absolute|hexadecimal|stype|binary} オプションと以下のオプションを同時に指定した場合
 - compress, -delete, -rename, -map, -stack オプションのいずれか
 - list オプションと -show[={reference|xreference}] オプションを同時指定
- -form=library オプションと以下のオプションを同時に指定した場合
 - delete, -rename, -extract, -hide, -replace オプションのいずれか

- -form={object|relocate} オプションと以下のオプションを同時に指定した場合

-extract オプション

また、入力ファイルや出力ファイルの形式によっても無効となる組み合わせがあります。

詳細については、「[表 B—5 出力形式と入力ファイル, 他のオプションとの関係](#)」を参照してください。

-REName

外部シンボル名，セクション名を変更します。

[指定形式]

```
-REName={ [file] (name1=name2[, name1=name2] ...) | [module] (name1=name2[, name1=name2] ...) }  
[[, { [file] (name1=name2[, name1=name2] ...) | [module] (name1=name2[, name1=name2] ...) } ] ... ]
```

- 省略時解釈

なし

[詳細説明]

- 外部シンボル名，セクション名を変更します。
- *name1* には変更対象のシンボル名，またはセクション名，*name2* には変更後のシンボル名，またはセクション名を指定します。
- 特定のファイル *file*，または特定のライブラリ内モジュール *module* に含まれるシンボル名，セクション名を変更することもできます。
- C 変数名を指定する場合は，プログラム中での定義名の先頭に “_” を付加します。
- 関数名を変更した場合の動作を保証することはできません。
- 指定した名前がセクション，シンボルの両方に存在した場合は，シンボル名を優先します。
- 同一のファイル名，モジュール名が複数存在する場合は，先に入力した方を優先します。

[使用例]

- シンボル名 *_sym1* を *data* に変更します。

```
>rlink a.obj b.obj -rename=(_sym1=data)
```

- ライブラリ・モジュール *lib1* 内の *.text* セクションを *P* セクションに変更します。

```
>rlink -library=lib.lib -rename=lib1(.text=P) -form=library
```

[備考]

- 本オプションは，*-extract* オプション，または *-strip* オプションと同時に指定した場合は無効となります。
- *-form=absolute* オプションを指定した場合は，入力されたライブラリのセクション名を変更することはできません。
- コンパイル・オプション *-Xmerge_files* と本オプションを組み合わせて使用した場合，動作は保証されません。

-DElete

外部シンボル名、またはライブラリ・モジュールを削除します。

[指定形式]

```
-DElete={ [file] (symbol[,symbol]...) | module} [, { [file] (symbol[,symbol]...) | module}] ...
```

- 省略時解釈
なし

[詳細説明]

- 外部シンボル名 *symbol*、またはライブラリ・モジュール *module* を削除します。
- 特定のファイル *file* に含まれるシンボル名、モジュールを削除することもできます。
- C 変数名、C 関数名を指定する場合は、プログラム中での定義名の先頭に “_” を付加します。
- 同一ファイル名が複数存在する場合は、先に入力した方を優先します。
- 本オプションでシンボル名の削除を指定した場合、オブジェクトは削除されず、属性が内部シンボルに変更されます。

[使用例]

- 全ファイル中のシンボル名 *_sym1* を削除します。

```
>rlink a.obj -delete=(_sym1)
```

- b.obj 内のシンボル名 *_sym2* を削除します。

```
>rlink a.obj b.obj -delete=b.obj(_sym2)
```

[備考]

- 本オプションは、-extract オプション、または -strip オプションと同時に指定した場合は無効となります。
- -form=library オプションを指定した場合は、ライブラリ・モジュールを削除することができます。
- -form={absolute|relocate|hexadecimal|stype|binary} オプションを指定した場合は、外部シンボルを削除することができます。
- コンパイル・オプション -Xmerge_files と本オプションを組み合わせて使用した場合、動作は保証されません。

-REPlace

ライブラリ・モジュールを置換します。

[指定形式]

```
-REPlace=file[(module[,module]...)][,file[(module[,module]...)]]...
```

- 省略時解釈
なし

[詳細説明]

- 指定したファイル *file*, またはライブラリ・モジュール *module* と *-library* オプションで指定したライブラリ・ファイル内の同名モジュールを置換します。

[使用例]

- *file1.obj* とライブラリ・ファイル *lib1.lib* 内のモジュール *file1* を置換します。

```
>rlink -library=lib1.lib -replace=file1.obj -form=library
```

- モジュール *mdl1* とライブラリ・ファイル *lib1.lib* 内のモジュール *mdl1* を置換します。

```
>rlink -library=lib1.lib,lib2.lib -replace=lib1.lib(mdl1) -form=library
```

[備考]

- 本オプションは、*-form={object|relocate|absolute|hexadecimal|stype|binary}* オプション、および *-extract*、*-strip* オプションを指定した場合は無効となります。
- コンパイル・オプション *-Xmerge_files* と本オプションを組み合わせで使用した場合、動作は保証されません。

-EXtract

ライブラリ・モジュールを抽出します。

[指定形式]

```
-EXtract=module[,module]...
```

- 省略時解釈
なし

[詳細説明]

- ライブラリ・モジュール *module* を -library オプションで指定したライブラリ・ファイルから抽出します。

[使用例]

- ライブラリ・ファイル lib1.lib からモジュール file1 を抽出し、オブジェクト・ファイル出力形式のファイルを出します。

```
>rlink -library=lib1.lib -extract=file1 -form=obj
```

[備考]

- 本オプションは、-form={absolute|hexadecimal|stype|binary} オプション、および -strip オプションを指定した場合は無効となります。
- -form=library オプションを指定した場合は、ライブラリ・モジュールを削除することができます。
- -form={absolute|relocate|hexadecimal|stype|binary} オプションを指定した場合は、外部シンボルを削除することができます。

-STRip

ロード・モジュール・ファイル、ライブラリ・ファイルのデバッグ情報を削除します。

[指定形式]

```
-STRip
```

- 省略時解釈
なし

[詳細説明]

- ロード・モジュール・ファイル、ライブラリ・ファイルのデバッグ情報を削除します。
- デバッグ情報を削除する前のファイルは、“ファイル名.abk”という名前のファイルにバックアップします。
- 本オプションを指定した場合、入力ファイルと出力ファイルは1対1対応になります。

[使用例]

- file1.abs, file2.abs, file3.abs のデバッグ情報を削除し、それぞれ file1.abs, file2.abs, file3.abs に出力します。
デバッグ情報を削除する前のファイルは、file1.abk, file2.abk, file3.abk にバックアップします。

<コマンド・ライン>

```
>rlink -subcommand=sub.txt
```

<サブコマンド・ファイル sub.txt >

```
-input=file1.abs file2.abs file3.abs  
-strip
```

[備考]

- 本オプションは、-form={object|relocate|hexadecimal|stype|binary} オプションを指定した場合は無効となります。

-CHange_message

インフォメーション、ワーニング、エラーのメッセージ種別を変更します。

[指定形式]

```
-CHange_message=level [=num [-num] [, num [-num]] ...] [, level [=num [-num] [, num [-num]] ...]] ...
```

- 省略時解釈
なし

[詳細説明]

- インフォメーション、ワーニング、エラーのメッセージ種別 *level* を変更します。
- メッセージ出力時の処理継続／中断を変更することができます。
- *level* に指定可能なものを以下に示します。

Information	インフォメーション
Warning	ワーニング
Error	エラー

- メッセージ番号 *num* を指定すると、指定した番号のメッセージの種別を変更します。
また、ハイフン (-) を使用して、メッセージ番号の範囲を指定することもできます。
- *num* には、コンポーネント番号 (05)、発生フェーズ (6) に続けて出力される 4 桁の数値 (E0562310 の場合は 2310) を指定します。
- メッセージ番号の指定を省略した場合は、すべてのメッセージの種別を指定したものに変わります。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の [メッセージ] カテゴリの [\[ワーニング・メッセージをインフォメーション・メッセージに変更する\]](#)、[\[ワーニング・メッセージの番号\]](#)、[\[インフォメーション・メッセージをワーニング・メッセージに変更する\]](#)、[\[インフォメーション・メッセージの番号\]](#)、[\[インフォメーション、ワーニング・メッセージをエラー・メッセージに変更する\]](#)、[\[インフォメーション、ワーニング・メッセージの番号\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の [メッセージ] カテゴリの [\[ワーニング・メッセージをインフォメーション・メッセージに変更する\]](#)、[\[ワーニング・メッセージの番号\]](#)、[\[インフォメーション・メッセージをワーニング・メッセージに変更する\]](#)、[\[インフォメーション・メッセージの番号\]](#)、[\[インフォメーション、ワーニング・メッセージをエラー・メッセージに変更する\]](#)、[\[インフォメーション、ワーニング・メッセージの番号\]](#)

【使用例】

- E0561310 をワーニングに変更し，E0561310 出力時も処理を継続します。

```
>rlink a.obj b.obj -change_message=warning=1310
```

- すべてのインフォメーション，ワーニングをエラーに変更します。
メッセージを1つでも出力すると，処理を中断します。

```
>rlink a.obj b.obj -change_message=error
```

-Hide

出力ファイル内のローカル・シンボル名情報を消去します。

[指定形式]

```
-Hide
```

- 省略時解釈
なし

[詳細説明]

- 出力ファイル内のローカル・シンボル名情報を消去します。
- ローカル・シンボルに関する名前の情報が消去されるため、バイナリ・エディタなどでファイルを開いてもローカル・シンボル名を確認することができなくなります。
なお、生成されるファイルの動作に対する影響は一切ありません。
- 本オプションは、ローカル・シンボル名を機密扱いにしたい場合などに指定してください。
- 秘匿対象となるシンボルの種類を以下に示します。
 - C ソース：static 型修飾子を指定した変数名、関数名など
 - C ソース：goto 文のラベル名
 - アセンブリ・ソース：外部定義（参照）シンボル宣言していないシンボル名なお、エントリ関数名は秘匿対象にはなりません。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の[デバッグ情報] カテゴリの[ローカル・シンボル名情報を消去する]
 - [\[ライブラリ生成オプション\] タブ](#)の[デバッグ情報] カテゴリの[ローカル・シンボル名情報を消去する]

[使用例]

- 出力ファイル内のローカル・シンボル名情報を消去します。

```
>rlink a.obj b.obj -hide
```

本オプションの機能が有効となる C ソース記述の例を以下に示します。

```
int g1;
int g2=1;
const int g3=3;

static int s1;          //<--- static 変数名は秘匿対象
static int s2=1;        //<--- static 変数名は秘匿対象
static const int s3=2;   //<--- static 変数名は秘匿対象

static int sub1()        //<--- static 関数名は秘匿対象
{
    static int s1;        //<--- static 変数名は秘匿対象
    int l1;

    s1 = l1; l1 = s1;
    return(l1);
}

int main()
{
    sub1();
    if (g1==1)
        goto L1;
    g2=2;
L1:                          //<--- goto 文のラベル名は秘匿対象
    return(0);
}
```

[備考]

- 本オプションは、-form={absolute|relocate|library} オプションを指定した場合のみ有効となります。
- 外部変数アクセス最適化を行う状況で本オプションを指定する場合は、一度目のリンク時には指定せず、二度目のリンク時にのみ指定してください。
- デバッグ情報内のシンボル名は、本オプションを指定しても削除されません。

-Total_size

リンク後のセクションの合計サイズを標準エラー出力に表示します。

[指定形式]

```
-Total_size
```

- 省略時解釈
なし

[詳細説明]

- リンク後のセクションの合計サイズを標準エラー出力に表示します。
- 以下の3種類のセクションに分けて、合計サイズを表示します。
 - 実行可能なプログラム・セクション
 - プログラム・セクション以外のROM領域配置セクション
 - RAM領域配置セクション
- 本オプションを指定することにより、ROM/RAMに配置する合計のセクション・サイズを容易に認識することができます。
- 本オプションは、CubeSuite+の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の[その他]カテゴリの[\[合計セクション・サイズを表示する\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の[その他]カテゴリの[\[合計セクション・サイズを表示する\]](#)

[使用例]

- リンク後のセクションの合計サイズを標準エラー出力に表示します。

```
>rlink a.obj b.obj -total_size
```

[備考]

- リンク・マップ・ファイルに合計サイズを表示するためには、-show=total_size オプションを指定する必要があります。
- ROM化支援機能(-rom オプション)対象のセクションの場合、転送元(ROM)と転送先(RAM)の両方で領域を使用するため、双方の合計サイズに対してセクション・サイズを加算します。

-LOgo

コピーライトを出力します。

[指定形式]

```
-LOgo
```

- 省略時解釈

コピーライトを出力します。

[詳細説明]

- コピーライトを出力します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の [\[その他\]](#) カテゴリの [\[コピーライト情報を表示する\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の [\[その他\]](#) カテゴリの [\[コピーライト情報を表示する\]](#)

[使用例]

- コピーライトを出力します。

```
>rlink a.obj b.obj -logo
```

-NOLOgo

コピーライトの出力を抑止します。

[指定形式]

```
-NOLOgo
```

- 省略時解釈

コピーライトを出力します（-logo オプションの指定と同じです）。

[詳細説明]

- コピーライトの出力を抑止します。
- 本オプションは、CubeSuite+ の以下のプロパティに相当します。
 - [\[リンク・オプション\] タブ](#)の [\[その他\]](#) カテゴリの [\[コピーライト情報を表示する\]](#)
 - [\[ライブラリ生成オプション\] タブ](#)の [\[その他\]](#) カテゴリの [\[コピーライト情報を表示する\]](#)

[使用例]

- コピーライトの出力を抑止します。

```
>rlink a.obj b.obj -nologo
```

-END

本オプションより前に指定したオプション列を実行します。

[指定形式]

```
-END
```

- 省略時解釈
なし

[詳細説明]

- 本オプションより前に指定したオプション列を実行します。
リンク処理の終了後に、本オプションより後に指定したオプション列の入力、リンク処理を継続します。

[注意]

- 本オプションは、サブコマンド・ファイル内のみで使用することができます。

[使用例]

- サブコマンド・ファイル test.sub を以下の内容で作成します。

```
input=a.obj,b.obj          ; (1)
start=P,C,D/100,B/8000     ; (2)
output=a.abs               ; (3)
end
input=a.abs                 ; (4)
form=stype                  ; (5)
output=a.mot                ; (6)
```

コマンド・ライン上でサブコマンド・ファイル test.sub を指定します。

```
>rlink -subcommand=test.sub
```

- (1) ~ (3) の処理を実行し、a.abs を出力します。
その後、(4) ~ (6) の処理を実行し、a.mot を出力します。

-EXIt

オプション指定の終了を指定します。

[指定形式]

```
-EXIt
```

- 省略時解釈
なし

[詳細説明]

- オプション指定の終了を指定します。

[注意]

- 本オプションは、サブコマンド・ファイル内のみで使用することができます。

[使用例]

- サブコマンド・ファイル test.sub を以下の内容で作成します。

```
input=a.obj,b.obj          ; (1)
start=P,C,D/100,B/8000     ; (2)
output=a.abs               ; (3)
exit
```

コマンド・ライン上でサブコマンド・ファイル test.sub を指定します。

```
>rlink -subcommand=test.sub -nodebug
```

(1) ～ (3) の処理を実行し、a.abs を出力します。

本オプションの実行後にコマンド・ライン上で指定している -nodebug オプションは無効になります。

B. 6 オプションの複数指定

ここでは、ccrh コマンドに対してオプションを同時に 2 つ以上指定した場合について説明します。

B. 6.1 優先順序

以下のオプションは、ほかの特定のオプションを無効とします。

-V/-h	ほかのすべてのオプションは無効となります。 このとき、警告を出力しません。
-P	ブリプロセス処理で終了するため、それ以降の処理に関するオプションは無効となります。 このとき、警告を出力します。
-S	コンパイル処理で終了するため、それ以降の処理に関するオプションは無効となります。 このとき、警告を出力します。
-c	アセンブル処理で終了するため、それ以降の処理に関するオプションは無効となります。 このとき、警告を出力します。
-Xcpu=g3k	-Xfloat オプションは無効となります。 このとき、警告を出力します。

以下の組み合わせでオプションを指定した場合は、警告を出力して、最後に指定したものが有効となります。

- -P, -S, -c
- -D, -U（シンボル名が同じ場合）
- -Onothing, -Odefault, -Osize, -Ospeed

なお、オプションの指定順序により、以下のオプションは無効となります。

- -Onothing, -Odefault, -Osize, -Ospeed の前に指定した *-Oitem* ^注

注 *-Oitem* : -Ounroll, -Oinline, -Oinline_size, -Odelete_static_func, -Opipeline, -Otail_call, -Omap, -Osmap

B. 6.2 機能矛盾

以下の組み合わせでオプションを指定した場合は、エラーとなります。

- -Omap, -Xep=fix
- -Osmap, -Xep=fix

B. 6.3 依存関係

以下のオプションは、ほかの特定のオプションにより動作が変わります。

-Xpreprocess	-P オプションと同時に指定しない場合、無効となります。 このとき、警告を出力しません。
-o	-P/-S/-c オプションと同時に指定した場合、生成ファイルの種別は、それぞれプリプロセス処理済みファイル／アセンブリ・ソース・ファイル／オブジェクト・ファイルとなります。
-g	-O オプションと同時に指定した場合、デバッグ情報が正しくないことがあります。
-Oinline	-Xmerge_files オプションと同時に指定した場合、ファイル間インライン展開を行う場合があります。

B. 6.4 #pragma 指定との関係

以下のオプションは、#pragma 指定との関係により動作が変わります。

- -Xep=callee, または -Xep の指定なし

#pragma section で以下の属性指定文字を同時に指定した場合は、エラーとなります。

edata, edata23, tdata, tdata4, tdata5, tdata7, tdata8, sdata, sdata23, ep_auto, ep_disp4,
ep_disp5, ep_disp7, ep_disp8, ep_disp16, ep_disp23, gp_disp16, gp_disp23

付録C 索引

【記号】

@ (アセンブル・オプション) ... 459
@ (コンパイル・オプション) ... 437

【A】

-ALIGNED_SECTION (リンク・オプション) ... 501

【B】

-Binary (リンク・オプション) ... 467
-Byte_count (リンク・オプション) ... 487

【C】

-CHange_message (リンク・オプション) ... 517
-COmpress (リンク・オプション) ... 508
-CPu (リンク・オプション) ... 502
-c (コンパイル・オプション) ... 382

【D】

-DEBug (リンク・オプション) ... 475
-DEFine (リンク・オプション) ... 469
-DElete (リンク・オプション) ... 513
-D (アセンブル・オプション) ... 448
-D (コンパイル・オプション) ... 383

【E】

-END (リンク・オプション) ... 524
-ENTry (リンク・オプション) ... 470
-EXIt (リンク・オプション) ... 525
-EXtract (リンク・オプション) ... 515

【F】

Far Jump ファイルを指定 ダイアログ ... 340
-FOrm (リンク・オプション) ... 472
-FSymbol (リンク・オプション) ... 499

【G】

-g (アセンブル・オプション) ... 445
-g (コンパイル・オプション) ... 377

【H】

-Hide (リンク・オプション) ... 519
-h (アセンブル・オプション) ... 441
-h (コンパイル・オプション) ... 371

【I】

-Input (リンク・オプション) ... 463
-I (アセンブル・オプション) ... 450
-I (コンパイル・オプション) ... 386

【L】

-LIBrary (リンク・オプション) ... 465
-LISt (リンク・オプション) ... 493
-LOgo (リンク・オプション) ... 522

【M】

-MAp (リンク・オプション) ... 481
-MEMory (リンク・オプション) ... 510
-Message (リンク・オプション) ... 484
MISRA-C:2004 ルール・ファイルの指定 ダイアログ ... 342
-MSg_unused (リンク・オプション) ... 486

【N】

-NOCompress (リンク・オプション) ... 509
-NODEBug (リンク・オプション) ... 476
-NOLOgo (リンク・オプション) ... 523
-NOMessage (リンク・オプション) ... 485

【O】

-OUTput (リンク・オプション) ... 479
-OVERRUN_FETCH (リンク・オプション) ... 490
-o (アセンブル・オプション) ... 442
-O (コンパイル・オプション) ... 402
-o (コンパイル・オプション) ... 372

【P】

-PADDING (リンク・オプション) ... 488
-P (コンパイル・オプション) ... 380

[R]

-RECOrd (リンク・オプション) ... 477
 -REName (リンク・オプション) ... 512
 -REPlace (リンク・オプション) ... 514
 -ROm (リンク・オプション) ... 478

[S]

-S9 (リンク・オプション) ... 505
 -SHow (リンク・オプション) ... 494
 -SPace (リンク・オプション) ... 482
 -STACK (リンク・オプション) ... 507
 -STARt (リンク・オプション) ... 496
 -STRip (リンク・オプション) ... 516
 -SUBcommand (リンク・オプション) ... 504
 -S (コンパイル・オプション) ... 381

[T]

-Total_size (リンク・オプション) ... 521

[U]

-U (アセンブル・オプション) ... 449
 -U (コンパイル・オプション) ... 385

[V]

-V (アセンブル・オプション) ... 440
 -V (コンパイル・オプション) ... 370

[X]

-Xalias (コンパイル・オプション) ... 410
 -Xansi (コンパイル・オプション) ... 391
 -Xasm_far_jump (アセンブル・オプション) ... 456
 -Xasm_option (コンパイル・オプション) ... 435
 -Xasm_path (コンパイル・オプション) ... 375
 -Xbit_order (コンパイル・オプション) ... 415
 -Xcall_jump (コンパイル・オプション) ... 423
 -Xcharacter_set (アセンブル・オプション) ... 451
 -Xcharacter_set (コンパイル・オプション) ... 401
 -Xcheck_div_ov (コンパイル・オプション) ... 427
 -Xcheck_language_extention (コンパイル・オプション)
 ... 400
 -Xcheck (コンパイル・オプション) ... 395
 -Xcommon (アセンブル・オプション) ... 446

-Xcommon (コンパイル・オプション) ... 378
 -Xcpu (コンパイル・オプション) ... 379, 447
 -Xcref (コンパイル・オプション) ... 432
 -Xdiv (コンパイル・オプション) ... 426
 -Xenum_type (コンパイル・オプション) ... 393
 -Xep (アセンブル・オプション) ... 455
 -Xep (コンパイル・オプション) ... 421
 -Xerror_file (アセンブル・オプション) ... 457
 -Xerror_file (コンパイル・オプション) ... 433
 -Xfar_jump (コンパイル・オプション) ... 424
 -Xfloat (コンパイル・オプション) ... 422
 -Xignore_files_misra (コンパイル・オプション) ...
 399
 -Xinline_strcpy (コンパイル・オプション) ... 408
 -Xintermodule (コンパイル・オプション) ... 406
 -Xlk_option (コンパイル・オプション) ... 436
 -Xmerge_files (コンパイル・オプション) ... 411
 -Xmerge_string (コンパイル・オプション) ... 409
 -Xmisra2004 (コンパイル・オプション) ... 397
 -Xmulti_level (コンパイル・オプション) ... 430
 -Xno_warning (アセンブル・オプション) ... 458
 -Xno_warning (コンパイル・オプション) ... 434
 -Xobj_path (アセンブル・オプション) ... 443
 -Xobj_path (コンパイル・オプション) ... 374
 -Xpack (コンパイル・オプション) ... 413
 -Xpass_source (コンパイル・オプション) ... 416
 -Xpatch (コンパイル・オプション) ... 431
 -Xpreinclude (コンパイル・オプション) ... 388
 -Xprep_path (コンパイル・オプション) ... 376
 -Xpreprocess (コンパイル・オプション) ... 389
 -Xprn_path (アセンブル・オプション) ... 444
 -Xreg_mode (アセンブル・オプション) ... 452
 -Xreg_mode (コンパイル・オプション) ... 418
 -Xreserve_r2 (アセンブル・オプション) ... 454
 -Xreserve_r2 (コンパイル・オプション) ... 420
 -Xswitch (コンパイル・オプション) ... 417
 -Xunordered_cmpf (コンパイル・オプション) ... 429
 -Xuse_fmaf (コンパイル・オプション) ... 428
 -Xvolatile (コンパイル・オプション) ... 394
 -Xwhole_program (コンパイル・オプション) ... 412

【あ行】

- アクティブ・プロジェクト … 73
- アセンブル・オプションの設定 … 38
- アセンブル・リストの出力 … 28
- アセンブル・リスト・ファイル … 93
 - アセンブル・リスト … 94
 - アセンブル・リストの構成 … 94
 - コマンド・ライン情報 … 95
 - セクション・リスト … 95
- 一括ビルド … 87
- インテル拡張ヘキサ・ファイル … 111
 - インテル拡張ヘキサ・ファイルの構成 … 111
 - エンド・レコード … 116
 - 拡張アドレス・レコード … 114
 - 拡張リニア・アドレス・レコード … 113
 - スタート・アドレス・レコード … 114
 - スタート・リニア・アドレス・レコード … 113
 - データ・レコード … 115
- インポートするファイルを選択 ダイアログ … 352
- エクスポートするファイルを選択 ダイアログ … 354
- エディタ パネル … 287
- オーバーレイ配置セクションの追加 ダイアログ … 312
- オプション ダイアログ … 329
 - [全般 - ビルド／デバッグ] カテゴリ … 331

【か行】

- カテゴリ … 20
- 既存のファイルを追加 ダイアログ … 334
- クリーン … 90
- コンパイル・オプションの設定 … 32

【さ行】

- 削除するセクション ダイアログ … 316
- システム・インクルード・パス順設定 ダイアログ … 305
- 出力 パネル … 288
- 出力ファイル名の変更 … 25
- 使用するライブラリ・ファイルを指定 ダイアログ … 346
- 処理中表示 ダイアログ … 328

- セクション設定 ダイアログ … 309
- セクション追加 ダイアログ … 312
- セクションのアドレス ダイアログ … 314
- セクション編集 ダイアログ … 312
- [全般 - ビルド／デバッグ] カテゴリ … 331

【た行】

- 対象外ファイルの追加 ダイアログ … 344
- タグ・ジャンプ … 289
- テキスト編集 ダイアログ … 298

【な行】

- 名前を付けて保存 ダイアログ … 348

【は行】

- パス編集 ダイアログ … 301
- バッチ・ビルド … 80, 86
- バッチ・ビルド ダイアログ … 325
- ビルド … 80, 83
- ビルド・オプションのインポート ダイアログ … 336
- ビルド・ツールのバージョン … 14
- ビルドの実行 … 80
- ビルド・モード … 75, 77
- ビルド・モード設定 ダイアログ … 322
- ビルド・モードの削除 … 78
- ビルド・モードの追加 … 75
- ビルド・モードの変更 … 77
- ファイル追加 ダイアログ … 291
- ファイルの依存関係 … 22
- ファイルの追加 … 15
- ファイルの表示順 … 21
- ファイルの保存設定 ダイアログ … 318
- フォルダとファイル追加 ダイアログ … 294
- フォルダの参照 ダイアログ … 338
- プログラムから開く ダイアログ … 350
- プロジェクト・ツリー パネル … 129
- プロパティ パネル … 145
 - [アセンブル・オプション] タブ … 198
 - [カテゴリ情報] タブ … 285
 - [共通オプション] タブ … 148
 - [個別アセンブル・オプション] タブ … 272
 - [個別コンパイル・オプション] タブ … 248

[コンパイル・オプション] タブ … 173
[ビルド設定] タブ … 245
[ファイル情報] タブ … 283
[ヘキサ出力オプション] タブ … 224
[ライブラリ生成オプション] タブ … 231
[リンク・オプション] タブ … 205
ヘキサ出力オプションの設定 … 54

【ま行】

マップ情報の出力 … 29
メイン・ウインドウ … 124
文字列入力 ダイアログ … 296
モトローラ・S タイプ・ファイル … 117
 S0 レコード … 119
 S1 レコード … 119
 S2 レコード … 119
 S3 レコード … 120
 S7 レコード … 120
 S8 レコード … 121
 S9 レコード … 121
モトローラ・S タイプ・ファイルの構成 … 117

【ら行】

ライブラリ情報の出力 … 30
ライブラリ生成オプションの設定 … 60
ライブラリ・リスト・ファイル … 107
 エラー情報 … 109
 オプション情報 … 108
 ライブラリ情報 … 109
 ライブラリ内モジュール、セクション、シンボル情報 … 110
 ライブラリ・リストの構成 … 107
ラピッド・ビルド … 80, 85
リビルド … 80, 84
リンク・オプションの設定 … 42
リンク順設定 ダイアログ … 320
リンク・マップ・ファイル … 96
 エラー情報 … 97
 オプション情報 … 97
クロス・リファレンス情報 … 101
合計セクション・サイズ … 98

シンボル情報 … 99
リンク・マップ情報 … 98
リンク・マップの構成 … 96
リンク・マップ・ファイル（オブジェクト結合時） … 103
アドレス重複情報 … 106
エラー情報 … 105
エントリ情報 … 105
オプション情報 … 104
結合アドレス情報 … 105
ヘッダ情報 … 104
リンク・マップの構成 … 103
ルール番号の指定 ダイアログ … 307

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.04.01	－	初版発行

CubeSuite+ V2.01.00 ユーザーズマニュアル
RH850 ビルド編

発行年月日 2013年4月1日 Rev.1.00

発行 ルネサス エレクトロニクス株式会社
〒211-8668 神奈川県川崎市中原区下沼部 1753



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/contact/>

CubeSuite+ V2.01.00