User's Manual

CubeSuite+ V1.02.00

統合開発環境

ユーザーズマニュアル　起動編

対象デバイス

78K0マイクロコントローラ

RL78ファミリ

78K0Rマイクロコントローラ

V850マイクロコントローラ

RXファミリ

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。

ルネサス エレクトロニクス
www.renesas.com

Rev.1.00　2012.03

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。
5. 輸出に際しては、「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。本資料に記載されている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の目的で使用しないでください。また、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に使用することができません。
6. 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ものですが、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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されてい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生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   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

   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、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準：　航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社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。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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，RX ファミリ，V850 マイクロコントローラ，RL78 ファミリ，78K0R マイクロコントローラ，78K0 マイクロコントローラ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，RX ファミリ，V850 マイクロコントローラ，RL78 ファミリ，78K0R マイクロコントローラ，78K0 マイクロコントローラの統合開発環境（ソフトウエア開発における，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+ の持つソフトウエ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エア，ソフトウエ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
付録 A　ウインドウ・リファレンス
付録 B　用語説明
付録 C　マニュアル構成
付録 D　入力規約
付録 E　正規表現の構文
付録 F　外部ビルド・ツールの使用
付録 G　Python コンソール /Python 関数
付録 H　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	：$\overline{\text{XXX}}$（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	... 0xXXXX

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	このマニュアル	R20UT0975E
	V850 設計編	R20UT0549J	R20UT0549E
	RL78 設計編	R20UT0976J	R20UT0976E
	78K0R 設計編	R20UT0547J	R20UT0547E
	78K0 設計編	R20UT0546J	R20UT0546E
	RX コーディング編	R20UT0767J	R20UT0767E
	V850 コーディング編	R20UT0553J	R20UT0553E
	コーディング編（CX コンパイラ）	R20UT0825J	R20UT0825E
	RL78, 78K0R コーディング編	R20UT0977J	R20UT0977E
	78K0 コーディング編	R20UT0782J	R20UT0782E
	RX ビルド編	R20UT0768J	R20UT0768E
	V850 ビルド編	R20UT0557J	R20UT0557E
	ビルド編（CX コンパイラ）	R20UT0826J	R20UT0826E
	RL78, 78K0R ビルド編	R20UT0730J	R20UT0730E
	78K0 ビルド編	R20UT0783J	R20UT0783E
	RX デバッグ編	R20UT1143J	R20UT1143E
	V850 デバッグ編	R20UT0734J	R20UT0734E
	RL78 デバッグ編	R20UT0978J	R20UT0978E
	78K0R デバッグ編	R20UT0732J	R20UT0732E
	78K0 デバッグ編	R20UT0731J	R20UT0731E
	解析編	R20UT0979J	R20UT0979E
	メッセージ編	R20UT0980J	R20UT0980E

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　…　9

1. 1　概　　要　…　9
1. 2　特　　長　…　10
1. 3　システム構成　…　11
1. 4　動作環境　…　12

第２章　機　　能　…　13

2. 1　CubeSuite+ をインストールする　…　13
2. 2　CubeSuite+ をアンインストールする　…　20
2. 3　ライセンス管理機能を使用する　…　22
　2. 3. 1　ライセンス情報を表示する　…　22
　2. 3. 2　ライセンスを追加する　…　23
　2. 3. 3　ライセンスを削除する　…　23
2. 4　アップデート機能を使用する　…　24
　2. 4. 1　手動アップデートを行う　…　24
　2. 4. 2　自動アップデートを行う　…　30
　2. 4. 3　アップデートを中断する　…　32
　2. 4. 4　アップデートを再開する　…　32
　2. 4. 5　アップデートの履歴を表示する　…　33
　2. 4. 6　非表示にしたアップデートを復元する　…　36
2. 5　CubeSuite+ を起動する　…　38
2. 6　プロジェクトを作成する　…　40
　2. 6. 1　プロジェクトをスタートする　…　44
　2. 6. 2　新しいプロジェクトを作成する　…　45
　2. 6. 3　新しいサブプロジェクトを追加する　…　49
2. 7　プロジェクトを操作する　…　51
　2. 7. 1　プロジェクトを開く　…　51
　2. 7. 2　既存のサブプロジェクトを追加する　…　53
　2. 7. 3　プロジェクトをお気に入りメニューに追加する　…　53
　2. 7. 4　プロジェクトからサブプロジェクトを外す　…　54
　2. 7. 5　プロジェクト名を変更する　…　55
　2. 7. 6　プロジェクト・フォルダをエクスプローラで開く　…　55
　2. 7. 7　CA850 のプロジェクトを CX のプロジェクトに変換する　…　57
　2. 7. 8　CubeSuite のプロジェクトを CubeSuite+ のプロジェクトに変換する　…　62
　2. 7. 9　HEW のプロジェクトを CubeSuite+ のプロジェクトに変換する　…　65
　2. 7. 10　PM+ のプロジェクトを CubeSuite+ のプロジェクトに変換する　…　73
2. 8　プロジェクト・ファイルを保存する　…　81
　2. 8. 1　プロジェクト・ファイルを別名で保存する　…　81
　2. 8. 2　すべてのファイルを保存する　…　82
　2. 8. 3　プロジェクトと本製品をパックして保存する　…　83
　2. 8. 4　プロジェクトを閉じる　…　87

2. 9　ウインドウ・レイアウトを変更する　…　88
2. 9. 1　パネルを自動的に隠す　…　88
2. 9. 2　ウインドウを独立させる　…　88
2. 9. 3　ウインドウをドッキングする　…　89
2. 9. 4　パネルを複数表示する　…　89
2. 9. 5　ウインドウ・レイアウトをリセットする　…　90
2. 10　Python 関数を実行する　…　91
2. 11　CubeSuite+ をコマンドラインで操作する　…　92

付録 A　ウインドウ・リファレンス　…　97

A. 1　説　　明　…　97

付録 B　用語説明　…　315

付録 C　マニュアル構成　…　316

付録 D　入力規約　…　318

D. 1　入力規約　…　318
D. 2　入力不備箇所に対するアイコン表示　…　319

付録 E　正規表現の構文　…　320

E. 1　文字のエスケープ　…　320
E. 2　文字クラス　…　320
E. 3　サポートされている Unicode 一般カテゴリ　…　321
E. 4　量指定子　…　322
E. 5　アトミック・ゼロ幅アサーション　…　323
E. 6　グループ化構成体　…　323
E. 7　置換構成体　…　324
E. 8　その他の構成体　…　324

付録 F　外部ビルド・ツールの使用　…　325

F. 1　概　　要　…　325
F. 2　デバッグ専用プロジェクトを作成する　…　326
F. 3　プロジェクトにファイルを追加する　…　327
F. 3. 1　ダウンロード・ファイルを追加する　…　327
F. 3. 2　C ソース・ファイル／その他のファイルを追加する　…　329
F. 3. 3　追加したファイルをプロジェクトから外す　…　333
F. 4　ビルドの設定をする　…　334
F. 4. 1　コマンドを設定する　…　334
F. 4. 2　ビルド・モードを設定する　…　338
F. 4. 3　ビルド対象プロジェクトを設定する　…　341

F. 5　ビルドを実行する　…　343
F. 5. 1　ビルドを実行する　…　344
F. 5. 2　リビルドを実行する　…　345
F. 5. 3　クリーンを実行する　…　346
F. 5. 4　ラピッド・ビルドを実行する　…　348
F. 5. 5　バッチ・ビルドを実行する　…　349
F. 5. 6　ビルドの実行を中止する　…　350
F. 5. 7　ビルド結果をファイルに保存する　…　351

付録 G　Python コンソール /Python 関数　…　352

G. 1　概　　要　…　352
G. 2　関連ファイル　…　352
G. 3　CubeSuite+ Python 関数／クラス／プロパティ　…　353
G. 3. 1　CubeSuite+ Python 関数（共通）　…　354
G. 3. 2　CubeSuite+ Python 関数（プロジェクト用）　…　362
G. 3. 3　CubeSuite+ Python 関数（ビルド・ツール用）　…　371
G. 3. 4　CubeSuite+ Python 関数（デバッグ・ツール用）　…　377
G. 3. 5　CubeSuite+ Python クラス　…　455
G. 3. 6　CubeSuite+ Python プロパティ（共通）　…　460
G. 3. 7　CubeSuite+ Python プロパティ（プロジェクト用）　…　468
G. 3. 8　CubeSuite+ Python プロパティ（ビルド・ツール用）　…　474
G. 3. 9　CubeSuite+ Python プロパティ（デバッグ・ツール用）　…　477
G. 3. 10　CubeSuite+ Python イベント　…　483
G. 4　Python コンソールの注意事項　…　486

付録 H　索　　引　…　487

第 1 章　概　　説

この章では，マイクロコントローラ（RX，V850，RL78，78K0R，78K0）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+ の役割，および機能概要について説明します。

1. 1　概　　要

CubeSuite+ は，マイクロコントローラ（RX，V850，RL78，78K0R，78K0）用の統合開発環境[注]（IDE：Integrated Development Environment）です。

各開発フェーズに必要なツールを統合することで，さまざまなツールを使い分ける必要がなくなり，本製品のみでソフトウエア開発における全フェーズを行うことができます。

また，統合されたツールは連携性を重視した設計が行われているため，設計フェーズのアウトプットがデバッグ・フェーズでも自動的に反映されるなど，さまざまなシーンにおける開発効率の向上を実現しています。

さらに，ネットワークを介して本製品のバージョンアップを自動的に行うためのアップデート機能を有し，ソフトウエア開発に必要な環境を容易に構築することができます（フリー・ダウンロード・ツール除く）。

注　統合開発環境は，ソフトウエア開発における，設計，実装（コーディング，ビルド），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フレームワークに統合した開発環境です。

備考　CubeSuite+ のほかに，用途にあわせて，エミュレータやオンチップ・デバッグ・エミュレータ（オンチップ・デバッグ機能内蔵マイクロコントローラ用に接続するエミュレータ），リアルタイム OS パッケージ（リアルタイム OS を使ったシステムの開発に利用）を用意することで，幅広い開発を行うことができます。

1.2 特 長

CubeSuite+ の特長を次に示します。

(1) プロジェクト管理

ソース・ファイルの構成，ビルド・オプション，およびデバッグ・ツールとの接続設定といったプロジェクト情報を管理することができます。

(2) 設計

端子配置機能では，マイクロコントローラの端子配置状況を入力することにより，端子配置表，端子配置図といったレポート・ファイルを出力することができます。

コード生成機能では，マイクロコントローラが提供している周辺機能（システム，ポート，割り込みなど）を制御する際に必要な情報を CubeSuite+ のパネル上で選択／入力することにより，対応するソース・コード（デバイス・ドライバ・プログラム）を出力することができます。

(3) コーディング

CubeSuite+ のパネル上でプロジェクトに含まれるファイルをツリー表示し，各種エディタと連携することにより，ファイルを編集することができます。

(4) ビルド

CubeSuite+ のパネル上で最適化などのビルド・オプションを設定することにより，効率の良いロード・モジュール・ファイル，またはライブラリ・ファイルを作成することができます。

備考 CubeSuite+ が提供するビルド・ツールを使用せず，外部のビルド・ツールと連携して使用することもできます（「付録 F 外部ビルド・ツールの使用」参照）。

(5) デバッグ

CubeSuite+ のパネル上で，デバッグ・ツールの接続設定やデバッグ情報の表示を行うことができます。また，様々なプログラム実行方法があり，効率良くデバッグすることができます。

(6) 解析

ソース・プログラム，およびプログラム実行時の情報を解析し，関数情報／変数情報の表示を行うことができます。

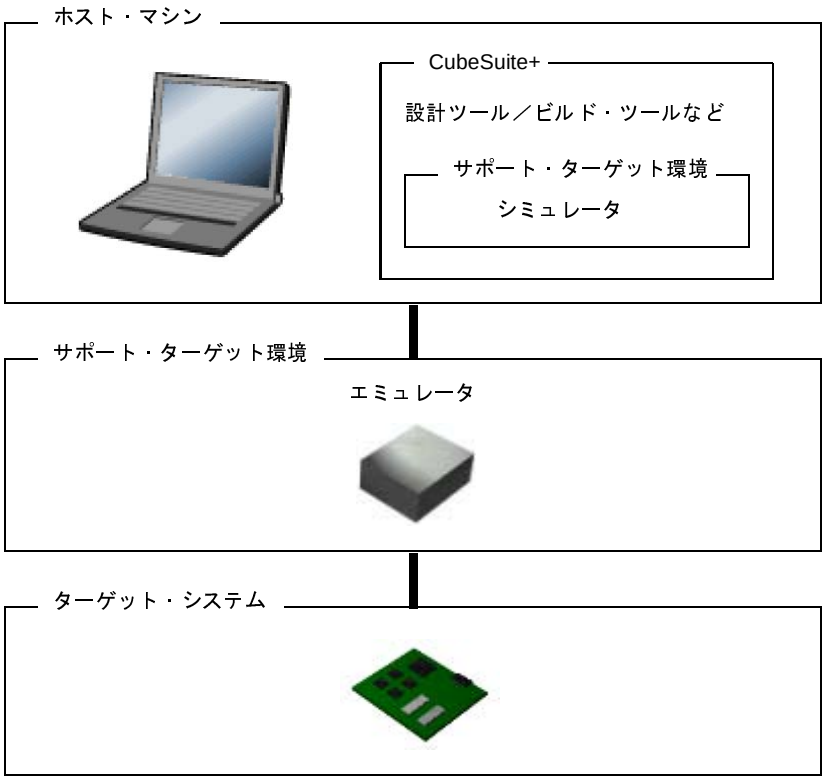
(7) アップデート

アップデート・サーバと通信することにより，本製品の最新版を入手することができます。

1. 3 システム構成

以下に，システム構成例を示します。

図 1—1 システム構成



備考 接続可能なエミュレータは，使用するマイクロコントローラによって異なります。
詳細については，「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

1. 4 動作環境

本製品は，以下の環境において動作します。

（1）ハードウエア環境

プロセッサ	：	1GHz 以上（ハイパー・スレッディング，マルチコア CPU に対応）
メイン・メモリ	：	1G バイト以上（Windwos 7（64 ビット版）は 2G バイト以上），推奨 2G バイト以上
ディスプレイ	：	1024 × 768 以上の解像度，65536 色以上
インタフェース	：	USB2.0

（2）ソフトウエア環境

- Windows XP（32 ビット版のみ）
- Windows Vista（32 ビット版，64 ビット版）
- Windows 7（32 ビット版，64 ビット版）
- Microsoft .NET Framework 3.5 SP1 ＋言語パック
- Microsoft Visual C++ 2008 SP1 ランタイム・ライブラリ
- Internet Explorer 6.0 以上

（3）サポート・ターゲット環境

- エミュレータ
  - IECUBE
  - IECUBE2
  - MINICUBE
  - MINICUBE2
  - E1
  - E20
  - EZ Emulator
- シミュレータ

第 2 章　機　　能

この章では，CubeSuite+ のインストール方法，ライセンス管理機能やアップデート機能の使用方法，および開発を始めるまでに必要な準備作業について説明します。

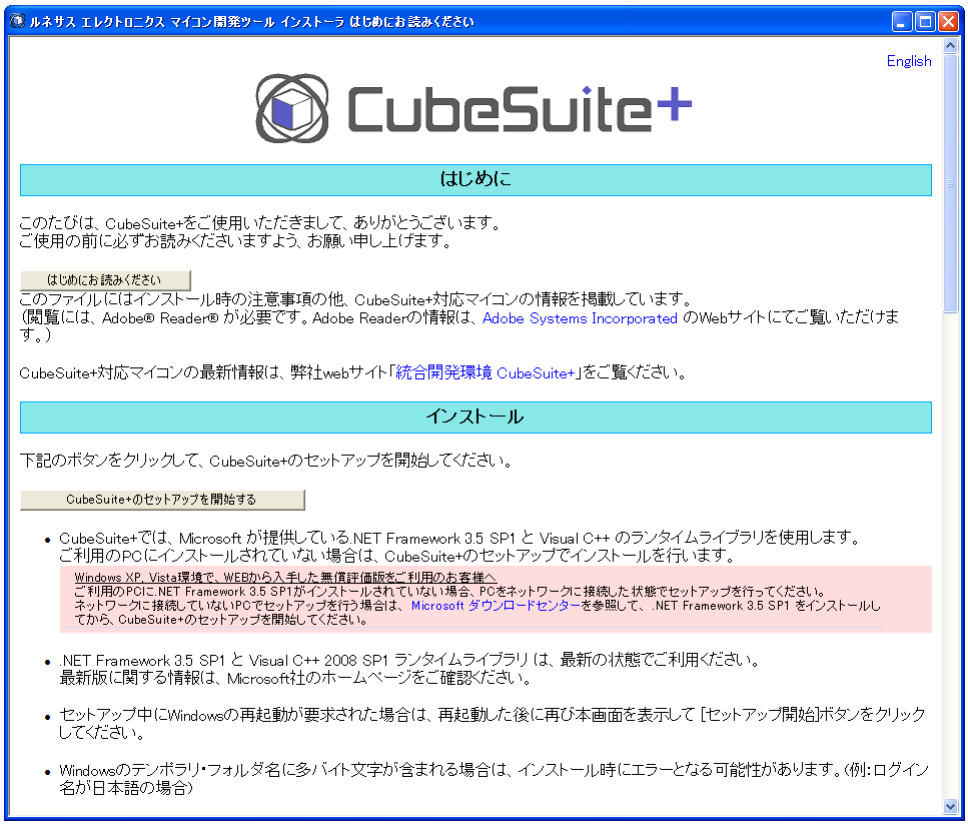
2. 1　CubeSuite+ をインストールする

CubeSuite+ のインストール方法について説明します。

(1) DVD をドライブに挿入します。

自動的に，インストール準備画面が表示されます。

図 2—1　インストール準備画面



［CubeSuite+ のセットアップを開始する］ボタンをクリックして，CubeSuite+ のセットアップを開始してください。

注意　CubeSuite+ のセットアップを行う場合，管理者権限が必要となります。

備考 自動的に表示されない場合，DVD 中の Install.hta を開いてください。

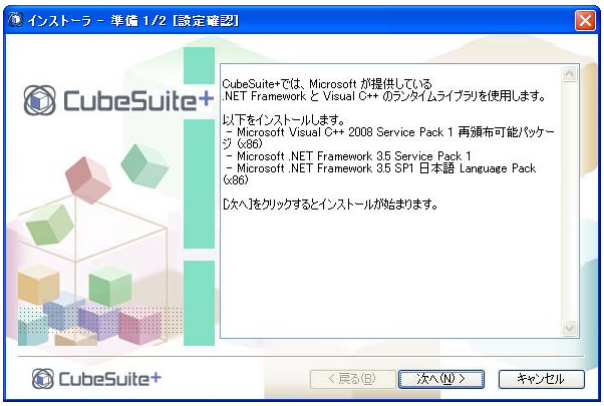
(2) インストールが必要な環境を確認します。

本製品が動作するためにインストールが必要な以下の環境のうち，未インストールのものが表示されます。

- Microsoft .NET Framework 3.5 SP1
- Microsoft .NET Framework 3.5 SP1 日本語言語パック
- Microsoft Visual C++ 2008 SP1 ランタイム・ライブラリ

注意 上記がすべてインストール済みの場合，本ウインドウは表示されません。

図 2—2 設定確認 ウインドウ



表示内容を確認し，［次へ］ボタンをクリックしてください。

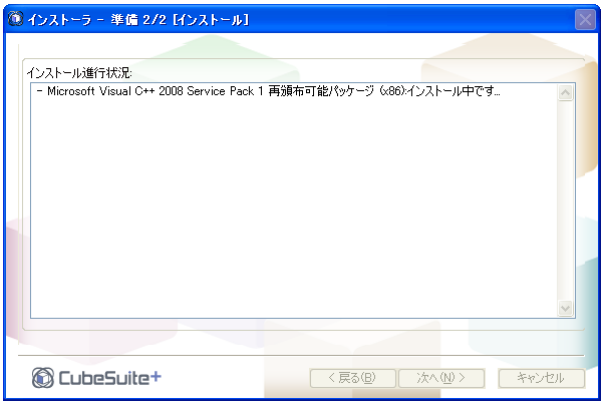
(3) 各環境のインストールの進行状況を確認します。

各環境のインストール・ウィザードにしたがって，インストールしてください。

インストール状況は，本ウインドウの［インストール進行状況］エリアに表示されます。

注意 必要な環境がすべてインストール済みの場合，本ウインドウは表示されません。

図 2—3 インストール ウインドウ



必要な環境のインストールがすべて完了した際，［次へ］ボタンが有効になります。
表示内容を確認し，［次へ］ボタンをクリックしてください。

備考 環境のインストールに失敗，または再起動が必要だった場合は，その旨のメッセージが表示され，インストールの継続はできません（［次へ］ボタンがクリックできません）。
再起動が必要な場合は，Windows を再起動した後，（1）のインストール準備画面の［CubeSuite+ のセットアップを開始する］ボタンからセットアップを再度始めてください。

（4）製品情報などを確認します。

製品情報などが表示されます。

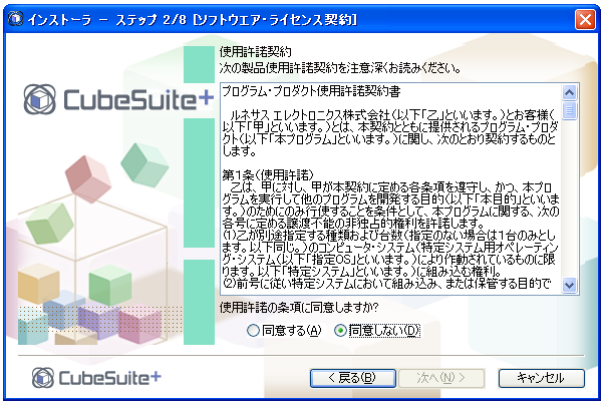
図 2—4 セットアップ開始時画面

表示内容を確認し，［次へ］ボタンをクリックしてください。

（5）使用許諾契約を確認します。

使用許諾契約に関するプログラム・プロダクト使用許諾契約書が表示されます。

図 2—5 ソフトウエア・ライセンス契約 ウインドウ



表示内容を確認し，使用許諾の条項に同意する場合は，「同意する」を選択後，［次へ］ボタンをクリックしてください。

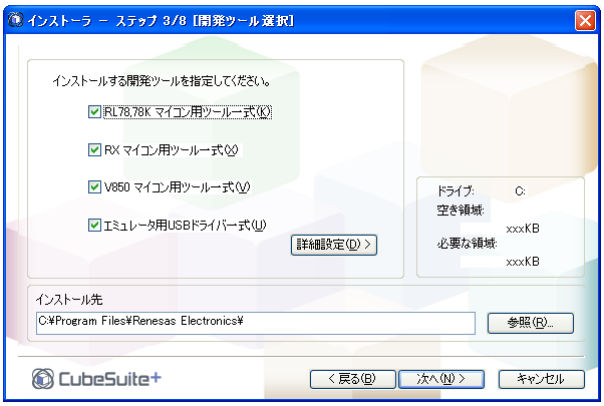
注意 ［同意しない］を選択した場合，インストールを続けることができません。

(6) インストール対象とインストール先を指定します。

インストール対象を選択してください。

また，インストール先を変更する場合は，［インストール先］エリアの内容を編集してください。

図 2—6 開発ツール選択 ウインドウ



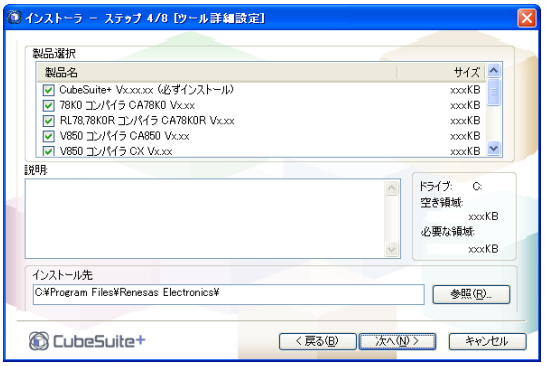
インストール対象，およびインストール先を指定後，［次へ］ボタンをクリックしてください。

注意 CubeSuite+ がインストールされているホスト・マシンでは，インストール先の変更はできません。インストール先を変更する場合は，CubeSuite+ をアンインストールしてから，インストールを行ってください。

備考 インストール対象を細かく指定する場合は，［詳細設定］ボタンをクリックしてください。ツール詳細設定 ウインドウが表示されます。［製品選択］エリアには，開発ツール選択 ウインドウで選択したインストール対象に対応した製品の詳細が表示されます。

［製品選択］エリアでツールを選択後，［次へ］ボタンをクリックしてください。

図 2—7 ツール詳細設定 ウインドウ



注意 必ずインストールしなければならない製品のチェックを外すことはできません。

(7) ライセンスを登録します。

ライセンスを登録することにより，以下の制限が解除されます。

- 開発可能なコード・サイズが限られます。

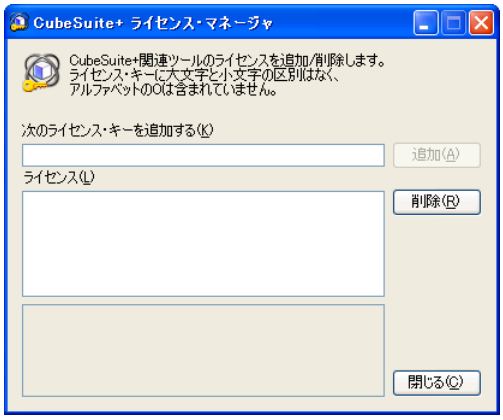
RX ファミリ	：128K バイトまで
V850 マイクロコントローラ	：256K バイトまで
RL78 ファミリ	：64K バイトまで（使用可能な RAM は内部 RAM のみ）
78K0R マイクロコントローラ	：64K バイトまで（使用可能な RAM は内部 RAM のみ）
78K0 マイクロコントローラ	：64K バイトまで（使用可能な RAM は内部 RAM のみ）

備考 ライセンスの登録は，本製品のインストール後，CubeSuite+ ライセンス・マネージャ ウインドウから行うことも可能です。

図 2—8 ライセンス登録 ウインドウ

［ライセンス・マネージャ起動］ボタンをクリックします。

図 2—9　CubeSuite+ ライセンス・マネージャ ウインドウ



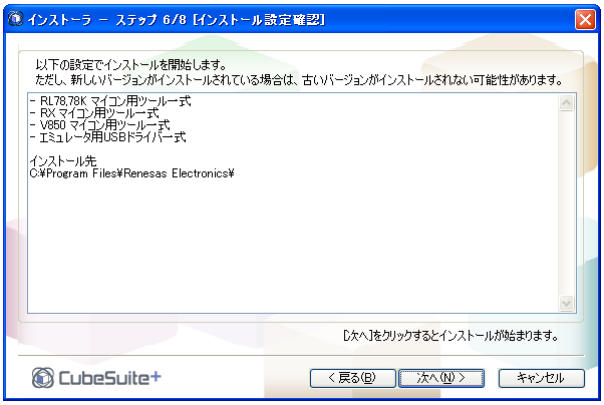
CubeSuite+ ライセンス・マネージャ ウインドウでライセンスを登録後，ライセンス登録 ウインドウの［次へ］ボタンをクリックします。

注意　CubeSuite+ ライセンス・マネージャ ウインドウ起動中は，インストーラ操作はできません。

（8）インストール対象とインストール先を確認します。

開発ツール選択 ウインドウ，またはツール詳細設定 ウインドウで設定した内容が表示されます。

図 2—10　インストール設定確認 ウインドウ



表示内容を確認し，［次へ］ボタンをクリックしてください。

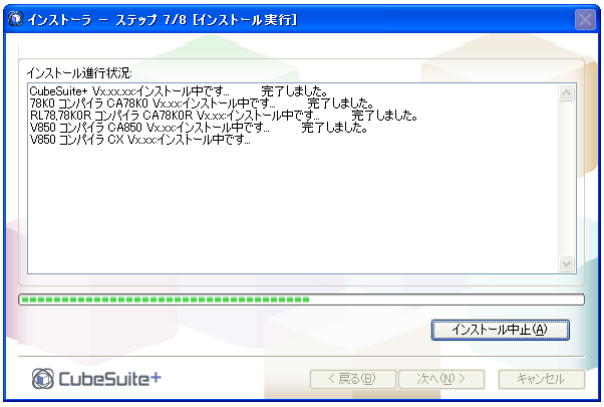
注意　Windows Vista，Windows 7 上でエミュレータ用 USB ドライバー式のインストールを行った場合，警告のダイアログが表示される場合があります。

（9）インストールの進行状況を確認します。

プログレスバーによる進行状況表示が行われます。

各ツールのインストール状況は，［インストール進行状況］エリアに表示されます。

図 2—11 インストール実行 ウインドウ



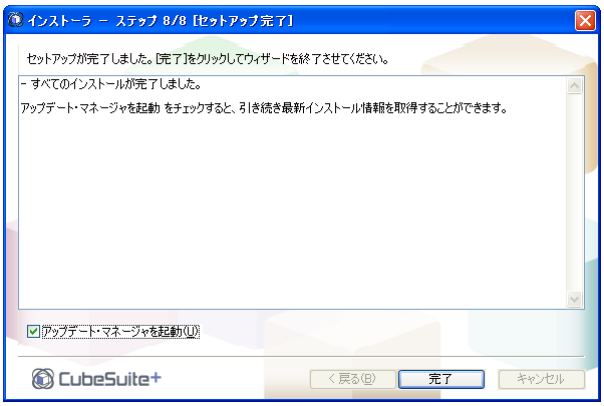
インストール対象のインストールがすべて完了した際，［次へ］ボタンが有効になります。
表示内容を確認し，［次へ］ボタンをクリックしてください。

備考 ［インストール中止］ボタンをクリックすると，［インストール進行状況］エリアに「インストール中止を受け付けました。」と表示し，インストール中のツールのインストールが終了した時点で，インストールを中止します。

（10）インストールの実行結果を確認します。

セットアップに関する一連の操作が終了すると，インストールの実行結果が表示されます。

図 2—12 セットアップ完了 ウインドウ



［完了］ボタンをクリックし，インストールを完了します。

備考 1. ［アップデート・マネージャを起動］チェック・ボックスにチェックがある場合は，インストール完了後に CubeSuite+ アップデート・マネージャ ウインドウが表示されます。
アップデート方法についての詳細は，「2. 4 アップデート機能を使用する」を参照してください。

2. （2）～（3）にて必要な環境をインストールした場合は，最新版へのアップデートを促すメッセージが表示されます。また，再起動が必要な場合は，その旨のメッセージも表示されます。

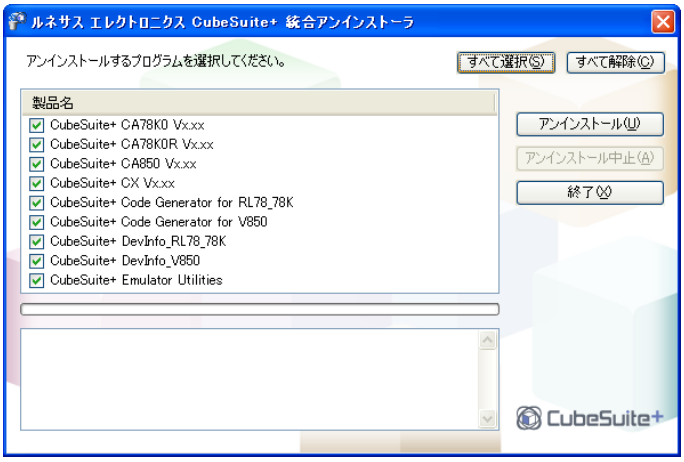
2. 2 CubeSuite+ をアンインストールする

CubeSuite+ をアンイストールするには，“統合アンインストーラ”から削除してください。

(1) 統合アンインストーラを起動します。

Windows の［スタート］メニューから［すべてのプログラム］→［Renesas Electronics CubeSuite+］→［統合アンインストーラ］を選択すると，統合アンインストーラ ウインドウが表示されます。

図 2—13 統合アンインストーラ ウインドウ



［すべて選択］ボタンをクリックすることにより，すべてのチェック・ボックスにチェックが入ります。

［すべて解除］ボタンをクリックすることにより，チェック・ボックスのすべてのチェックを外します。

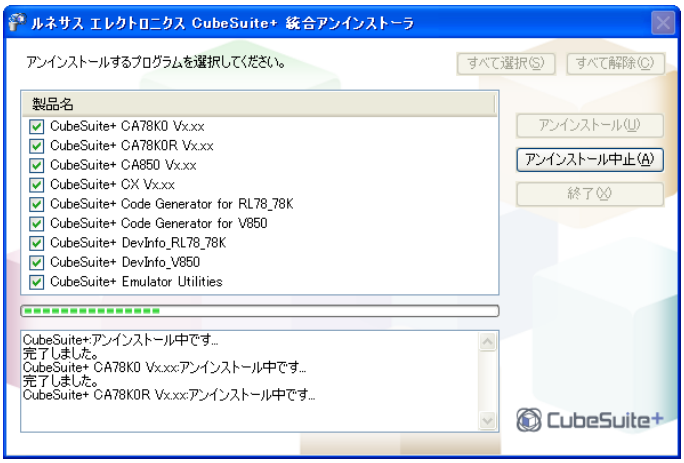
［アンインストール］ボタンをクリックしてください。

(2) アンインストールの進行状況を確認します。

プログレスバーによる進行状況表示が行われます。

各ツールのアンインストール状況は，進行状況詳細エリアに表示されます。

図 2—14 統合アンインストーラ ウインドウ（進行状況）

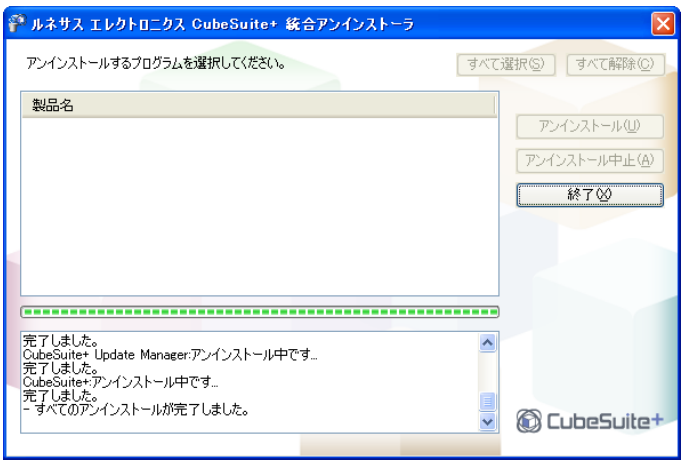


備考 ［アンインストール中止］ボタンをクリックすると，進行状況詳細エリアに「アンインストール中止を受け付けました。」と表示し，アンインストール中のツールのアンインストールが終了した時点で，アンインストールを中止します。

(3) アンインストールの実行結果を確認します。

アンインストールに関する一連の操作が終了すると，アンインストールの実行結果が表示されます。

図 2—15 統合アンインストーラ ウインドウ（実行結果）



［終了］ボタンをクリックし，アンインストールを完了します。

注意 全製品のアンインストールを行うと，統合アンインストーラも自動的にアンインストールされます。

2. 3 ライセンス管理機能を使用する

ライセンス管理機能とは，下記の方法で，CubeSuite+ ライセンス・マネージャ ウインドウを起動することにより，正規に取得した CubeSuite+ や関連ツールのライセンスをホスト・マシン に登録したり，登録されているライセンスの管理などを行う機能です。

- Windows の［スタート］メニューから ［すべてのプログラム］→ ［Renesas Electronics CubeSuite+］→ ［CubeSuite+ ライセンス・マネージャ］ を選択
- インストール時に，ライセンス登録 ウインドウの［ライセンス・マネージャ起動］ボタンをクリック

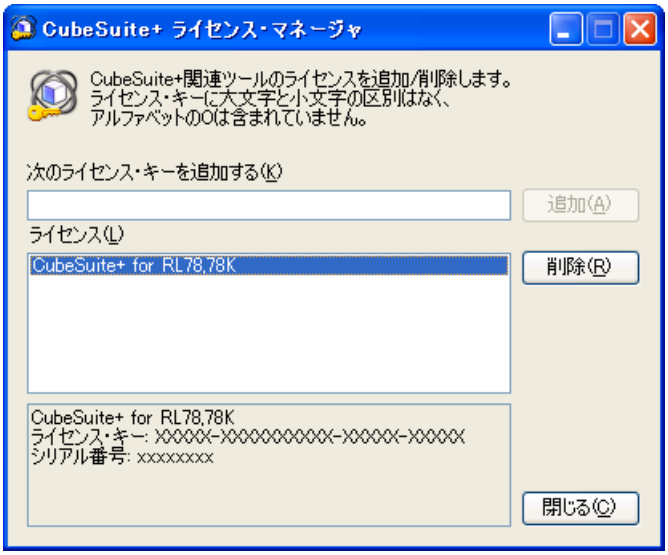
ライセンスを登録しない場合，以下の制限が生じます。

- 開発可能なコード・サイズ
  - RX ファミリ ： 128K バイトまで
  - V850 マイクロコントローラ ： 256K バイトまで
  - RL78 ファミリ ： 64K バイトまで
  - 78K0R マイクロコントローラ ： 64K バイトまで
  - 78K0 マイクロコントローラ ： 64K バイトまで

2. 3. 1 ライセンス情報を表示する

CubeSuite+ ライセンス・マネージャ ウインドウを起動すると，［ライセンス］エリアに有効な登録済みライセンスのリストが表示されます。

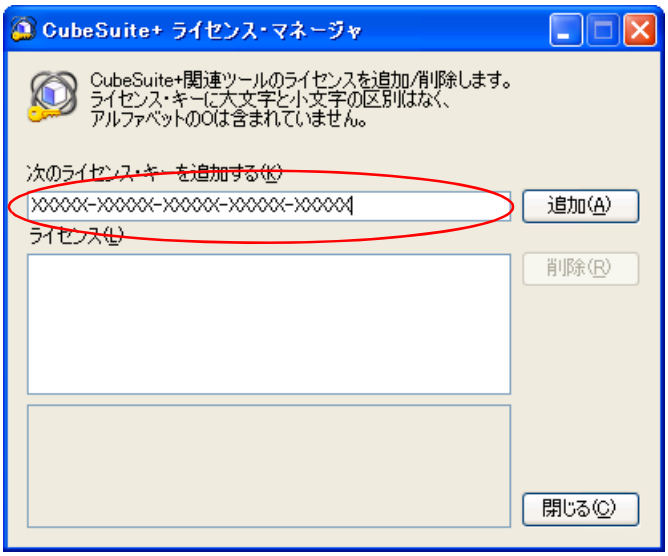
図 2—16 CubeSuite+ ライセンス・マネージャ ウインドウ



2. 3. 2 ライセンスを追加する

CubeSuite+ ライセンス・マネージャ ウインドウの［次のライセンス・キーを追加する］テキスト・ボックスに，ライセンス・キーを入力し，［追加］ボタンをクリックすることによりライセンスが追加されます。

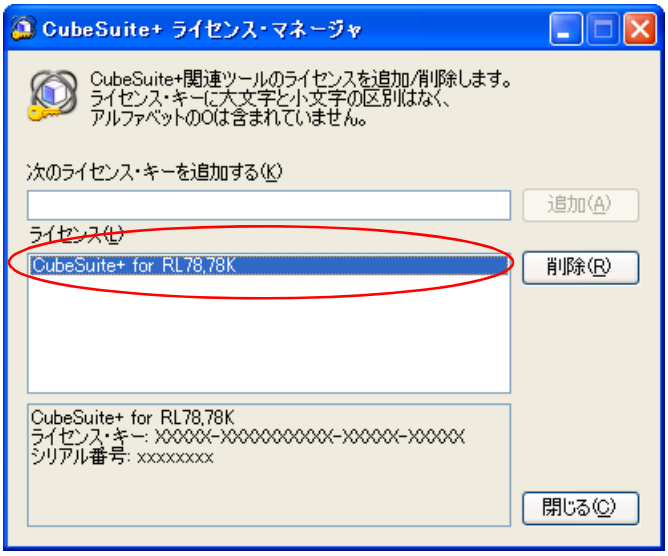
図 2—17 CubeSuite+ ライセンス・マネージャ ウインドウ



2. 3. 3 ライセンスを削除する

CubeSuite+ ライセンス・マネージャ ウインドウの［ライセンス］エリアに表示されているリストから，削除したいライセンスを選択し，［削除］ボタンをクリックすることにより，ライセンスの削除が行われます。

図 2—18 CubeSuite+ ライセンス・マネージャ ウインドウ



2.4 アップデート機能を使用する

アップデート機能とは，インストールされているツールやドキュメントを最新バージョンに更新する機能です。

インターネットを介して最新のバージョン情報を取得し，ダウンロードとインストールを行います。

アップデートの方法には，表示情報の指示にしたがってアップデートを行う手動アップデートと，設定された内容にしたがって自動的にアップデートを行う自動アップデートがあります。

注意 アップデート機能を利用する際には，ホスト・マシンがインターネットに接続している必要があります。

2.4.1 手動アップデートを行う

手動でアップデートを行う方法について説明します。

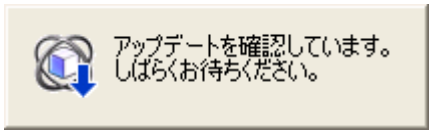
(1) アップデート・マネージャを起動します。

Windows の［スタート］メニューから［すべてのプログラム］→［Renesas Electronics CubeSuite+］→［CubeSuite+ アップデート・マネージャ］を選択，または CubeSuite+ の［ヘルプ］メニューから［アップデートの確認 ...］を選択すると，CubeSuite+ アップデート・マネージャ ウインドウが表示されます。

図 2—19 CubeSuite+ アップデート・マネージャ ウインドウ

備考 アップデート情報の取得中は，アップデートの確認中 ダイアログが表示されます。

図 2—20 アップデートの確認中 ダイアログ



(2) アップデート項目を選択します。

アップデートの選択 ページに適用可能なアップデート項目の一覧が表示されます。

図 2—21 アップデートの選択 ページ

アップデート対象選択エリアにおいて，アップデート対象をチェックし，［ダウンロードとインストール］または［ダウンロードのみ］をクリックしてください。

- ［ダウンロードとインストール］をクリックした場合：(3)
- ［ダウンロードのみ］をクリックした場合 ：(5)

備考 1. メニュー・エリアで，マイクロコントローラを選択すると，選択されたマイクロコントローラに限定されたアップデート項目のみが表示されます。
CubeSuite+ の［ヘルプ］メニューから［アップデートの確認 ...］を選択して CubeSuite+ アップデート・マネージャ ウインドウを起動した場合は，CubeSuite+ で開いているプロジェクトのマイクロコントローラに対応したアップデート項目のみが表示されます。

2. アップデート対象選択エリアにおいて，［すべて選択］をクリックすると，すべてのアップデート対象がチェックされます。

3. アップデート情報の取得に失敗した場合は，エラー ページが表示されます。

(3) ダウンロードとインストールを行います。

ダウンロードとインストール ページにアップデートの選択 ページで選択したアップデート項目が表示されます。

備考　アップデートの選択 ページで選択されたアップデート対象と同時にインストールする必要のあるアップデートがある場合は，メッセージ ダイアログが表示されます。
メッセージ ダイアログで［はい］ボタンをクリックすると，それらもアップデート対象となります。

図 2—22　ダウンロードとインストール ページ

表示内容を確認し，［インストール］ボタンをクリックしてください。

(4) ダウンロード，およびインストールの進行状況を確認します。

ダウンロードを行った後，インストールを行い，アップデート実行中 ダイアログにプログレスバーによる進行状況表示が行われます。

各ツールのダウンロード，およびインストール状況は，［アップデートの状態］エリアに表示されます。

図 2—23 アップデート実行中 ダイアログ（ダウンロード中）

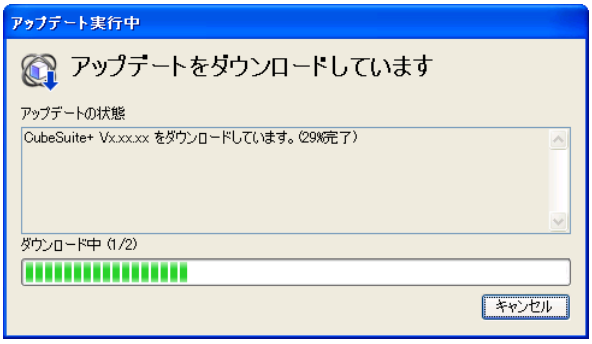
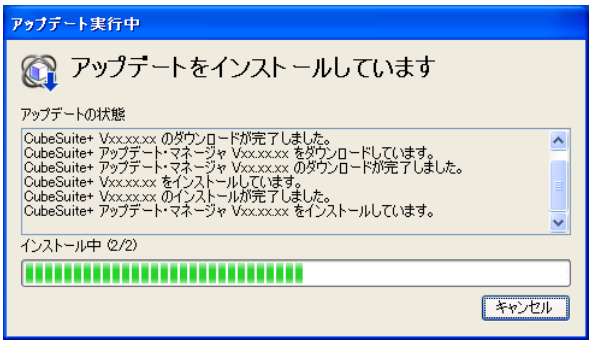


図 2—24 アップデート実行中 ダイアログ（インストール中）



注意 1. インストールを行うには管理者権限が必要です。

2. Windows Vista，Windows 7 上でエミュレータ用 USB ドライバー式のインストールを行った場合，警告のダイアログが表示される場合があります。

（7）へ進みます。

（5）ダウンロードのみを行います。

ダウンロードのみ ページにアップデートの選択 ページで選択したアップデート項目が表示されます。

図 2—25 ダウンロードのみ ページ

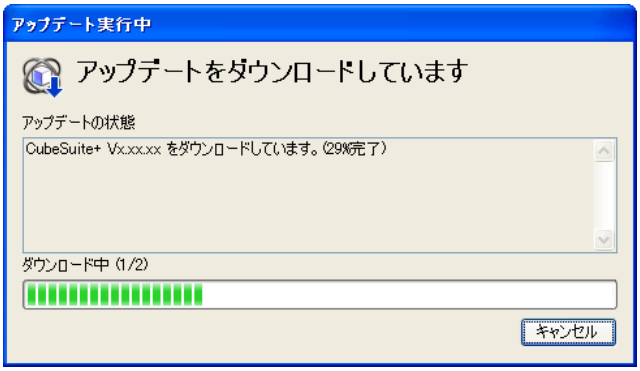
表示内容を確認し，［ダウンロード］ボタンをクリックします。

(6) ダウンロードの進行状況を確認します。

ダウンロードを行い，アップデート実行中 ダイアログにプログレスバーによる進行状況表示が行われます。

各ツールのダウンロード状況は，［アップデートの状態］エリアに表示されます。

図 2—26 アップデート実行中 ダイアログ

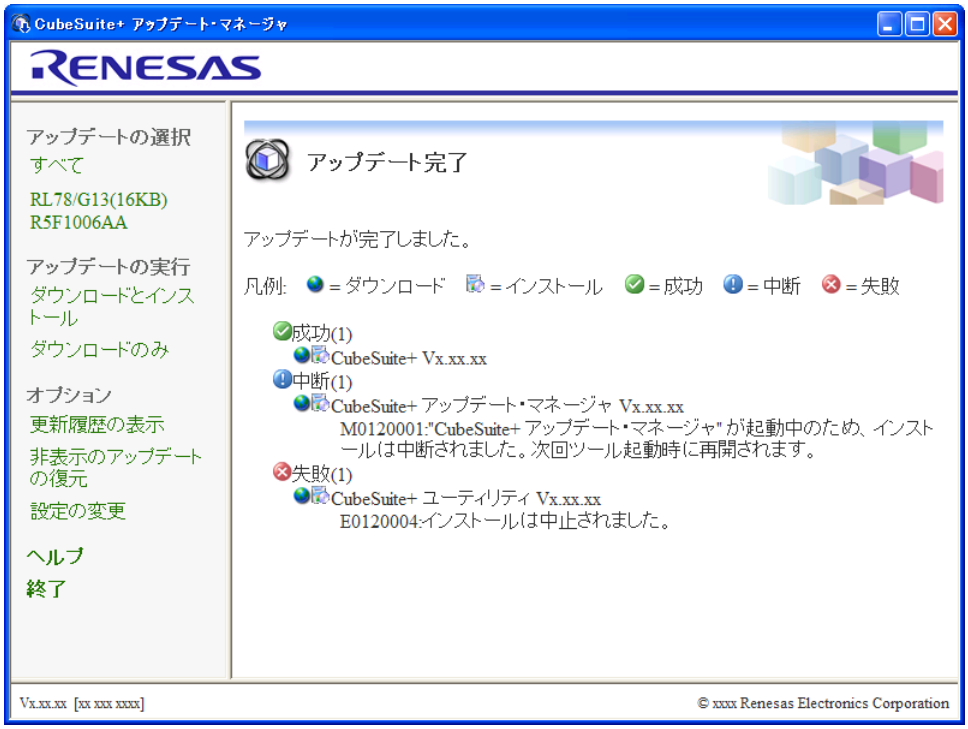


備考 ダウンロードしたアップデート項目をあとからインストールする方法については，「2. 4. 5 アップデートの履歴を表示する」を参照してください。

(7) ダウンロードとインストールの結果を表示します。

アップデートに関する一連の操作が終了すると，アップデート完了 ページに実行結果が表示されます。

図 2—27 アップデート完了 ページ



メニュー・エリアで［終了］をクリックし，アップデートを完了します。

備考 1. ダウンロード，またはインストールの結果は，下記のアイコンで表示されます。

	成功したアップデートの数，アップデート対象のタイトルを表示します。
	中断したアップデートの数，アップデート対象のタイトルとメッセージを表示します。
	失敗したアップデートの数，アップデート対象のタイトルとメッセージを表示します。

2. アップデート対象のタイトルの先頭には，以下のアイコンが表示されます。

	アップデート対象のダウンロードを行いました。
	アップデート対象のインストールを行いました。

2. 4. 2 自動アップデートを行う

自動でアップデートを行う方法について説明します。

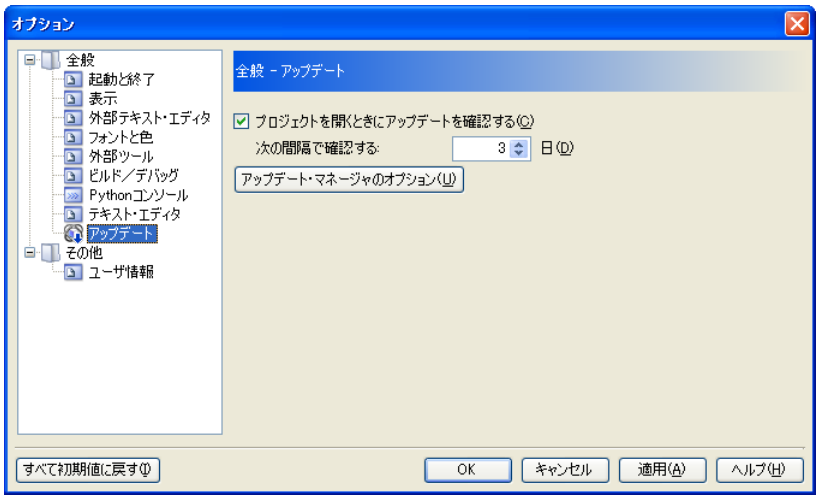
自動アップデート機能は，CubeSuite+ でプロジェクトを開くとき，手動アップデート機能の一部を自動的に行います。

(1) 自動確認機能

自動確認機能は，CubeSuite+ でプロジェクトを開くときに，アップデート対象の確認を行います。

この機能は，オプション ダイアログの［全般 - アップデート］カテゴリで［プロジェクトを開くときにアップデートを確認する］をチェックし，［次の間隔で確認する］で設定した条件に一致した場合に実行されます。

図 2—28 オプション ダイアログ（[全般 - アップデート］カテゴリ）



備考 1. アップデート対象の確認は，CubeSuite+ の処理と並行して行われます。アップデート対象の確認中はタスクトレイに が表示されます。

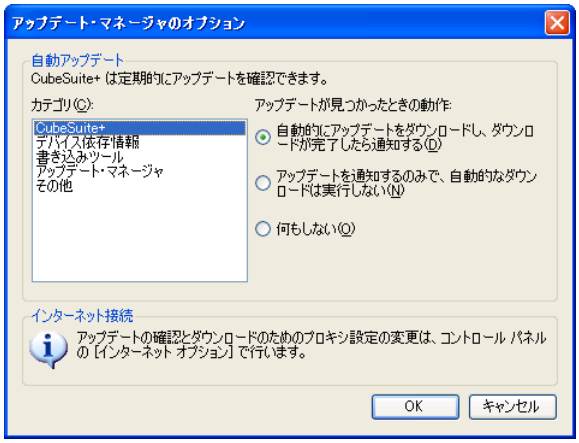
2. アップデート対象が 1 つもなかった場合は，自動確認機能は終了します。

(2) 自動ダウンロード機能

自動ダウンロード機能は，(1) によるアップデート対象の確認後，見つかったアップデート対象をダウンロードします。

この機能は，アップデート・マネージャのオプション ダイアログで［自動的にアップデートをダウンロードし，ダウンロードが完了したら通知する］を選択した場合に，該当カテゴリごとに実行されます。

図 2—29　アップデート・マネージャのオプション ダイアログ



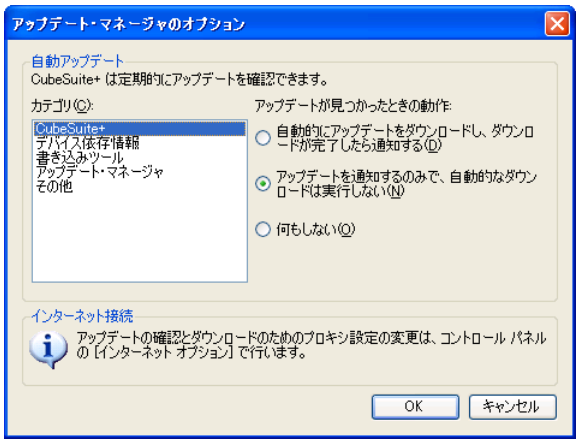
備考　アップデート対象のダウンロード中はタスクトレイに が表示されます。

(3) アップデート通知機能

アップデート通知機能は，(2) によるアップデート対象のダウンロード後，タスクトレイにアップデートを通知するアイコン を表示します。見つかったアップデート対象の中に，今回の自動確認で初めて見つかったアップデート対象が含まれている場合は を表示します。

この機能は，アップデート・マネージャのオプション ダイアログで［アップデートを通知するのみで，自動的なダウンロードは実行しない］を選択した場合に，ダウンロードをせずに実行されます。これは，各カテゴリごとに指定できます。

図 2—30　アップデート・マネージャのオプション ダイアログ



備考　，または をクリックすると，CubeSuite+ アップデート・マネージャ ウインドウが表示されます。

「2. 4. 1　手動アップデートを行う」の (3) 以降の手順にしたがって，インストールを行ってください。

図 2—31 CubeSuite+ アップデート・マネージャ ウインドウ

2. 4. 3 アップデートを中断する

アップデート実行中 ダイアログで［キャンセル］ボタンを選択することで，アップデートを中断することができます。アップデートが中断された場合は，アップデート完了 ページにダウンロード中断として表示されます。

また，アップデートを行うとき，関連ツールが起動しているとファイルの更新をすることができません。そのため，関連ツールが起動している場合はインストールを保留し，アップデート完了 ページにインストール中断として表示されます。

2. 4. 4 アップデートを再開する

アップデート・マネージャ，またはアップデート機能に対応する CubeSuite+ 関連ツールを起動したとき，インストール中断がないかが確認されます。

インストール中断があった場合で，関連ツールが起動している場合は，メッセージ ダイアログにメッセージを表示しますので，関連ツールを終了後，［再試行］ボタンを選択してください。

関連ツールが起動していない場合は，メッセージ ダイアログにメッセージを表示しますので，［はい］ボタンを選択してください。アップデート・マネージャにより，アップデートを行います。アップデート実行後に CubeSuite+ アップデート・マネージャ ウインドウが閉じられ，起動途中で終了した関連ツールが再起動されます。

2. 4. 5 アップデートの履歴を表示する

CubeSuite+ アップデート・マネージャ ウインドウのメニュー・エリアの［更新履歴の表示］を選択すると，更新履歴 ページが表示されます。

このページでは，ダウンロード・ファイルのインストール（他の PC のアップデートを含む），およびダウンロード・ファイルの削除を行うことができます。

図 2—32 更新履歴 ページ

(1) ダウンロード・ファイルをインストールする場合

ダウンロード・ファイルをインストールする方法について説明します。

(a) ダウンロード・ファイルを選択します。

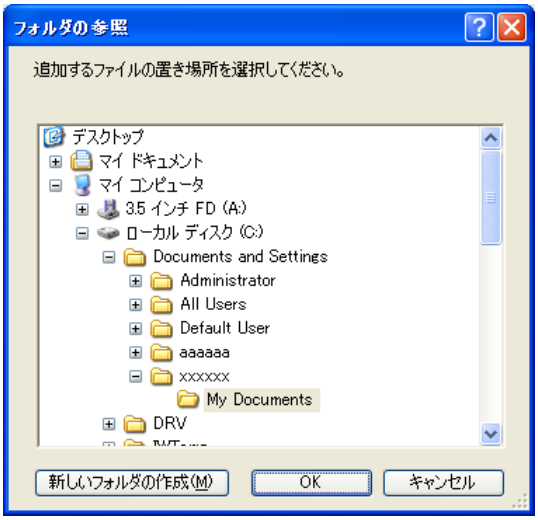
更新履歴 ページでインストールしたいダウンロード・ファイルをチェックし，［コピー］ボタンをクリックしてください。

図 2—33 更新履歴 ページ

(b) ダウンロード・ファイルのコピー先を選択します。

フォルダの参照 ダイアログが表示されます。

図 2—34 フォルダの参照 ダイアログ



ダウンロード・ファイルのコピー先のフォルダを選択し，［OK］ボタンをクリックしてください。

(c) コピー先フォルダを確認し，インストールを行います。

ダウンロード・ファイルが指定フォルダにコピーされます。

エクスプローラでコピー先のフォルダを開き，ダウンロード・ファイルからインストールを行ってください。

(2) ダウンロード・ファイルを削除する場合

(a) ダウンロード・ファイルを選択します。

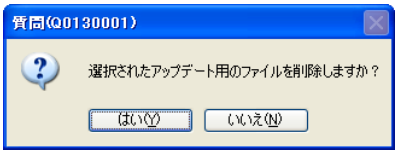
更新履歴 ページで削除したいダウンロード・ファイルをチェックし，［削除］ボタンをクリックしてください。

図 2—35 更新履歴 ページ

(b) ダウンロード・ファイルの削除可否を確認し，削除します。

メッセージ ダイアログに削除可否の確認メッセージが表示されます。

図 2—36 メッセージ ダイアログ



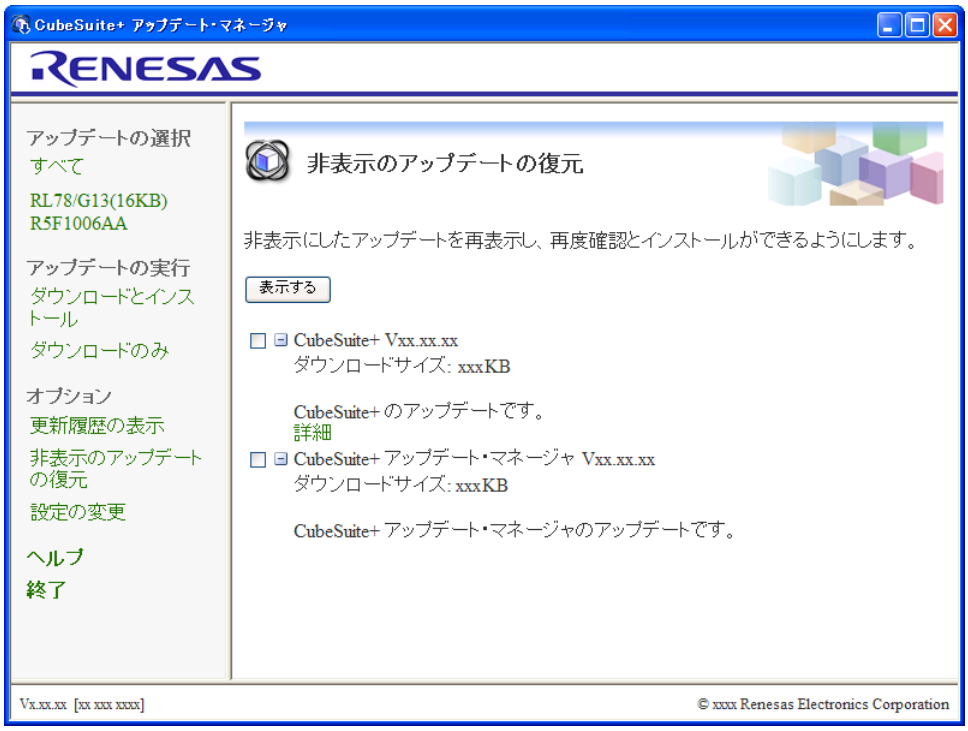
［はい］ボタンをクリックすると，ダウンロード・ファイルを削除します。

2. 4. 6 非表示にしたアップデートを復元する

CubeSuite+ アップデート・マネージャ ウインドウのメニュー・エリアの［非表示のアップデートの復元］を選択すると，非表示のアップデートの復元 ページが表示されます。

このページでは，アップデートの選択 ページで非表示にしたアップデートを復元することができます。

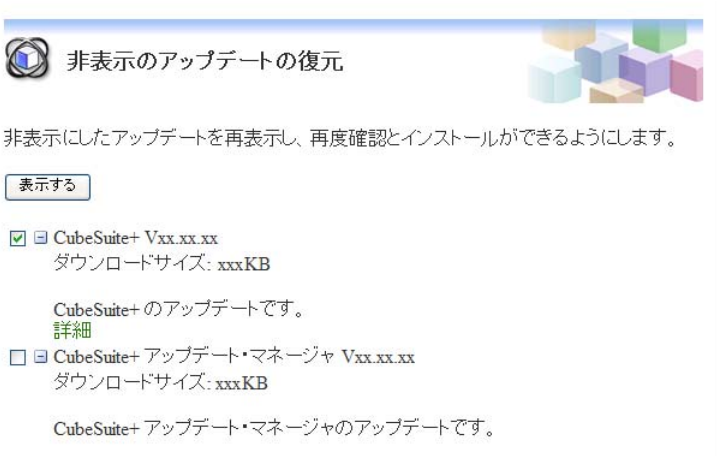
図 2—37 非表示のアップデートの復元 ページ



(1) アップデートを選択します。

非表示のアップデートの復元 ページで復元したいアップデートをチェックし，［表示する］ボタンをクリックしてください。

図 2—38 非表示のアップデートの復元 ページ



(2) アップデートの復元を確認します。

非表示のアップデートの復元 ページで選択したアップデートが非表示状態となり，アップデートの選択 ページで表示状態となったことを確認します。

図 2—39 非表示のアップデートの復元 ページ

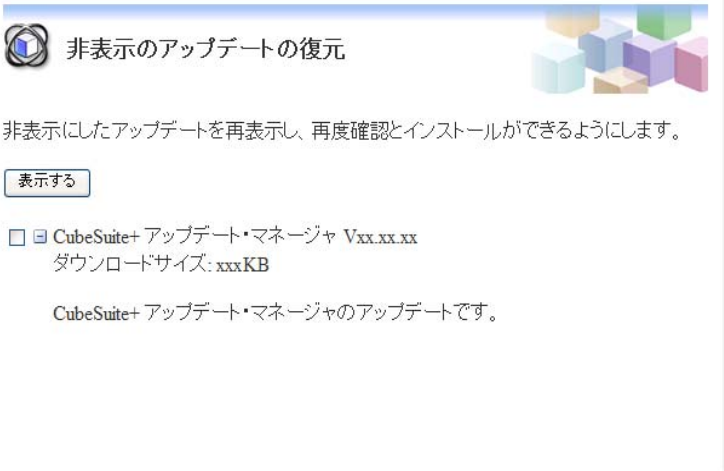


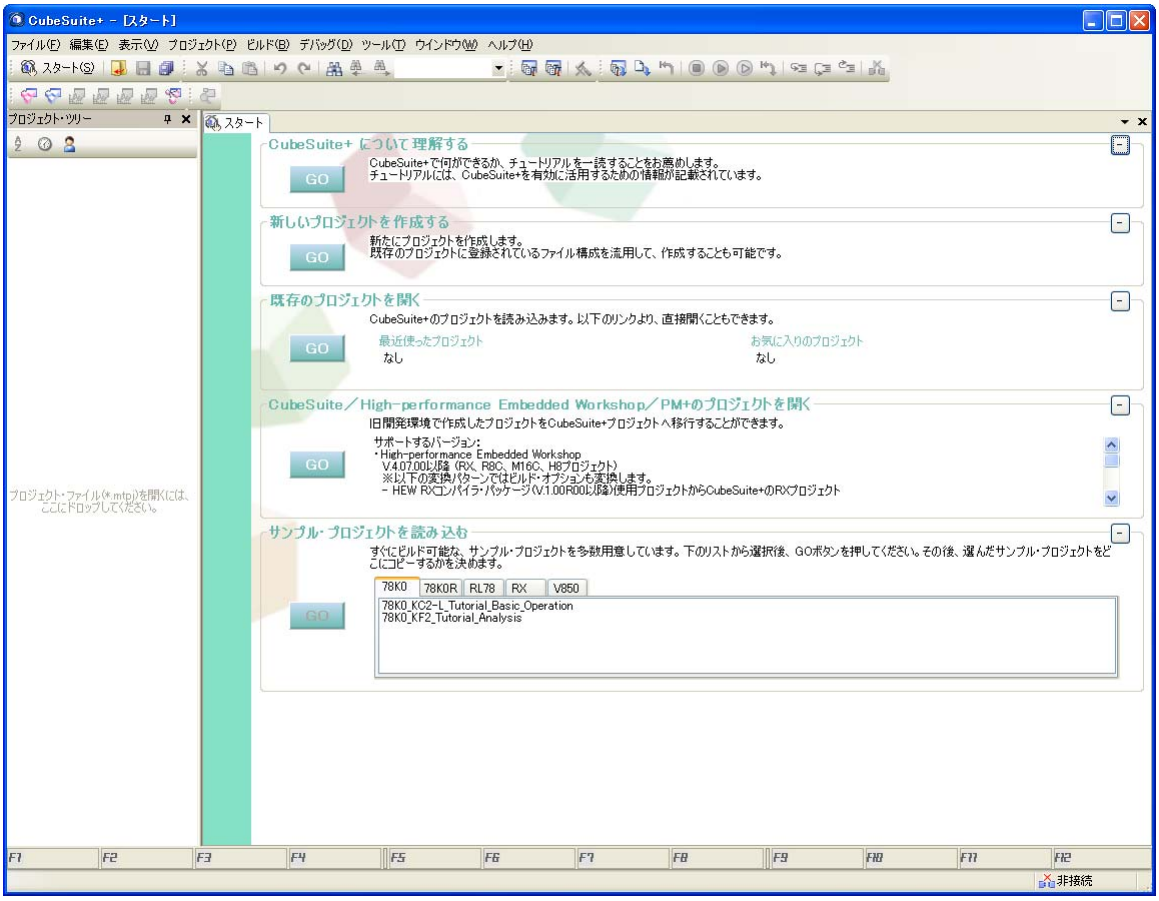
図 2—40 アップデートの選択 ページ

2. 5 CubeSuite+ を起動する

CubeSuite+ をインストール後，初めて起動する場合は，Windows の［スタート］メニューから［すべてのプログラム］→［Renesas Electronics CubeSuite+］→［CubeSuite+］を選択します。

CubeSuite+ を起動すると，以下のメイン・ウインドウが表示されます。

図 2—41 メイン・ウインドウ（スタート パネル）



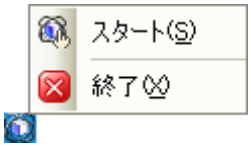
2 回目以降は，CubeSuite+ を常駐起動することにより，起動時間を短縮することができます（以降，“ラピッド・スタート”と呼びます）。

備考 ラピッド・スタートの有効／無効は，オプション ダイアログの［全般 - 起動と終了］カテゴリの［ラピッド・スタートを有効にして起動を高速化する］で設定します（デフォルト：有効）。

ラピッド・スタートが有効である場合，ログイン後に CubeSuite+ が待機状態となり，タスクトレイに アイコンが表示されます。

CubeSuite+ の待機状態を解除し，メイン・ウインドウを表示するには，アイコンのコンテキスト・メニュー→［スタート］を選択，またはアイコンをダブルクリックします。

図 2—42 ラピッド・スタート・アイコンのコンテキスト・メニュー



メイン・ウインドウの ボタンをクリックすると，CubeSuite+ は待機状態となります（待機中は，タスクトレイに アイコンが表示されます）。

CubeSuite+ を終了するには， アイコンのコンテキスト・メニュー→［終了］を選択します。

2. 6 プロジェクトを作成する

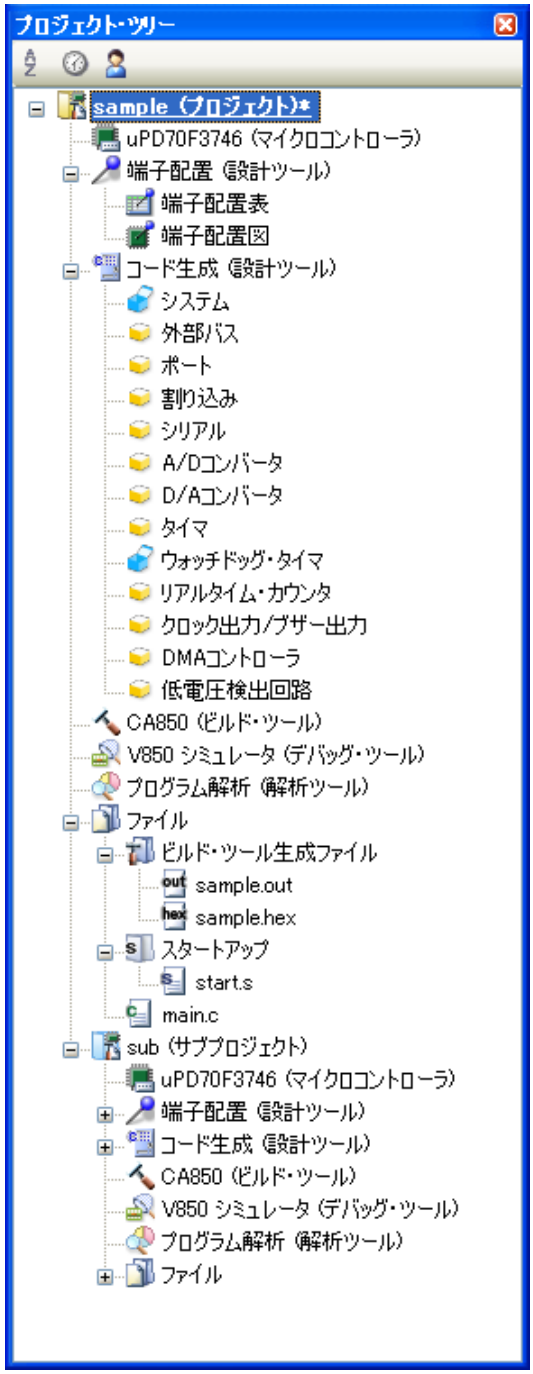
プロジェクトとは，CubeSuite+ がアプリケーション・システム開発の単位として管理するものです。

CubeSuite+ は，プロジェクトで使用するマイクロコントローラ，ビルド・ツール，ソース・ファイル等の設定情報をプロジェクト・ファイル（*.mtpj）に保存し，参照します。

(1) プロジェクト・ツリーの構成と詳細設定

プロジェクトの設定は，プロジェクト・ツリー パネルで行います。

図 2—43 プロジェクト・ツリー パネル



プロジェクト・ツリーでは，プロジェクトの構成要素が以下のノードでツリー表示されます。

なお，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）がプロパティ パネルに表示され，設定の変更を行うことができます。

また，各構成要素（ノード，またはファイル）のコンテキスト・メニューからもプロジェクトの設定を行うことができます。

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。
マイクロコントローラ名 (マイクロコントローラ) （以降，“マイクロコントローラ・ノード”と呼びます。）	プロジェクトで使用するマイクロコントローラです。
設計ツール名 (設計・ツール) （以降，“設計ツール・ノード”と呼びます。）	使用する設計ツール（端子配置，コード生成等）です。 なお，デバッグ専用プロジェクト注を対象としている場合，コード生成 (設計ツール) ノードは表示されません。
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（コンパイラ，アセンブラ等）です。 なお，デバッグ専用プロジェクト注を対象としている場合，*ビルド・ツール名*は“なし”と表示されます。
デバッグ・ツール名 (デバッグ・ツール) （以降，“デバッグ・ツール・ノード”と呼びます。）	使用するデバッグ・ツール（エミュレータ，シミュレータ等）です。
プログラム解析 (解析ツール) （以降，“解析ツール・ノード”と呼びます。）	使用する解析ツールです。 なお，デバッグ専用プロジェクト注を対象としている場合，このノードは表示されません。
ファイル （以降，“ファイル・ノード”と呼びます。）	プロジェクトに登録しているファイルが，直下に表示されます。
ダウンロード・ファイル （以降，“ダウンロード・ファイル・ノード”と呼びます。）	プロジェクトにダウンロード・ファイルを登録するためのノードです。 なお，デバッグ専用プロジェクト注を対象としている場合のみ，このノードは表示されます。
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。）	ビルド時に作成されるノードで，ビルド・ツールによって生成されたファイル（オブジェクト・ファイルを除く）が直下に表示されます。 なお，デバッグ専用プロジェクト注を対象としている場合，このノードは表示されません。
スタートアップ【V850】【RL78】【78K0R】【78K0】 （以降，“スタートアップ・ノード”と呼びます。）	プロジェクトに標準以外のスタートアップ・ルーチンを登録するためのノードです。このノードは常にファイル・ノード以下に表示されます。 なお，デバッグ専用プロジェクト注を対象としている場合，このノードは表示されません。
カテゴリ名 （以降，“カテゴリ・ノード”と呼びます。）	ファイルをモジュール単位などに分類するために，ユーザが作成するカテゴリです。

ノード	説明
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。 サブプロジェクトについては，「(2) プロジェクトとサブプロジェクト」を参照してください。

注 デバッグ専用プロジェクトについての詳細は，「付録F 外部ビルド・ツールの使用」を参照してください。

備考 1. 使用するマイクロコントローラに対応しているツールのみ表示されます。

2. 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されます。
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。

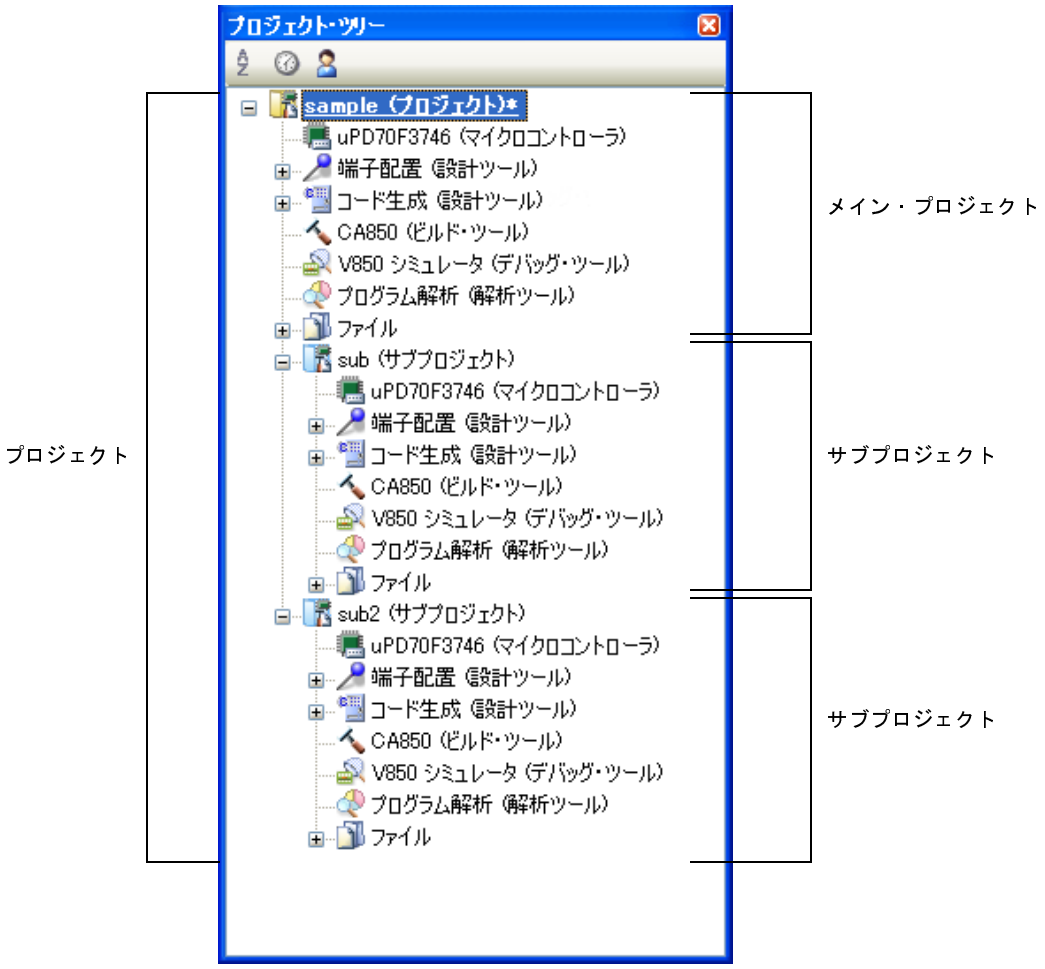
(2) プロジェクトとサブプロジェクト

プロジェクトは，その下位階層にサブプロジェクトを追加することができます。
サブプロジェクトの設定情報は，サブプロジェクト・ファイル（*.mtsp）に保存されます。
サブプロジェクトは，たとえば，以下のような目的で使用します。

- プロジェクトで使用するライブラリ・ファイルを作成するプロジェクトも作成するとき，ライブラリ・ファイルを作成するプロジェクトをサブプロジェクトとして作成する
- 異なるマイクロコントローラ対して同じアプリケーション・システムを開発するとき，マイクロコントローラの異なるプロジェクトをサブプロジェクトとして作成する

サブプロジェクトを追加した場合，本マニュアルでは，プロジェクトのことをサブプロジェクトに対して“メイン・プロジェクト”と呼びます。“プロジェクト”は，メイン・プロジェクトとサブプロジェクトの総称となります。

図 2—44 プロジェクト・ツリー パネル（サブプロジェクトを 2 個追加した場合）



なお，プロジェクトの設定と，プロジェクトに追加したサブプロジェクトの設定は独立しており，お互いに影響はしません。メイン・プロジェクトとサブプロジェクト間，および異なるサブプロジェクト間で同じ設定を行う場合は，プロジェクト・ツリーで設定するノードを複数選択し，プロパティ パネルで設定を行ってください。

注意 サブプロジェクトの下にサブプロジェクトを作成することはできません。

備考 サブプロジェクトの追加方法については，「2. 6. 3 新しいサブプロジェクトを追加する」，および「2. 7. 2 既存のサブプロジェクトを追加する」を参照してください。

2. 6. 1 プロジェクトをスタートする

ツールバーの スタート(S) をクリックすると，スタート パネルがオープンします。パネル上のボタンをクリックすることにより，新しいプロジェクトを作成したり，既存のプロジェクトを開くことができます。

なお，スタート パネルは，CubeSuite+ の初回起動時に自動的にオープンします。

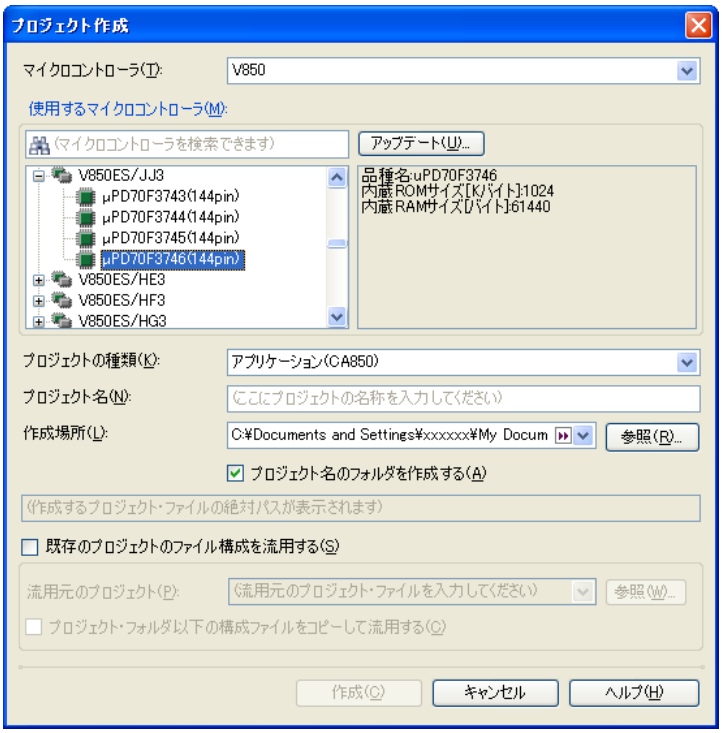
図 2—45 スタート パネル

2. 6. 2 新しいプロジェクトを作成する

新規にプロジェクトを作成します。

［プロジェクト］メニュー→［新しいプロジェクトを作成 ...］を選択すると，プロジェクト作成 ダイアログがオープンします。

図 2—46 プロジェクト作成 ダイアログ（初回起動時）



以下の順番で項目を設定します。

(1) マイクロコントローラの種類を選択

［マイクロコントローラ］において，プロジェクトで使用するマイクロコントローラの種類を選択します。

以下の項目を選択することができます。

- RX
- V850
- RL78
- 78K0R
- 78K0

(2) マイクロコントローラを選択

［使用するマイクロコントローラ］エリアにおいて，プロジェクトで使用するマイクロコントローラを選択します。

［使用するマイクロコントローラ］エリアに使用するマイクロコントローラがない場合は，［アップデート ...］ボタンをクリックしてください。

CubeSuite+ アップデート・マネージャ ウインドウがオープンし，ネットワークを介してマイクロコントローラの最新情報を検索することができます。

注意 ［アップデート ...］ボタンは，インストーラを使用して本製品をインストールした場合のみ有効となります。パックしたものを使用する場合は無効となります。

(3) プロジェクトの種類を選択

［プロジェクトの種類］において，作成するプロジェクトの種類を選択します。

以下の項目を選択することができます。

- アプリケーション（CC-RX/CA850/CX/CA78K0R/CA78K0）

  CubeSuite+ が提供するビルド・ツールを使用して，C ソース・ファイルから ROM 化用モジュール・ファイル【CA850】【CA78K0R】，ロード・モジュール・ファイル，またはヘキサ・ファイルを生成する場合に選択します。

  ここで生成するファイルが自動的にデバッグ対象となります。

- C++ アプリケーション（CC-RX）

  CubeSuite+ が提供するビルド・ツール CC-RX を使用して，C++ ソース・ファイル（main 関数のファイルのみ），および C ソース・ファイルからロード・モジュール・ファイル，またはヘキサ・ファイルを生成する場合に選択します。

  ここで生成するファイルが自動的にデバッグ対象となります。

- 空のアプリケーション（CC-RX）

  CubeSuite+ が提供するビルド・ツール CC-RX を使用して，ロード・モジュール・ファイルを生成する場合に選択します。

  ここで生成するファイルが自動的にデバッグ対象となります。

  なお，プロジェクト作成時にサンプルのスタートアップ・プログラムは生成しません。

- ライブラリ（CC-RX/CA850/CX/CA78K0R/CA78K0）

  CubeSuite+ が提供するビルド・ツールを使用して，ユーザ・ライブラリ用のライブラリ・ファイルを生成する場合に選択します。

- デバッグ専用

  CubeSuite+ が提供するビルド・ツール以外のビルド・ツールが生成したロード・モジュール・ファイル，またはヘキサ・ファイルをデバッグ対象とする場合に選択します（デバッグ専用プロジェクト）。

  デバッグ専用プロジェクトの作成方法，および使用方法についての詳細は，「付録 F 外部ビルド・ツールの使用」を参照してください。

(4) プロジェクト名とプロジェクト・ファイルの作成場所を指定

［プロジェクト名］，および［作成場所］に，プロジェクト名とプロジェクト・ファイルの作成場所を指定します。

指定した作成場所の下にプロジェクト名のフォルダを作成しない場合は，［プロジェクト名のフォルダを作成する］のチェックを外してください。

注意 プロジェクト・ファイルの作成場所を直接入力する場合は，絶対パスで入力してください。

(5) 既存のプロジェクトのファイル構成の流用を指定

既存のプロジェクトのファイル構成を流用してプロジェクトを作成する場合は，［既存のプロジェクトのファイル構成を流用する］をチェックし，［流用元のプロジェクト］に流用元のプロジェクト・ファイル名を指定してください。

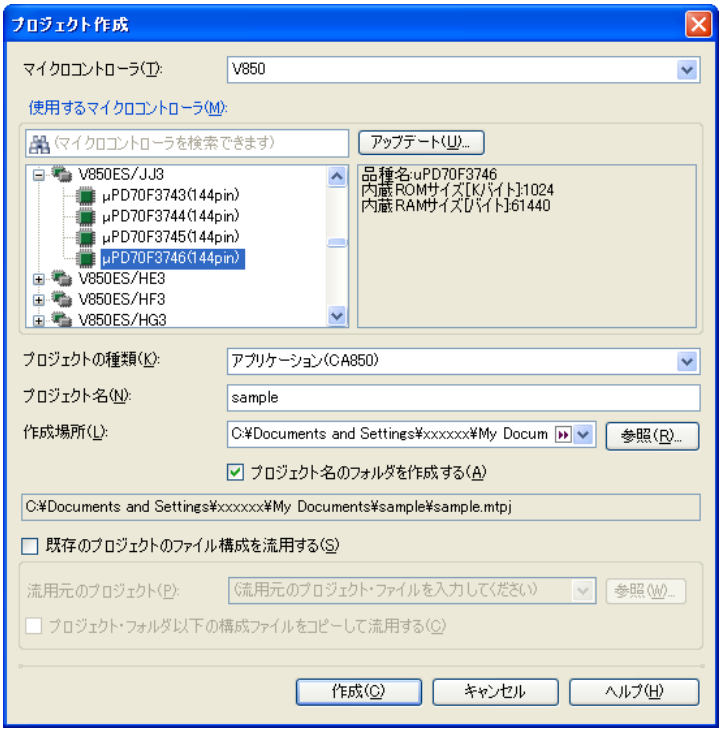
注意　CubeSuite，High-performance Embedded Workshop，PM+ のプロジェクト・ファイルを指定することはできません。
CubeSuite，High-performance Embedded Workshop，PM+ のプロジェクトを流用したい場合は，いったん，CubeSuite+ でそのプロジェクトを開いて，CubeSuite+ のプロジェクトとして保存してください（「2. 7. 8　CubeSuite のプロジェクトを CubeSuite+ のプロジェクトに変換する」，「2. 7. 9　HEW のプロジェクトを CubeSuite+ のプロジェクトに変換する」，「2. 7. 10　PM+ のプロジェクトを CubeSuite+ のプロジェクトに変換する」参照）。そのあと，保存したプロジェクト・ファイルをこのエリアに指定してください。

備考 1. 流用元のプロジェクトで使用しているビルド・ツールのバージョンと，作成するプロジェクトのバージョンが異なる場合は，自動的に変換します（［プロジェクトの種類］において“デバッグ専用”を指定した場合を除く）。

2. ビルド・ツールが CA850 のプロジェクトのファイル構成を流用して，ビルド・ツールが CX のプロジェクトを作成することができます（「2. 7. 7　CA850 のプロジェクトを CX のプロジェクトに変換する」参照）。

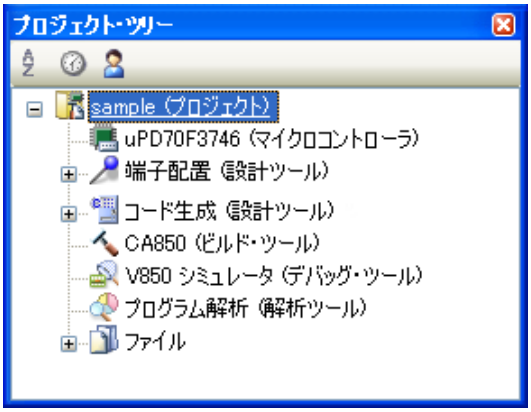
項目設定後のイメージを以下に示します。

図 2—47　プロジェクト作成 ダイアログ（項目設定後）



［作成］ボタンをクリックすると，（4）で指定した場所にプロジェクト・ファイルを作成し，プロジェクト・ツリー パネルに作成したプロジェクトの構成がツリー表示されます。

図 2—48 プロジェクト・ツリー パネル（新規プロジェクト作成後）



備考 プロジェクト作成後，ビルド・フェーズ，またはデバッグ・フェーズに進むためには，対象となるファイルをプロジェクトに追加する必要があります。

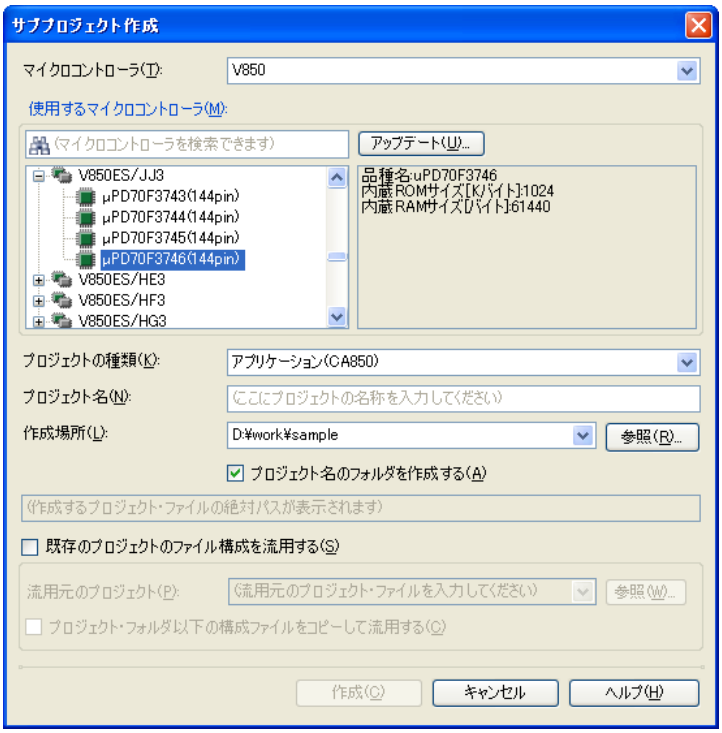
ファイルの追加方法については，以下を参照してください。

- ［プロジェクトの種類］において“アプリケーション”または“ライブラリ”を選択した場合
  →「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」参照
- ［プロジェクトの種類］において“デバッグ専用”を選択した場合
  →「F. 3 プロジェクトにファイルを追加する」参照

2. 6. 3　新しいサブプロジェクトを追加する

プロジェクト・ツリーでプロジェクト・ノードを選択し，コンテキスト・メニューの［追加］→［新しいサブプロジェクトを追加 ...］を選択すると，プロジェクト作成 ダイアログがオープンします。

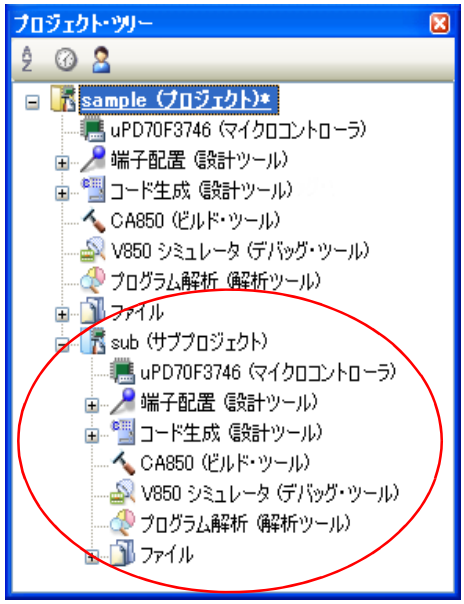
図 2—49　プロジェクト作成 ダイアログ（新しいサブプロジェクトを追加する場合）



ダイアログ上で，各項目を設定したのち，［作成］ボタンをクリックしてください（各項目の設定については，「2. 6. 2　新しいプロジェクトを作成する」を参照してください）。

備考　ビルド・ツールが CA850 のサブプロジェクトのファイル構成を流用して，ビルド・ツールが CX のサブプロジェクトを追加することもできます（「2. 7. 7　CA850 のプロジェクトを CX のプロジェクトに変換する」参照）。

図 2—50 プロジェクト・ツリー パネル（サブプロジェクト追加後）



2. 7 プロジェクトを操作する

プロジェクトに関する操作方法について説明します。

2. 7. 1 プロジェクトを開く

プロジェクトを開くには，以下の方法があります。

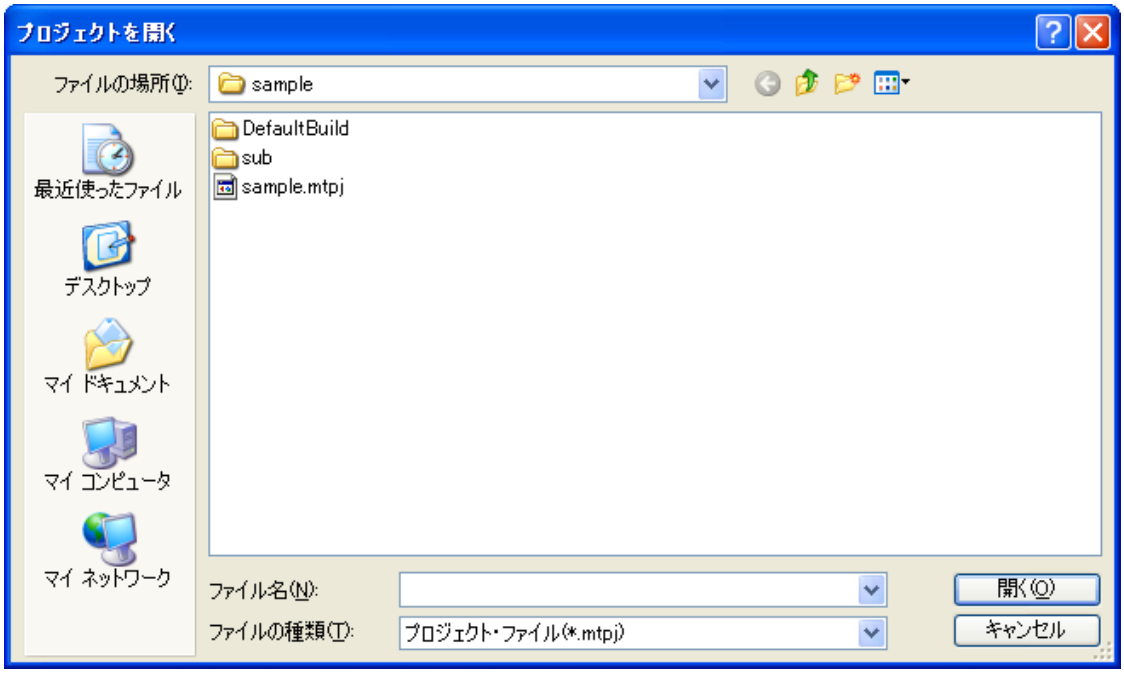
- 既存のプロジェクトを開く
- 最近使ったプロジェクトを開く
- お気に入りメニューからプロジェクトを開く

(1) 既存のプロジェクトを開く

既存のプロジェクトは，プロジェクト・ファイルを指定して開きます。

［プロジェクト］メニュー→［プロジェクトを開く ...］を選択すると，プロジェクトを開く ダイアログがオープンします。

図 2—51 プロジェクトを開く ダイアログ



ダイアログ上で，該当プロジェクト・ファイルを指定し，［開く］ボタンをクリックしてください。

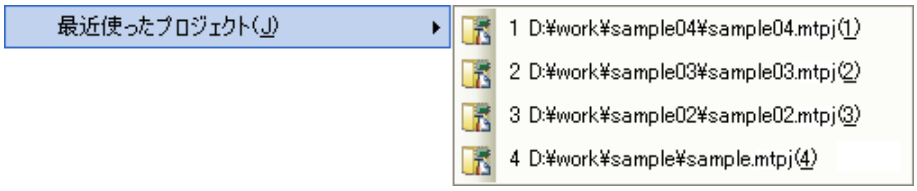
備考 CubeSuite+ を起動していない場合は，エクスプローラでプロジェクト・ファイルをダブルクリックするなどの操作により，CubeSuite+ を起動し，該当プロジェクトを開くことができます。

(2) 最近使ったプロジェクトを開く

最近使ったプロジェクト（最も新しいものから 4 つまで）については，メニューから直接開くことができます。

［ファイル］メニュー→［最近使ったプロジェクト］を選択すると，最近使ったプロジェクトのパスが新しいものから順に 4 つまでカスケード・メニュー表示されます。開きたいプロジェクトのパスを選択してください。

図 2—52 ［最近使ったプロジェクト］項目

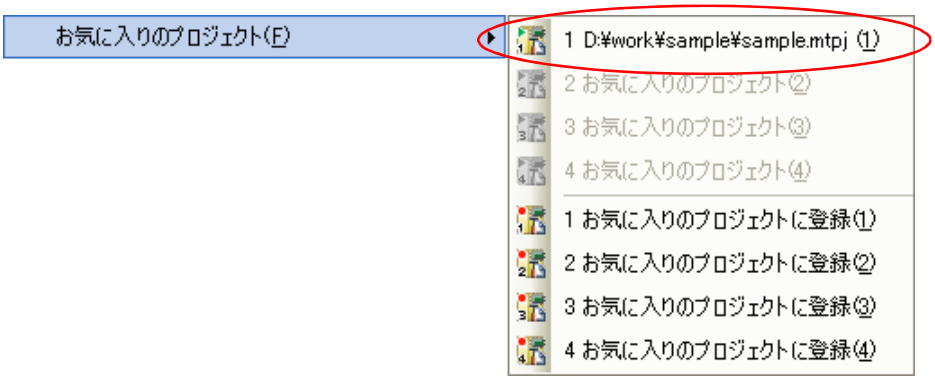


(3) お気に入りメニューからプロジェクトを開く

お気に入りメニューに登録しているプロジェクトを開きます。

［プロジェクト］メニュー→［お気に入りのプロジェクト］を選択すると，お気に入りメニューに登録しているプロジェクトのパスがカスケード・メニュー表示されます。開きたいプロジェクトのパスを選択してください。

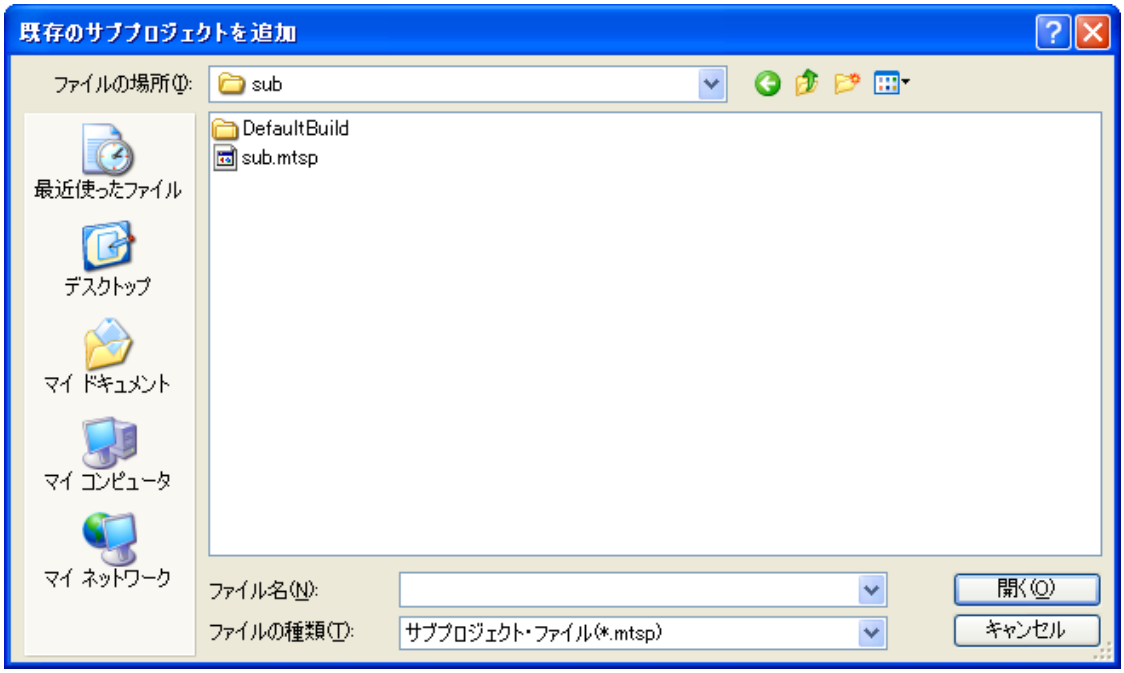
図 2—53 ［お気に入りのプロジェクト］項目



2. 7. 2 既存のサブプロジェクトを追加する

プロジェクト・ツリーでプロジェクト・ノードを選択し，コンテキスト・メニューの［追加］→［既存のサブプロジェクトを追加 ...］を選択すると，既存のサブプロジェクトを追加 ダイアログがオープンします。

図 2—54 既存のサブプロジェクトを追加 ダイアログ



ダイアログ上で，追加するサブプロジェクトのサブプロジェクト・ファイルを指定し，［開く］ボタンをクリックしてください。

2. 7. 3 プロジェクトをお気に入りメニューに追加する

現在開いているプロジェクトを“お気に入りのプロジェクト”としてメニューに追加することができます（4 つまで）。

［プロジェクト］メニュー→［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］～［4 お気に入りのプロジェクトに登録］を選択すると，現在開いているプロジェクトのパスが［プロジェクト］メニュー→［お気に入りのプロジェクト］以下に登録されます。

図 2—55 ［お気に入りのプロジェクトに登録］項目

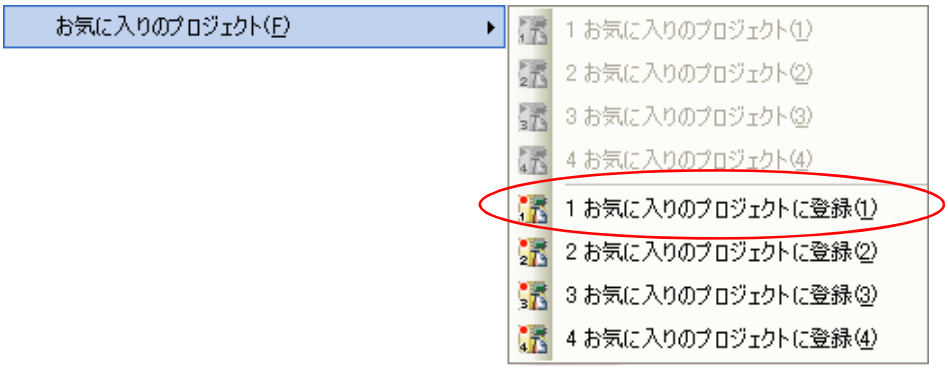
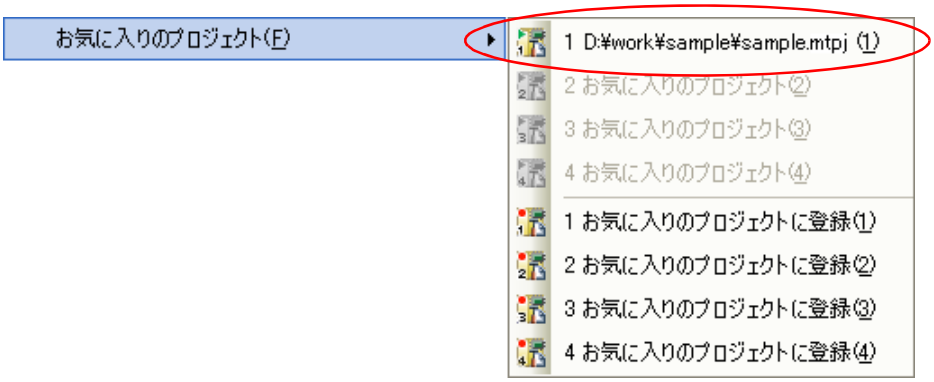


図 2—56 ［お気に入りのプロジェクト］項目（プロジェクト登録後）

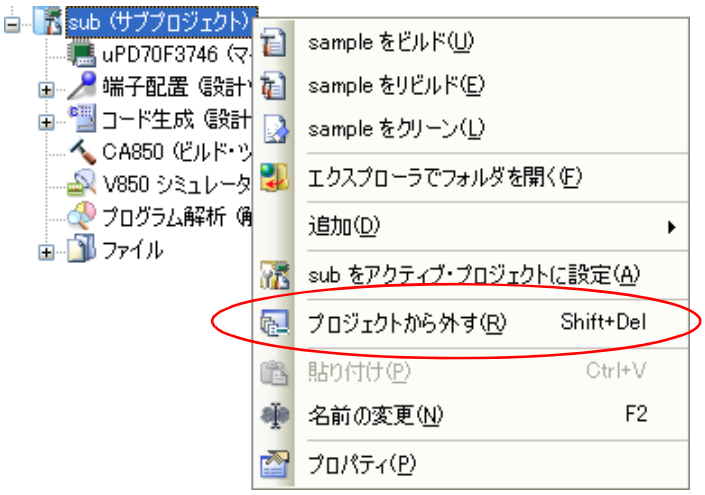


2. 7. 4 プロジェクトからサブプロジェクトを外す

プロジェクトに登録しているサブプロジェクトをプロジェクトから外すには，プロジェクト・ツリーでサブプロジェクト・ノードを選択し，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。

なお，サブプロジェクト・ファイル自体はファイル・システム上からは削除されません。

図 2—57 ［プロジェクトから外す］項目

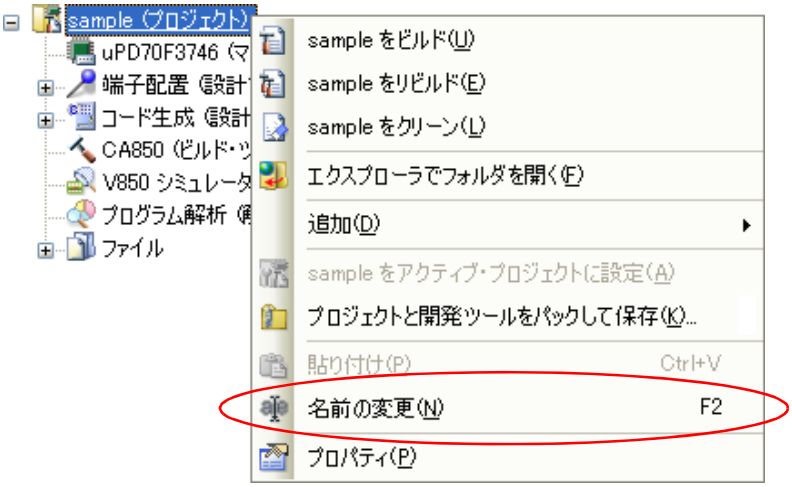


2.7.5 プロジェクト名を変更する

プロジェクト・ツリーでプロジェクト（メイン・プロジェクト，またはサブプロジェクト）の名前を変更することができます。

プロジェクト・ノード，またはサブプロジェクト・ノードを選択し，コンテキスト・メニューの［名前の変更］を選択してください。

図 2—58 ［名前の変更］項目（プロジェクトの場合）



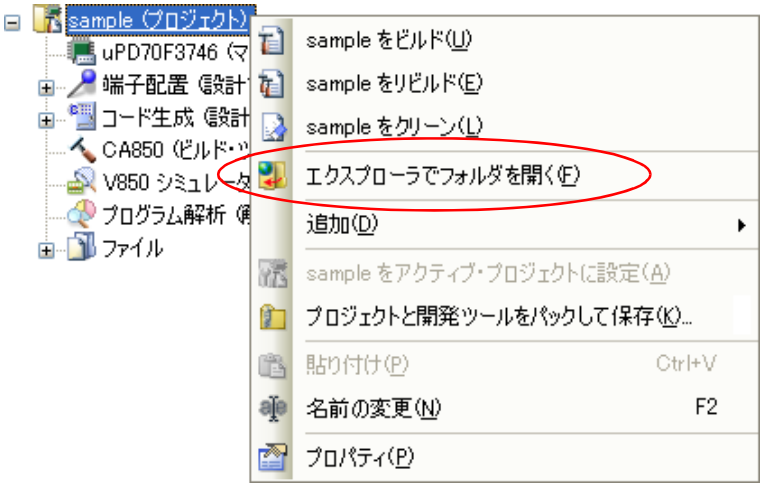
備考 プロジェクト名を変更したのち，プロジェクトを保存すると，実際のプロジェクト・ファイル名も変更されます。

2.7.6 プロジェクト・フォルダをエクスプローラで開く

プロジェクト・ツリーから，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開くことができます。

プロジェクト・ノード，またはサブプロジェクト・ノードを選択し，コンテキスト・メニューの［エクスプローラでフォルダを開く］を選択してください。

図 2—59 ［エクスプローラでフォルダを開く］項目（プロジェクトの場合）



備考 各ファイルのコンテキスト・メニューの［エクスプローラでフォルダを開く］を選択した場合は，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開きます。

2. 7. 7 CA850 のプロジェクトを CX のプロジェクトに変換する

ビルド・ツールが CA850 である CubeSuite+ のプロジェクトは，そのファイル構成を流用してプロジェクトを作成することにより，ビルド・ツールが CX であるプロジェクトに変換することができます。

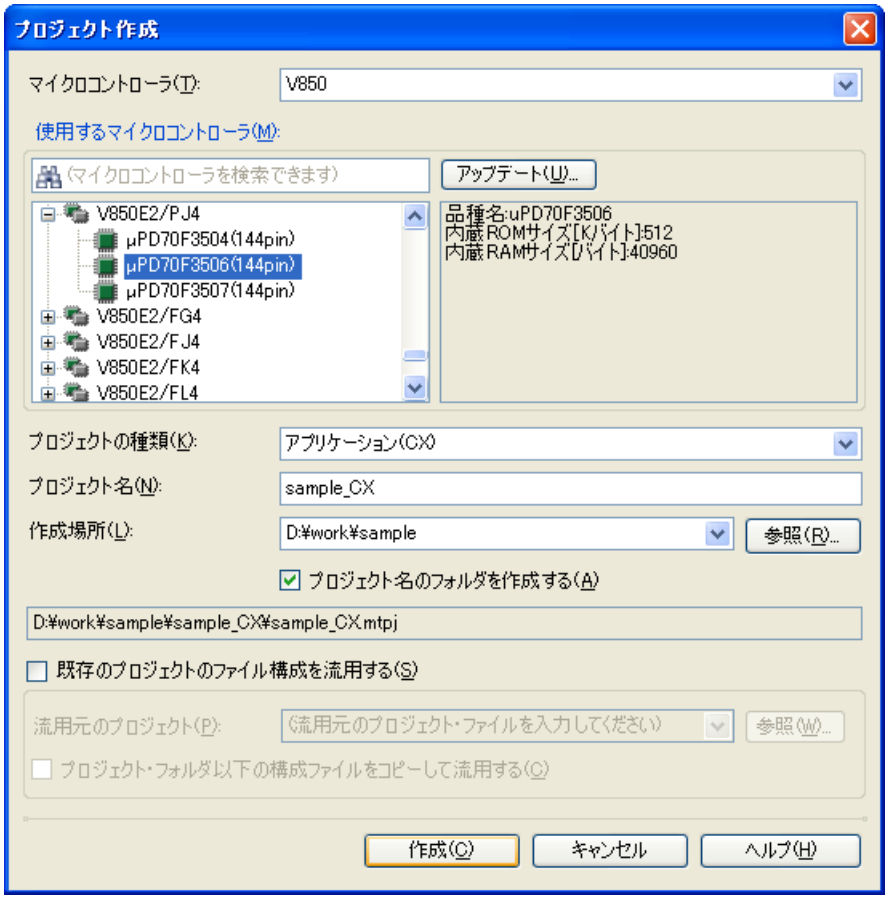
プロジェクトを作成する際，ソース・ファイルを CX 用にコンバートすることもできるため，ビルド・ツールに依存するソース記述を行う必要がありません。

また，ビルド・ツールのプロパティの設定も，CX 用に変換して引き継ぎます。

注意 PM+ のプロジェクトを流用したい場合は，いったん，CubeSuite+ で PM+ のプロジェクトを開いて，CubeSuite+ のプロジェクトとして保存してください（「2. 7. 10 PM+ のプロジェクトを CubeSuite+ のプロジェクトに変換する」参照）。

始めに，［プロジェクト］メニュー→［新しいプロジェクトを作成 ...］を選択し，プロジェクト作成 ダイアログをオープンします。

図 2—60 プロジェクト作成 ダイアログ



以下の順番で項目を設定し，［作成］ボタンをクリックします。

(1) マイクロコントローラの種類を選択

［マイクロコントローラ］において，“V850”を選択します。

(2) マイクロコントローラを選択

［使用するマイクロコントローラ］において，プロジェクトで使用するマイクロコントローラを選択します。

(3) プロジェクトの種類を選択

［プロジェクトの種類］において，流用元のプロジェクトに合致する，“アプリケーション（CX）”，または“ライブラリ（CX）”を選択します。

(4) プロジェクト名とプロジェクト・ファイルの作成場所を指定

［プロジェクト名］，および［作成場所］に，プロジェクト名とプロジェクト・ファイルの作成場所を指定します。

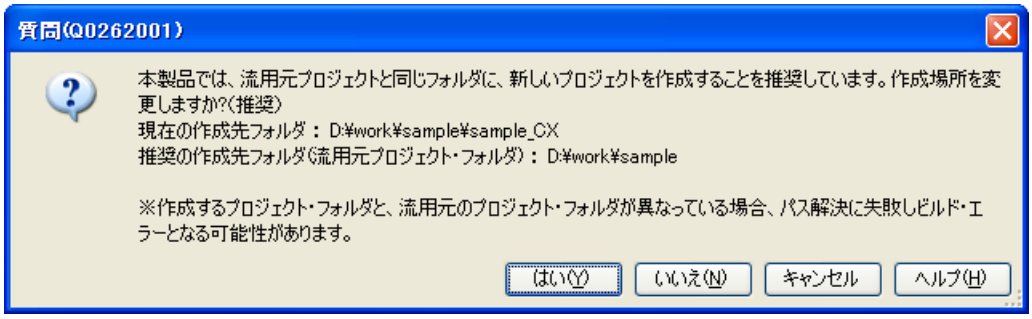
指定した作成場所の下にプロジェクト名のフォルダを作成しない場合は，［プロジェクト名のフォルダを作成する］のチェックを外してください。

(5) CA850 のプロジェクトの流用を指定

［既存のプロジェクトのファイル構成を流用する］をチェックし，［流用元のプロジェクト］に流用元のプロジェクト・ファイル名を指定してください。

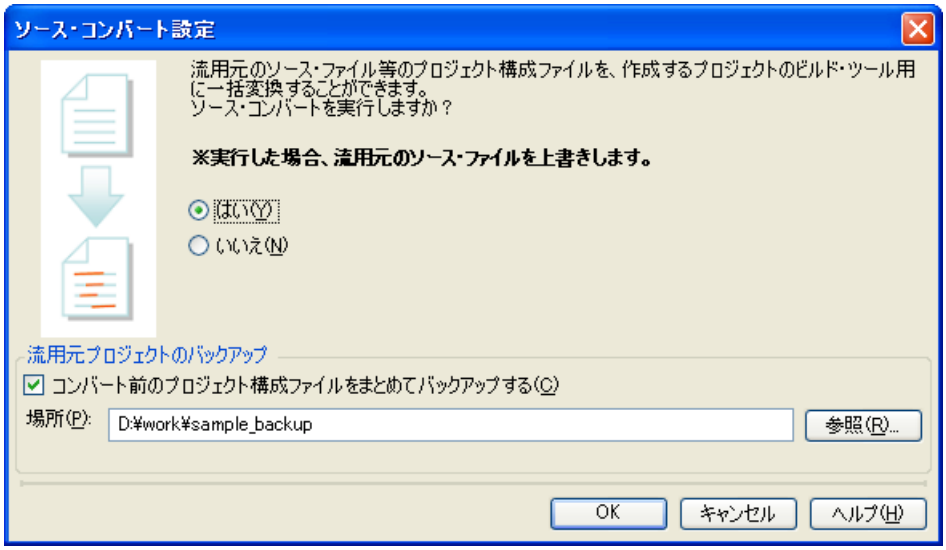
注意 プロジェクト・フォルダとソース・ファイルの関係を崩さないようにするため，流用元と作成するプロジェクト・フォルダは同一にすることを推奨します。
流用元と作成するプロジェクト・フォルダが異なる場合，［作成］ボタンをクリックすると，以下のメッセージ ダイアログがオープンします。
そのまま処理を続ける場合は，［はい］をクリックしてください。

図 2—61 メッセージ ダイアログ



［作成］ボタンをクリックすると，ソース・コンバート設定 ダイアログ【CX】がオープンします。

図 2—62 ソース・コンバート設定 ダイアログ



ソース・ファイルのコンバートを実行するには，［はい］を選択します。

また，流用元のプロジェクトのソース・ファイル，およびプロジェクト一式をバックして保存する場合は，［コンバート前のプロジェクト構成ファイルをまとめてバックアップする］をチェックし，保存先を指定します。

［OK］ボタンをクリックすると，ソース・ファイルのコンバートを実行し，CX のプロジェクトを作成します。

備考 CA850 のプロジェクトを流用して CX のプロジェクトを作成した結果（IDE やコンパイラ・パッケージのバージョン，およびオプションの変換情報）は，プロジェクト流用情報としてファイルに出力されます。

- プロジェクト流用情報ファイル名は，“プロジェクト流用情報 *n*.txt”（*n*=2 ～ 100）となります。
  n は通常付加しません。作成するファイルが既に存在する場合のみ付加します。
- プロジェクト流用情報ファイルは，作成したプロジェクト（サブプロジェクト）ごとに出力されます。
- プロジェクト流用情報ファイルの出力先は，プロジェクト（サブプロジェクト）のプロジェクト・フォルダです。
- 出力されたプロジェクト流用情報ファイルは，プロジェクト・ツリー パネルにおいて，プロジェクト（サブプロジェクト）のファイル・ノードに追加されます。

プロジェクト流用情報ファイルの出力フォーマットを以下に示します。

(1) *プロジェクト作成日時*

(2) <CubeSuite+ IDE（統合開発環境フレームワーク）バージョン>
CubeSuite+ IDE: *CA850 のプロジェクトの IDE バージョン -> CX のプロジェクトの IDE バージョン*

(3) < コンパイラ・パッケージ・バージョン >
CA850: *バージョン* -> CX: *バージョン*

(4) < 利用しないオプション (*ビルド・モード*)>
コマンド名 (ビルド・ツールのプロパティのタブ名)
オプション
:

(5) < 変更したオプション (*ビルド・モード*)>
コマンド名 (ビルド・ツールのプロパティのタブ名)
CA850 のプロジェクトのオプション -> CX のプロジェクトのオプション
:

項番	説明
(1)	プロジェクト作成日時 CA850 のプロジェクトを流用して CX のプロジェクトを作成した日時を，“*YYYY* 年 *MM* 月 *DD* 日 *hh*:*mm*:*ss*”の形式で出力します。
(2)	IDE（統合開発環境フレームワーク）バージョン CA850 のプロジェクトの IDE バージョン，および CX のプロジェクトの IDE バージョンを出力します。
(3)	コンパイラ・パッケージ・バージョン CA850 のバージョン，および CX のバージョンを出力します。 なお，CX のバージョンは，CubeSuite+ 環境にインストールしているパッケージのうち，最新バージョンとなります。
(4)	利用しないオプション (*ビルド・モード*) CA850 のプロジェクトで設定していたが，CX のプロジェクトでは利用しないオプションが存在する場合，その情報をビルド・モードごとに以下の形式で出力します。 *コマンド名 (ビルド・ツールのプロパティのタブ名)* *オプション* : - 本項目は，該当するオプションが存在する場合のみ出力されます。 - CA850 のオプションは，同じ機能を持つ CX のオプションに変換されます。 同じ機能を持ち，名前のみが異なるオプションについては，本項目の出力対象外となります。 - ビルド・モードは，“DefaultBuild”，ユーザ作成ビルド・モードの順に出力されます（“DefaultBuild”は，CubeSuite+ がデフォルトで用意しているビルド・モードです）。 ビルド・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 ビルド編」を参照してください。

項番	説明
(5)	変更したオプション(*ビルド・モード*) CA850 のプロジェクトで設定していたオプションで，CX のプロジェクトで利用するために変更したオプションが存在する場合（CA850 とオプションの機能は同じだが，CX には相当するパラメータが存在しないために，別のパラメータに設定を変更した場合など），その情報をビルド・モードごとに以下の形式で出力します。 *コマンド名*(*ビルド・ツールのプロパティのタブ名*) *CA850 のプロジェクトのオプション -> CX のプロジェクトのオプション* : - 本項目は，該当するオプションが存在する場合のみ出力されます。 - CA850 のオプションは，同じ機能を持つ CX のオプションに変換されます。 同じ機能を持ち，名前のみが異なるオプションについては，本項目の出力対象外となります。 - ビルド・モードは，“DefaultBuild”，ユーザ作成ビルド・モードの順に出力されます（“DefaultBuild”は，CubeSuite+ がデフォルトで用意しているビルド・モードです）。 ビルド・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 ビルド編」を参照してください。

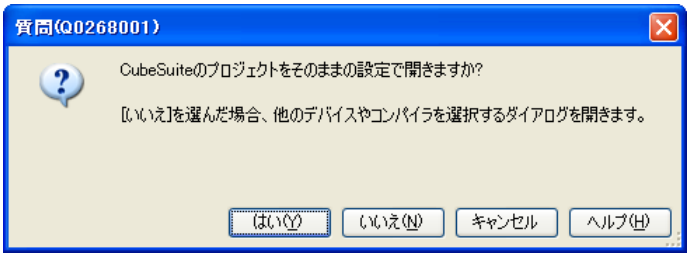
2. 7. 8 CubeSuite のプロジェクトを CubeSuite+ のプロジェクトに変換する

CubeSuite のプロジェクトは，CubeSuite+ の［プロジェクト］メニュー→［プロジェクトを開く ...］でプロジェクト・ファイル（*.cspj）を選択することにより，CubeSuite+ のプロジェクトに変換することができます。

（1）変換設定の実施を選択

CubeSuite のプロジェクト・ファイルを選択すると，以下のメッセージ ダイアログがオープンします。

図 2—63 メッセージ ダイアログ



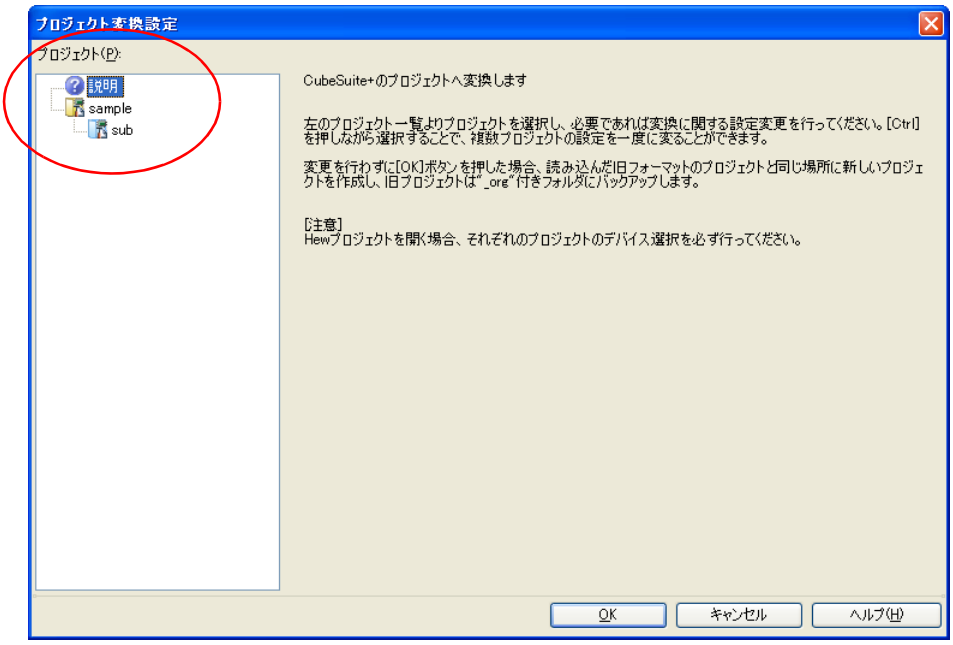
［はい］ボタンを選択すると，CubeSuite のプロジェクト設定を変更せずに，CubeSuite のプロジェクトに変換します。

使用するマイクロコントローラやプロジェクト名などを変更する場合は，［いいえ］ボタンをクリックします（（2）に進んでください）。

（2）変換設定対象プロジェクトを選択

プロジェクト変換設定 ダイアログがオープンします。

図 2—64 プロジェクト変換設定 ダイアログ



［プロジェクト］エリアに変換先プロジェクトの構成がツリー表示されますので，変換設定を行うプロジェクトを選択します。

(3) 変換先プロジェクトを設定

プロジェクトを選択すると，右側のエリアに変換先プロジェクトの設定項目が表示されます。

図 2—65 プロジェクト変換設定 ダイアログ（メイン・プロジェクトを選択した場合）

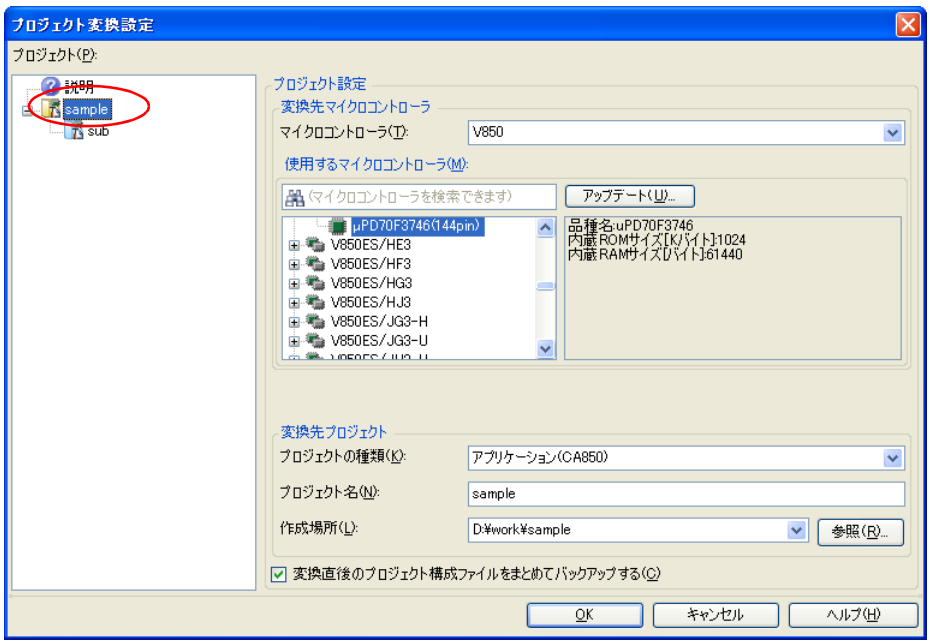
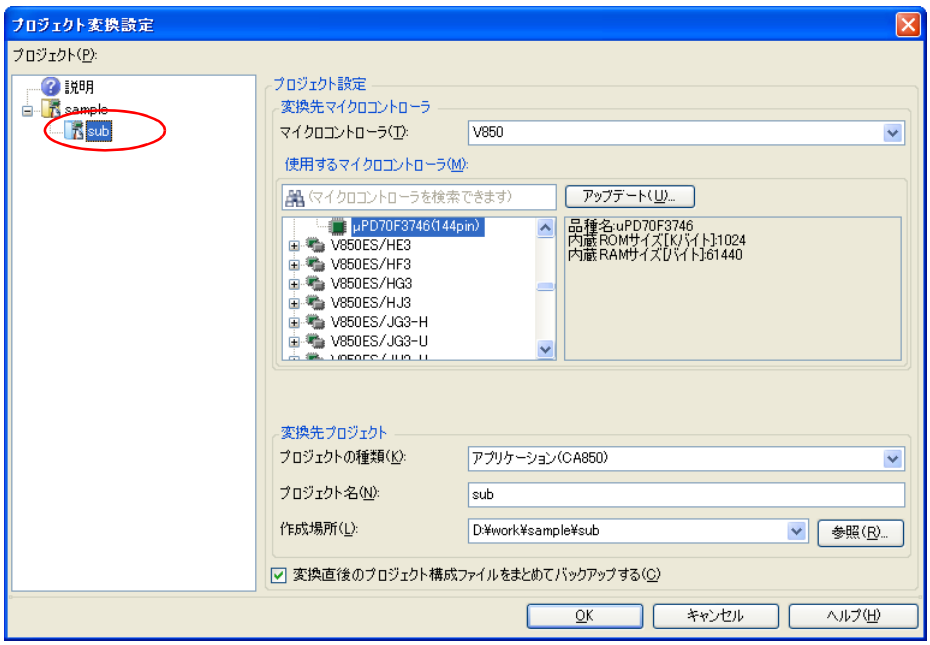


図 2—66 プロジェクト変換設定 ダイアログ（サブプロジェクトを選択した場合）



変換先プロジェクトで使用するマイクロコントローラ，およびプロジェクトの種類，名前，作成場所を設定した後，［OK］ボタンをクリックします。

備考 1. CubeSuite のプロジェクトにサブプロジェクトが存在しない場合，サブプロジェクトは表示されません。

2. ［変換直後のプロジェクト構成ファイルをまとめてバックアップする］をチェックすると，変換直後のプロジェクトのソース・ファイル，およびプロジェクト一式をパックして保存することができます。

3. 各設定項目についての詳細は，「プロジェクト変換設定 ダイアログ」を参照してください。

（4）CubeSuite+ のプロジェクトに変換

CubeSuite のプロジェクトが CubeSuite+ のプロジェクトに変換されます。

なお，CubeSuite のプロジェクトは，以下の規則で CubeSuite+ のプロジェクトに変換されます。

- CubeSuite のプロジェクトは，元のプロジェクトと同名のプロジェクトに変換されます。変換後のプロジェクト・ファイル名は，“*プロジェクト名* .mtpj”となります。

備考 1. 変換後のプロジェクト・ファイルは，CubeSuite のプロジェクト・ファイルが置かれていたフォルダに作成されます。すでに同名のプロジェクト・ファイルが存在する場合は，ファイル名に“_*数字*_”（*数字* : 1，2，…）が付加されます。

2. ビルド・ツールが異なる場合には，ビルド・モード，ビルド・オプションは変換対象にはなりません。

2. 7. 9 HEW のプロジェクトを CubeSuite+ のプロジェクトに変換する

High-performance Embedded Workshop（以降，“HEW”と略します）の RX プロジェクトは，CubeSuite+ の［プロジェクト］メニュー→［プロジェクトを開く ...］でワークスペース・ファイル（*.hws），またはプロジェクト・ファイル（*.hwp）を選択することにより，CubeSuite+ の RX プロジェクトに変換することができます。

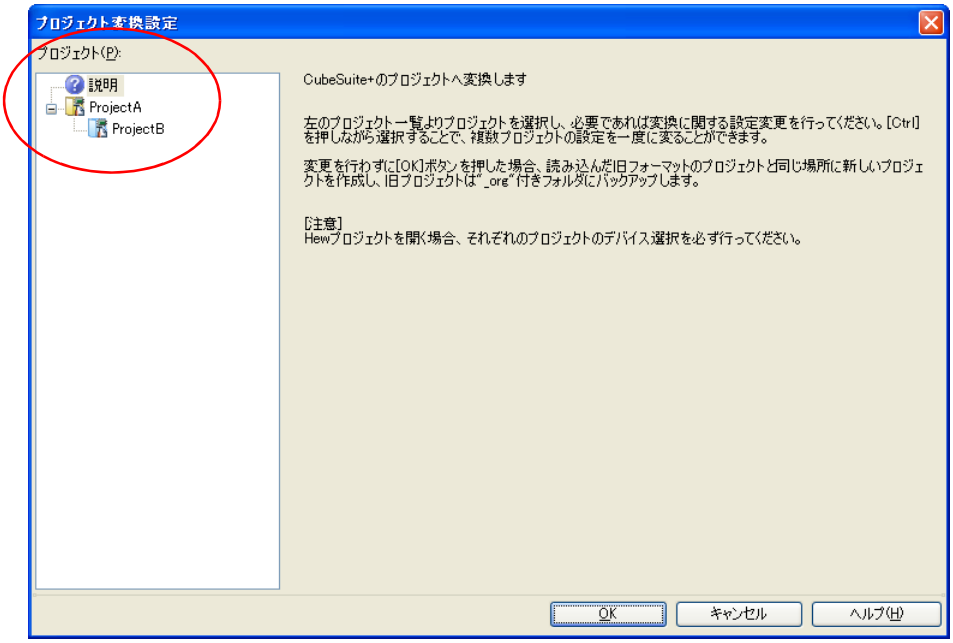
- HEW のワークスペース・ファイル（*.hws）から開く場合
- HEW のプロジェクト・ファイル（*.hwp）から開く場合

（1）HEW のワークスペース・ファイル（*.hws）から開く場合

（a）変換設定対象プロジェクトを選択

HEW のワークスペース・ファイルを選択すると，プロジェクト変換設定 ダイアログがオープンします。

図 2—67 プロジェクト変換設定 ダイアログ



［プロジェクト］エリアに変換先プロジェクトの構成がツリー表示されますので，変換設定を行うプロジェクトを選択します。

(b) 変換先プロジェクトを設定

プロジェクト名を選択すると，右側のエリアに変換先プロジェクトの設定項目が表示されます。

図 2—68 プロジェクト変換設定 ダイアログ（メイン・プロジェクトを選択した場合）

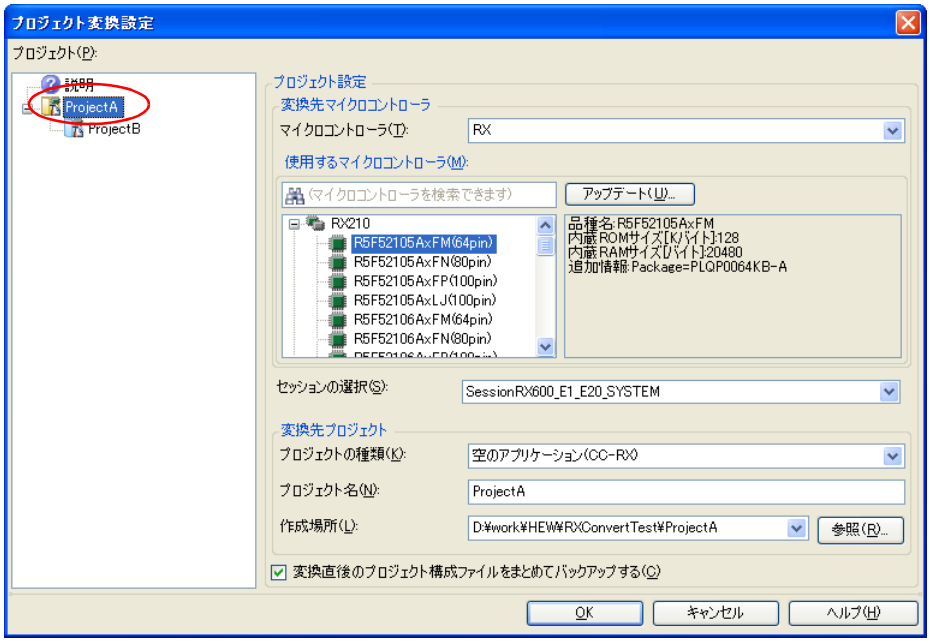
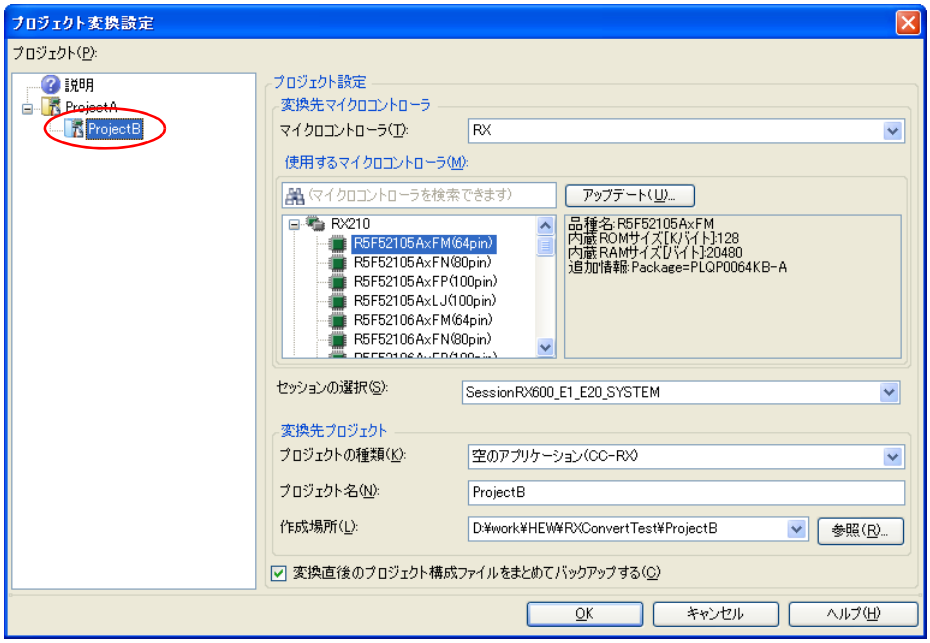


図 2—69 プロジェクト変換設定 ダイアログ（サブプロジェクトを選択した場合）



変換先プロジェクトで使用するマイクロコントローラ，セッション，およびプロジェクトの種類，名前，作成場所を設定した後，［OK］ボタンをクリックします。

備考 1. HEW のワークスペースにプロジェクトが 1 つしか存在しない場合，サブプロジェクトは表示されません。

2. ［セッションの選択］は，プロジェクトに複数のセッションが存在する場合のみ表示されます。

3. ［変換直後のプロジェクト構成ファイルをまとめてバックアップする］をチェックすると，変換直後のプロジェクトのソース・ファイル，およびプロジェクト一式をパックして保存することができます。

4. 各設定項目についての詳細は，「プロジェクト変換設定 ダイアログ」を参照してください。

注意 CPU オプション，およびセクション（start）オプションは，選択デバイスにあわせて変更しません。HEW のプロジェクトとは異なるデバイスを選択する場合は，プロジェクトを変換後，CPU オプション，およびセクション（start）オプションを確認（変更）してください。

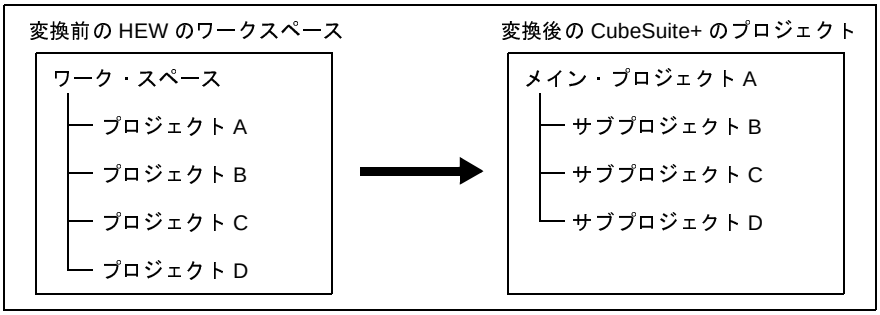
（c）CubeSuite+ のプロジェクトに変換

HEW のプロジェクトが CubeSuite+ のプロジェクトに変換されます。

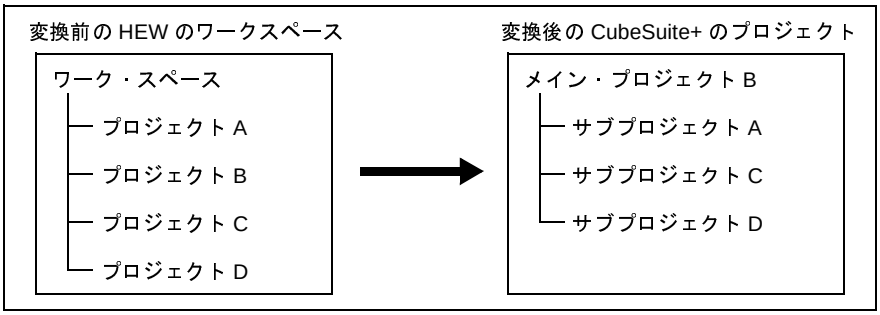
なお，HEW のワークスペースから開いた場合，プロジェクトは以下の規則で変換されます。

- HEW のワークスペースは変換されません。
- HEW のプロジェクトは，ワークスペース・ファイルに記載されているプロジェクト間の依存関係に応じて，メイン・プロジェクト，またはサブプロジェクトに変換されます。
  - プロジェクト間に依存関係がない場合は，ワークスペース・ファイル内におけるプロジェクトの出現順に，プロジェクト・ツリーに追加されます。
  - プロジェクト間に依存関係がある場合は，他のプロジェクトが依存していない，最初に出現するプロジェクトがメイン・プロジェクトとなります。
    サブプロジェクトは，プロジェクト・ツリーの先頭からビルドするため，依存関係とは逆の順番でプロジェクト・ツリーに追加されます。

例 1. プロジェクト A，B，C，D の間に依存関係がない場合



2. プロジェクトCがプロジェクトAに依存し，プロジェクトB，Dには依存関係がない場合



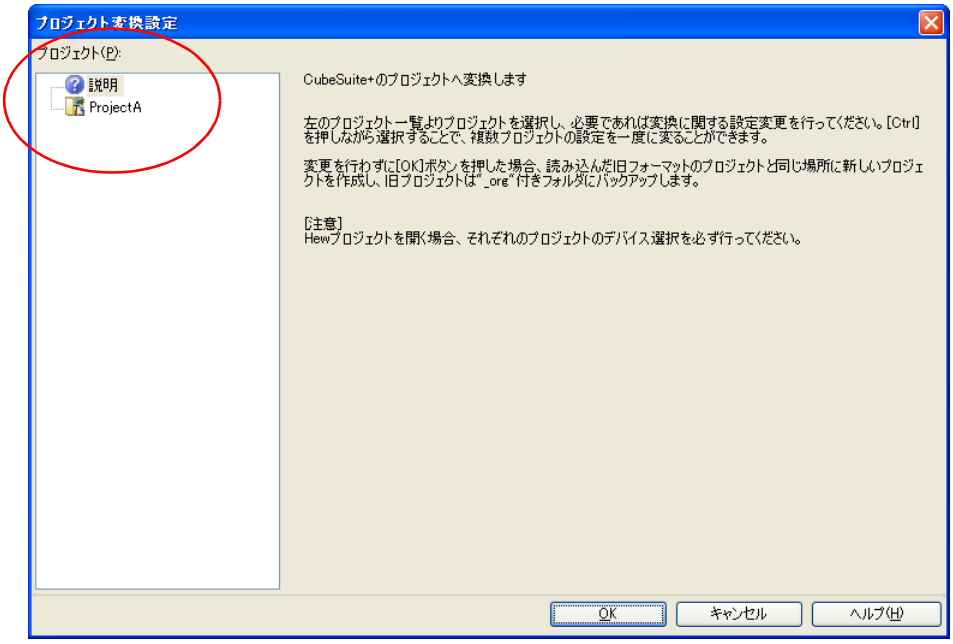
- ビルド・オプションに使用していた HEW のプレースホルダは，CubeSuite+ のプレースホルダに置き換わります。ただし，HEW と CubeSuite+ では，ワークスペース，およびプロジェクトの概念が異なるため，変換後のプレースホルダが正しい情報を指していない場合があります。必要に応じてビルド・オプションを変更してください。
- 変換後のメイン・プロジェクト・ファイル名は，“*プロジェクト名*.mtpj”，サブプロジェクト・ファイル名は，“*プロジェクト名*.mtsp”となります。
- 変換後のプロジェクトでは，ワークスペース・ファイル内で最初に出現するプロジェクトがアクティブ・プロジェクトとなります。

(2) HEW のプロジェクト・ファイル（*.hwp）から開く場合

(a) 変換設定対象プロジェクトを選択

HEW のプロジェクト・ファイルを選択すると，プロジェクト変換設定 ダイアログがオープンします。

図 2—70 プロジェクト変換設定 ダイアログ

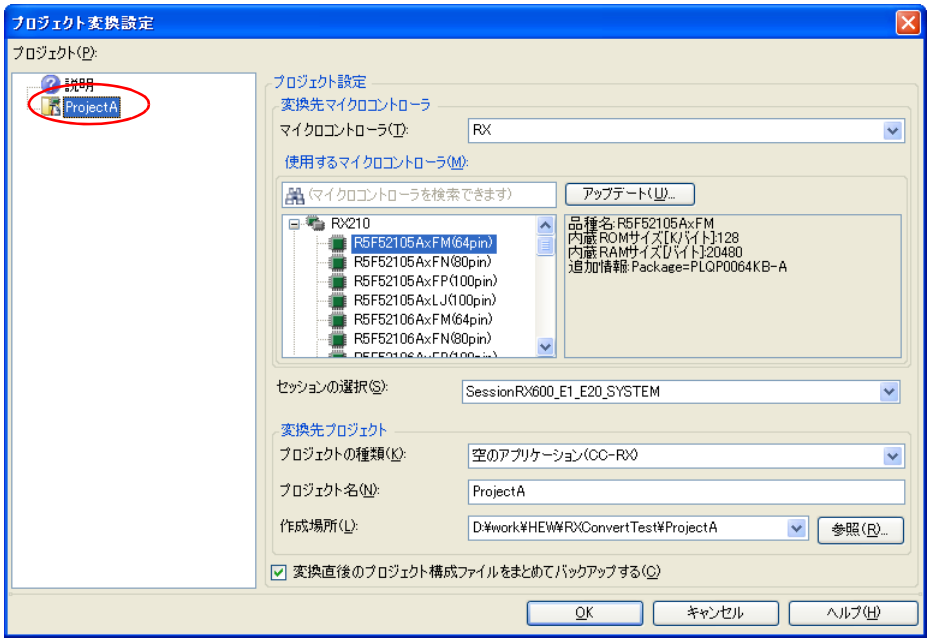


［プロジェクト］エリアに変換先プロジェクトの構成がツリー表示されますので，変換設定を行うプロジェクトを選択します。

(b) 変換先プロジェクトを設定

プロジェクト名を選択すると，右側のエリアに変換先プロジェクトの設定項目が表示されます。

図 2—71 プロジェクト変換設定 ダイアログ（プロジェクトを選択した場合）



変換先プロジェクトで使用するマイクロコントローラ，セッション，およびプロジェクトの種類，名前，作成場所を設定した後，［OK］ボタンをクリックします。

備考 1. ［セッションの選択］は，プロジェクトに複数のセッションが存在する場合のみ表示されます。

2. ［変換直後のプロジェクト構成ファイルをまとめてバックアップする］をチェックすると，変換直後のプロジェクトのソース・ファイル，およびプロジェクト一式をパックして保存することができます。

3. 各設定項目についての詳細は，「プロジェクト変換設定 ダイアログ」を参照してください。

(c) CubeSuite+ のプロジェクトに変換

HEW のプロジェクトが CubeSuite+ のプロジェクトに変換されます。

なお，HEW のプロジェクトから開いた場合，以下の規則で変換されます。

- HEW のプロジェクトは，元のプロジェクトと同名のプロジェクトに変換されます。変換後のプロジェクト・ファイル名は，“*プロジェクト名*.mtpj”となります。

備考 1. 変換対象は，HEW のプロジェクト・ファイルのファイル構成情報，コンフィグレーション，ビルド・オプションの情報です。
ただし，ビルド・ツールが異なる場合には，コンフィグレーション，ビルド・オプションは変換対象にはなりません。

2. HEW のプロジェクトの対象ビルド・ツールのバージョン以降から，変換後のビルド・ツールのバージョンまでの間にビルド・オプションが追加されていた場合，追加されたオプションは変換後のビルド・ツールのデフォルト値となります。

3. 変換後のプロジェクト・ファイルは，HEW のプロジェクト・ファイルが置かれていたフォルダに作成されます。すでに同名のプロジェクト・ファイルが存在する場合は，変換処理を中断します。

4. HEW のプロジェクトのコンフィグレーションは，変換後，CubeSuite+ のビルド・モードに置き換わります。変換後のプロジェクトには，HEW のプロジェクトのコンフィグレーション以外に DefaultBuild ビルド・モードが追加され，それがデフォルトで選択されます。HEW のプロジェクトを使用する場合は，ビルド・モードを変更してください。
ただし，CubeSuite+ におけるフォルダ名として使用できない文字（\，/，:，*，?，"，<，>，|）が含まれている場合は，“_”に置き換えます。
コンフィグレーション等でビルド対象ファイルを切り替えているプロジェクトは変換できません。

5. 変換をサポートする HEW のバージョンは，4.07 以降です。なお，使用するビルド・ツールには，インストールされているコンパイラ・パッケージの最新版が設定されます。

6. HEW のプロジェクトを CubeSuite+ のプロジェクトに変換した結果は，プロジェクト変換情報としてファイルに出力されます。
- プロジェクト変換情報ファイル名は，“プロジェクト変換情報 _ *プロジェクト名* .txt”となります。
- プロジェクト変換情報ファイルは，変換後の各プロジェクト（サブプロジェクト）ごとに出力されます。
- プロジェクト変換情報ファイルの出力先は，各プロジェクト（サブプロジェクト）のプロジェクト・フォルダです。
- 出力されたプロジェクト変換情報ファイルは，プロジェクト・ツリー パネルにおいて，各プロジェクト（サブプロジェクト）のファイル・ノードにも表示されます。

7. カスタム・ビルド・フェーズを使用しているプロジェクトは変換できません。

8. 変換対象は，ルネサス エレクトロニクス 社製のコンパイラの HEW のプロジェクト・ファイルのみです。

プロジェクト変換情報ファイルの出力例を以下に示します。

(1) *プロジェクト変換日時*

(2) <IDE バージョン>

CubeSuite+ IDE: *バージョン* [*日付*]

HewProjectConverterLibrary: *バージョン* [日付]

HewProjectConverter: *バージョン* [日付]

(3) < コンパイラ・パッケージ・バージョン >

HEW プロジェクトのコンパイラ・パッケージ: バージョン -> CubeSuite+ プロジェクトのコンパイラ・パッケージ: バージョン

(4) < 利用しないオプション (*ビルド・モード名*)>

ツール名 (*CubeSuite+ のビルド・ツールのプロパティのタブ*)

オプション

:

(5) < 変更したオプション (*ビルド・モード名*)>

ツール名 (*CubeSuite+ のビルド・ツールのプロパティのタブ*)

HEW プロジェクトのオプション -> CubeSuite+ プロジェクトのオプション

:

項番	説明
(1)	プロジェクト変換日時 HEW のプロジェクトを CubeSuite+ のプロジェクトに変換した日時を，“*YYYY* 年 *MM* 月 *DD* 日 *hh*:*mm*:*ss*”の形式で出力します。
(2)	IDE バージョンと日付 CubeSuite+ の IDE，HewProjectConverterLibrary，HewProjectConverter のバージョンと日付を出力します。
(3)	コンパイラ・パッケージ・バージョン HEW プロジェクトで使用しているコンパイラ・パッケージとそのバージョン，および CubeSuite+ プロジェクトで使用しているコンパイラ・パッケージとそのバージョンを出力します。 なお，CubeSuite+ プロジェクトのコンパイラ・パッケージのバージョンは，CubeSuite+ 環境にインストールしているパッケージのうち，最新バージョンとなります。

項番	説明
(4)	利用しないオプション (*ビルド・モード名*) HEW プロジェクトで設定していたが，CubeSuite+ プロジェクトでは利用しないオプション（コンパイラ・パッケージのバージョンアップに伴い削除されたオプション，および HEW 自身のオプション）が存在する場合，その情報をビルド・モードごとに以下の形式で出力します。 *ツール名* (*CubeSuite+ のビルド・ツールのプロパティのタブ名*) *オプション* : - ビルド・モードは，“Debug Build”，“Release Build”，ユーザ作成ビルド・モードの順に出力されます（“Debug Build”，“Release Build”は，HEW がデフォルトで用意しているビルド・モードで，これらはデバッグ情報を出力するオプションの有無の点で異なります）。 - *ツール名*に該当する *CubeSuite+ のビルド・ツールのプロパティのタブ名*が存在しない場合は，“その他のオプション”を出力します。
(5)	変更したオプション (*ビルド・モード名*) HEW プロジェクトで設定していたオプションで，CubeSuite+ プロジェクトで利用するために変更したオプション（パラメータの指定可能範囲が変更されたオプション，コンパイラ・パッケージのバージョンアップに伴い変更されたオプション）が存在する場合，ビルド・モードごとに以下の形式で出力します。 *ツール名* (*CubeSuite+ のビルド・ツールのプロパティのタブ名*) *HEW プロジェクトのオプション -> CubeSuite+ プロジェクトのオプション* : - ビルド・モードは，“Debug Build”，“Release Build”，ユーザ作成ビルド・モードの順に出力されます（“Debug Build”，“Release Build”は，HEW がデフォルトで用意しているビルド・モードで，これらはデバッグ情報を出力するオプションの有無の点で異なります）。 - *ツール名*に該当する *CubeSuite+ のビルド・ツールのプロパティのタブ名*が存在しない場合は，“その他のオプション”を出力します。

2. 7. 10　PM+ のプロジェクトを CubeSuite+ のプロジェクトに変換する

PM+ のプロジェクトは，CubeSuite+ の［プロジェクト］メニュー→［プロジェクトを開く ...］でワークスペース・ファイル（*.prw），またはプロジェクト・ファイル（*.prj）を選択することにより，CubeSuite+ のプロジェクトに変換することができます。

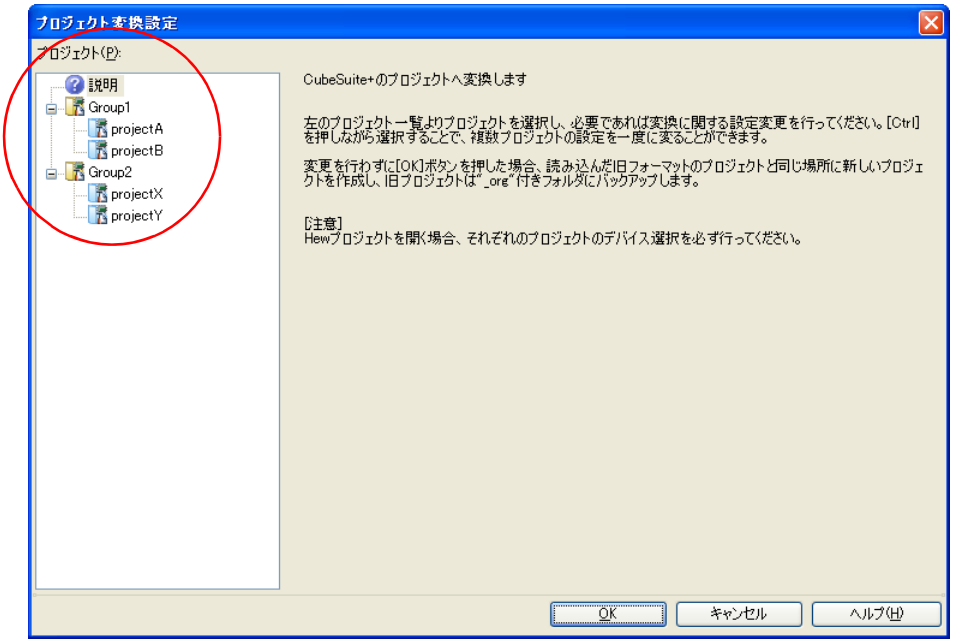
- PM+ のワークスペース・ファイル（*.prw）から開く場合
- PM+ のプロジェクト・ファイル（*.prj）から開く場合

（1）PM+ のワークスペース・ファイル（*.prw）から開く場合

（a）変換設定対象プロジェクトを選択

PM+ のワークスペース・ファイルを選択すると，プロジェクト変換設定 ダイアログがオープンします。

図 2—72　プロジェクト変換設定 ダイアログ



［プロジェクト］エリアに変換先プロジェクトの構成がツリー表示されますので，変換設定を行うプロジェクトを選択します。

(b) 変換先プロジェクトを設定

プロジェクト名を選択すると，右側のエリアに変換先プロジェクトの設定項目が表示されます。

図 2—73 プロジェクト変換設定 ダイアログ（メイン・プロジェクトを選択した場合）

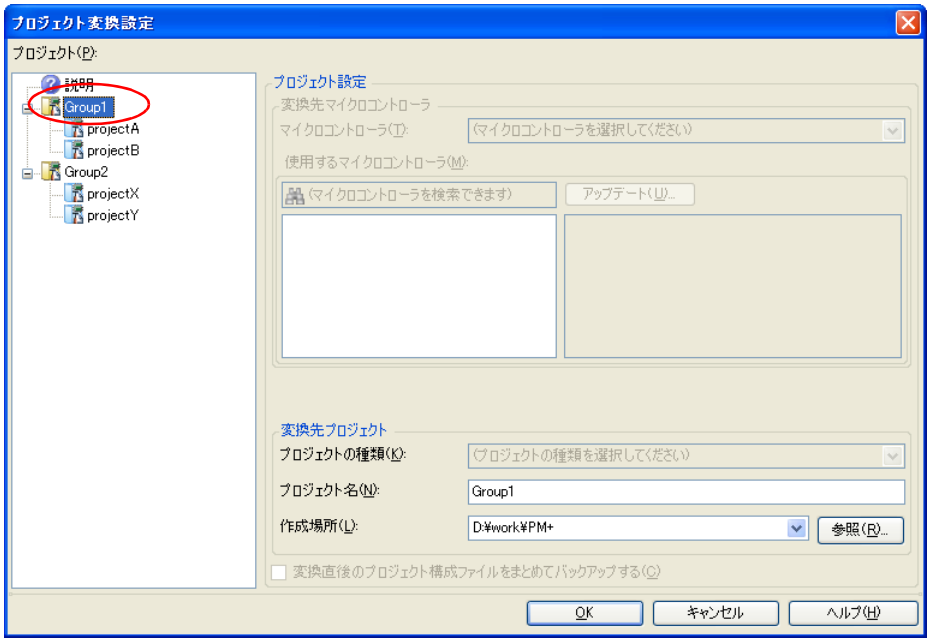
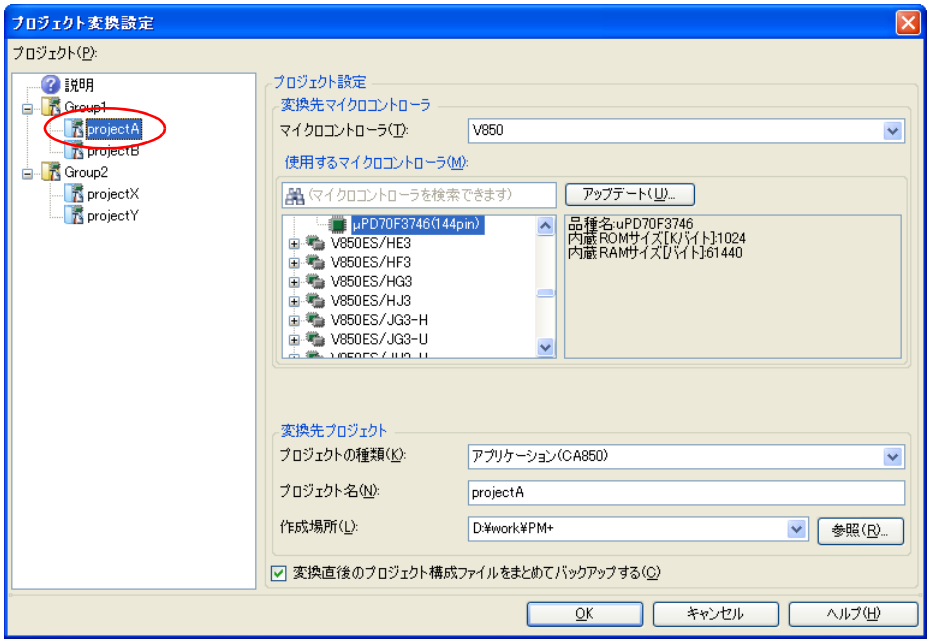


図 2—74 プロジェクト変換設定 ダイアログ（サブプロジェクトを選択した場合）



変換先プロジェクトで使用するマイクロコントローラ，およびプロジェクトの種類，名前，作成場所を設定した後，［OK］ボタンをクリックします。

備考 1. メイン・プロジェクトを選択した場合，［変換先マイクロコントローラ］エリア，および［変更先プロジェクト］エリアの［プロジェクトの種類］は無効となります。

2. ［変換直後のプロジェクト構成ファイルをまとめてバックアップする］をチェックすると，変換直後のプロジェクトのソース・ファイル，およびプロジェクト一式をパックして保存することができます。
ただし，メイン・プロジェクトを選択した場合，本項目は無効となります。

3. 各設定項目についての詳細は，「プロジェクト変換設定 ダイアログ」を参照してください。

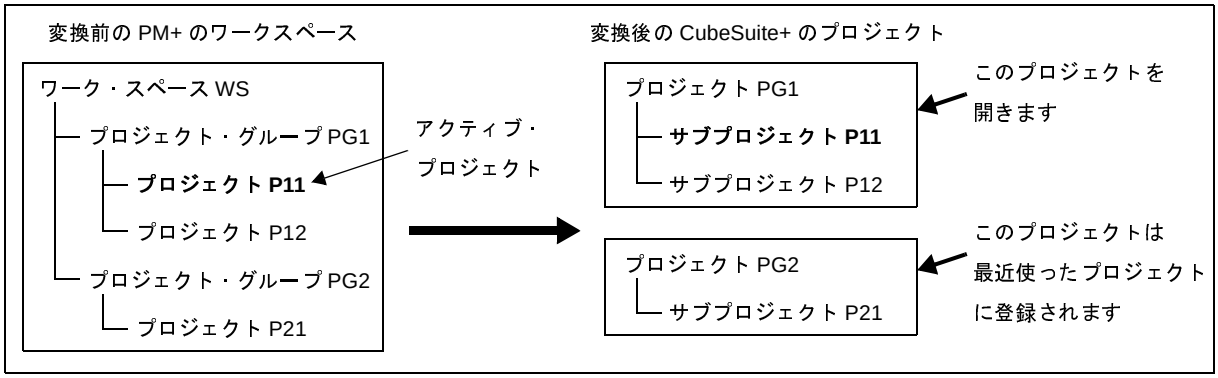
（c）CubeSuite+ のプロジェクトに変換

PM+ のプロジェクトが CubeSuite+ のプロジェクトに変換されます。

なお，PM+ のプロジェクトは，以下の規則で CubeSuite+ のプロジェクトに変換されます。

- PM+ のワークスペースは変換されません。
- PM+ のプロジェクト・グループは，プロジェクト・グループと同名のプロジェクトに分割されます。変換後のプロジェクト・ファイル名は，“*プロジェクト・グループ名* .mtpj”となります。
- PM+ のプロジェクトは，元のプロジェクトと同名のサブプロジェクトに変換されます。変換後のサブプロジェクト・ファイル名は，“*プロジェクト名* .mtsp”となります。
- 変換後のプロジェクトにおいて，プロジェクト・ツリーの先頭に表示されるサブプロジェクトがアクティブ・プロジェクトとなります。
- 変換後のプロジェクトにおいて，PM+ のリンク順は保持されません。

図 2—75 PM+ のプロジェクトの変換イメージ



備考 1. 変換後，CubeSuite+ は PM+ のアクティブ・プロジェクトを含むプロジェクト・グループを変換したプロジェクトを開きます。それ以外のプロジェクトは，最近使ったプロジェクトに登録されます。ただし，登録数が 4 つを越えた分は登録されません。

2. 変換後のプロジェクトの設定は，PM+ のプロジェクト・グループの中で，ワークスペース・ファイルの一番はじめに記載されているプロジェクトの設定と同じになります。なお，ファイルは登録されていない状態となります。

3. 変換後の各プロジェクトを 1 つのプロジェクトにまとめたい場合は，1 つのプロジェクトにサブプロジェクトとして別のプロジェクトを追加してください。

4. ビルド・ツールが指定されてないプロジェクトを含むワーク・スペースを読み込んだ場合，マイクロコントローラに応じたビルド・ツールが設定されます。

注意 1. CubeSuite+ で読み込めるプロジェクトは，PM+ で読み込めてビルドが通る状態のプロジェクトのみです。

2. 同一フォルダに，同じファイル名（拡張子を除いたファイル名）のプロジェクト・ファイルとサブプロジェクト・ファイルが存在している場合，プロジェクト・ファイルが正しく保存されません。メイン・プロジェクト，またはサブプロジェクトのどちらかに対して，プロジェクト・ツリー上で名前を変更してください。

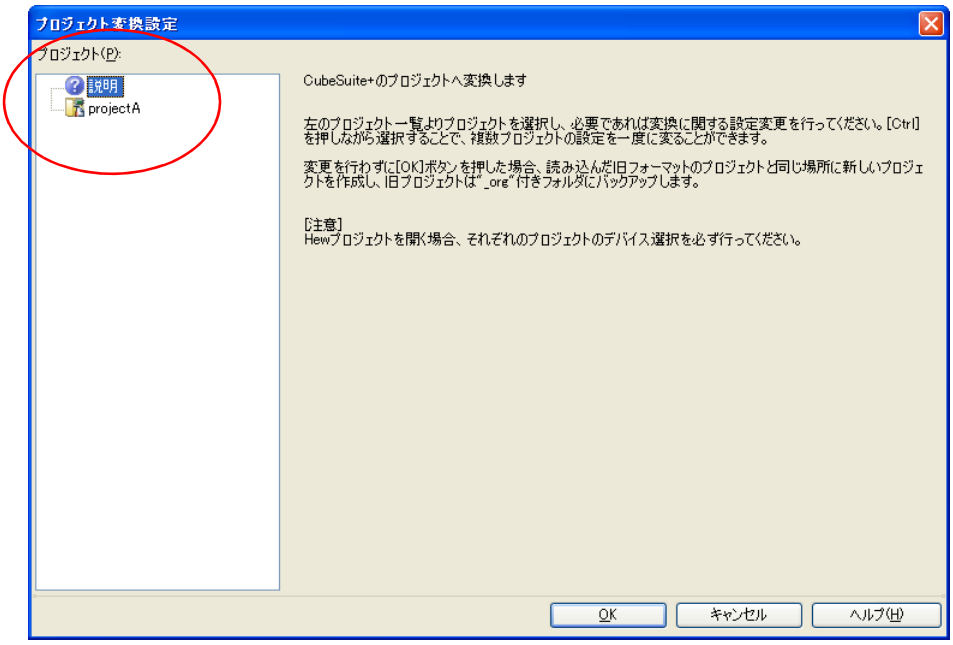
3. PM+ のワークスペースにプロジェクトが 1 つしか存在しない場合，そのプロジェクトはメイン・プロジェクトに変換されます。

（2）PM+ のプロジェクト・ファイル（*.prj）から開く場合

（a）変換設定対象プロジェクトを選択

PM+ のプロジェクト・ファイルを選択すると，プロジェクト変換設定 ダイアログがオープンします。

図 2—76 プロジェクト変換設定 ダイアログ

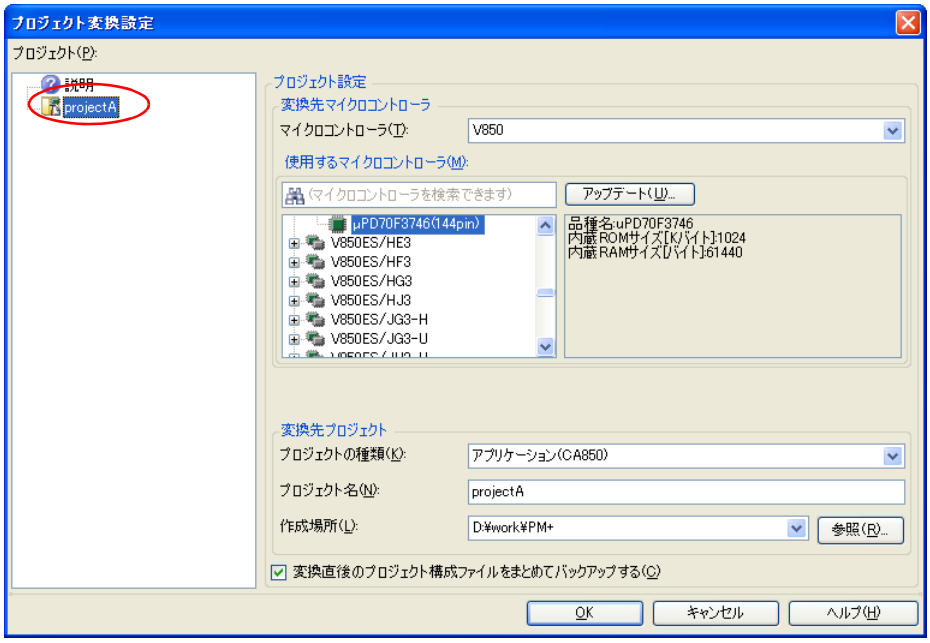


［プロジェクト］エリアに変換先プロジェクトの構成がツリー表示されますので，変換設定を行うプロジェクトを選択します。

(b) 変換先プロジェクトを設定

プロジェクト名を選択すると，右側のエリアに変換先プロジェクトの設定項目が表示されます。

図 2—77 プロジェクト変換設定 ダイアログ（プロジェクトを選択した場合）



変換先プロジェクトで使用するマイクロコントローラ，およびプロジェクトの種類，名前，作成場所を設定した後，［OK］ボタンをクリックします。

備考 1. ［変換直後のプロジェクト構成ファイルをまとめてバックアップする］をチェックすると，変換直後のプロジェクトのソース・ファイル，およびプロジェクト一式をパックして保存することができます。

2. 各設定項目についての詳細は，「プロジェクト変換設定 ダイアログ」を参照してください。

(c) CubeSuite+ のプロジェクトに変換

PM+ のプロジェクトが CubeSuite+ のプロジェクトに変換されます。

なお，PM+ のプロジェクトは，以下の規則で CubeSuite+ のプロジェクトに変換されます。

- PM+ のプロジェクトは，元のプロジェクトと同名のプロジェクトに変換されます。変換後のプロジェクト・ファイル名は，“*プロジェクト名*.mtpj”となります。

備考 ビルド・ツールが指定されてないプロジェクトを読み込んだ場合，マイクロコントローラに応じたビルド・ツールが設定されます。

注意 CubeSuite+ で読み込めるプロジェクトは，PM+ で読み込めてビルドが通る状態のプロジェクトのみです。

備考 1. 変換対象は，PM+ のプロジェクト・ファイルのファイル構成情報，ビルド・モード，ビルド・オプションの情報です。
ただし，ビルド・ツールが異なる場合には，ビルド・モード，ビルド・オプションは変換対象にはなりません。

2. PM+ のプロジェクトの対象ビルド・ツールのバージョン以降から，変換後のビルド・ツールのバージョンまでの間にビルド・オプションが追加されていた場合，追加されたオプションは変換後のビルド・ツールのデフォルト値となります。

3. 変換後のプロジェクト・ファイルは，PM+ のプロジェクト・ファイルが置かれていたフォルダに作成されます。すでに同名のプロジェクト・ファイルが存在する場合は，ファイル名に“_ *数字* _”（*数字* : 1，2，…）が付加されます。

4. 変換後のビルド・モードは，PM+ のプロジェクトのビルド・モードと同名となります。
ただし，CubeSuite+ におけるフォルダ名として使用できない文字（\，/，:，*，?，"，<，>，|）が含まれている場合は，“_”に置き換えます。
また，同名のビルド・モードが変換後のプロジェクトに用意される場合は，ビルド・モード名に“_ *数字* _”（*数字* : 1，2，…）が付加されます。

5. 変換をサポートする PM+ とビルド・ツールのバージョンは，PM+ V6.30 および CC78K0 V4.00/RA78K0 V4.00 以上【78K0】，PM+ V6.20 および CC78K0R V1.00/RA78K0R V1.00 以上【78K0R】，PM+ V6.00 および CA850 V3.00 以上【V850】です。なお，使用するビルド・ツールには，インストールされているコンパイラ・パッケージの最新版が設定されます。

6. PM+ のプロジェクトを CubeSuite+ のプロジェクトに変換した結果は，プロジェクト変換情報としてファイルに出力されます。
- プロジェクト変換情報ファイル名は，“プロジェクト変換情報 _ *プロジェクト名* .txt”となります。
- プロジェクト変換情報ファイルは，変換後の各プロジェクト（サブプロジェクト）ごとに出力されます。
- プロジェクト変換情報ファイルの出力先は，各プロジェクト（サブプロジェクト）のプロジェクト・フォルダです。
- 出力されたプロジェクト変換情報ファイルは，プロジェクト・ツリー パネルにおいて，各プロジェクト（サブプロジェクト）のファイル・ノードにも表示されます。

プロジェクト変換情報ファイルの出力例を以下に示します。

(1) *プロジェクト変換日時*

(2) <IDE バージョン>

CubeSuite+ IDE: *バージョン* [*日付*]

(3) < コンパイラ・パッケージ・バージョン >

PM+ プロジェクトのコンパイラ・パッケージ: バージョン -> CubeSuite+ プロジェクトのコンパイラ・パッケージ: バージョン

(4) < 利用しないオプション (*ビルド・モード名*)>

ツール名 (*CubeSuite+ のビルド・ツールのプロパティのタブ*)

オプション

:

(5) < 変更したオプション (*ビルド・モード名*)>

ツール名 (*CubeSuite+ のビルド・ツールのプロパティのタブ*)

PM+ プロジェクトのオプション -> CubeSuite+ プロジェクトのオプション

:

項番	説明
(1)	プロジェクト変換日時 PM+ のプロジェクトを CubeSuite+ のプロジェクトに変換した日時を，“*YYYY* 年 *MM* 月 *DD* 日 *hh*:*mm*:*ss*”の形式で出力します。
(2)	IDE バージョンと日付 CubeSuite+ の IDE バージョンと日付を出力します。
(3)	コンパイラ・パッケージ・バージョン PM+ プロジェクトで使用しているコンパイラ・パッケージとそのバージョン，および CubeSuite+ プロジェクトで使用しているコンパイラ・パッケージとそのバージョンを出力します。 なお，CubeSuite+ プロジェクトのコンパイラ・パッケージのバージョンは，CubeSuite+ 環境にインストールしているパッケージのうち，最新バージョンとなります。
(4)	利用しないオプション (*ビルド・モード名*) PM+ プロジェクトで設定していたが，CubeSuite+ プロジェクトでは利用しないオプション（コンパイラ・パッケージのバージョンアップに伴い削除されたオプション，および PM+ 自身のオプション）が存在する場合，その情報をビルド・モードごとに以下の形式で出力します。 *ツール名* (*CubeSuite+ のビルド・ツールのプロパティのタブ名*) *オプション* : - ビルド・モードは，“Debug Build”，“Release Build”，ユーザ作成ビルド・モードの順に出力されます（“Debug Build”，“Release Build”は，PM+ がデフォルトで用意しているビルド・モードで，これらはデバッグ情報を出力するオプションの有無の点で異なります）。 - *ツール名*に該当する *CubeSuite+ のビルド・ツールのプロパティのタブ名*が存在しない場合は，“その他のオプション”を出力します。

項番	説明
(5)	変更したオプション (*ビルド・モード名*) PM+ プロジェクトで設定していたオプションで，CubeSuite+ プロジェクトで利用するために変更したオプション（パラメータの指定可能範囲が変更されたオプション，コンパイラ・パッケージのバージョンアップに伴い変更されたオプション）が存在する場合，ビルド・モードごとに以下の形式で出力します。 *ツール名* (*CubeSuite+ のビルド・ツールのプロパティのタブ名*) *PM+ プロジェクトのオプション -> CubeSuite+ プロジェクトのオプション* : - ビルド・モードは，“Debug Build”，“Release Build”，ユーザ作成ビルド・モードの順に出力されます（“Debug Build”，“Release Build”は，PM+ がデフォルトで用意しているビルド・モードで，これらはデバッグ情報を出力するオプションの有無の点で異なります）。 - *ツール名*に該当する *CubeSuite+ のビルド・ツールのプロパティのタブ名*が存在しない場合は，“その他のオプション”を出力します。

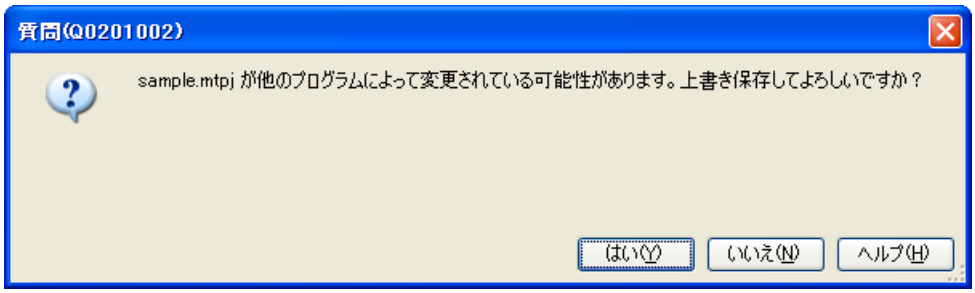
2.8 プロジェクト・ファイルを保存する

プロジェクトの設定情報は，プロジェクト・ファイル（*.mtpj）に保存します。

［ファイル］メニュー，または［プロジェクト］メニュー→［プロジェクトを保存］を選択してください。

プロジェクト・ファイルが他のプログラムによって変更されている場合，以下のメッセージ ダイアログがオープンします。

図 2—78 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［はい］をクリックしてください。

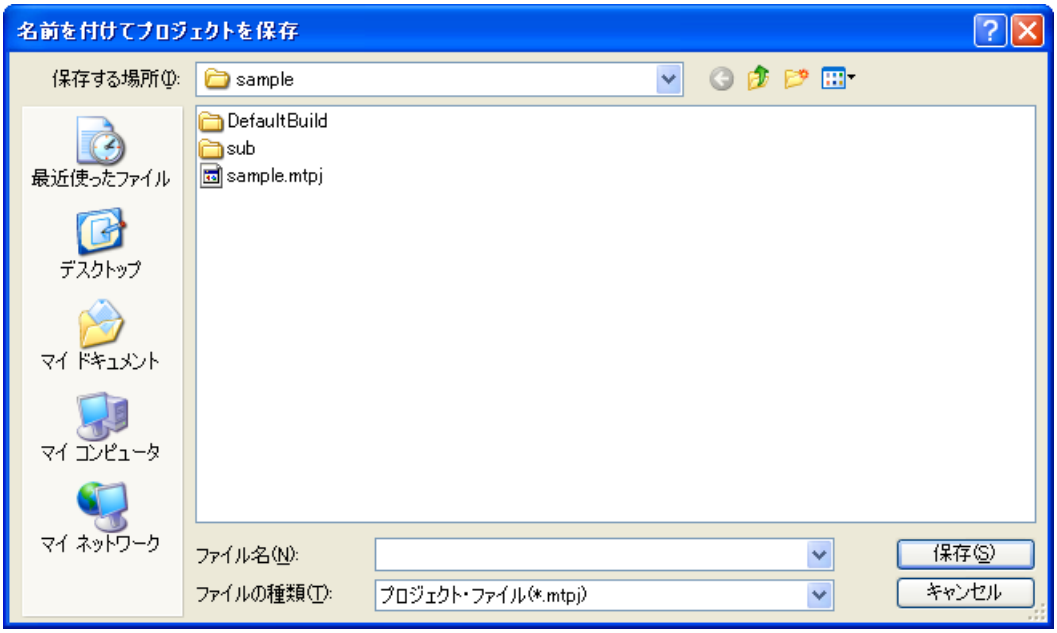
現在の設定情報でプロジェクト・ファイルを上書きします。

2.8.1 プロジェクト・ファイルを別名で保存する

プロジェクト・ファイルを別名で保存することができます。

［ファイル］メニュー，または［プロジェクト］メニュー→［名前を付けてプロジェクトを保存 ...］を選択すると，名前を付けてプロジェクトを保存 ダイアログがオープンします。

図 2—79 名前を付けてプロジェクトを保存 ダイアログ



ダイアログ上で，保存先フォルダと別のプロジェクト・ファイル名（*.mtpj）を指定してください。指定したフォルダとファイル名でプロジェクト・ファイルを保存します。

注意 1. プロジェクトに登録しているファイルは，元のプロジェクト同一のものになります（プロジェクトに登録しているファイルはコピーされません）。

2. 同一フォルダに，同じファイル名（拡張子を除いたファイル名）のプロジェクト・ファイルとサブプロジェクト・ファイルが存在している場合，プロジェクト・ファイルが正しく保存されません。メイン・プロジェクト，またはサブプロジェクトの名前を変更してください。

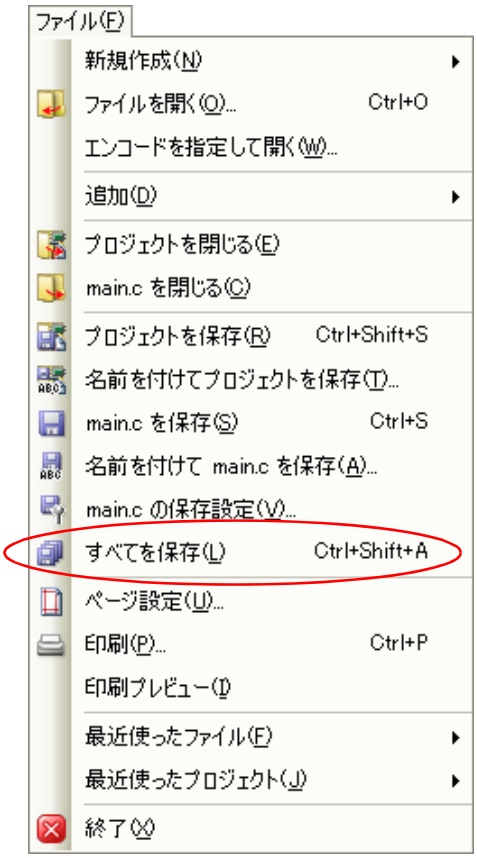
備考 元のプロジェクト・ファイルと別名で保存したプロジェクト・ファイルをそれぞれ変更した場合，お互いの内容も変わってしまいます。

2. 8. 2 すべてのファイルを保存する

プロジェクト・ファイル，および編集中のすべてのファイルを保存することができます。

［ファイル］メニュー→［すべてを保存］を選択すると，現在の設定情報でプロジェクト・ファイルを上書きし，編集中のすべてのファイルを保存します。

図 2—80 ［すべてを保存］項目



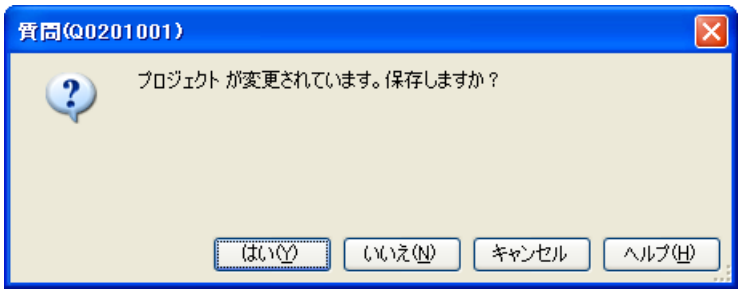
2. 8. 3 プロジェクトと本製品をパックして保存する

プロジェクト一式（サブプロジェクトが存在する場合はサブプロジェクトも含める）と，本製品一式（IDE，言語ツール，マイクロコントローラ情報など）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することができます。

プロジェクト・ツリーでプロジェクト・ノードを選択し，コンテキスト・メニューの［プロジェクトと開発ツールをパックして保存 ...］メニューを選択します。

プロジェクトの構成やプロパティに変更がある場合は，以下のメッセージ ダイアログがオープンします。

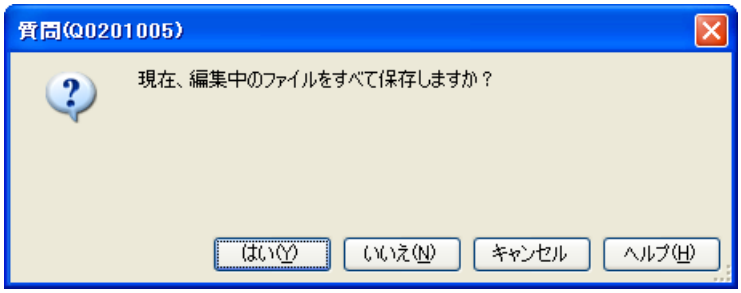
図 2—81 メッセージ ダイアログ



プロジェクトを保存する場合は［はい］，保存しない場合は［いいえ］をクリックしてください。

編集中のファイルがある場合は，以下のメッセージ ダイアログがオープンします。

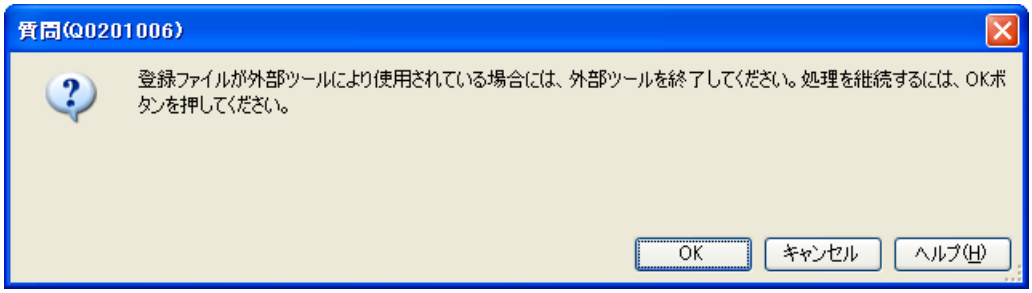
図 2—82 メッセージ ダイアログ



ファイルを保存する場合は［はい］，保存しない場合は［いいえ］をクリックしてください。

外部テキスト・エディタなどの外部ツールを使用し，編集中のファイルがある場合は，以下のメッセージ ダイアログがオープンします。

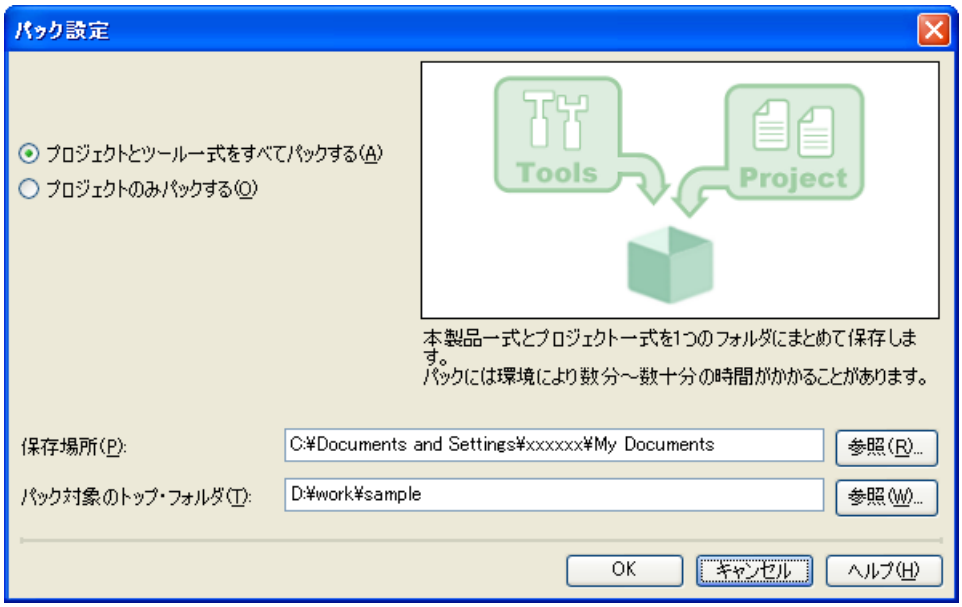
図 2—83 メッセージ ダイアログ



備考 本ダイアログは，オプション ダイアログの［全般 - テキスト・エディタ］カテゴリの［外部テキスト・エディタを使用する］プロパティをチェックしている場合のみ表示されます。

処理を継続するには，［OK］ボタンをクリックしてください。
パック設定 ダイアログがオープンします。

図 2—84 パック設定 ダイアログ

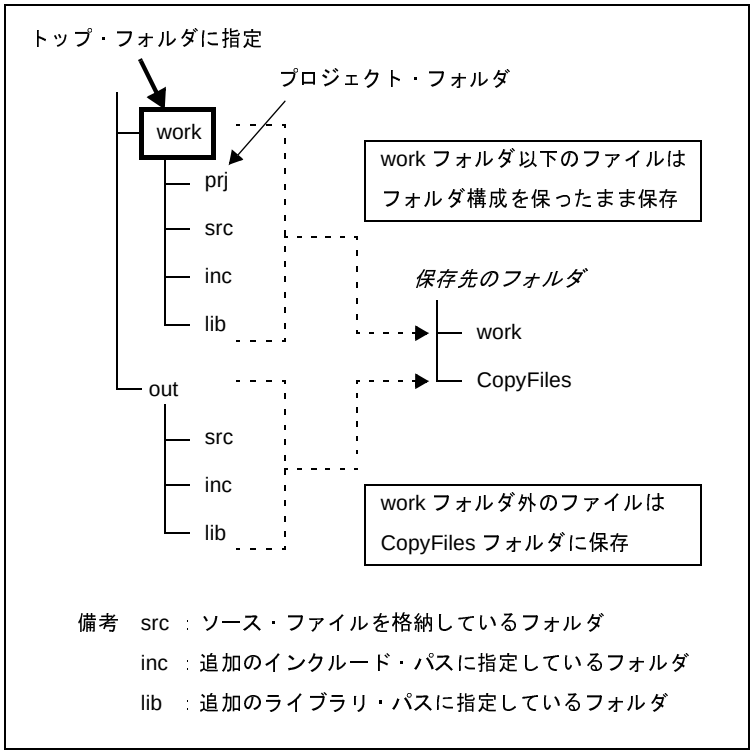


ダイアログ上で，パック方法（プロジェクトとツール一式，またはプロジェクトのみ），保存場所，パック対象のトップ・フォルダを指定します。

注意 保存場所には，プロジェクト・フォルダ以外のフォルダを指定してください。

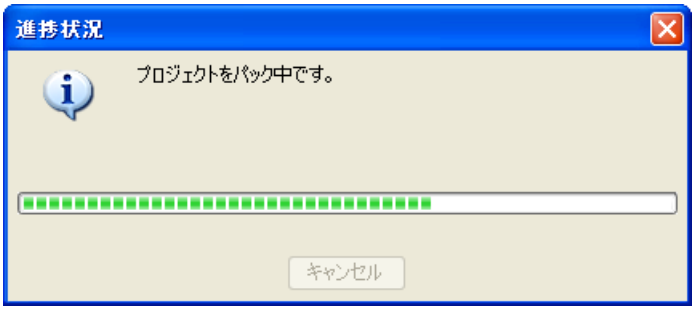
備考 パック対象のトップ・フォルダを指定することにより，プロジェクトのフォルダ構成を保ったまま保存することができます。

図 2—85 プロジェクトのパック例



［OK］ボタンをクリックすると，コピー処理を開始します。
処理中の間，以下のダイアログが表示されます。

図 2—86 コピー処理中



保存先のフォルダ構成を以下に示します。

図 2—87　保存先のフォルダ構成

```
保存先のフォルダ
 ├─ readme.txt        : 保存先のフォルダ構成等を説明文書
 ├─ errorlog.txt      : コピーに失敗したファイルの記録
 ├─ トップ・フォルダ  : パック対象のトップ・フォルダとして指定したフォルダ注1
 ├─ Tools             : 本製品がインストールされているフォルダ注2
 └─ Copy_Files        : パック対象のトップ・フォルダ内に置かれていないファイルのコピー先フォルダ注3
     ├─ copylog.txt   : コピーしたファイルの記録
     ├─ 1             : コピーしたファイルの格納フォルダ 1 注4
     ├─ 2             : コピーしたファイルの格納フォルダ 2 注4
     :
```

注 1. パック設定 ダイアログで指定したパック対象のトップ・フォルダ以下に存在する以下のファイルが格納されます。

- プロジェクト・フォルダ以下のすべてのファイル
- ビルド・ツールの［追加のインクルード・パス］プロパティ，および［追加のライブラリ・パス］プロパティで指定したフォルダ以下すべて
- プロジェクトに追加しているファイル

2. パック設定 ダイアログでプロジェクトのみを保存した場合は，本フォルダは作成しません。

3. パック対象のトップ・フォルダ内に置かれていないファイルが，同一パスごとにフォルダに分けて格納されます。

4. フォルダ名は，フォルダの必要数により桁数を調整します。たとえば，2 桁の場合は“01”，“02”…となります。

注意 1. **スタート メニューに関する情報は，保存の対象となりません。保存先のツールを使用する際は，保存先フォルダの readme.txt の内容に従ってください。**

2. **ツールのインストールに関する情報は，保存の対象となりません。保存先のツールをアンインストールするには，保存先のフォルダごと削除してください。**

3. **オプション ダイアログやユーザ設定 ダイアログで行ったカスタマイズに関する情報は，保存の対象となりません。保存先のツールを使用する際は，デフォルトの環境設定となります。**

4. **保存先のフォルダのツールは，アップデートを行うことができません。**

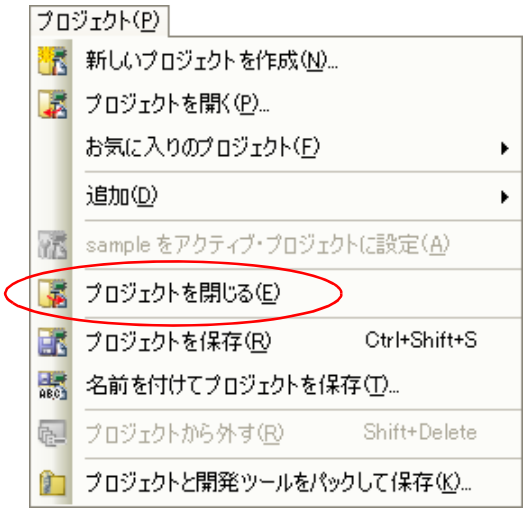
5. **パック設定 ダイアログで指定したパック対象のトップ・フォルダ内に置かれていないファイルは，保存時に Copy_files フォルダにコピーされるため，保存先のプロジェクトを使用する際に，再度プロジェクトへの登録を行う必要があります。**

6. **エミュレータのドライバは，保存の対象となりません。別のパソコンで，パックしたものを使用する場合は，別途，ドライバのインストールが必要になります。**

2. 8. 4 プロジェクトを閉じる

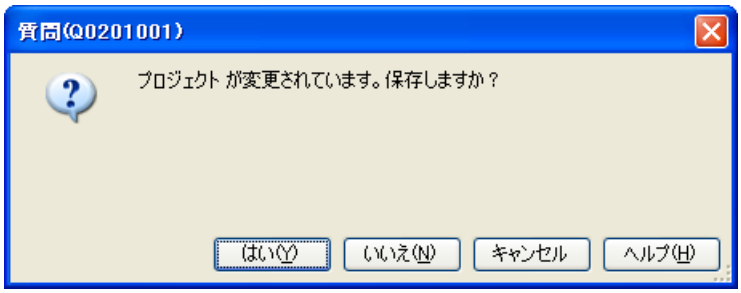
プロジェクトを閉じるには，［ファイル］メニュー，または［プロジェクト］メニュー→［プロジェクトを閉じる］を選択してください。

図 2—88 ［プロジェクトを閉じる］項目



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，メッセージ ダイアログが表示されます。

図 2—89 メッセージ ダイアログ



保存する場合は［はい］ボタン，保存しない場合は［いいえ］ボタンをクリックしてください。

2.9 ウインドウ・レイアウトを変更する

CubeSuite+ のウインドウ・レイアウトの変更方法について説明します。

2.9.1 パネルを自動的に隠す

パネルは，自動非表示機能をサポートしています。

自動非表示を使用すると，使用していないパネルがフレームの端に最小化されるため，一度により多くの情報を表示できるようになります。

(1) 自動非表示機能を有効にする

自動非表示機能を有効にするには，非表示にするパネルをクリックして選択します。

該当パネルのタイトル バーを右クリックしたのち［自動的に隠す］を選択，またはタイトル バーの をクリックします。

自動非表示に設定されたパネルがフォーカスを失うと，該当パネルのフレーム端に最小化されたアイコンとパネル名がタブとして表示されます。

自動非表示に設定されたパネルを表示するには，マウス・カーソルを該当タブ上に移動します。タブからパネルがオープンし，使用可能な状態になります。

該当パネルがフォーカスを失うと，パネルは再びタブとして最小化されます。

(2) 自動非表示機能を無効にする

自動非表示機能を無効にするには，表示しておきたいパネルをクリックして選択します。

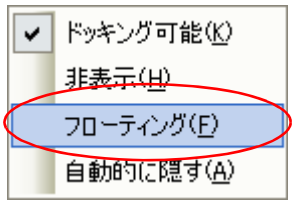
パネルのタイトル バーを右クリックし，［自動的に隠す］を再び選択，またはパネルのタイトル バーの をクリックします。

2.9.2 ウインドウを独立させる

独立させたいウインドウのタイトル バーをクリックして，移動します。

または，独立させたいウインドウのタイトル バーをダブルクリックするか，右クリックし，［フローティング］をクリックします。

図 2—90 コンテキスト・メニュー



2. 9. 3 ウインドウをドッキングする

独立したウインドウをメイン・ウインドウにドッキングすることができます。

ドッキングさせたいウインドウのタイトル バーをクリックして移動すると，メイン・ウインドウ上の中央と上下左右に自動的にドッキング・インジケータが表示されます。

ドッキング・インジケータ上にポイントを移すとメイン・ウインドウの一部の背景が青色になります。そこで，マウス ボタンを離すと青色部分だったところにウインドウがドッキングされます。

インジケータの選択により，以下のように自由自在に配置することが可能です。

	CubeSuite+ の上側に配置されます。
	CubeSuite+ の左側に配置されます。
	CubeSuite+ の右側に配置されます。
	CubeSuite+ の下側に配置されます。
の上	対象パネルの上側に配置されます。
の左	対象パネルの左側に配置されます。
の右	対象パネルの右側に配置されます。
の下	対象パネルの下側に配置されます。
の中心	対象パネルにタブとして配置されます。

また，ウインドウのタイトル バーをダブルクリックするか，右クリックし，［フローティング］をクリックすることにより，以前ドッキングされていた位置にドッキングされます。

メイン・ウインドウにドッキングされたものは，スプリッタをクリックして移動させることにより，ウインドウの大きさを調整することができます。

2. 9. 4 パネルを複数表示する

参照したいパネルをクリックして同時に参照したいパネル上に移動します。パネル上に「2. 9. 3 ウインドウをドッキングする」で説明したドッキング・インジケータが表示されるので，にポイントを移し，配置させたい位置によってマウスボタンをの上下左右のいずれかで離してください。

2. 9. 5 ウインドウ・レイアウトをリセットする

メイン・ウインドウの［表示］メニューの［レイアウトをリセット］を選択すると，ウインドウ・レイアウトを初期状態に戻すことができます。

図 2—91 ［レイアウトをリセット］項目

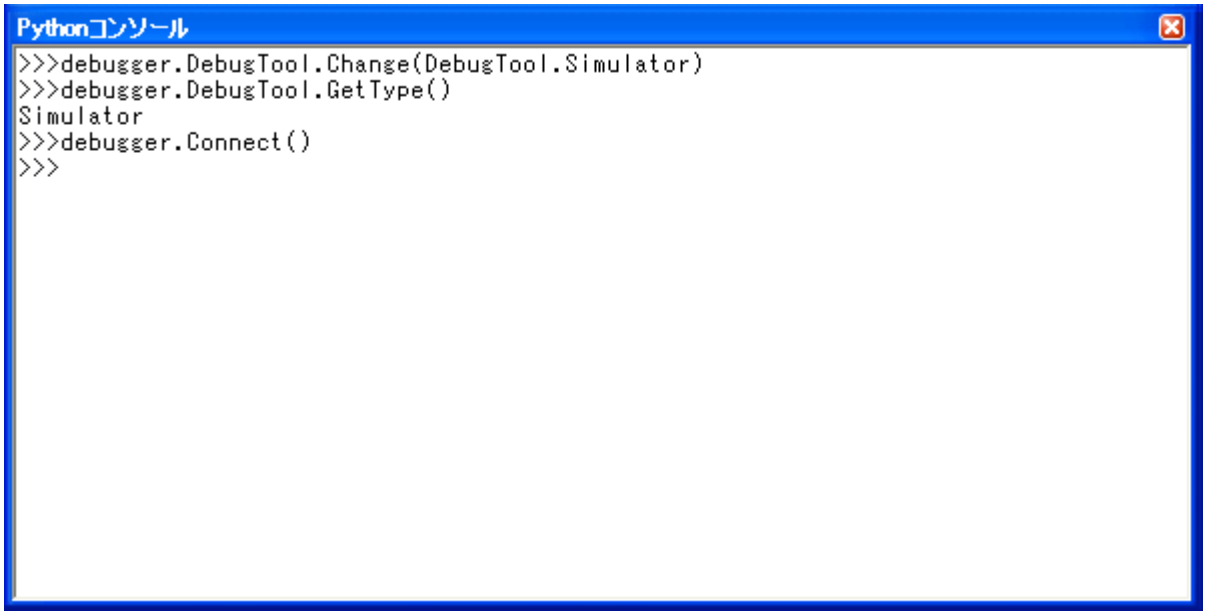
2. 10 Python 関数を実行する

CubeSuite+ では，IronPython 関数や制御文，および CubeSuite+ を制御するために追加された CubeSuite+ Python 関数（「G. 3 CubeSuite+ Python 関数／クラス／プロパティ」参照）をコマンド入力方式で実行することができます。

［表示］メニュー→［Python コンソール］を選択すると，Python コンソール パネルがオープンします。

パネル上で Python 関数や制御文を実行することにより，CubeSuite+，およびデバッグ・ツールを操作することができます。

図 2—92 Python コンソール パネル



備考 Python コンソール，および Python 関数の詳細については，「付録 G Python コンソール /Python 関数」を参照してください。

2. 11 CubeSuite+ をコマンドラインで操作する

CubeSuite+ は，コマンドライン（Windows のコマンド・プロンプト等）から起動したり，メイン・ウインドウを表示せずに，コマンド入力方式で操作することができます。

ここでは，Windows のコマンド・プロンプトから CubeSuite+ を起動するものとして説明します。

コマンド・プロンプト上で，CubeSuite+ のインストール・フォルダに格納されている CubeSuite+.exe，または CubeSuiteW+.exe を実行します。

(1) CubeSuite+.exe を実行する場合

CubeSuite+.exe を実行する場合は，メイン・ウインドウを表示せずに，CubeSuite+ の起動，プラグインの読み込み，ビルドの実行を行うことができます。

以下に，コマンドラインにおける指定形式を示します。

```
CubeSuite+.exe △ [[[/bb|/br|/bcb|/bc △ [ ビルド・モード名 ]]] △ [[/np △プラグイン名 [, プラグイン名 ,...]]] △ [/npall] △ [ プラグインに対するオプション△ [ プラグイン・オプションのパラメータ ,...]] △プロジェクト・ファイル名 ]
```

△ ：1 個以上の空白を示します。

[ ] ：[ ] 内は省略可能です。

| ：| で区切ったオプションのうち，いずれか 1 つが指定可能です。

... ：直前の [ ] 内のパターンの繰り返しが可能です。

以下に，各オプションの説明を示します。

オプション	説明
指定なし	CubeSuite+ をメイン・ウインドウを表示せずに起動して，何も行わずに終了します。
/bb △ [*ビルド・モード名*]	ビルドを実行します。 メイン・ウインドウを表示せずに，*プロジェクト・ファイル名*で指定したプロジェクトに含まれているすべてのプロジェクトを，*ビルド・モード名*で指定したビルド・モードでビルドして終了します。 *ビルド・モード名*で指定したビルド・モードを持たないプロジェクトの場合は，DefaultBuild をもとにビルド・モードを複製してビルドします。 *ビルド・モード名*で指定したビルド・モードが*プロジェクト・ファイル名*で指定したプロジェクトにて定義されていない場合は，エラーを表示して，処理を終了します。 *ビルド・モード名*を省略した場合は，DefaultBuild でビルドします。 *プロジェクト・ファイル名*を省略した場合は，エラーを表示して，処理を終了します。

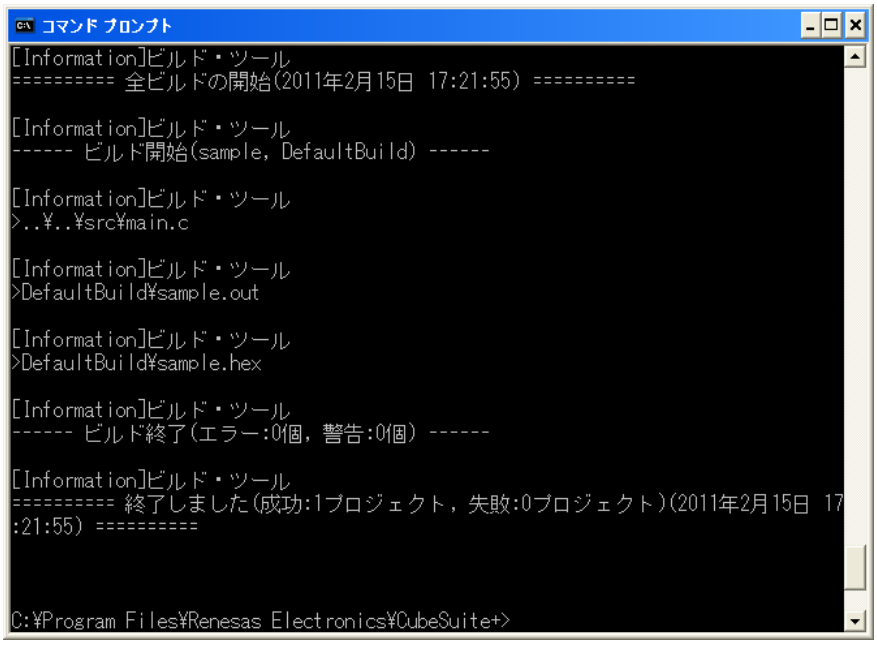
オプション	説明
/br △ [*ビルド・モード名*]	リビルドを実行します。 メイン・ウインドウを表示せずに，*プロジェクト・ファイル名*で指定したプロジェクトに含まれているすべてのプロジェクトを，*ビルド・モード名*で指定したビルド・モードでビルドして終了します。 *ビルド・モード名*で指定したビルド・モードを持たないプロジェクトの場合は，DefaultBuild をもとにビルド・モードを複製してビルドします。 *ビルド・モード名*で指定したビルド・モードが*プロジェクト・ファイル名*で指定したプロジェクトにて定義されていない場合は，エラーを表示して，処理を終了します。 *ビルド・モード名*を省略した場合は，DefaultBuild でビルドします。 *プロジェクト・ファイル名*を省略した場合は，エラーを表示して，処理を終了します。
/bcb △ [*ビルド・モード名*]	クリーンの後にビルドを実行します。 メイン・ウインドウを表示せずに，*プロジェクト・ファイル名*で指定したプロジェクトに含まれているすべてのプロジェクトを，*ビルド・モード名*で指定したビルド・モードでビルドして終了します。 *ビルド・モード名*で指定したビルド・モードを持たないプロジェクトの場合は，DefaultBuild をもとにビルド・モードを複製してビルドします。 *ビルド・モード名*で指定したビルド・モードが*プロジェクト・ファイル名*で指定したプロジェクトにて定義されていない場合は，エラーを表示して，処理を終了します。 *ビルド・モード名*を省略した場合は，DefaultBuild でビルドします。 *プロジェクト・ファイル名*を省略した場合は，エラーを表示して，処理を終了します。
/bc △ [*ビルド・モード名*]	クリーンを実行します。 メイン・ウインドウを表示せずに，*プロジェクト・ファイル名*で指定したプロジェクトに含まれているすべてのプロジェクトを，*ビルド・モード名*で指定したビルド・モードでビルドして終了します。 *ビルド・モード名*で指定したビルド・モードを持たないプロジェクトの場合は，DefaultBuild をもとにビルド・モードを複製してビルドします。 *ビルド・モード名*で指定したビルド・モードが*プロジェクト・ファイル名*で指定したプロジェクトにて定義されていない場合は，エラーを表示して，処理を終了します。 *ビルド・モード名*を省略した場合は，DefaultBuild でビルドします。 *プロジェクト・ファイル名*を省略した場合は，エラーを表示して，処理を終了します。

オプション	説明
/np △ *プラグイン名* [, *プラグイン名* ,...]	Plugins フォルダ以下のプラグインのうち，指定したプラグインを読み込まずに，メイン・ウインドウを表示せずに CubeSuite+ を起動します。 *プラグイン名*には，各 dll ファイルを格納しているフォルダ名を指定します。なお，フォルダ名の大文字／小文字は区別しません。 カンマで区切ることにより，複数プラグイン名を指定可能です。 /np オプション自体を複数記述した場合も許容し，各 DLL をロードしません。 本オプションで指定したプラグインが存在しない場合は，無視します。 *プロジェクト・ファイル名*を省略した場合は，エラーを表示して，処理を終了します。
/npall	Plugins フォルダ以下のプラグインを 1 つもロードしない状態で，CubeSuite+ をメイン・ウインドウを表示せずに起動します。 /np オプションと同時に指定した場合は，本オプションを優先します。 *プロジェクト・ファイル名*を省略した場合は，エラーを表示して，処理を終了します。
プラグイン・オプション△ [*プラグイン・オプションのパラメータ* ,...]	プラグインに対するオプションを指定します（「(3) プラグイン・オプション」参照）。
プロジェクト・ファイル名	指定したプロジェクト・ファイルを読み込んだ状態で，CubeSuite+ をメイン・ウインドウを表示せずに起動します。

備考　CubeSuite+ の実行中に［Ctrl］＋［C］キーを押下すると，CubeSuite+ を強制終了します。

なお，実行結果は，コマンド・プロンプト上に出力します。

図 2—93　コマンド・プロンプト（CubeSuite+.exe でビルドを実行した場合）



(2) CubeSuiteW+.exe を実行する場合

CubeSuiteW+.exe を実行する場合は，［スタート］メニューから起動した場合と同様，メイン・ウインドウを表示した CubeSuite+ の起動，プラグインの読み込みを行うことができます。

以下に，コマンドラインにおける指定形式を示します。

```
CubeSuiteW+.exe △ [[/np △プラグイン名 [, プラグイン名 ,...]]] △ [/npall] △ [/noproj] △ [/nrs]
△ [ プラグインに対するオプション△ [ プラグイン・オプションのパラメータ ,...]][ △プロジェクト・ファイル名 ]
```

△ ：1 個以上の空白を示します。

[ ] ：[ ] 内は省略可能です。

... ：直前の [ ] 内のパターンの繰り返しが可能です。

以下に，各オプションの説明を示します。

オプション	説明
指定なし	［スタート］メニューから起動した場合と同様，CubeSuite+ をメイン・ウインドウを表示して起動します。
/np △ *プラグイン名*△ [, *プラグイン名* ,...]	Plugins フォルダ以下のプラグインのうち，指定したプラグインを読み込まずに，メイン・ウインドウを表示して CubeSuite+ を起動します。 *プラグイン名*には，各 dll ファイルを格納しているフォルダ名を指定します。なお，フォルダ名の大文字／小文字は区別しません。 カンマで区切ることにより，複数プラグイン名を指定可能です。 /np オプション自体を複数記述した場合も許容し，各 DLL をロードしません。 本オプションで指定したプラグインが存在しない場合は，無視します。
/npall	Plugins フォルダ以下のプラグインを 1 つもロードしない状態で，CubeSuite+ をメイン・ウインドウを表示して起動します。 不正なオプションを指定した場合は，無視します。
/noproj	CubeSuite+ の起動直後に読み込むプロジェクト（コマンドラインに指定したプロジェクト，または前回使用したプロジェクト）を読み込まずに，メイン・ウインドウを表示して起動します。
/nrs	ラピッド・スタート（「2. 5 CubeSuite+ を起動する」参照）によりすでに待機している CubeSuite+ が起動していても無視して，そのまま自プロセスを起動します。
プラグイン・オプション△ [*プラグイン・オプションのパラメータ* ,...]	プラグインに対するオプションを指定します（「(3) プラグイン・オプション」参照）。
プロジェクト・ファイル名	指定したプロジェクト・ファイルを読み込んだ状態で，CubeSuite+ をメイン・ウインドウを表示して起動します。

(3) プラグイン・オプション

以下に，指定可能なプラグイン・オプションを示します。

オプション	説明
/ps △*スクリプト・ファイル名*	Python コンソール・プラグインのオプションです。 CubeSuite+ でプロジェクト・ファイルを読み込んだ後，*スクリプト・ファイル名*に記述している内容を実行します。 不要なパラメータを指定している場合，およびスクリプト・ファイルが読み込めない場合は，エラーを表示して，スクリプト・ファイルを実行しません。

スクリプト・ファイルの記述例を以下に示します。

```
debugger.Connect()
debugger.Download.LoadModule(r"C:¥project¥test¥DefaultBuild¥test.out")
debugger.Register.GetValue("pc")
breakpoint = BreakCondition()
breakpoint.Address = "func"
debugger.Breakpoint.Set(breakpoint)
debugger.Go(GoOption.WaitBreak)
debugger.Register.GetValue("pc")
```

付録 A　ウインドウ・リファレンス

ここでは，インストール，アップデート，ライセンスの設定，および CubeSuite+ の起動などに関連したウインドウ／パネル／ダイアログについて説明します。

A. 1　説　　明

以下に，インストール，アップデート，ライセンスの設定，および CubeSuite+ の起動などに関するウインドウ／パネル／ダイアログの一覧を示します。

表 A—1　ウインドウ／パネル／ダイアログ一覧

ウインドウ／パネル／ダイアログ名	機能概要
CubeSuite+ アップデート・マネージャ ウインドウ	アップデート機能の状態表示と各種操作を行います。
アップデートの確認中 ダイアログ	アップデートの確認中に表示されるダイアログです。
アップデート実行中 ダイアログ	アップデートのダウンロードとインストールの進捗を表示します。
アップデート・マネージャのオプション ダイアログ	アップデート・マネージャのオプションを表示，変更します。
タスクトレイ	バックグラウンドでアップデートの確認，ダウンロードが行われているときに，Windows の［タスクトレイ］内にアイコンを表示します。
CubeSuite+ ライセンス・マネージャ ウインドウ	ライセンスの一覧表示とライセンスの追加と削除を行います。
メイン・ウインドウ	CubeSuite+ を起動した際，最初にオープンするウインドウです。
スタート パネル	“チュートリアルを開く”，“プロジェクトへのアクセス（作成，開く）”，“サンプル・プロジェクトを読み込む”を簡単に行うことができます
プロジェクト作成 ダイアログ	プロジェクト，またはサブプロジェクトの新規作成を行います。
ソース・コンバート設定 ダイアログ【CX】	流用元のプロジェクトの構成ファイル（ソース・ファイル等）を，作成するプロジェクトのビルド・ツール用にコンバートする際の設定を行います。
プロジェクト変換設定 ダイアログ	旧開発環境（CubeSuite，Hew，PM+）のプロジェクト（またはワークスペース）を CubeSuite+ のプロジェクトに変換するための設定を行います。
プロジェクト・ツリー パネル	プロジェクトを構成するマイクロコントローラ，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。
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ノードの種類について，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。
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。

ウインドウ／パネル／ダイアログ名	機能概要
出力 パネル	各ツール（ビルド・ツール／デバッグ・ツール／プラグイン製品）から出力されるメッセージ，または検索・置換 ダイアログによる一括検索を行った際の結果を表示します。
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。
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。
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。
ファイル・エンコードの選択 ダイアログ	ファイル・エンコードを選択します。
指定行へのジャンプ ダイアログ	指定した行にキャレットを移動します。
検索・置換 ダイアログ	指定した文字列の検索，および置換を行います。
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。
Print Preview ウインドウ	印刷する前のソース・ファイルのプレビューを行います。
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。
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。
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／クリーンを行います。
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，その進捗状況を表示します。
パック設定 ダイアログ	プロジェクトと本製品をパックして保存する際の設定を行います。
オプション ダイアログ	CubeSuite+ の各種環境設定を行います。
プラグインの管理 ダイアログ	本製品の起動時に読み込むプラグインを設定します。
ユーザ設定 ダイアログ	メイン・ウインドウのツールバー，およびメニューバーをカスタマイズします。
新しいツール ダイアログ	メイン・ウインドウに表示する“新規ツールバー”を作成します。
ツールバーの名前の変更 ダイアログ	ユーザが作成したツールバーの名称を編集します。
キーボードのユーザ設定 ダイアログ	各コマンドに対して，ショートカット・キーを割り当てます。
コマンドの配置の変更 ダイアログ	メイン・ウインドウのメニュー項目／ボタン配置を変更（追加／削除を含む）します。
バージョン情報 ダイアログ	CubeSuite+，および各プラグイン製品のバージョン情報を表示します。
詳細バージョン情報 ダイアログ	本製品の詳細バージョンや現在のプロジェクトの情報を表示します。
ワン・ポイント・アドバイス ダイアログ	CubeSuite+ を使用する際のワン・ポイント・アドバイスを表示します。
その他の分割パネル ... ダイアログ	分割パネルとしてメイン・ウインドウに表示しているパネルの内，アクティブにするパネル，もしくは閉じるパネルを選択します。
プロジェクトを開く ダイアログ	既存のプロジェクトを開く，または新しいプロジェクトの作成時に流用元のプロジェクトを指定するために，プロジェクト・ファイルを選択します。

ウインドウ／パネル／ダイアログ名	機能概要
ファイルを開く ダイアログ	オープンするファイルの選択を行います。
既存のサブプロジェクトを追加 ダイアログ	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，サブプロジェクト・ファイルを選択します。
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，またはファイル（ソース・コード，レポート・ファイルなど）の出力先を設定します。
名前を付けてプロジェクトを保存 ダイアログ	プロジェクト・ファイルを別名で保存します。
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名前を付けてファイルに保存します。
プログラムを選択 ダイアログ	外部ツールの実行ファイルを選択します。
外部テキスト・エディタの選択 ダイアログ	外部テキスト・エディタの実行ファイルを選択します。
Python コンソール パネル	IronPython を利用して，コマンド入力方式で CubeSuite+，およびデバッグ・ツールを操作します。
統合アンインストーラ ウインドウ	インストールされている CubeSuite+ 製品のアンインストールを一括で指定します。

CubeSuite+ アップデート・マネージャ ウインドウ

アップデート機能の状態表示と各種操作を行います。

図 A—1　CubeSuite+ アップデート・マネージャ ウインドウ

ここでは，以下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］メニューから［すべてのプログラム］→［Renesas Electronics CubeSuite+］→［アップデート・マネージャ］を選択
- CubeSuite+ の［ヘルプ］メニューから［アップデートの確認 ...］を選択
- Windows の［タスクトレイ］内の，または をクリック

[各エリアの説明]

(1) メニュー・エリア

メイン・エリアの表示変更や，設定を変更するメニューが表示されます。項目を選択することによって項目に対応する操作を行うことができます。

すべて	アップデートの表示を行います。
マイクロコントローラ名	対応するマイクロコントローラのアップデートの表示を行います。 (過去にマイクロコントローラ指定機能により指定されたマイクロコントローラの名称を最近指定された順に 5 つまで表示されています。)
ダウンロードとインストール	選択されたアップデート対象のダウンロードを行い，ダウンロード完了後にインストールを行います。
ダウンロードのみ	選択されたアップデート対象のダウンロードのみを行います。
更新履歴の表示	アップデートの履歴を表示します。
非表示のアップデートの復元	非表示のアップデートの復元 ページを表示します。
設定の変更	アップデート・マネージャのオプション ダイアログを表示します。
ヘルプ	ヘルプを表示します。
終了	アップデート・マネージャを終了します。

(2) メイン・エリア

動作中の機能に応じて表示される内容が以下のいずれかのページに切り替わります。

- アップデートの選択 ページ
- ダウンロードとインストール ページ
- ダウンロードのみ ページ
- アップデート完了 ページ
- 更新履歴 ページ
- 非表示のアップデートの復元 ページ
- エラー ページ

アップデートの選択 ページ

このページでは，適用可能なアップデートが一覧表示され，アップデートのダウンロード，インストールを選択します。

図 A—2　アップデートの選択 ページ

ここでは，以下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］

- CubeSuite+ アップデート・マネージャ ウインドウのメニューエリアで［すべて］をクリック
- CubeSuite+ アップデート・マネージャ ウインドウのメニューエリアで［*マイクロコントローラ名*］をクリック

［各エリアの説明］

(1) アップデートの選択エリア

マイクロコントローラが指定されている場合は，マイクロコントローラ名が表示されます。

ダウンロードとインストール	選択されたアップデート対象のダウンロードを行い，ダウンロード完了後にインストールを行います。
ダウンロードのみ	選択されたアップデート対象のダウンロードのみを行います。

(2) アップデート対象選択エリア

アップデート対象が一覧表示されます。

［すべて選択］をクリックすると，すべてのアップデート対象がチェックされます。

それぞれのアップデート対象で，表示される情報は以下のとおりです。

カテゴリ	アップデートのカテゴリ名を表示します。 同じカテゴリをもつアップデートはまとめて表示します。
タイトル	アップデートのタイトルを表示します。
ダウンロードサイズ	アップデートのダウンロードサイズを表示します。
概要	アップデートの概要情報を表示します。
詳細	アップデートの詳細情報がある場合に表示し，選択するとブラウザ上に詳細情報を表示します。

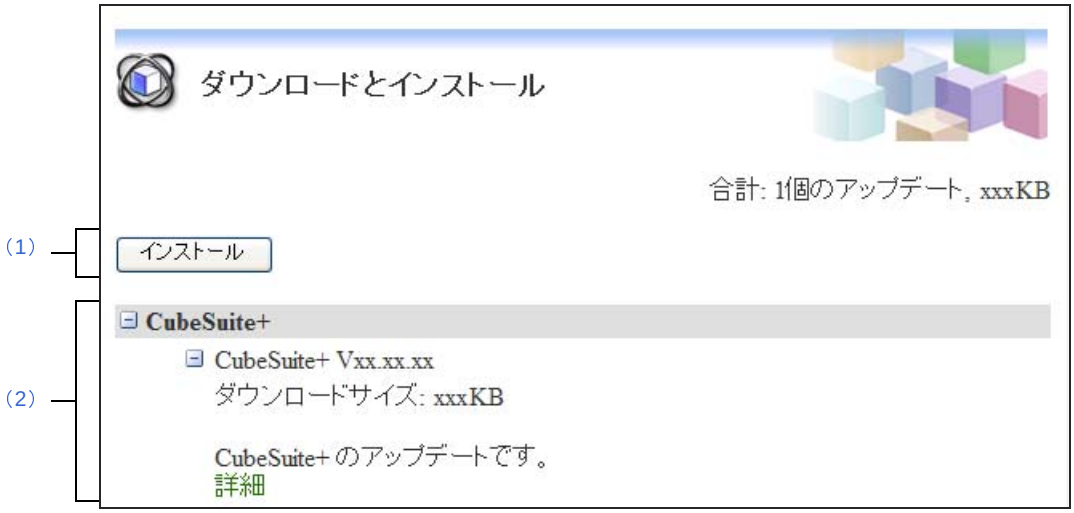
(3)［非表示にする］ボタン

アップデート対象選択エリアで，アップデートを選択している場合，選択されたアップデート全体が非表示となります。

ダウンロードとインストール ページ

このページでは，選択されたアップデートが一覧表示され，ダウンロードとインストールを開始します。

図 A—3　ダウンロードとインストール ページ



ここでは，以下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］

- CubeSuite+ アップデート・マネージャ ウインドウのメニューエリアで［ダウンロードとインストール］をクリック
- アップデートの選択 ページのメインエリアで［ダウンロードとインストール］をクリック

［各エリアの説明］

(1)［インストール］ボタン

ダウンロードとインストールを開始します。

アップデートがひとつも選択されていない場合は無効です。

注意　Windows Vista，Windows 7 上でエミュレータ用 USB ドライバー式のインストールを行った場合，警告のダイアログが表示される場合があります。

(2) 選択されているアップデート表示エリア

選択されているアップデートが一覧表示されます。

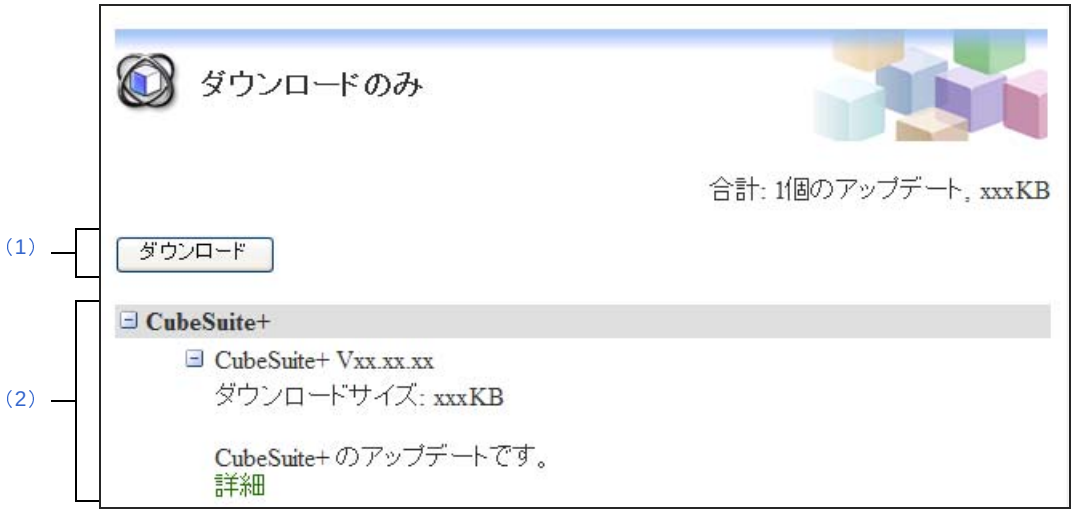
それぞれのアップデートで，表示される情報は以下のとおりです。

カテゴリ	アップデートのカテゴリ名を表示します。 同じカテゴリをもつアップデートはまとめて表示します。
タイトル	アップデートのタイトルを表示します。
ダウンロードサイズ	アップデートのダウンロードサイズを表示します。
概要	アップデートの概要情報を表示します。
詳細	アップデートの詳細情報がある場合に表示し，選択するとブラウザ上に詳細情報を表示します。

ダウンロードのみ ページ

このページでは，選択されたアップデートを一覧表示し，ダウンロードを開始します。

図 A—4　ダウンロードのみ ページ



ここでは，以下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］

- CubeSuite+ アップデート・マネージャ ウインドウのメニューエリアで［ダウンロードのみ］をクリック
- アップデートの選択 ページのメインエリアで［ダウンロードのみ］をクリック

［各エリアの説明］

(1)［ダウンロード］ボタン

ダウンロードを開始します。

アップデートがひとつも選択されていない場合は無効です。

(2) 選択されているアップデート表示エリア

選択されているアップデートが一覧表示されます。

それぞれのアップデートで，表示される情報は以下のとおりです。

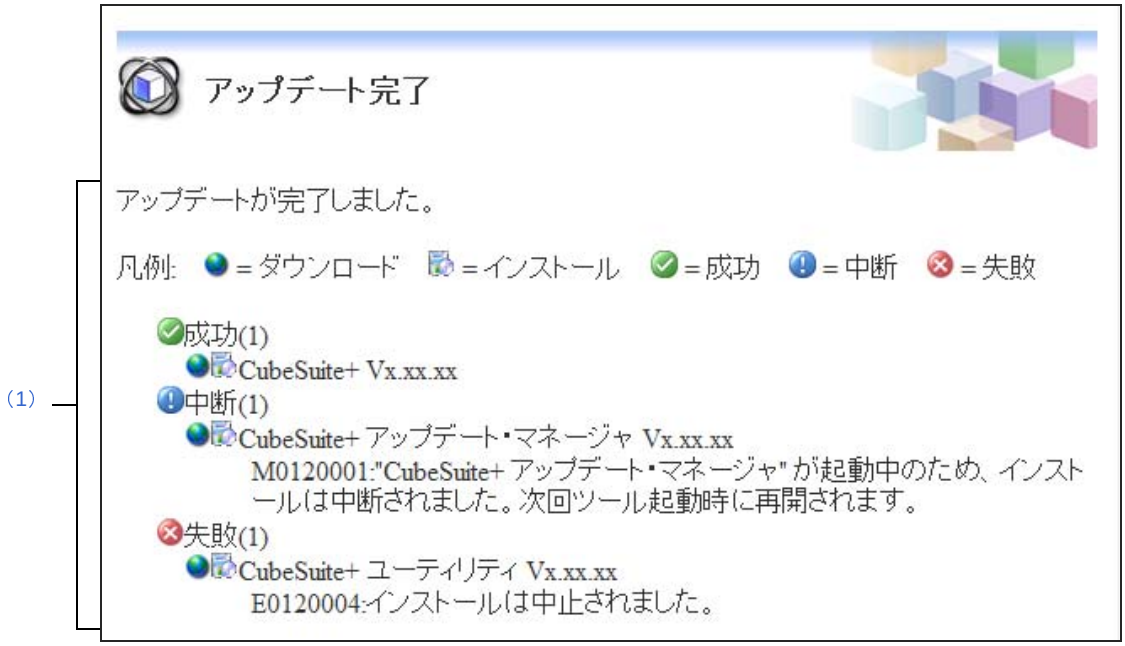
カテゴリ	アップデートのカテゴリ名を表示します。 同じカテゴリをもつアップデートはまとめて表示されます。
タイトル	アップデートのタイトルが表示されます。

ダウンロードサイズ	アップデートのダウンロードサイズが表示されます。
概要	アップデートの概要情報が表示されます。
詳細	アップデートの詳細情報がある場合に表示され，選択するとブラウザ上に詳細情報が表示されます。

アップデート完了 ページ

このページでは，アップデートの結果が成功，中断，失敗ごとに一覧表示されます。

図 A—5　アップデート完了 ページ



ここでは，以下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］

- アップデートが終了したときに自動的にオープン

［各エリアの説明］

(1) アップデートの結果表示エリア

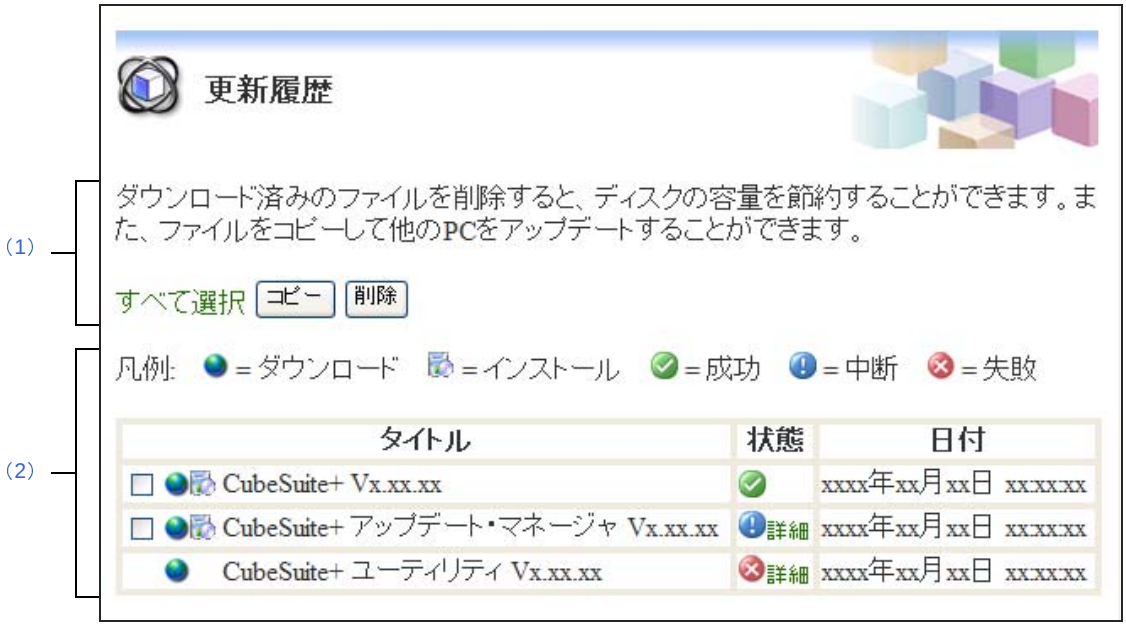
アップデートの結果を一覧表示します。

	成功したアップデートの数とアップデートのタイトルを表示します。
	中断したアップデートの数，アップデートのタイトルとメッセージを表示します。
	失敗したアップデートの数，アップデートのタイトルとメッセージを表示します。

更新履歴 ページ

このページでは，アップデートの実行履歴が表示され，アップデート用ファイルのコピーと削除を行います。

図 A—6　更新履歴 ページ



ここでは，以下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］

- CubeSuite+ アップデート・マネージャ ウインドウのメニューエリアで［更新履歴の表示］をクリック

［各エリアの説明］

(1) ボタンエリア

［すべて選択］をクリックすることで，一覧のすべてのチェック・ボックスをチェック状態にします。

ボタン	機能
コピー	チェックしたアップデートを指定したフォルダにコピーします。
削除	チェックしたアップデートを削除します。

（2）アップデートの実行履歴表示エリア

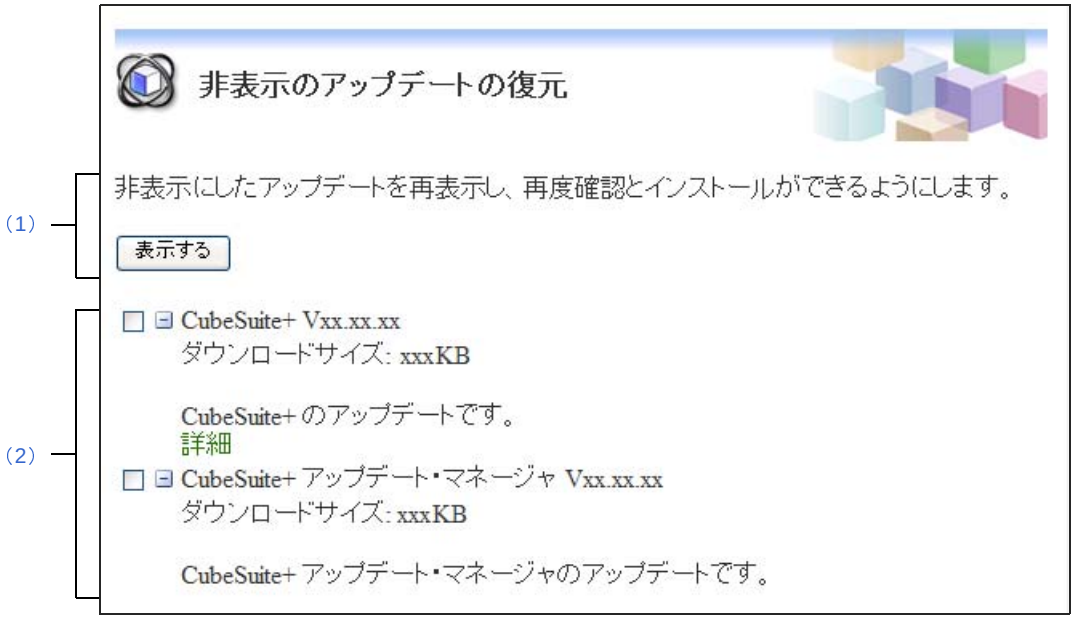
アップデートの実行履歴が表示されます。

一覧の状態欄の［詳細］をクリックすることで，メッセージ ダイアログが表示され，対象アップデートの結果に対応するメッセージが表示されます。

非表示のアップデートの復元 ページ

このページでは，アップデートの選択 ページで非表示にしたアップデートを一覧表示し，選択したアップデートの表示を復元し，再度確認とインストールができるようにします。

図 A—7　非表示のアップデートの復元 ページ



ここでは，以下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］

- CubeSuite+ アップデート・マネージャ ウインドウのメニューエリアで［非表示のアップデートの復元］をクリック

［各エリアの説明］

(1)［表示する］ボタン

チェックしたアップデートを表示状態にし，再度確認とインストールができるようにします。

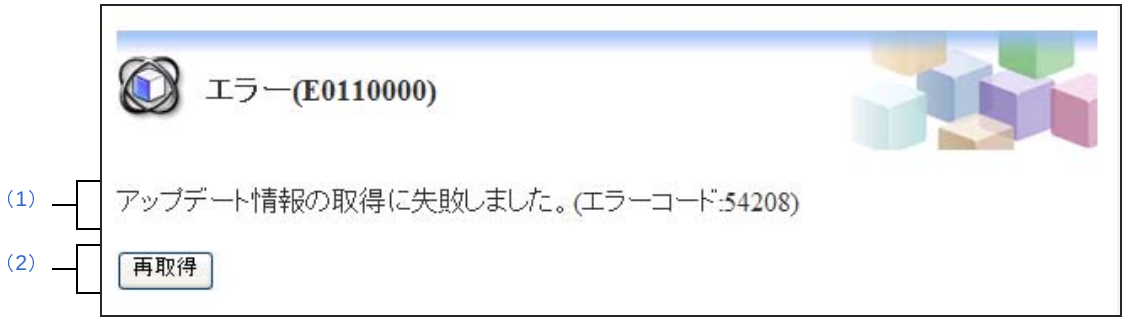
(2) 非表示のアップデート表示エリア

アップデートの選択 ページで非表示にしたアップデートが一覧表示されます。

エラー ページ

このページは，アップデート情報の取得が失敗したときに表示されます。

図 A—8　エラー ページ



ここでは，以下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］

- アップデート情報の取得に失敗したときに自動的にオープン

［各エリアの説明］

(1) メッセージエリア

アップデート情報の取得に失敗した旨のメッセージが表示されます。

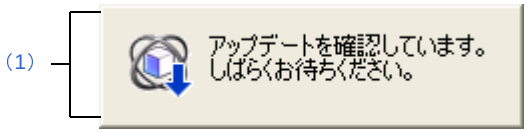
(2)［再取得］ボタン

アップデート情報の再取得を行います。

アップデートの確認中 ダイアログ

アップデートの確認中に表示されるダイアログです。

図 A—9　アップデートの確認中 ダイアログ



ここでは，以下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］

- アップデートの確認中に自動的にオープン

［各エリアの説明］

(1) アップデート確認中メッセージエリア

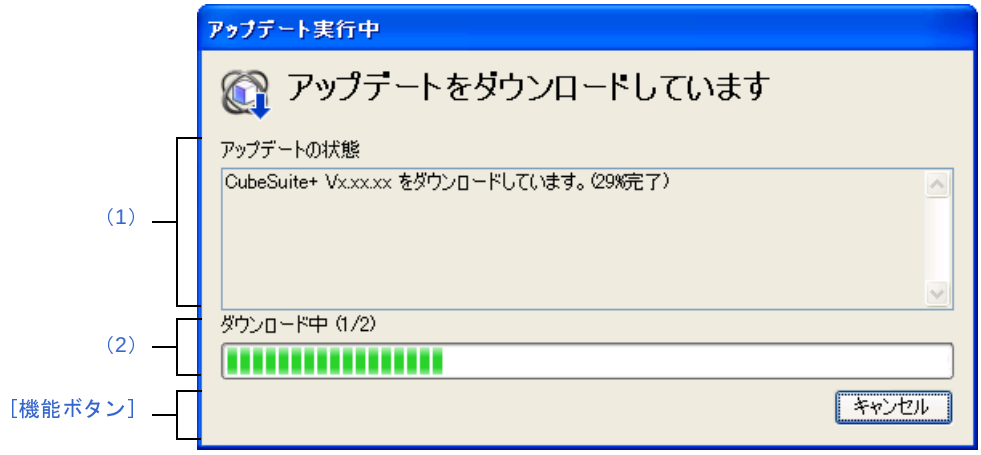
アップデートの確認中にメッセージが表示されます。

アップデートの確認が終わると，自動的に閉じます。

アップデート実行中 ダイアログ

アップデートのダウンロードとインストールの進捗を表示します。

図 A—10　アップデート実行中 ダイアログ



ここでは，以下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]

- アップデートのダウンロード中，インストール中に自動的にオープン

[各エリアの説明]

(1) [アップデートの状態] エリア

現在のアップデートおよびインストールの進捗状況を表示します。

アップデートおよびインストールが完了すると自動的にダイアログは閉じられます。

(2) [ダウンロード中／インストール中] エリア

ダウンロード中はダウンロードしているファイルのサイズに対するダウンロード済みサイズの割合を表示します。

インストール中はインストールを行うアップデートの数に対するインストール済みのアップデートの割合を表示します。

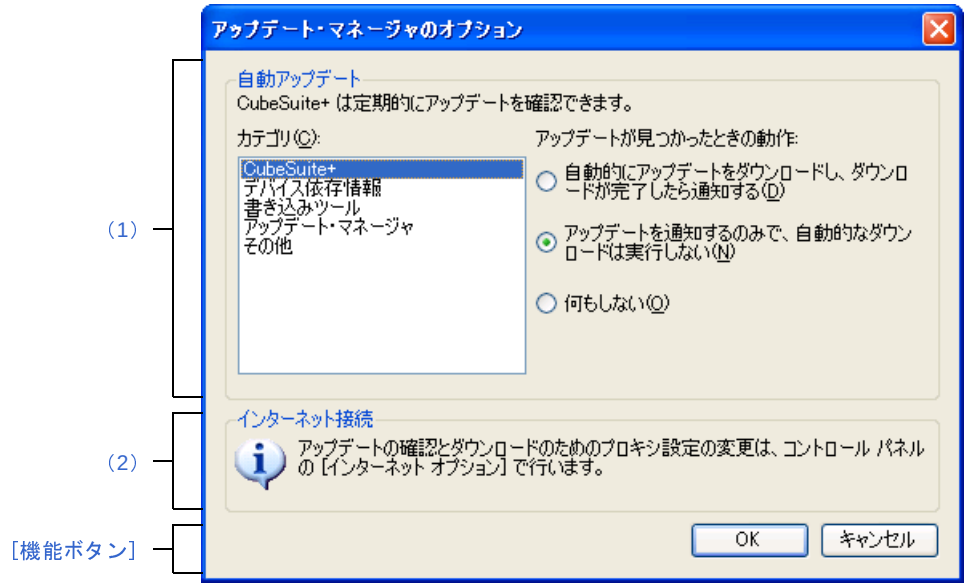
[機能ボタン]

ボタン	機能
キャンセル	アップデートを中止します。

アップデート・マネージャのオプション ダイアログ

アップデート・マネージャのオプションを表示，変更します。

図 A—11　アップデート・マネージャのオプション ダイアログ



ここでは，以下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］

- CubeSuite+ アップデート・マネージャ ウインドウのメニューエリアで［設定の変更］をクリック

［各エリアの説明］

(1)［自動アップデート］エリア

(a)［カテゴリ］

アップデートが見つかったときの動作を設定するカテゴリを選択します。

(b)［アップデートが見つかったときの動作］

アップデートが見つかったときの動作を設定します。

選択したカテゴリ種別に対する動作を選択します。

- ［自動的にアップデートをダウンロードし，ダウンロードが完了したら通知する］
  定期的なアップデートの確認のときに該当するアップデートが存在した場合，確認することなく自動的にダウンロードを行い，ダウンロードが完了してから通知を行います。
- ［アップデートを通知するのみで，自動的なダウンロードを実行しない］
  定期的なアップデートの確認のときに該当するアップデートが存在した場合，通知を行います。
- ［何もしない］
  定期的なアップデートの確認のときに該当するアップデートが存在しても通知を行いません。

(2)［インターネット接続］エリア

インターネット接続に関する説明です。

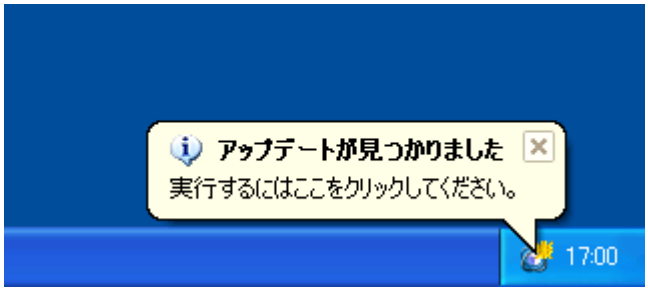
［機能ボタン］

ボタン	機能
OK	設定項目を確定し，反映してダイアログを閉じます。
キャンセル	設定項目の変更を取り消してダイアログを閉じます。

タスクトレイ

バックグラウンドでアップデートの確認，ダウンロードが行われているときに，Windows の［タスクトレイ］内にアイコンを表示します。

図 A—12　タスク トレイ表示例



ここでは，以下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］

- バックグラウンドでアップデートの確認，ダウンロードが行われているときに自動的に表示

［各エリアの説明］

(1) タスクトレイ

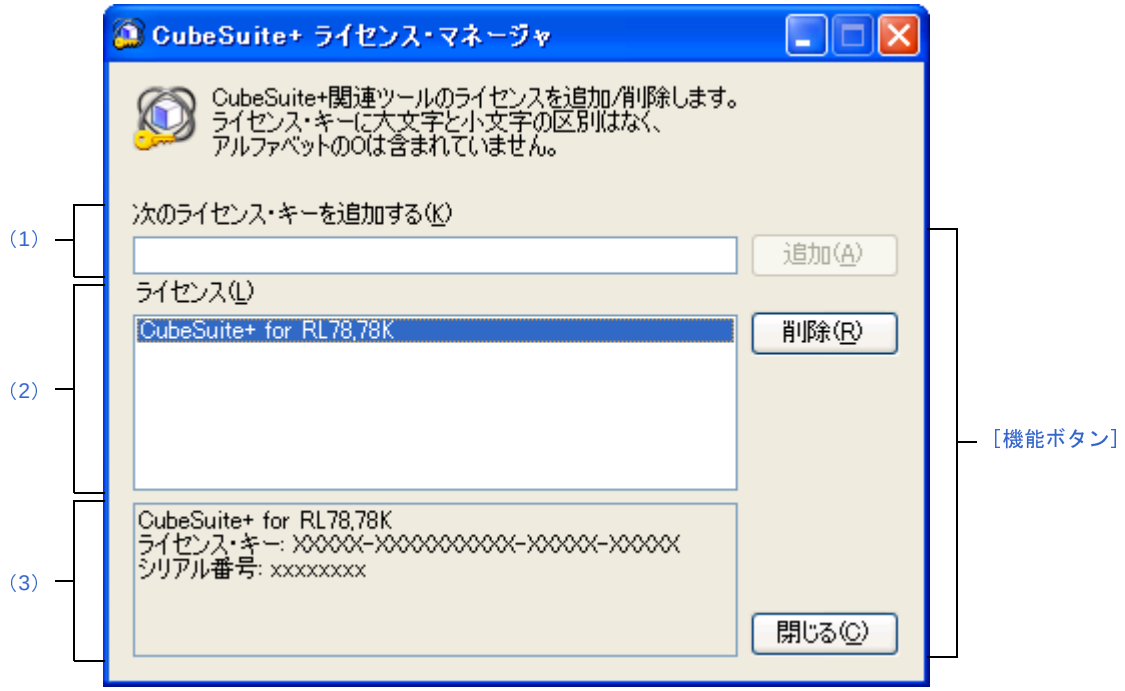
バックグラウンドでアップデートの確認，ダウンロードが行われているときに以下のアイコンを表示します。

	アップデートの確認中アイコンです。
	ダウンロード中アイコンです。進捗率が%で表示されます。
	アップデート通知アイコンです。とともに，通知内容を表示します。 クリックすると，CubeSuite+ アップデート・マネージャ ウインドウを表示します。

CubeSuite+ ライセンス・マネージャ ウインドウ

ライセンスの一覧表示とライセンスの追加と削除を行います。

図 A—13　CubeSuite+ ライセンス・マネージャ ウインドウ



ここでは，以下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］

- Windows の［スタート］メニューから ［すべてのプログラム］→ ［Renesas Electronics CubeSuite+］→ ［CubeSuite+ ライセンス・マネージャ］ を選択
- インストール時に，ライセンス登録 ウインドウの［ライセンス・マネージャ起動］ボタンをクリック

［各エリアの説明］

(1)［次のライセンス・キーを追加する］エリア

ライセンス・キーの入力を行うエリアです。

(2)［ライセンス］エリア

有効な登録済みライセンスのリストを表示します。

(3) ライセンス情報エリア

ライセンス エリアのリストを選択すると，選択したライセンスの詳細情報を表示します。

[機能ボタン]

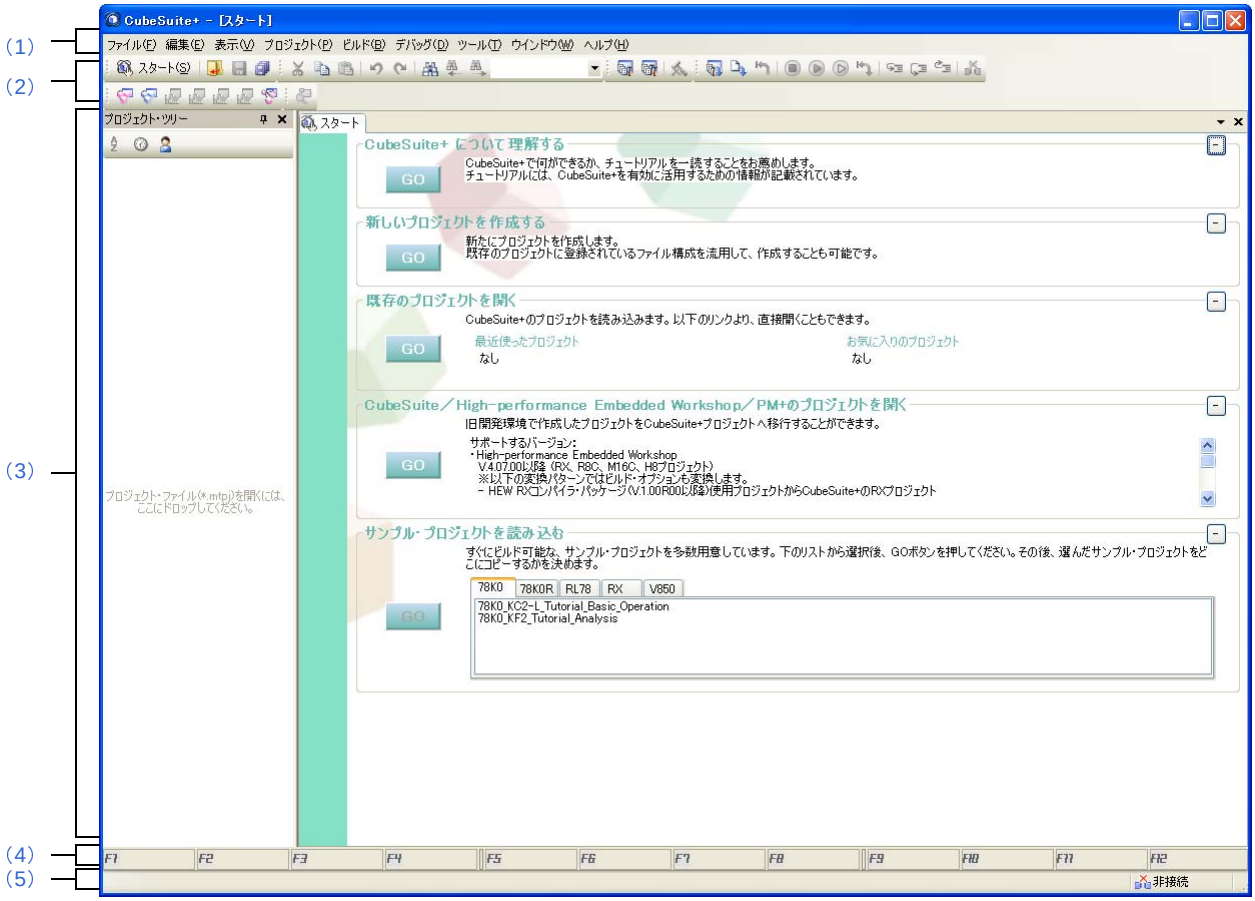
ボタン	機能
追加	入力したライセンスの追加を行います。
削除	選択したライセンスの削除を行います。
閉じる	ライセンス・マネージャを終了します。

メイン・ウインドウ

CubeSuite+ を起動した際，最初にオープンするウインドウです。

本ウインドウからプログラムの実行制御や各パネルのオープンなどを行います。

図 A—14　メイン・ウインドウ



ここでは，以下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

[各エリアの説明]

(1) メニューバー

共通のメニューを示します。

なお，各メニューから引き出される項目は，ユーザ設定 ダイアログでカスタマイズすることができます。

(a) [ファイル]

[ファイル] メニューでは，ファイルに関するメニュー項目を表示します。

新規作成	新規作成のためのカスケード・メニューを表示します。
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。
新しいファイルを作成	新しい空のファイルを作成し，編集できるようにします。 新規作成し，保存されていないエディタ パネルのデフォルトのタイトルは，“Source*X*”です（*X* は，新規ソース・ファイルの作成により 1 ずつ増加する固有の番号です）。
ファイルを開く ...	ファイル，およびプロジェクトを開くためのファイルを開く ダイアログを開きます。
エンコードを指定して開く ...	ファイルを開く ダイアログをオープンし，特定のエンコードで開くファイルを選択してオープンしたのち，ファイル・エンコードの選択 ダイアログでエンコードを選択します。
追加	追加のためのカスケード・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されます。
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。
ファイルを閉じる	現在フォーカスのあるエディタ パネルを閉じます。 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。
*対象*を保存	現在フォーカスのあるファイルの内容を保存します。
*対象*の保存設定 ...	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するために，ファイルの保存設定 ダイアログをオープンします。
名前を付けて*対象*を保存 ...	現在フォーカスのあるファイルを別名のファイルに保存するために，名前を付けて保存 ダイアログをオープンします。
すべてを保存	ファイルを更新中のエディタ パネルおよびプロジェクトをすべて保存します。
ページ設定 ...	Windows で用意されている印刷用の，ページ設定 ダイアログを開きます。
印刷 ...	アクティブなエディタ パネルの内容を印刷するために，Windows で用意されている，印刷 ダイアログを開きます。
印刷プレビュー	Print Preview ウインドウをオープンし，印刷するソース・ファイルのプレビューを行います。
最近使ったファイル	最近使ったファイルの一覧を表示し，ファイルを開くためのカスケード・メニューを表示します。
1 *パス*	もっとも最近使ったファイルを開くメニューです。
2 *パス*	2 番目に最近使ったファイルを開くメニューです。
3 *パス*	3 番目に最近使ったファイルを開くメニューです。
4 *パス*	4 番目に最近使ったファイルを開くメニューです。
最近使ったプロジェクト	最近使ったプロジェクトの一覧を表示し，プロジェクトを開くためのカスケード・メニューを表示します。
1 *パス*	もっとも最近使ったプロジェクトを開くメニューです。
2 *パス*	2 番目に最近使ったプロジェクトを開くメニューです。
3 *パス*	3 番目に最近使ったプロジェクトを開くメニューです。
4 *パス*	4 番目に最近使ったプロジェクトを開くメニューです。
終了	本アプリケーションを終了します。未保存のソース・ファイル，メイン・プロジェクト，またはサブプロジェクト，ファイルがある場合は保存するかどうかを確認するためにメッセージ ダイアログを開きます。

(b)［編集］

［編集］メニューでは，編集に関するメニュー項目を表示します。

元に戻す	直前に行った操作を取り消します。
やり直し	元に戻した操作をやり直します。
切り取り	選択範囲を切り取ってクリップボードに貼り付けます。
コピー	選択範囲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貼り付けます。
削除	選択範囲を削除します。
すべて選択	すべての項目を選択します。
名前の変更	選択された項目の名前を変更します。
検索 ...	検索・置換 ダイアログ，またはトレース パネルにフォーカスがある場合はトレース検索 ダイアログを，メモリ パネルにフォーカスがある場合は，メモリ検索 ダイアログを開き，指定された項目を検索します。
置換 ...	検索・置換 ダイアログを開き，指定された文字列を，他の文字列に置換します。
移動 ...	エディタ パネルにフォーカスがある場合は，指定行へのジャンプ ダイアログをオープンし，指定された行に移動します。 逆アセンブルパネル／ SFR パネル【RL78】【78K0R】【78K0】／ IOR パネル【RX】【V850】にフォーカスがある場合は，指定位置へ移動 ダイアログをオープンし，指定された位置に移動します。
コードのアウトライン	ソース・ファイルのコードの展開／折りたたみ表示を行うためのカスケード・メニューを表示します。
定義を折りたたむ	関数定義など，実装ブロックとして登録されているすべてのノードを折りたたみます。
アウトラインを切り替える	折りたたまれた部分で，カーソルが置かれている最も内側のアウトライン部分の現在の状態を切り替えます。
すべてのアウトラインを切り替える	すべてのノードの状態を切り替え，すべて同じ状態（展開または折りたたみ）に設定します。折りたたまれているノードと展開されたノードが混在している場合，すべてを展開します。
アウトラインを中止する	コードのアウトラインを中止します。現在の ソース・ファイルからすべてのアウトライン情報を削除します。
自動アウトラインを開始する	コードの自動アウトラインを開始します。サポートしているソース・ファイルのアウトライン情報を自動的に表示します。
高度な設定	エディタ パネルに関する高度な操作を行うためのカスケード・メニューを表示します。

行インデントを増やす	現在カーソルのある行のインデントをタブ１個分増やします。
行インデントを減らす	現在カーソルのある行のインデントをタブ１個分減らします。
行コメントを削除する	現在カーソルのある行の先頭から，言語（C++ など）に応じた行コメントの区切り記号の最初のセットを削除します。現在のソース・ファイルが行コメントの区切り記号が指定されている言語（C++ など）を使用している場合のみ使用できます。
行コメントを付ける	現在カーソルのある行の先頭に，言語（C++ など）に応じた行コメントの区切り記号を設定します。現在のソース・ファイルが行コメントの区切り記号が指定されている言語（C++ など）を使用している場合のみ使用できます。
タブをスペースに変換する	現在カーソルのある行のすべてのタブをスペースに変換します。
スペースをタブに変換する	現在カーソルのある行の連続したスペースの一組をタブに変換します。ただし，そのスペースの各組がタブ１個以上の幅に等しい場合に限ります。
選択行をタブ化する	現在の行をタブ化します。行の先頭にある（テキストの前の）すべてのスペースを可能な限りタブに変換します。
選択行を非タブ化する	現在の行を非タブ化します。行の先頭にある（テキストの前の）すべてのタブをスペースに変換します。
大文字にする	選択しているすべての文字を大文字に変換します。
小文字にする	選択しているすべての文字を小文字に変換します。
大文字／小文字を切り替える	選択しているすべての文字を，大文字または小文字に切り替えます。
先頭を大文字にする	選択しているすべての単語の先頭文字を大文字に変換します。
前後の空白を削除する	カーソル位置の前後にある余分な空白を削除し，空白文字を１個だけ残します。カーソルが単語内にある場合，または前後に空白文字がない場合，何も行いません。
末尾の空白を削除する	カーソルのある行で，最後の非空白文字の後にある空白を削除します。
行を削除する	現在カーソルのある行を完全に削除します。
行をコピーする	現在カーソルのある行をコピーして，その直後に挿入します。
空白行を削除する	カーソルのある行が空である場合，または空白文字しかない場合，その行を削除します。

(c)［表示］

［表示］メニューでは，パネルなどの表示に関するメニュー項目を表示します。

プロジェクト・ツリー	プロジェクト・ツリー パネルを表示し，フォーカスをプロジェクト・ツリー パネルに移動します。
プロパティ	プロパティ パネルを表示し，フォーカスをプロパティ パネルに移動します。このとき，プロジェクト・ツリー パネルで選択されている項目のプロパティを表示します。
出力	出力 パネルを表示し，フォーカスを出力 パネルに移動します。
デバッグ・マネージャ	デバッグ・マネージャ パネルを表示します。
ウォッチ	ウォッチ パネルを開くためのカスケード・メニューを表示します。
ウォッチ 1	ウォッチ 1 パネルを表示します。
ウォッチ 2	ウォッチ 2 パネルを表示します。
ウォッチ 3	ウォッチ 3 パネルを表示します。
ウォッチ 4	ウォッチ 4 パネルを表示します。
ローカル変数	ローカル変数パネルを表示します。
コール・スタック	コール・スタック パネルを表示します。
メモリ	メモリ パネルを開くためのカスケード・メニューを表示します。
メモリ 1	メモリ 1 パネルを表示します。
メモリ 2	メモリ 2 パネルを表示します。
メモリ 3	メモリ 3 パネルを表示します。
メモリ 4	メモリ 4 パネルを表示します。
SFR【RL78】【78K0R】【78K0】／IOR【RX】【V850】	SFR パネル【RL78】【78K0R】【78K0】／IOR パネル【RX】【V850】を表示します。
CPU レジスタ	CPU レジスタ パネルを表示します。
トレース	トレース パネルを表示します。【IECUBE】【IECUBE2】【シミュレータ】
逆アセンブル	逆アセンブル パネルを開くためのカスケード・メニューを表示します。
逆アセンブル 1	逆アセンブル 1 パネルを表示します。
逆アセンブル 2	逆アセンブル 2 パネルを表示します。
逆アセンブル 3	逆アセンブル 3 パネルを表示します。
逆アセンブル 4	逆アセンブル 4 パネルを表示します。
イベント	イベント パネルを表示します。
現在の PC 位置を開く	現在の PC 位置をエディタ パネルで表示します。 PC 位置にソース情報，またはソース・ファイルが存在しない場合は，逆アセンブル パネルで表示します。 開く対象のエディタ パネルおよび逆アセンブル パネルがすでに表示されている場合は，フォーカスをそのパネルに移動します。
ジャンプ先の位置へ進む	元のジャンプ先へ進みます。
ジャンプ前の位置へ戻る	定義箇所へジャンプする前の位置へ戻ります。
タグ・ジャンプ	キャレット行のメッセージに対応するエディタ（ファイル，行，桁）へジャンプします。

ドッキング・レイアウトの保存と復帰	ドッキング・レイアウトの保存と復帰に関するカスケード・メニューを表示します。
レイアウト保存 1	ドッキング・レイアウトを 1 つ目の記憶場所に保存します。保存したレイアウトは［レイアウト復帰 1］により復元します。
レイアウト保存 2	ドッキング・レイアウトを 2 つ目の記憶場所に保存します。保存したレイアウトは［レイアウト復帰 2］により復元します。
レイアウト保存 3	ドッキング・レイアウトを 3 つ目の記憶場所に保存します。保存したレイアウトは［レイアウト復帰 3］により復元します。
レイアウト保存 4	ドッキング・レイアウトを 4 つ目の記憶場所に保存します。保存したレイアウトは［レイアウト復帰 4］により復元します。
レイアウト復帰 1	保存したドッキング・レイアウトを復元します。復元するレイアウトは［レイアウト保存 1］によって保存したレイアウトです。 一度も保存していない場合は，初回起動時のレイアウトを復元します。
レイアウト復帰 2	保存したドッキング・レイアウトを復元します。復元するレイアウトは［レイアウト保存 2］によって保存したレイアウトです。 一度も保存していない場合は，初回起動時のレイアウトを復元します。
レイアウト復帰 3	保存したドッキング・レイアウトを復元します。復元するレイアウトは［レイアウト保存 3］によって保存したレイアウトです。 一度も保存していない場合は，初回起動時のレイアウトを復元します。
レイアウト復帰 4	保存したドッキング・レイアウトを復元します。復元するレイアウトは［レイアウト保存 4］によって保存したレイアウトです。 一度も保存していない場合は，初回起動時のレイアウトを復元します。
レイアウトをリセット	メイン・ウインドウのドッキング・パネルに関するレイアウトを既定の状態に戻します。
Python コンソール	Python コンソール パネルを表示します。

備考　レイアウトの保存／復帰は，以下の 3 つの状態ごとに行われます。

- プロジェクト読み込み前
- プロジェクト読み込み時，かつ，デバッグ・ツール接続前
- プロジェクト読み込み時，かつ，デバッグ・ツール接続時

(d)［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”が表示されます。
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”が表示されます。
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”が表示されます。
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”が表示されます。
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されます。
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，このメニューは，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，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・システム上からは削除されません。
プロジェクトと開発ツールをパックして保存 ...	プロジェクト一式と本製品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。

(e)［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。

なお，ここでは，対象となるプロジェクトの種類がデバッグ専用プロジェクト（「付録F　外部ビルド・ツールの使用」参照）の場合のメニュー項目のみ示します。

デバッグ専用プロジェクト以外を対象としたビルド関連のメニューについては，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください。

ビルド・プロジェクト	プロジェクトのビルド（「F. 5. 1　ビルドを実行する」参照）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（「F. 5. 2　リビルドを実行する」参照）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（「F. 5. 3　クリーンを実行する」参照）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。
ラピッド・ビルド	ラピッド・ビルド機能（「F. 5. 4　ラピッド・ビルドを実行する」参照）の有効（デフォルト）／無効を選択します（トグル）。
*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（「F. 5. 1　ビルドを実行する」参照）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（「F. 5. 2　リビルドを実行する」参照）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（「F. 5. 3　クリーンを実行する」参照）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。
ビルドを中止	実行中のビルド／リビルド／クリーン／バッチ・ビルドを中止します。
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（「F. 5. 5　バッチ・ビルドを実行する」参照）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。

(f)［デバッグ］

デバッグ関連のメニューについては，「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

(g)［ツール］

［ツール］メニューでは，ツールに関するメニュー項目を表示します。

オプション ...	オプション ダイアログを開きます。
プラグインの管理 ...	プラグインの管理 ダイアログを開きます。
ユーザ設定 ...	ユーザ設定 ダイアログを開きます。

(h)［ウインドウ］

［ウインドウ］メニューでは，ウインドウ操作に関するメニュー項目を表示します。

分割	アクティブなエディタ パネルを水平方向に分割します。 分割の対象は，アクティブなエディタ パネルのみで，他のパネルは分割されません。分割の上限は，2 分割までです。
分割の解除	エディタ パネルの 2 分割表示を解除します。
すべての分割パネルを閉じる	すべての分割パネルを閉じます。
1 *対象*	1 *対象*のパネルにフォーカスを移動します。
2 *対象*	2 *対象*のパネルにフォーカスを移動します。
3 *対象*	3 *対象*のパネルにフォーカスを移動します。
4 *対象*	4 *対象*のパネルにフォーカスを移動します。
5 *対象*	5 *対象*のパネルにフォーカスを移動します。
6 *対象*	6 *対象*のパネルにフォーカスを移動します。
7 *対象*	7 *対象*のパネルにフォーカスを移動します。
8 *対象*	8 *対象*のパネルにフォーカスを移動します。
9 *対象*	9 *対象*のパネルにフォーカスを移動します。
その他の分割パネル ..	分割パネルが 10 個以上開いている場合に，その他の分割パネル ... ダイアログを開きます。

(i)［ヘルプ］

［ヘルプ］メニューでは，製品情報などに関するメニュー項目を表示します。

ヘルプ	本製品のヘルプを開きます。
*対象*のヘルプを開く	フォーカスの位置しているウインドウ，パネル，ダイアログ，エラー表示等に応じたヘルプを開きます。
ワン・ポイント・アドバイス ...	ワン・ポイント・アドバイス ダイアログを開きます。
チュートリアル	本製品のヘルプを開きます。
ルネサス エレクトロニクス・マイコン Web を開く	ブラウザで弊社製のマイクロコントローラに関する Web ページを開きます。

詳細バージョン情報 ...	詳細バージョン情報 ダイアログを表示します。
アップデートの確認 ...	CubeSuite+ アップデート・マネージャ ウインドウを開き，アップデートの確認を開始します。
バージョン情報 ...	バージョン情報 ダイアログを表示します。

(2) ツールバー

共通のボタン群を示します。ビルド，デバッグ，解析関連のボタンについては，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」，「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」，「CubeSuite+ 統合開発環境 ユーザーズマニュアル 解析編」を参照してください。

なお，ツールバー上のボタンは，ユーザ設定 ダイアログでカスタマイズすることができます。また，同ダイアログにより，新規にツールバーを作成することもできます。

(a) スタートと保存

スタートと保存では，スタートボタンとファイル操作，編集に関するボタン群を表示します。

スタート(S)	スタート パネルを開きます。
	ファイル，およびプロジェクトを開くためのファイルを開く ダイアログを開きます。
	現在フォーカスのあるファイルの内容を保存します。
	ファイルを更新中のエディタ パネルおよびプロジェクトをすべて保存します。
	選択範囲を切り取ってクリップボードに貼り付けます。
	選択範囲をクリップボードにコピーします。
	クリップボードの内容を貼り付けます。
	直前に行った操作を取り消します。
	元に戻した操作をやり直します。
	検索・置換 ダイアログ，またはトレース パネルにフォーカスがある場合はトレース検索 ダイアログを，メモリ パネルにフォーカスがある場合は，メモリ検索 ダイアログを開き，指定された項目を検索します。
	検索対象となっているパネルに対して，指定されている条件で逆方向に検索します。
	検索対象となっているパネルに対して，指定されている条件で順方向に検索します。

(b) 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／クリーン／バッチ・ビルドを中止します。

備考　ビルド／リビルド／クリーン／バッチ・ビルドについての詳細は，以下を参照してください。

- デバッグ専用プロジェクト以外を対象とする場合：　「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」
- デバッグ専用プロジェクトを対象とする場合：　「F. 5　ビルドを実行する」

(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
- 出力 パネル

表示内容の詳細については，各パネルの項を参照してください。

(4) ファンクション・キーバー

現在アクティブなウインドウのファンクション・キーの役割と，メイン・ウインドウで使用することが可能なファンクション・キーの役割をボタンで表示します。

ファンクション・キーバーのボタン上にマウス・カーソルを置くと，メッセージが表示されます。

(5) ステータスバー

現在選択しているメニュー項目の簡易説明，およびデバッグを行う際に必要な情報などを表示します。

詳細は，「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

スタート パネル

チュートリアルのオープン，プロジェクトへのアクセス（作成／オープン），およびサンプル・プロジェクトの読み込みなどの操作をワン・クリックで行います。

図 A—15　スタート パネル

ここでは，以下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］

- ツールバーの スタート(S) ボタンをクリック
- メニューバーの［ウインドウ］→［スタート］を選択

［各エリアの説明］

(1)［CubeSuite+ について理解する］エリア

(a)［CubeSuite+ について理解する］

CubeSuite+ について理解を深めるため， GO ボタンのクリックでにチュートリアルをオープンします。

(2)［新しいプロジェクトを作成する］エリア

(a)［新しいプロジェクトを作成する］

CubeSuite+ 用プロジェクト・ファイルを新規に作成するため，GO ボタンのクリックでプロジェクト作成 ダイアログをオープンします。

(3)［既存のプロジェクトを開く］エリア

(a)［最近使ったプロジェクト］

最近読み込んだプロジェクト・ファイルの名前（プロジェクト名）を最大 4 個まで表示します。

(b)［お気に入りのプロジェクト］

お気に入りに登録されているプロジェクト・ファイルの名前（プロジェクト名）を最大 4 個まで表示します。

(c)［既存のプロジェクトを開く］

既存の CubeSuite+ 用プロジェクト・ファイル（拡張子：.mtpj）を開くため，GO ボタンのクリックでプロジェクトを開く ダイアログをオープンします。

(4)［CubeSuite ／ High-performance Embedded Workshop ／ PM+ のプロジェクトを開く］エリア

(a)［CubeSuite ／ High-performance Embedded Workshop ／ PM+ のプロジェクトを開く］

既存の CubeSuite 用プロジェクト・ファイル（拡張子：.cspj）／ HEW 用ワークスペース・ファイル（拡張子：.hws）／ HEW 用プロジェクト・ファイル（拡張子：.hwp）／ PM+ 用ワークスペース・ファイル（拡張子：.prw）／ PM+ 用プロジェクト・ファイル（拡張子：.prj）を開くため，GO ボタンのクリックでプロジェクトを開く ダイアログをオープンします。

(5)［サンプル・プロジェクトを読み込む］エリア

(a)［サンプル・プロジェクトを読み込む］

CubeSuite+ が提供しているサンプル・プロジェクト・ファイル（拡張子：.mtpj）を読み込みます。

［*マイクロコントローラ*］タブを選択し，サンプル・プロジェクトのリストからサンプルを選択して，GO ボタンをクリックすると選択しているサンプル・プロジェクトをコピーするためのファルダを指定するため，フォルダの参照 ダイアログをオープンします。指定されたフォルダにサンプルをコピーしてから，サンプル・プロジェクトを読み込みます。

(6) ボタン

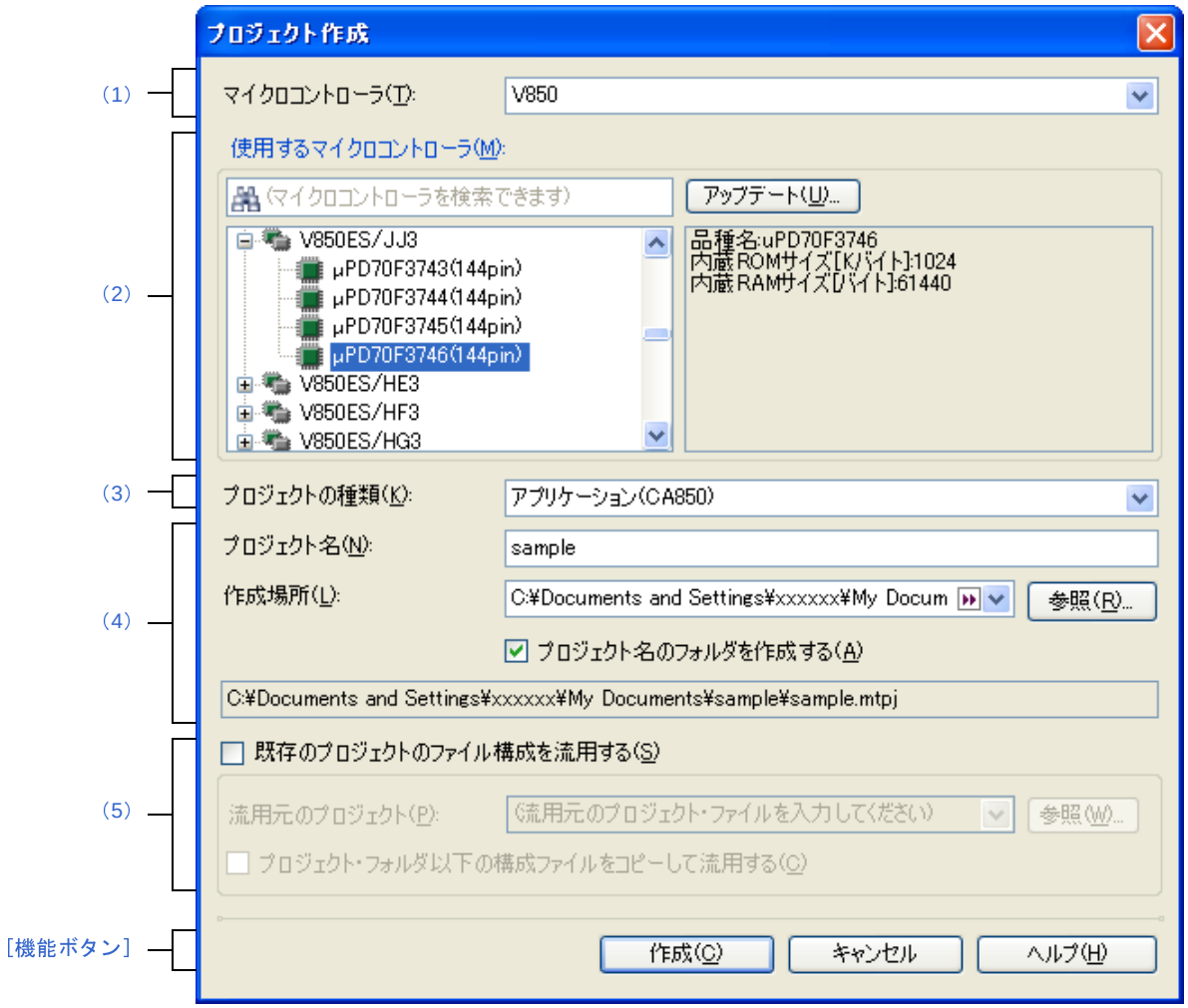
−	エリアを最小化し，タイトル文字列と，最小化ボタンのみ表示します。
+	最小化したエリアを元に戻します。

プロジェクト作成 ダイアログ

プロジェクト，またはサブプロジェクトの新規作成を行います。

注意　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。

図 A—16　プロジェクト作成 ダイアログ



備考　サブプロジェクトを作成する場合，タイトルバーの表示は“サブプロジェクト作成”となります。

ここでは，以下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]

(1) メイン・プロジェクトを作成する場合

- ツールバーの［スタート］ボタンをクリックしてスタート パネルをオープンしたのち，［新しいプロジェクトを作成する］エリアの［GO］ボタンをクリック
- ［ファイル］メニュー→［新規作成］→［新しいプロジェクトを作成 ...］を選択
- ［プロジェクト］メニュー→［新しいプロジェクトを作成 ...］を選択

(2) サブプロジェクトを作成する場合

- ［プロジェクト］メニュー→［追加］→［新しいサブプロジェクトを追加 ...］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，またはサブプロジェクトのコンテキスト・メニュー→［追加］→［新しいサブプロジェクトを追加 ...］を選択

[各エリアの説明]

(1)［マイクロコントローラ］

新規に作成するプロジェクトのマイクロコントローラの種類を選択します。

以下の項目を選択することができます。

- RX
- V850
- RL78
- 78K0R
- 78K0

(2)［使用するマイクロコントローラ］エリア

プロジェクトで使用するマイクロコントローラを選択します。

(a) マイクロコントローラ検索ボックス

マイクロコントローラ一覧で選択するマイクロコントローラを簡易検索することができます。

(b) マイクロコントローラ一覧

マイクロコントローラを選択します。

［マイクロコントローラ］で選択したマイクロコントローラの種類に対応するマイクロコントローラを愛称で分類し，一覧がツリー表示されます。

一覧からプロジェクトで使用するマイクロコントローラを選択すると，右側のボックスに以下の情報を表示します。

- 品種名
- 内蔵 ROM サイズ［K バイト］【RX】【V850】
- 内部 ROM サイズ［K バイト］【RL78】【78K0R】【78K0】
- 内蔵 RAM サイズ［バイト］【RX】【V850】

- 内部 RAM サイズ［バイト］【RL78】【78K0R】【78K0】
- 追加情報注

注　上記以外に情報がある場合のみ表示します。

(c) ボタン

アップデート ...	CubeSuite+ アップデート・マネージャ ウインドウを表示します。 使用するマイクロコントローラが一覧に表示されない場合，ネットワークを介してマイクロコントローラを検索することができます。

注意　本ボタンは，インストーラを使用して本製品をインストールした場合のみ有効となります。パックしたものを使用する場合，本ボタンは無効となります。

(3)［プロジェクトの種類］

作成するプロジェクトの種類を選択します。

以下の項目を選択することができます。

- アプリケーション（CC-RX/CA850/CX/CA78K0R/CA78K0）

  CubeSuite+ が提供するビルド・ツールを使用して，C ソース・ファイルから ROM 化用モジュール・ファイル【CA850】【CA78K0R】，ロード・モジュール・ファイル，またはヘキサ・ファイルを生成する場合に選択します。

  ここで生成するファイルが自動的にデバッグ対象となります。

- C++ アプリケーション（CC-RX）

  CubeSuite+ が提供するビルド・ツール CC-RX を使用して，C++ ソース・ファイル（main 関数のファイルのみ），および C ソース・ファイルからロード・モジュール・ファイル，またはヘキサ・ファイルを生成する場合に選択します。

  ここで生成するファイルが自動的にデバッグ対象となります。

- 空のアプリケーション（CC-RX）

  CubeSuite+ が提供するビルド・ツール CC-RX を使用して，ロード・モジュール・ファイルを生成する場合に選択します。

  ここで生成するファイルが自動的にデバッグ対象となります。

  なお，プロジェクト作成時にサンプルのスタートアップ・プログラムは生成しません。

- ライブラリ（CC-RX/CA850/CX/CA78K0R/CA78K0）

  CubeSuite+ が提供するビルド・ツールを使用して，ユーザ・ライブラリ用のライブラリ・ファイルを生成する場合に選択します。

- デバッグ専用

  CubeSuite+ が提供するビルド・ツール以外のビルド・ツールが生成したロード・モジュール・ファイル，またはヘキサ・ファイルをデバッグ対象とする場合に選択します（デバッグ専用プロジェクト）。

  デバッグ専用プロジェクトの作成方法，および使用方法についての詳細は，「付録 F　外部ビルド・ツールの使用」を参照してください。

(4) 作成プロジェクト・エリア

作成するプロジェクトの設定を行います。

(a) [プロジェクト名]

作成するプロジェクトの名前を直接入力します。

ここで入力した*プロジェクト名*＋“.mtpj”がプロジェクト・ファイル名となります。

備考　サブプロジェクトの場合は，*プロジェクト名*＋“.mtsp”がサブプロジェクト・ファイル名となります。

(b) [作成場所]

プロジェクト・ファイルの作成場所を絶対パスで直接入力，または[参照 ...]ボタンから選択します。

デフォルトでは，前回指定した作成場所を表示します。はじめて作成する場合は，“C:¥Documents and Settings¥*ユーザ名*¥My Documents”を表示します。

(c) 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，[作成場所]に絶対パスを表示します。

(d) [プロジェクト名のフォルダを作成する]

[作成場所]で指定した場所の下にプロジェクト名のフォルダを作成するかどうかをチェック・ボックスにより選択します。

デフォルトでは，チェック・ボックスをチェックします。

(e) プロジェクト・ファイル・パス

[プロジェクト名]，および[作成場所]で指定したプロジェクト・ファイルのフル・パスを表示します。

備考 1.　[プロジェクト名]，[作成場所]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。

2.　[プロジェクト名]の入力内容が正しくない場合，以下のメッセージがツールチップ表示されます。

メッセージ	説明
プロジェクト名が不正です。ファイル名として使用できる文字を入力してください。	プロジェクト・ファイルとして使用できない文字が使用されています。
同一 ID を持つサブプロジェクトがすでに登録されています。	同一 ID を持つサブプロジェクトがすでに登録されています。
サブプロジェクト名が不正です。メイン・プロジェクトと同じ場所に、同名のサブプロジェクトは作成できません。	指定された作成場所に，同名（拡張子を除く）のメイン・プロジェクトが存在しています。

3. ［作成場所］の入力内容が正しくない場合，以下のメッセージがツールチップ表示されます。

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。
プロジェクトの作成場所が存在しません。	フォルダの作成がキャンセル，または失敗し，存在しないフォルダが指定されている状態になりました。
プロジェクトの作成場所が不正です。パス名として使用できる文字を入力してください。	不正なパスを含むファイル名が指定されました。ファイル名，およびフォルダ名に文字（¥ , /, :, *, ?, ", <, >, |）は使用できません。

4. 各エリアにプロジェクト名やパス名を表示しきれない場合は，⏭が表示されます。⏭にマウス・カーソルをあわせると，絶対パスをポップアップ表示します。

（5）流用元プロジェクト・エリア

既存のプロジェクト（メイン・プロジェクト，またはサブプロジェクト）のファイル構成を流用してプロジェクトを作成する場合の設定を行います。

（a）［既存のプロジェクトのファイル構成を流用する］

既存のプロジェクトのファイル構成を流用してプロジェクトを作成する場合，このチェック・ボックスを選択します。

デフォルトでは，チェックされません。

なお，ファイルのカテゴリも流用の対象となります。

（b）［流用元のプロジェクト］

既存のプロジェクトのファイル構成を流用してプロジェクトを作成する場合，流用元のプロジェクト・ファイル名を指定します。

絶対パスで直接入力，または［参照 ...］ボタンを選択し，プロジェクトを開く ダイアログにおいて選択します。

デフォルトでは，前回指定したプロジェクト・ファイル名を表示します。はじめて作成する場合は，“C:¥Documents and Settings¥*ユーザ名*¥My Documents”を表示します。

なお，この項目は，［既存のプロジェクトのファイル構成を流用する］をチェックしている場合のみ有効となります。

注意　CubeSuite，HEW，PM+ のプロジェクト・ファイルを指定することはできません。

備考 1. 指定可能な文字数は，259 文字までです。

2. 入力内容が正しくない場合，以下のメッセージがツールチップ表示されます。

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。

メッセージ	説明
流用元のプロジェクトが存在しません。	パスに存在しないフォルダが含まれています。 または，指定したプロジェクト・ファイルが見つかりません。
流用元のプロジェクト名が不正です。ファイル名として使用できる文字を入力してください。	不正なパスを含むファイル名が指定されました。 ファイル名，およびフォルダ名に文字（¥ , /, :, *, ?, ", <, >, |）は使用できません。
現在開いているプロジェクトは流用元に指定できません。	現在開いているプロジェクト（メイン・プロジェクト，サブプロジェクトを含む）が流用元に指定されました。
流用元プロジェクトとしてプロジェクト・ファイルが指定されていません。	プロジェクト・ファイル，サブプロジェクト・ファイル以外の存在するファイルが指定されました。

3. パス名を表示しきれない場合は，⏩が表示されます。⏩にマウス・カーソルをあわせると，絶対パスをポップアップ表示します。

（c）ボタン

参照 ...	プロジェクトを開く ダイアログをオープンします。 プロジェクト・ファイルを選択すると，［流用元のプロジェクト］に絶対パスを表示します。

（d）［プロジェクト・フォルダ以下の構成ファイルをコピーして流用する］

［流用元のプロジェクト］で指定しているプロジェクトのプロジェクト・フォルダ以下にある構成ファイルを，新規に作成するプロジェクト・フォルダにコピーする場合，このチェック・ボックスを選択します。

デフォルトでは，チェックされません。

なお，この項目は，［既存のプロジェクトのファイル構成を流用する］をチェックしている場合のみ有効となります。

備考 1. 流用元のプロジェクトで使用しているビルド・ツールのバージョンと，作成するプロジェクトのバージョンが異なる場合は，自動的に変換します（デバッグ専用プロジェクトを除く）。

2. 作成するプロジェクトのビルド・ツールが，流用元のプロジェクトのノードに対応していない場合，該当ノードは通常のカテゴリ・ノードとしてプロジェクト・ツリーで表示されます（スタートアップ・ノード／ダウンロード・ファイル・ノードなど）。

［機能ボタン］

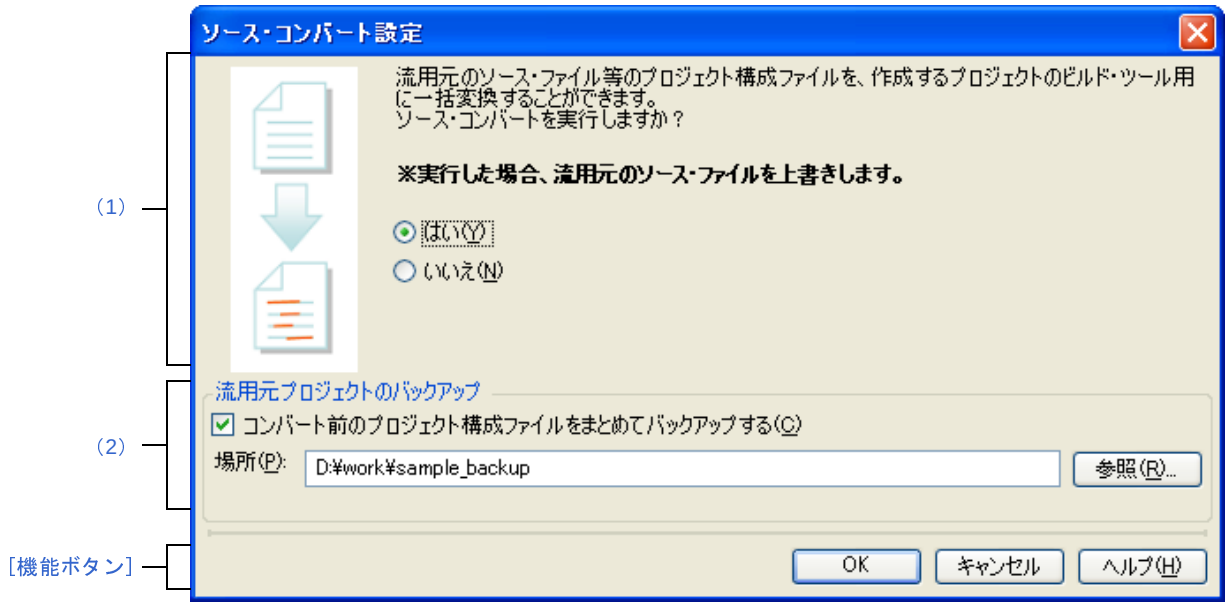
ボタン	機能
作成	設定した条件でプロジェクトを作成し，本ダイアログをクローズします。 ［既存のプロジェクトのファイル構成を流用する］をチェックしている場合は，［流用元のプロジェクト］で指定したプロジェクト（メイン・プロジェクト，またはサブプロジェクト）のファイル構成を流用してプロジェクトを作成します。 流用元のプロジェクトのビルド・ツールがCA850，作成するプロジェクトのビルド・ツールがCXの場合は，ソース・ファイル等のコンバートを実行するかどうかを選択するためのソース・コンバート設定 ダイアログ【CX】をオープンします。
キャンセル	設定した条件を破棄し，本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

ソース・コンバート設定 ダイアログ【CX】

流用元のプロジェクトの構成ファイル（ソース・ファイル等）を，作成するプロジェクトのビルド・ツール用にコンバートする際の設定を行います。

注意　本ダイアログは，流用元のプロジェクトのビルド・ツールが CA850，作成するプロジェクトのビルド・ツールが CX の場合のみ表示します。

図 A—17　ソース・コンバート設定 ダイアログ



ここでは，以下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］

- プロジェクト作成 ダイアログにおいて，流用元のプロジェクトのビルド・ツールに CA850，作成するプロジェクトのビルド・ツールに CX を設定し，［作成］ボタンをクリック

注意　プロジェクト作成 ダイアログにおいて，［プロジェクト・フォルダ以下の構成ファイルをコピーして流用する］をチェックしている場合は，本ダイアログをオープンすることはできません。

[各エリアの説明]

(1) ソース・コンバート実行選択エリア

プロジェクト作成時に，ソース・コンバートを実行するかどうかを選択します。

- ［はい］を選択した場合：　ソース・コンバートを実行します。
- ［いいえ］を選択した場合：　ソース・コンバートを実行しません。

(2)［流用元プロジェクトのバックアップ］エリア

流用元のプロジェクトのバックアップに関する設定を行います。

(a)［コンバート前のプロジェクト構成ファイルをまとめてバックアップする］

流用元のプロジェクトのソース・ファイル，およびプロジェクト一式をバックして保存する場合，このチェック・ボックスを選択します。

(b)［場所］

流用元のプロジェクトのソース・ファイル，およびプロジェクト一式の保存先を指定します。

絶対パスで直接入力，または［参照 ...］ボタンを選択し，フォルダの参照 ダイアログにおいて選択します。

デフォルトでは，“*作成中のプロジェクトのプロジェクト・フォルダ*_backup”を表示します。すでに同名のフォルダが存在する場合は，フォルダ名に数字（2，3，…，100）が付加されます。

なお，この項目は，［コンバート前のプロジェクト構成ファイルをまとめてバックアップする］をチェックしている場合のみ有効となります。

(c) 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・ボックスにパスを表示します。

備考 1. ［場所］で指定可能な文字数は，247 文字までです。入力内容が正しくない場合，以下のメッセージがツールチップ表示されます。

メッセージ	説明
フォルダ名が長すぎます。247 文字以内にしてください。	フォルダ名が 247 文字を越えています。
フォルダ名が不正です。文字 (:, *, ?, ", <, >, |) は使用できません。	不正なパスを含むフォルダ名が指定されました。ファイル名，およびフォルダ名に文字（:, *, ?, ", <, >, |）は使用できません。
流用元プロジェクト・フォルダ以下のフォルダは指定できません。	流用元プロジェクト・フォルダ以下のフォルダが指定されました。

2. パス名を表示しきれない場合は，⏩が表示されます。⏩にマウス・カーソルをあわせると，絶対パスをポップアップ表示します。

[機能ボタン]

ボタン	機能
OK	本ダイアログをクローズし，設定した条件でプロジェクトを作成します。
キャンセル	設定した条件を破棄し，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。

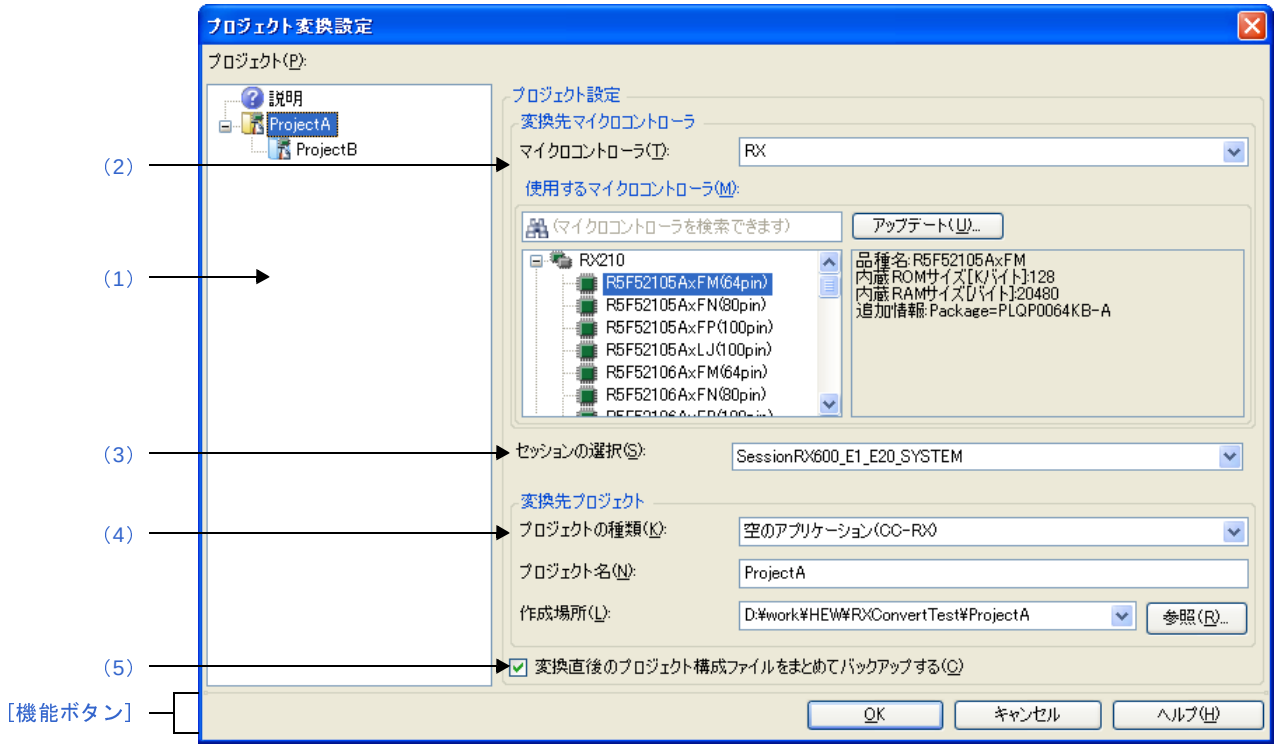
プロジェクト変換設定 ダイアログ

旧開発環境（CubeSuite，Hew，PM+）のプロジェクト（またはワークスペース）を CubeSuite+ のプロジェクトに変換するための設定を行います。

備考　旧開発環境のプロジェクトの変換方法についての詳細は，以下を参照してください。

- 「2. 7. 8　CubeSuite のプロジェクトを CubeSuite+ のプロジェクトに変換する」
- 「2. 7. 9　HEW のプロジェクトを CubeSuite+ のプロジェクトに変換する」
- 「2. 7. 10　PM+ のプロジェクトを CubeSuite+ のプロジェクトに変換する」

図 A—18　プロジェクト変換設定 ダイアログ



ここでは，以下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］メニュー→［プロジェクトを開く ...］を選択してプロジェクトを開く ダイアログをオープン
  → CubeSuite，Hew，PM+ のプロジェクト（またはワークスペース）を選択して［開く］ボタンをクリック
- スタート パネルの［CubeSuite ／ High-performance Embedded Workshop ／ PM+ のプロジェクトを開く］エリアにおいて，［GO］ボタンをクリックしてプロジェクトを開く ダイアログをオープン
  → CubeSuite，Hew，PM+ のプロジェクト（またはワークスペース）を選択して［開く］ボタンをクリック

[各エリアの説明]

(1)［プロジェクト］

変換先プロジェクトの構成をツリー表示し，変換設定を行うプロジェクトを選択します。

	説明ノード 選択すると，右側のエリアに本ダイアログの操作概要を表示します。
	メイン・プロジェクト・ノード 変換先プロジェクトのメイン・プロジェクトとなることを示します。 選択すると，右側のエリアにメイン・プロジェクトの設定項目を表示します。
	サブプロジェクト・ノード 変換先プロジェクトのサブプロジェクトとなることを示します。 選択すると，右側のエリアにサブプロジェクトの設定項目を表示します。

備考 1. ［Ctrl］キーを押下しながらメイン・プロジェクト・ノード，サブプロジェクト・ノードを選択すると，一度に複数プロジェクトの設定を行うことができます。

2. 以下を変換する場合，サブプロジェクト・ノードは表示されません。
- CubeSuite のプロジェクトで，サブプロジェクトが存在しない場合
- HEW のワークスペースで，プロジェクトが 1 つしか存在しない場合
- HEW のプロジェクトの場合
- PM+ のプロジェクトの場合

3. 設定内容が正しくない場合，以下のメッセージがツールチップ表示されます。

メッセージ	説明
プロジェクト・ファイルの絶対パスが長すぎるプロジェクトがあります。259 文字以内にしてください。	本エリアで選択していないプロジェクトの設定が，変換先プロジェクト・ファイルの絶対パスとして 259 文字を越えています。

(2)［変換先マイクロコントローラ］エリア

変換先プロジェクトのマイクロコントローラの設定を行います。

備考　PM+ のワークスペースを変換し，メイン・プロジェクト・ノードを選択した場合，本エリアは無効となります。

(a)［マイクロコントローラ］

変換先プロジェクトのマイクロコントローラの種類を選択します。

以下の項目を選択することができます。

- RX
- V850
- RL78
- 78K0R
- 78K0

(b)［使用するマイクロコントローラ］エリア

変換先プロジェクトで使用するマイクロコントローラを選択します。

- マイクロコントローラ検索ボックス

マイクロコントローラ一覧で選択するマイクロコントローラを簡易検索することができます。

- マイクロコントローラ一覧

マイクロコントローラを選択します。

［マイクロコントローラ］で選択したマイクロコントローラの種類に対応するマイクロコントローラを愛称で分類し，一覧がツリー表示されます。

一覧からプロジェクトで使用するマイクロコントローラを選択すると，右側のボックスに以下の情報を表示します。

- 品種名
- 内蔵 ROM サイズ［K バイト］【RX】【V850】
- 内部 ROM サイズ［K バイト］【RL78】【78K0R】【78K0】
- 内蔵 RAM サイズ［バイト］【RX】【V850】
- 内部 RAM サイズ［バイト］【RL78】【78K0R】【78K0】
- 追加情報注

注　上記以外に情報がある場合のみ表示します。

- ボタン

アップデート ...	CubeSuite+ アップデート・マネージャ ウインドウを表示します。 使用するマイクロコントローラが一覧に表示されない場合，ネットワークを介してマイクロコントローラを検索することができます。

注意　本ボタンは，インストーラを使用して本製品をインストールした場合のみ有効となります。
パックしたものを使用する場合，本ボタンは無効となります。

備考　選択内容が正しくない場合，以下のメッセージがツールチップ表示されます。

メッセージ	説明
デバイスを選択してください。	デバイスが選択されていないため（デバイス・カテゴリが選択されている等），プロジェクトの種類を決定することができません。

(3)［セッションの選択］

変換先プロジェクトで使用するセッションをドロップダウン・リストにより選択します。

ドロップダウン・リストには，HEW のプロジェクトに含まれるすべてのセッションのうち，ターゲットにリンクしたセッションのみを表示します。

備考　本項目は，HEW のプロジェクトを変換し，かつプロジェクトに複数のセッションが存在する場合のみ表示されます。

(4)［変換先プロジェクト］エリア

作成するプロジェクトの設定を行います。

(a)［プロジェクトの種類］

作成するプロジェクトの種類を選択します。

備考　PM+ のワークスペースを変換し，メイン・プロジェクト・ノードを選択した場合，本項目は無効となります。

以下の項目を選択することができます。

- アプリケーション（CC-RX/CA850/CX/CA78K0R/CA78K0）
  CubeSuite+ が提供するビルド・ツールを使用して，C ソース・ファイルから ROM 化用モジュール・ファイル【CA850】【CA78K0R】，ロード・モジュール・ファイル，またはヘキサ・ファイルを生成する場合に選択します。
  ここで生成するファイルが自動的にデバッグ対象となります。
- C++ アプリケーション（CC-RX）
  CubeSuite+ が提供するビルド・ツール CC-RX を使用して，C++ ソース・ファイル（main 関数のファイルのみ），および C ソース・ファイルからロード・モジュール・ファイル，またはヘキサ・ファイルを生成する場合に選択します。
  ここで生成するファイルが自動的にデバッグ対象となります。
- 空のアプリケーション（CC-RX）
  CubeSuite+ が提供するビルド・ツール CC-RX を使用して，ロード・モジュール・ファイルを生成する場合に選択します。
  ここで生成するファイルが自動的にデバッグ対象となります。
  なお，プロジェクト作成時にサンプルのスタートアップ・プログラムは生成しません。

- ライブラリ（CC-RX/CA850/CX/CA78K0R/CA78K0）

  CubeSuite+ が提供するビルド・ツールを使用して，ユーザ・ライブラリ用のライブラリ・ファイルを生成する場合に選択します。

- デバッグ専用

  CubeSuite+ が提供するビルド・ツール以外のビルド・ツールが生成したロード・モジュール・ファイル，またはヘキサ・ファイルをデバッグ対象とする場合に選択します（デバッグ専用プロジェクト）。

  デバッグ専用プロジェクトの作成方法，および使用方法についての詳細は，「付録Ｆ　外部ビルド・ツールの使用」を参照してください。

(b)［プロジェクト名］

作成するプロジェクトの名前を直接入力します。

ここで入力した*プロジェクト名*＋“.mtpj”がプロジェクト・ファイル名となります。

備考 1. サブプロジェクトの場合は，*プロジェクト名*＋“.mtsp”がサブプロジェクト・ファイル名となります。

2. プロジェクト・ファイルの作成場所にすでに同名のプロジェクト・ファイルが存在する場合は，ファイル名に“_*数字*_”（*数字*：1 ～ 99）が付加されます。

(c)［作成場所］

プロジェクト・ファイルの作成場所を絶対パスで直接入力，または［参照 ...］ボタンから選択します。

デフォルトでは，旧開発環境のプロジェクト・フォルダを表示します。

(d) 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，［作成場所］に絶対パスを表示します。

備考 1. ［プロジェクト名］，［作成場所］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。

2. ［プロジェクト名］の入力内容が正しくない場合，以下のメッセージがツールチップ表示されます。

メッセージ	説明
プロジェクト名が不正です。ファイル名として使用できる文字を入力してください。	プロジェクト・ファイルとして使用できない文字が使用されています。
サブプロジェクト名が不正です。メイン・プロジェクトと同じ場所に、同名のサブプロジェクトは作成できません。	指定された作成場所に，同名（拡張子を除く）のメイン・プロジェクトが存在しています。
プロジェクト名を指定してください。	プロジェクト名が指定されていません。

3. ［作成場所］の入力内容が正しくない場合，以下のメッセージがツールチップ表示されます。

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。
プロジェクトの作成場所が存在しません。	フォルダの作成がキャンセル，または失敗し，存在しないフォルダが指定されている状態になりました。
プロジェクトの作成場所が不正です。パス名として使用できる文字を入力してください。	不正なパスを含むファイル名が指定されました。 ファイル名，およびフォルダ名に文字（¥ , /, :, *, ?, ", <, >, |）は使用できません。
変換先プロジェクトの設定がまったく同一のプロジェクトがあります。プロジェクト名または作成場所を変更してください。	他のプロジェクトと同名の絶対パスのプロジェクト・ファイルを作成しようとしています。
作成場所を指定してください。	作成場所が指定されていません。

（5）［変換直後のプロジェクト構成ファイルをまとめてバックアップする］

変換直後のプロジェクトのソース・ファイル，およびプロジェクト一式をパックして保存する場合，このチェック・ボックスを選択します。

備考　PM+ のワークスペースを変換し，メイン・プロジェクト・ノードを選択した場合，本項目は無効となります。

保存先フォルダは，変換先プロジェクト・フォルダと同じ階層にある“*変換先プロジェクト・フォルダ*_org”となります。

すでに同名のフォルダが存在する場合は，フォルダ名に“_ *数字* _”（*数字*：1 ～ 99）が付加されます。

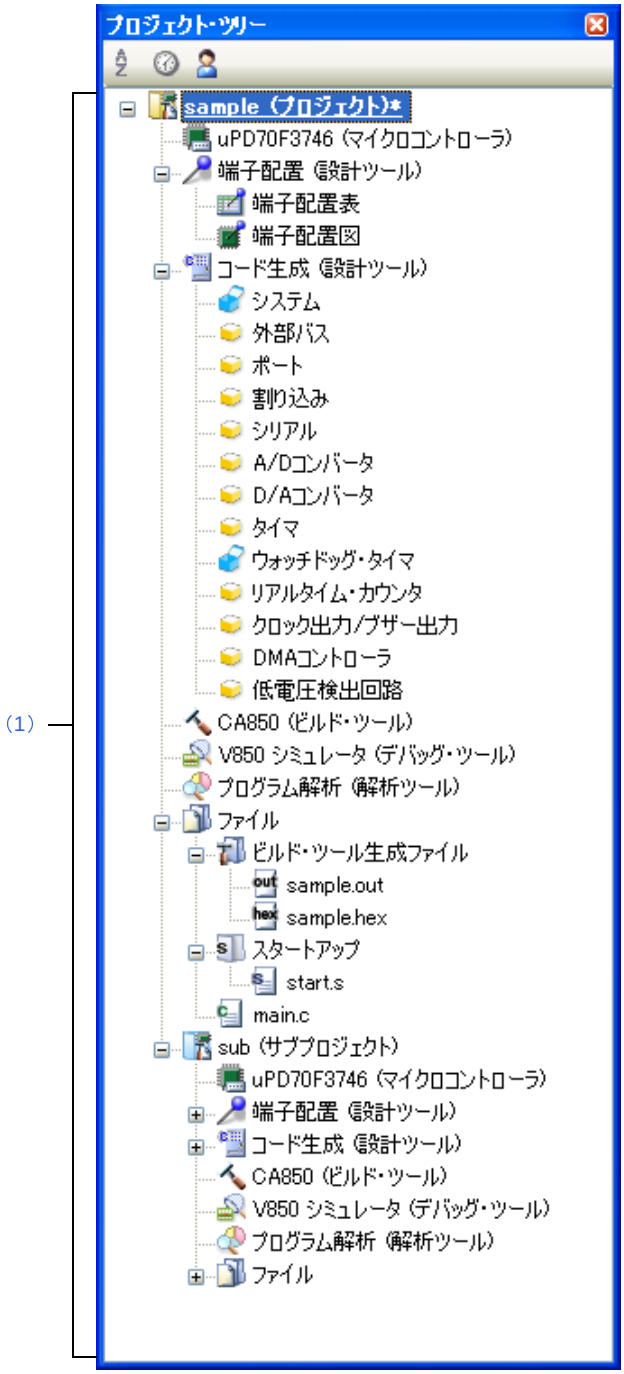
［機能ボタン］

ボタン	機能
OK	本ダイアログをクローズし，設定した条件でプロジェクトを変換します。
キャンセル	設定した条件を破棄し，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。

プロジェクト・ツリー パネル

プロジェクトを構成するマイクロコントローラ，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—19　プロジェクト・ツリー パネル



ここでは，以下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］
- ［ツールバー］
- ［［編集］メニュー（プロジェクト・ツリー パネル専用部分）］
- ［［ヘルプ］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。

なお，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）がプロパティ パネルに表示され，設定の変更を行うことができます。

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。
マイクロコントローラ名 (マイクロコントローラ) （以降，“マイクロコントローラ・ノード”と呼びます。）	プロジェクトで使用するマイクロコントローラです。
設計ツール名 (設計ツール) （以降，“設計ツール・ノード”と呼びます。）	使用する設計ツール（端子配置，コード生成等）です。 なお，コード生成 (設計ツール) ノードは，デバッグ専用プロジェクトを対象としている場合は表示されません。
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（コンパイラ，アセンブラ等）です。 なお，デバッグ専用プロジェクトを対象としている場合は，*ビルド・ツール名*は“なし”と表示されます。
デバッグ・ツール名 (デバッグ・ツール) （以降，“デバッグ・ツール・ノード”と呼びます。）	使用するデバッグ・ツール（エミュレータ，シミュレータ等）です。 新規プロジェクト作成時は，シミュレータが設定されます。
プログラム解析 (解析ツール) （以降，“解析ツール・ノード”と呼びます。）	使用する解析ツールです。 なお，このノードはデバッグ専用プロジェクトを対象としている場合は表示されません。
ファイル （以降，“ファイル・ノード”と呼びます。）	プロジェクトに登録しているファイルが，直下に表示されます。

ノード	説明
ダウンロード・ファイル （以降，“ダウンロード・ファイル・ノード”と呼びます。）	プロジェクトにダウンロード・ファイルを登録するためのノードです。 なお，このノードはデバッグ専用プロジェクトを対象としている場合のみ表示されます。
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。）	ビルド時に作成されるノードで，ビルド・ツールによって生成されたファイル（オブジェクト・ファイルを除く）が直下に表示されます。 なお，このノードはデバッグ専用プロジェクトを対象としている場合は表示されません。
スタートアップ【V850】【RL78】【78K0R】【78K0】 （以降，“スタートアップ・ノード”と呼びます。）	プロジェクトに標準以外のスタートアップ・ルーチンを登録するためのノードです。このノードは常にファイル・ノード以下に表示されます。 なお，このノードはデバッグ専用プロジェクトを対象としている場合は表示されません。
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。 なお，このノードは常にファイル・ノード以下に作成されます。
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。

備考 1.　使用するマイクロコントローラに対応しているツールのみ表示されます。

2.　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されます。
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。

3.　デバッグ専用プロジェクトについての詳細は，「付録Ｆ　外部ビルド・ツールの使用」を参照してください。

このエリアは，以下の機能を備えています。

(a) プロジェクト・ファイルの読み込み

エクスプローラなどからプロジェクト・ファイルをドラッグし，このエリアにドロップすると，メイン・ウインドウでそのプロジェクトを開くことができます。

(b) サブプロジェクトの追加

以下のいずれかの方法により，サブプロジェクトの追加を行うことができます。

- 既存のサブプロジェクトを追加する場合
  - プロジェクト・ノード，またはサブプロジェクト・ノードを選択し，［ファイル］メニュー，または［プロジェクト］メニュー→［追加］→［既存のサブプロジェクトを追加 ...］を選択する。既存のサブプロジェクトを追加 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，またはサブプロジェクト・ノードのコンテキスト・メニューの［追加］→［既存のサブプロジェクトを追加 ...］を選択する。既存のサブプロジェクトを追加 ダイアログがオープンし，追加するサブプロジェクト・ファイルを選択する。

- 新しいサブプロジェクトを追加する場合
  - プロジェクト・ノード，またはサブプロジェクト・ノードを選択し，［ファイル］メニュー，または［プロジェクト］メニュー→［追加］→［新しいサブプロジェクトを追加 ...］を選択する。プロジェクト作成 ダイアログがオープンし，新しく作成するサブプロジェクト・ファイルを作成する。
  - プロジェクト・ノード，またはサブプロジェクト・ノードのコンテキスト・メニューの［追加］→［新しいサブプロジェクトを追加 ...］を選択する。プロジェクト作成 ダイアログがオープンし，新しく作成するサブプロジェクト・ファイルを作成する。

(c) プロジェクトからサブプロジェクトを外す

以下のいずれかの方法により，サブプロジェクトをプロジェクトから外すことができます。

- 削除するサブプロジェクトを選択し，［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。
- 削除するサブプロジェクトを選択し，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。

注意　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。

(d) サブプロジェクトの移動

以下の方法により，サブプロジェクトの移動を行うことができます。

- 移動するサブプロジェクトをドラッグし，移動先でドロップする。

備考　CubeSuite+ を複数起動し，異なるプロジェクトにドロップすることができますが，その場合はサブプロジェクトは移動ではなく，コピーされます。

(e) 複数ノードの選択

以下の方法により，複数の連続したノードの選択を行うことができます。

- 始点となるノード上でクリック，またはカーソルキーで選択したのち，［Shift］キーを押下しながら終点となるノードをクリック，またはカーソルキーで移動する。

以下のいずれかの方法により，複数の連続していないノードの選択を行うことができます。

- ［Ctrl］キーを押下しながらノード上でをクリックする。
- ［Ctrl］キーを押下しながらカーソルキーでノードを選択したのち，［Space］キーを押下する。

(f) 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）
- C++ ソース・ファイル（*.cpp，*.cc，*.cp）【CC-RX】
- ヘッダ・ファイル（*.h，*.inc）
- ヘッダ・ファイル（*.hpp）【CC-RX】
- アセンブラ・ソース・ファイル（*.src）【CC-RX】
- アセンブラ・ソース・ファイル（*.s）【CA850】【CX】
- アセンブラ・ソース・ファイル（*.asm）【CX】【CA78K0R】【CA78K0】
- リンク・ディレクティブ・ファイル（*.dir，*.dr）【CA850】【CX】【CA78K0R】【CA78K0】
- リンク順指定ファイル（*.mtls）
- セクション・ファイル（*.sf）【CA850】
- シンボル情報ファイル（*.sfg）【CX】
- 変数／関数情報ファイル（*.vfi）【CA78K0R】
- 変数情報ファイル（*.vfi）【CA78K0】
- 関数情報ファイル（*.fin）【CA78K0】
- マップ・ファイル（*.map）
- マップ・ファイル（*.lbp）【CC-RX】
- シンボル・テーブル・ファイル（*.sym）【CA78K0R】【CA78K0】
- ヘキサ・ファイル（*.hex）
- ヘキサ・ファイル（*.hxb，*.hxf）【CA78K0R】【CA78K0】
- アセンブル・リスト・ファイル（*.lst）【CC-RX】
- S レコード・ファイル（*.mot）【CC-RX】
- クロス・リファレンス・ファイル（*.cref）【CC-RX】
- リンク・サブコマンド・ファイル（*.clnk）【CC-RX】
- 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。

［ツールバー］

	プロジェクト・ツリー エリアのカテゴリ・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
	プロジェクト・ツリー エリアの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
	プロジェクト・ツリー エリアのカテゴリ・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。

［［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，このメニューは，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。
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，このメニューは，クリップボードにファイル，カテゴリ・ノードが存在する場合のみ有効となります。ただし，クリップボート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。
削除	選択しているファイル，カテゴリ・ノードを削除します。 ファイルを選択している場合は，ファイル・システム上からも削除されます。 選択しているファイルを他のプロジェクトにも登録している場合は，ファイル・システム上からは削除せず，そのプロジェクトから外します。 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削除します。 なお，このメニューは，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。

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されます。 ファイルを選択し，そのファイルを他のプロジェクトにも登録している場合は，それらの名前も変更されます。 なお，このメニューは，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。

[[ヘルプ］メニュー（プロジェクト・ツリー パネル専用部分）]

プロジェクト・ツリー パネルのヘルプを開く	このパネルのヘルプを表示します。

[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，このメニューは，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，このメニューは，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，このメニューは，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。
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されます。
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，このメニューは，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*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パックして保存 ...	プロジェクト一式と本製品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。
貼り付け	この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，このメニューは，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，このメニューは，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，このメニューは，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。

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されます。
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，このメニューは，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・システム上からは削除されません。 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，このメニューは，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	この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) マイクロコントローラ・ノードを選択している場合

プロパティ	選択しているマイクロコントローラのプロパティをプロパティ パネルに表示します。

(4) ビルド・ツール・ノードを選択している場合

ここでは，デバッグ専用プロジェクト（「付録F　外部ビルド・ツールの使用」参照）を対象としたビルド・ツール・ノード（“なし(ビルド・ツール)”と表示）を選択している場合のメニューについてのみ示します。

デバッグ専用プロジェクト以外を対象としたビルド・ツール・ノードについては，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください。

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。
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。

(5) ダウンロード・ファイル・ノードを選択している場合

このノードは，デバッグ専用プロジェクト（「付録F　外部ビルド・ツールの使用」参照）を対象とした場合のみ表示されます。

追加	プロジェクトにダウンロード・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ダウンロード・ファイルとして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，選択した種類で空のファイルを作成したのち，ダウンロード・ファイルとしてプロジェクトに追加します。
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。
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。

(6) 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。
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，クリップボート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。
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) ファイルを選択している場合

ここでは，デバッグ専用プロジェクト（「付録Ｆ　外部ビルド・ツールの使用」参照）を対象としたファイルを選択している場合のメニューについてのみ示します。

デバッグ専用プロジェクト以外を対象としたファイルについては，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください。

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（「(f) エディタの起動」参照）。 なお，本メニューは，複数のファイルを選択している場合は無効となります。
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。 なお，本メニューは，複数のファイルを選択している場合は無効となります。
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。 ただし，複数のファイルを選択している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。

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。
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他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。
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。

(8) 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。
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。
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，クリップボート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。 カテゴリ名を編集中の場合は，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。

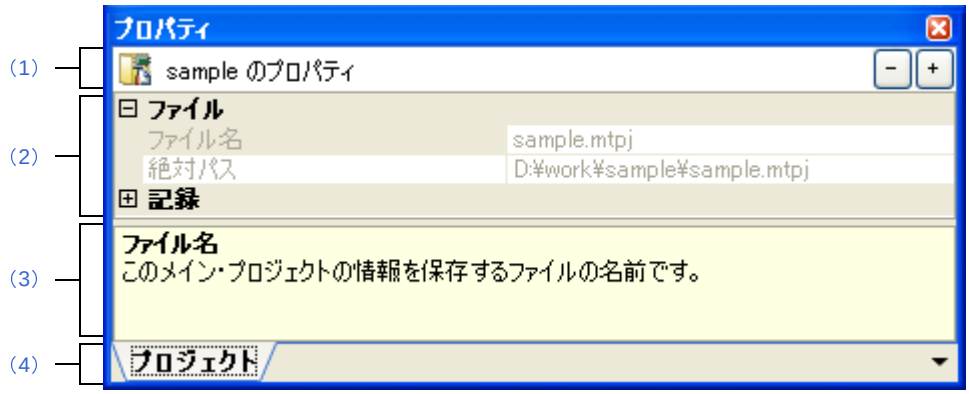
備考 1. 複数の構成要素を選択している場合は，最後に選択した構成要素に対応するコンテキスト・メニューを表示します。

2. プロジェクト・ツリー下部の空白部分では，現在選択している構成要素に対応するコンテキスト・メニューを表示します。

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ノードの種類について，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。また，コード生成パネルでクリックした［コード生成］ボタンの種類，コード生成プレビュー パネルで選択したファイルに対応した情報の表示，および設定の変更を行います。

図 A—20　プロパティ パネル



ここでは，以下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］
- ［プロパティ パネルからオープンするダイアログ］
- ［［編集］メニュー（プロパティ パネル専用部分）］
- ［［ヘルプ］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，プロジェクト・ノード，サブプロジェクト・ノード，マイクロコントローラ・ノード，設計ツール・ノード，ビルド・ツール・ノード，デバッグ・ツール・ノード，解析ツール・ノード，ファイル・ノード，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
- コード生成パネルにおいて，［コード生成］ボタンをクリックしたのち，［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
- コード生成プレビュー パネルにおいて，ファイルを選択したのち，［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，プロジェクト・ノード，サブプロジェクト・ノード，マイクロコントローラ・ノード，設計ツール・ノード，ビルド・ツール・ノード，デバッグ・ツール・ノード，解析ツール・ノード，ファイル・ノード，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) 対象名エリア，および展開状態変更ボタン

対象名エリアには，プロジェクト・ツリー パネルで選択しているノードの名称を表示します。

複数のノードを選択している場合，このエリアは空欄となります。

展開状態変更ボタンは，現在表示している詳細情報表示／変更エリアの展開状態を変更します。

−	現在表示している詳細情報表示／変更エリアをすべて閉じた状態にします。
+	現在表示している詳細情報表示／変更エリアをすべて展開した状態にします。

(2) 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プロジェクト・ノード，サブプロジェクト・ノード，マイクロコントローラ・ノード，ビルド・ツール・ノード，デバッグ・ツール・ノード，解析ツール・ノード，ファイル・ノード，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が展開表示されていることを示し，また，⊞マークは，カテゴリ内の項目が折りたたみ表示され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。

HEX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。

(3) プロパティの説明エリア

詳細情報表示／変更エリアで選択したカテゴリや項目の簡単な説明を表示します。

(4) 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。

このパネルには，以下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 パネルで以下のノードを選択した場合

プロジェクト・ノード	- ［プロジェクト］タブ
サブプロジェクト・ノード	- ［サブプロジェクト］タブ
マイクロコントローラ・ノード	- ［マイクロコントローラ情報］タブ
設計ツール・ノード	- ［端子配置の情報］タブ - ［端子配置図の設定］タブ - ［出力設定］タブ - ［マクロ設定］タブ タブの詳細については，「CubeSuite+ 統合開発環境 ユーザーズマニュアル 設計編」を参照してください。

ビルド・ツール・ノード （デバッグ専用プロジェクト注1以外）	- ［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 - ［ライブラリアン・オプション］タブ【CC-RX】 - ［ROM 化プロセス・オプション］タブ【CA850】【CA78K0R】 - ［ROM 化オプション］タブ【CX】 - ［ヘキサ・コンバート・オプション］タブ【CA850】 - ［ヘキサ出力オプション］タブ【CX】 - ［オブジェクト・コンバート・オプション］タブ【CA78K0R】【CA78K0】 - ［アーカイブ・オプション］タブ【CA850】 - ［ライブラリ生成オプション］タブ【CX】【CA78K0R】【CA78K0】 - ［ライブラリ・ジェネレート・オプション］タブ【CC-RX】 - ［セクション・ファイル・ジェネレート・オプション］タブ【CA850】 - ［ダンプ・オプション］タブ【CA850】 - ［クロス・リファレンス・ツール・オプション］タブ【CA850】 - ［メモリ・レイアウト視覚化ツール・オプション］タブ【CA850】 - ［変数／関数配置オプション］タブ【CA78K0R】 - ［変数配置オプション］タブ【CA78K0】 - ［メモリ・バンク配置オプション］タブ【CA78K0】 タブの詳細については，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください。
ビルド・ツール・ノード （デバッグ専用プロジェクト注1）	- ［ビルド・オプション］タブ
デバッグ・ツール・ノード	- ［接続用設定］タブ - ［デバッグ・ツール設定］タブ - ［フラッシュ・セルフ・エミュレーション設定］タブ【IECUBE】 - ［データフラッシュ・エミュレーション設定］タブ【IECUBE【V850】】【IECUBE【RL78】】【IECUBE【78K0R】】 - ［ダウンロード・ファイル設定］タブ - ［フラッシュ・オプション設定］タブ【V850E2】 - ［フック処理設定］タブ タブの詳細については，「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。
解析ツール・ノード	- ［設定］タブ タブの詳細については，「CubeSuite+ 統合開発環境 ユーザーズマニュアル 解析編」を参照してください。

ファイル	- ［ビルド設定］タブ（C ソース・ファイル，C++ ソース・ファイル【CC-RX】，アセンブラ・ソース・ファイル，オブジェクト・モジュール・ファイル，リンク・ディレクティブ・ファイル【CA850】【CX】【CA78K0R】【CA78K0】，シンボル情報ファイル【CX】，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）【CA850】【CX】【CA78K0R】【CA78K0】 - ［個別コンパイル・オプション(C)］タブ（C ソース・ファイルの場合）【CC-RX】 - ［個別コンパイル・オプション(C++)］タブ（C++ ソース・ファイルの場合）【CC-RX】 - ［個別アセンブル・オプション］タブ（アセンブラ・ソース・ファイルの場合）**注 2** - ［ファイル情報］タブ タブの詳細については，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください。
カテゴリ・ノード	- ［カテゴリ情報］タブ タブの詳細については，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください。

注 1. デバッグ専用プロジェクトについての詳細は，「付録 F　外部ビルド・ツールの使用」を参照してください。

2. このタブは，以下の場合も表示されます。

- 【CA78K0】【CA78K0R】
  ［個別コンパイル・オプション］タブの［アセンブリ・ファイル］カテゴリの［アセンブリ・ファイルを出力する］プロパティで［はい］を選択した場合
- 【CA850】
  ［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
- 【CA850】
  C ソース・ファイルを選択し，［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［アセンブリ・ファイルを出力する］プロパティで［はい (-Fs)］を選択した場合

(b) コード生成パネルの［コード生成］ボタンをクリックした場合

- ［マクロ設定］タブ タブの詳細については，「CubeSuite+ 統合開発環境 ユーザーズマニュアル 設計編」を参照してください。

(c) コード生成プレビュー パネルでファイルを選択している場合

- ［ファイル設定］タブ タブの詳細については，「CubeSuite+ 統合開発環境 ユーザーズマニュアル 設計編」を参照してください。

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されます。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されます。

[プロパティ パネルからオープンするダイアログ]

プロパティ パネルからオープンするダイアログには，次のものがあります。

- 文字列入力 ダイアログ

  詳細は，「文字列入力 ダイアログ（CubeSuite+ 統合開発環境 ユーザーズマニュアル 起動編）」「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

- テキスト編集 ダイアログ

  詳細は，「テキスト編集 ダイアログ（CubeSuite+ 統合開発環境 ユーザーズマニュアル 起動編）」「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」「CubeSuite+ 統合開発環境 ユーザーズマニュアル デバッグ編」を参照してください。

- パス編集 ダイアログ

  詳細は，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください。

[[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の場合，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の場合，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の場合，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の場合，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。

[[ヘルプ] メニュー（プロパティ パネル専用部分）]

プロパティ パネルのヘルプを開く	このパネルのヘルプを表示します。

[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の場合，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の場合，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の場合，選択している文字列を削除します。

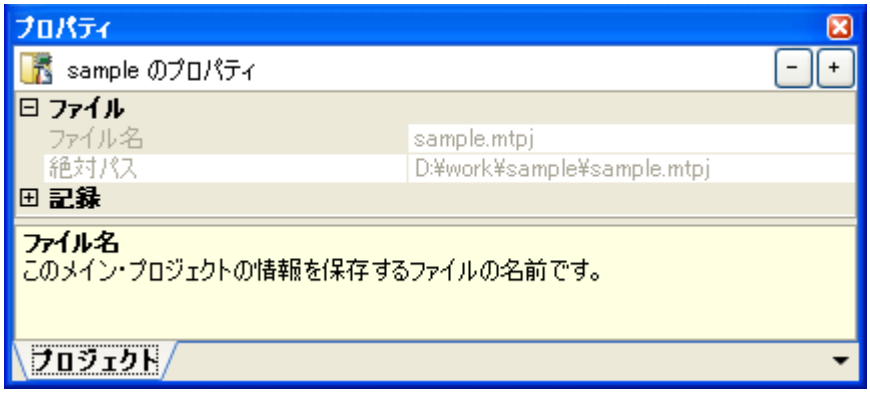
すべて選択	プロパティの値を編集中の場合，選択しているプロパティの値文字列をすべて選択します。
デフォルトに戻す	選択している項目の設定値をプロジェクトに設定しているデフォルト値に戻します。 ただし，［個別コンパイル・オプション］タブ，［個別アセンブル・オプション］タブにおいては，全体オプションの設定値に戻します。
すべてデフォルトに戻す	現在表示しているタブの設定値をすべてプロジェクトに設定しているデフォルト値に戻します。 ただし，［個別コンパイル・オプション］タブ，［個別アセンブル・オプション］タブにおいては，全体オプションの設定値に戻します。

［プロジェクト］タブ

このタブでは，プロジェクト（メイン・プロジェクト）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］
（2）［記録］

図 A—21　プロパティ パネル：［プロジェクト］タブ



［各カテゴリの説明］

（1）［ファイル］

ファイルに関する詳細情報の表示を行います。

ファイル名	プロジェクト（メイン・プロジェクト）の情報を保存するファイルのファイル名を表示します。	
	デフォルト	*プロジェクト・ファイル名*
	変更方法	変更不可
絶対パス	プロジェクト（メイン・プロジェクト）の情報を保存するプロジェクト・ファイルの絶対パス名を表示します。	
	デフォルト	*プロジェクト・ファイルの絶対パス名*
	変更方法	変更不可

(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。

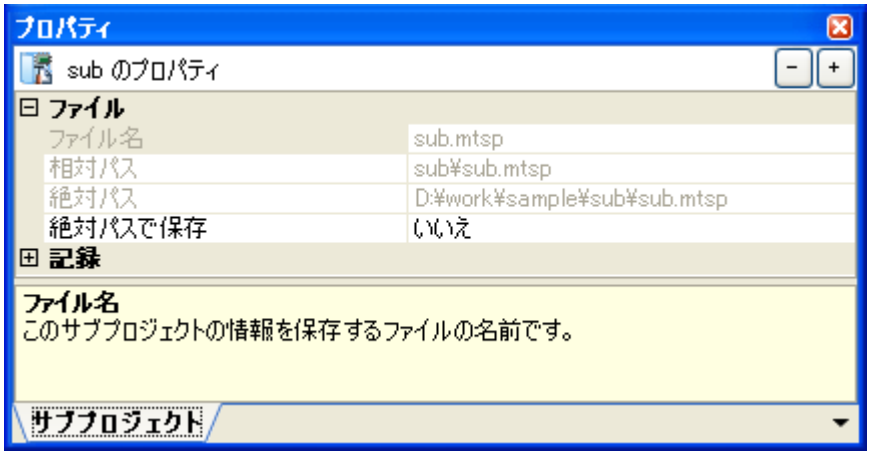
メモ	プロジェクト（メイン・プロジェクト）にメモを追加します。 1 行に 1 項目ずつ指定します。 追加したメモはサブ・プロパティとして表示されます。	
	デフォルト	メモ［*項目数*］
	変更方法	［...］ボタンをクリックし，テキスト編集 ダイアログによる編集 サブ・プロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

［サブプロジェクト］タブ

このタブでは，サブプロジェクトに対して，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］
（2）［記録］

図 A—22　プロパティ パネル：［サブプロジェクト］タブ



［各カテゴリの説明］

（1）［ファイル］

ファイルに関する詳細情報の表示を行います。

ファイル名	サブプロジェクトの情報を保存するファイルのファイル名を表示します。	
	デフォルト	*サブプロジェクト・ファイル名*
	変更方法	変更不可
相対パス	サブプロジェクトの情報を保存するサブプロジェクト・ファイルのメイン・プロジェクト・ファイルの場所からの相対パス名を表示します。 なお，このプロパティは，プロジェクト・ツリー パネルでサブプロジェクトを選択した場合のみ表示されます。ただし，相対パスで表現できない場合（メイン・プロジェクトとサブプロジェクトが異なるドライブに存在する場合など）は，絶対パスで表示されます。	
	デフォルト	*メイン・プロジェクトからの相対パス名*
	変更方法	変更不可
絶対パス	サブプロジェクトの情報を保存するサブプロジェクト・ファイルの絶対パス名を表示します。	
	デフォルト	*サブプロジェクト・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	サブプロジェクトの情報を保存するサブプロジェクト・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。		
	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。

(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。

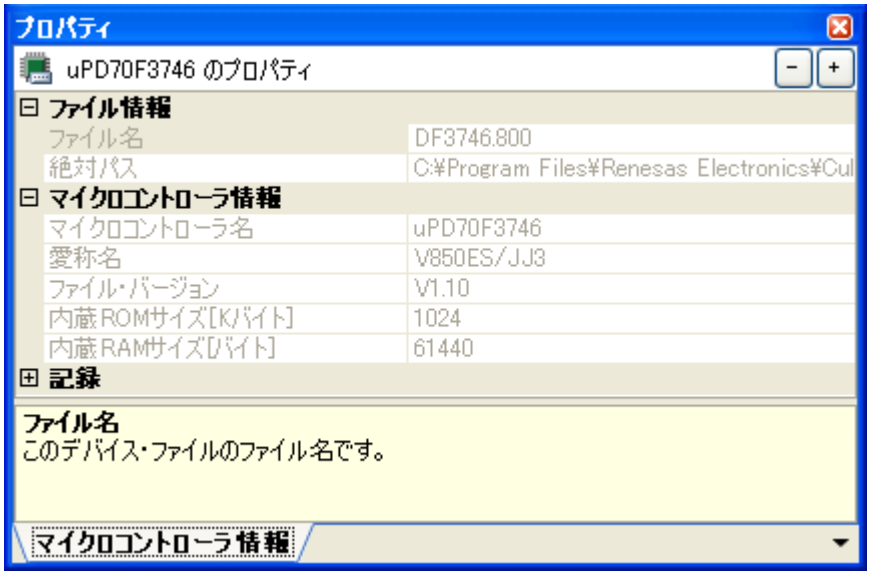
メモ	サブプロジェクトにメモを追加します。 1 行に 1 項目ずつ指定します。 追加したメモはサブ・プロパティとして表示されます。	
	デフォルト	メモ［*項目数*］
	変更方法	［...］ボタンをクリックし，テキスト編集 ダイアログによる編集 サブ・プロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

［マイクロコントローラ情報］タブ

このタブでは，プロジェクトで設定しているマイクロコントローラに対して，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）［マイクロコントローラ情報］
（3）［記録］

図 A—23　プロパティ パネル：［マイクロコントローラ情報］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を行います。

ファイル名	デバイス・ファイルのファイル名を表示します。	
	デフォルト	*デバイス・ファイルのファイル名*
	変更方法	変更不可
絶対パス	デバイス・ファイルの絶対パス名を表示します。	
	デフォルト	*デバイス・ファイルの絶対パス名*
	変更方法	変更不可

(2)［マイクロコントローラ情報］

マイクロコントローラに関する詳細情報の表示を行います。

マイクロコントローラ名	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マイクロコントローラを表示します。	
	デフォルト	*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マイクロコントローラ*
	変更方法	変更不可
愛称名	使用するマイクロコントローラの愛称名を表示します。	
	デフォルト	*使用するマイクロコントローラの愛称名*
	変更方法	変更不可
ファイル・バージョン	使用するマイクロコントローラのデバイス・ファイルのバージョンを表示します。	
	デフォルト	*使用するマイクロコントローラのデバイス・ファイルのバージョン*
	変更方法	変更不可
バンク・サイズ［K バイト］	使用するマイクロコントローラのバンク・サイズ［K バイト］が 10 進数表示されます。なお，このプロパティは，メモリ・バンク搭載品の場合のみ表示されます。	
	デフォルト	*使用するマイクロコントローラのデバイス・ファイルに記録されているバンク・サイズ*
	変更方法	変更不可
内部／内蔵 ROM サイズ［K バイト］	使用するマイクロコントローラの内部／内蔵 ROM のサイズ［K バイト］が 10 進数表示されます。	
	デフォルト	*使用するマイクロコントローラの内部／内蔵 ROM のサイズ*
	変更方法	変更不可
内部／内蔵 RAM サイズ［K バイト］	使用するマイクロコントローラの内部／内蔵 RAM のサイズ［K バイト］が 10 進数表示されます。	
	デフォルト	*使用するマイクロコントローラの内部／内蔵 RAM のサイズ*
	変更方法	変更不可

(3)［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。

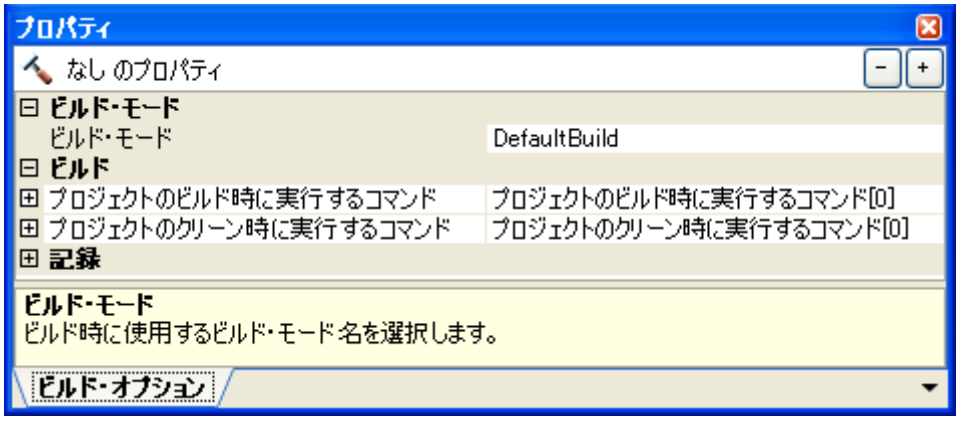
メモ	マイクロコントローラ情報にメモを追加します。 1 行に 1 項目ずつ指定します。 追加したメモはサブ・プロパティとして表示されます。	
	デフォルト	メモ［*項目数*］
	変更方法	［...］ボタンをクリックし，テキスト編集 ダイアログによる編集 サブ・プロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

［ビルド・オプション］タブ

このタブでは，デバッグ専用プロジェクト（「付録Ｆ　外部ビルド・ツールの使用」を参照）のビルド・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）［ビルド］
（3）［記録］

図 A—24　プロパティ パネル：［ビルド・オプション］タブ



［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	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 以外）	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) [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。

プロジェクトのビルド時に実行するコマンド	デバッグ専用プロジェクトのビルド時（F. 5. 1　ビルドを実行する）に実行するコマンドを指定します。 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。 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，ビルド時に 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。	
	デフォルト	プロジェクトのビルド時に実行するコマンド [0]
	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。

プロジェクトのクリーン時に実行するコマンド	デバッグ専用プロジェクトのクリーン時（F. 5. 3　クリーンを実行する）に実行するコマンドを指定します。 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。 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，クリーン時に 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。	
	デフォルト	プロジェクトのクリーン時に実行するコマンド [0]
	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。

(3)［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。

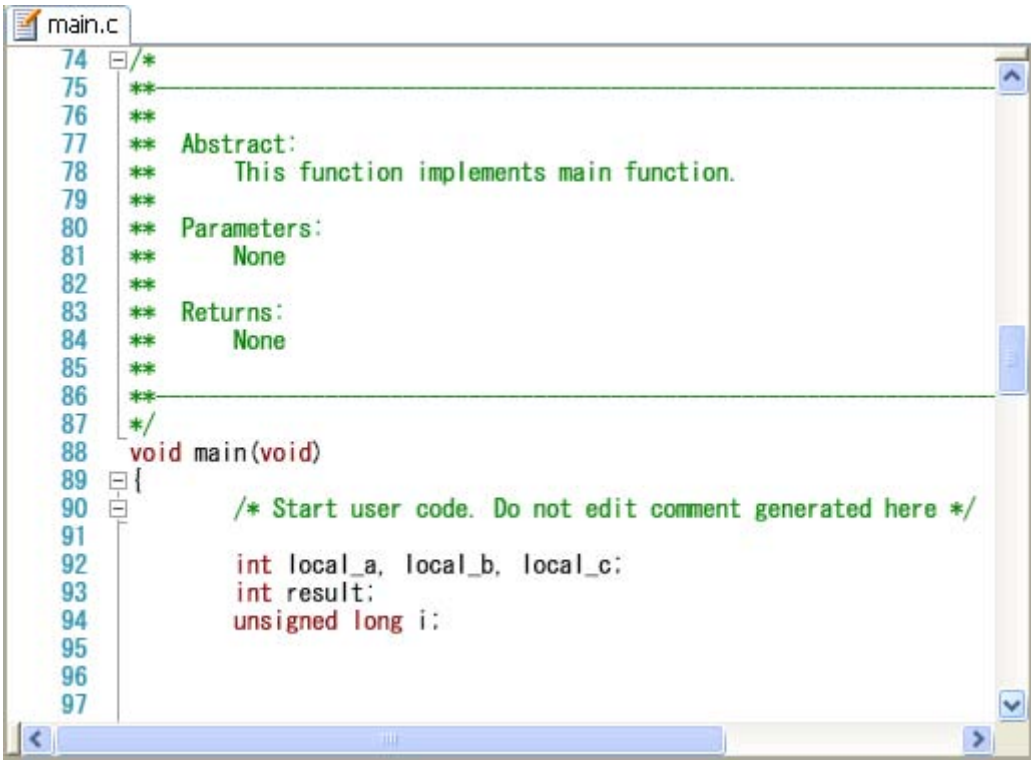
メモ	このビルド・ツールにメモを追加します。 1 行に 1 項目ずつ指定します。 この設定はすべてのビルド・モードで共通です。 追加したメモはサブプロパティとして表示します。	
	デフォルト	メモ［*項目数*］
	変更方法	[...] ボタンをクリックし，テキスト編集 ダイアログによる編集 サブ・プロパティはテキスト・ボックスによる直接入力も可能
	指定可能値	256 文字までの文字列 256 個まで指定可能です。

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 コーディング編」を参照してください。

図 A—25　エディタ パネル



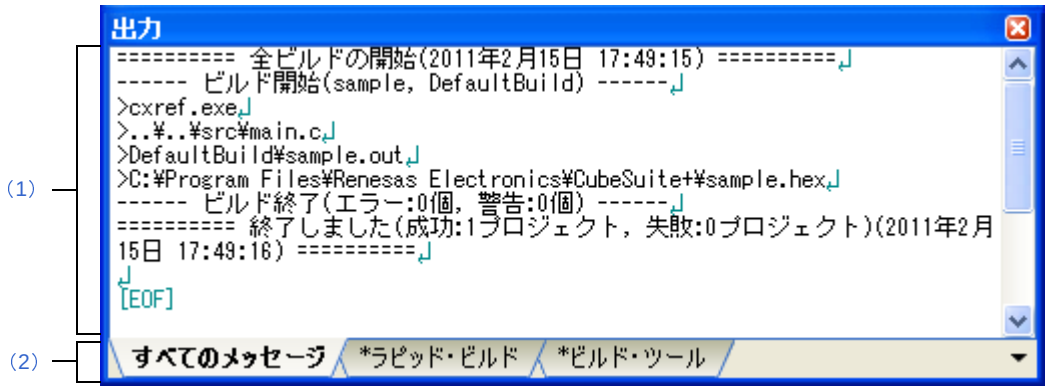
出力 パネル

各ツール（ビルド・ツール／デバッグ・ツール／プラグイン製品）から出力されるメッセージ，または検索・置換 ダイアログによる一括検索を行った際の結果を表示します。

メッセージは，出力元のツールごとに分類されたタブ上でそれぞれ個別に表示します。

［すべてのメッセージ］タブでは，すべての出力メッセージを集約して表示します。

図 A—26　出力 パネル



ここでは，以下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

各ツールから出力されたメッセージ，および検索結果を表示します。

［すべてのメッセージ］タブでは，すべての出力メッセージを集約して表示します。

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，以下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	黒	何らかの情報を通知する際に表示されます。
		背景色	白	

メッセージ種別	表示例（デフォルト）			説明
警告メッセージ	AaBbCc	文字色	青	操作に対して，何らかの警告を通知する際に表示されます。
		背景色	標準色	
エラー・メッセージ	AaBbCc	文字色	赤	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されます。
		背景色	薄グレー	

このエリアは，以下の機能を備えています。

(a) タグ・ジャンプ

出力され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あわせて［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，ビルド時に出力されたエラー・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・メッセージが表示されている時に，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，表示されたメッセージ，または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。

(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され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。

表示されるタブは以下のとおりです。

タブ名	説明
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・ツールから出力されたメッセージを表示します。
ラピッド・ビルド	ラピッド・ビルドの実行により，ビルド・ツールから出力されたメッセージを表示します。
デバッグ・ツール	デバッグ・ツールから出力されたメッセージを表示します。
コード生成	コード生成から出力されたメッセージを表示します。
プログラム解析	解析ツールから出力されたメッセージを表示します。
参照の検索	解析ツールによる関数／変数の参照箇所一覧を表示します。 タブの詳細については，「CubeSuite+ 統合開発環境 ユーザーズマニュアル 解析編」を参照してください。
検索・置換	検索・置換 ダイアログによる一括検索結果を表示します。
すべてのメッセージ	上記のすべてのメッセージを出力順に一括して表示します。

注意　新たなメッセージが非選択状態のタブ上に出力されても，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＊が付加し，新たなメッセージが出力され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］メニューは以下のとおりです（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されている内容を，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，起動後にはじめてこの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。 ただし，ビルド実行中は無効となります。
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されている内容を，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）。 ただし，プログラム実行中は無効となります。

[[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］メニューは以下のとおりです（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	この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。

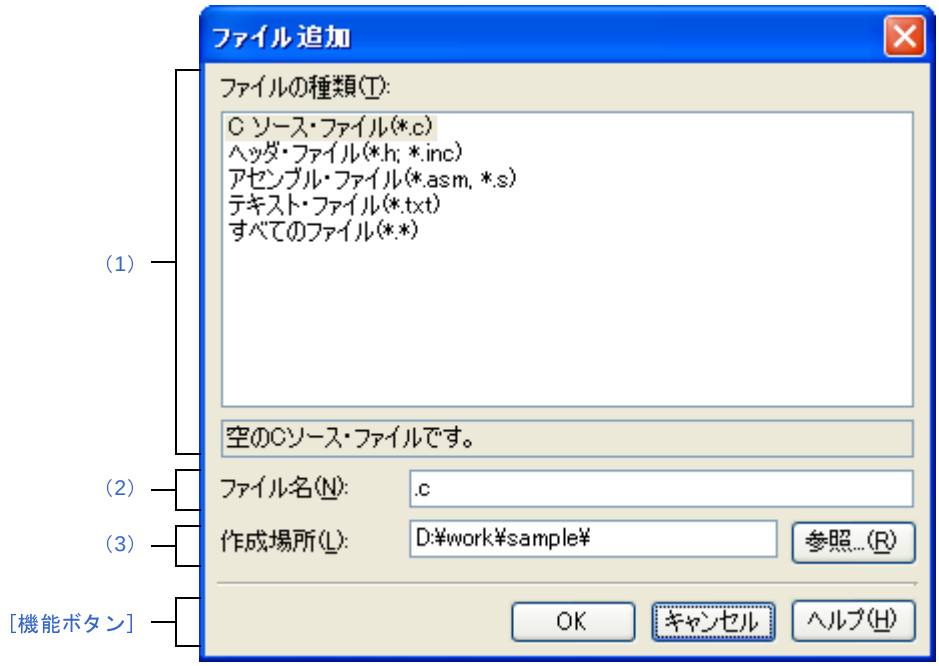
[コンテキスト・メニュー]

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	この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	この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	キャレット行のメッセージに対応するエディタ（ファイル，行，桁）へジャンプします。
メッセージに関するヘルプ	表示されたメッセージ，または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—27　ファイル追加 ダイアログ



ここでは，以下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，ダウンロード・ファイル・ノードを選択したのち，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）

- アセンブル・ファイル（*.asm; *.s）
- テキスト・ファイル（*.txt）
- すべてのファイル（*.*）

(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）。
ただし，［ファイルの種類］エリアで［すべてのファイル（*.*）］を選択した場合は，拡張子の付加は行いません。

(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。

(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]

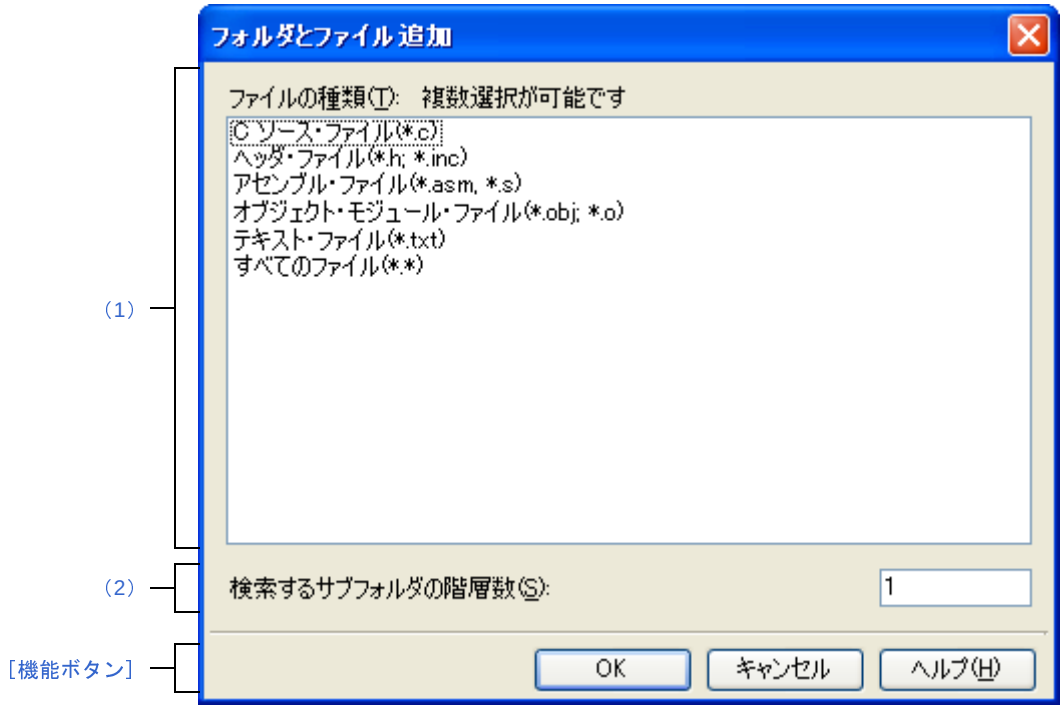
ボタン	機能
OK	入力したファイル名でファイルを作成して，プロジェクトに追加し，エディタ パネルでオープンします。そののち，本ダイアログをクローズします。
キャンセル	ファイルの生成を行わずに，本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

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—28　フォルダとファイル追加 ダイアログ



ここでは，以下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）
- アセンブル・ファイル（*.asm; *.s）
- オブジェクト・モジュール・ファイル（*.obj; *.o）
- テキスト・ファイル（*.txt）
- すべてのファイル（*.*）

(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，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。

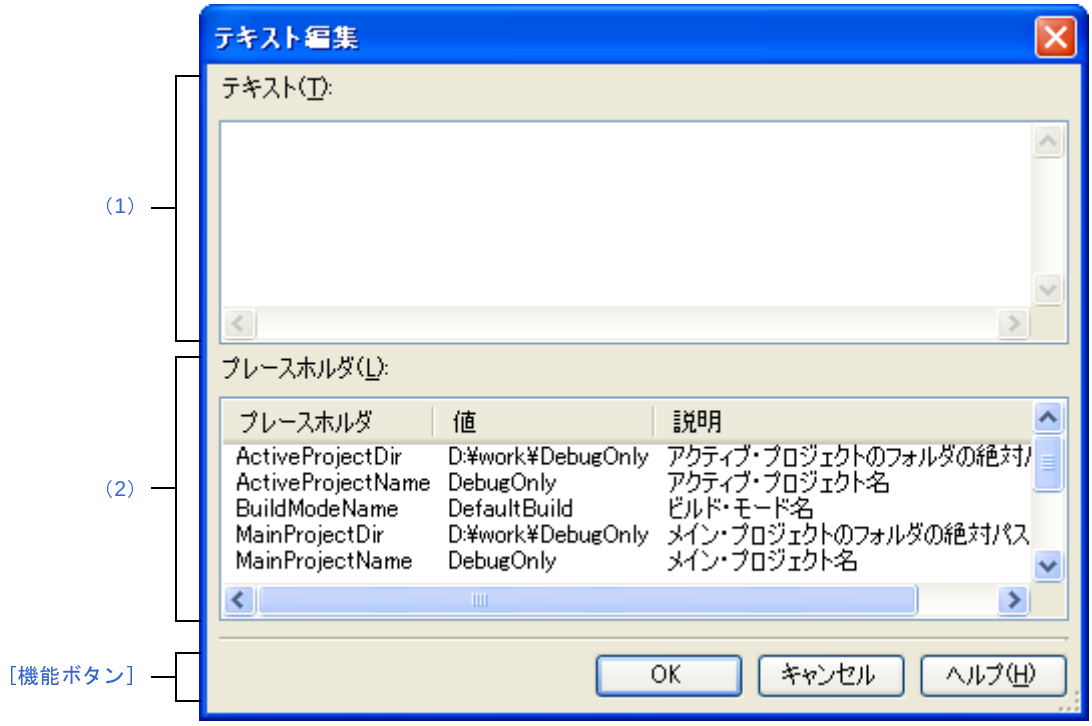
［機能ボタン］

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。

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—29　テキスト編集 ダイアログ



ここでは，以下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 パネルの［ビルド・オプション］タブにおいて，以下のプロパティを選択したのち，［...］ボタンをクリック
  - ［ビルド］カテゴリの［プロジェクトのビルド時に実行するコマンド］
  - ［ビルド］カテゴリの［プロジェクトのクリーン時に実行するコマンド］

[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］

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。

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。

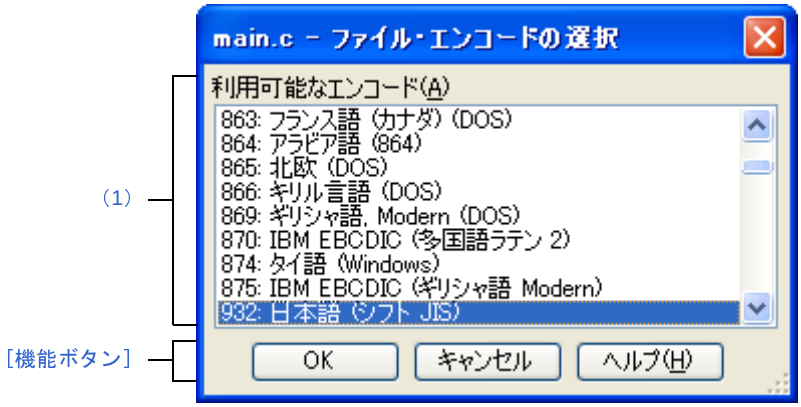
[機能ボタン]

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。

ファイル・エンコードの選択 ダイアログ

ファイル・エンコードの選択を行います。

図 A—30　ファイル・エンコードの選択 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］

- ［ファイル］メニュー→［エンコードを指定して開く ...］を選択してファイルを開く ダイアログをオープン→ダイアログ上で［開く］ボタンをクリック

［各エリアの説明］

(1)［利用可能なエンコード］

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。

現在の OS が対応するコード・ページ／エンコード名を，アルファベット順で表示します。

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されません。

デフォルトでは，オプション ダイアログにおける［全般 - テキスト・エディタ］カテゴリのデフォルトのエンコードを選択します。

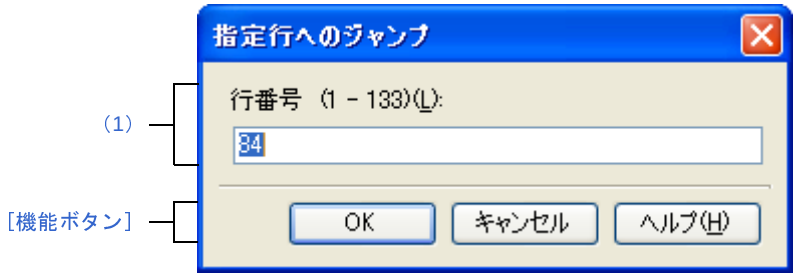
[機能ボタン]

ボタン	機能
OK	指定したファイル・エンコードを使用し，ファイルを開く ダイアログで選択したファイルをオープンします。
キャンセル	ファイルを開く ダイアログで選択したファイルをオープンせずに，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

指定行へのジャンプ ダイアログ

指定したソース行にキャレットを移動します。

図 A—31　指定行へのジャンプ 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］

- ［編集］メニュー→［移動 ...］を選択
- エディタ パネルにおいて，コンテキスト・メニューの［移動 ...］を選択

［各エリアの説明］

(1)［行番号（*有効な行の範囲*）］

“（*有効な行の範囲*）”には，現在のファイルの有効な行の範囲が表示されます。

キャレットを移動したい行番号を 10 進数で直接入力により指定します。

デフォルトでは，エディタ パネル上の現在のキャレット位置の行番号が表示されます。

［機能ボタン］

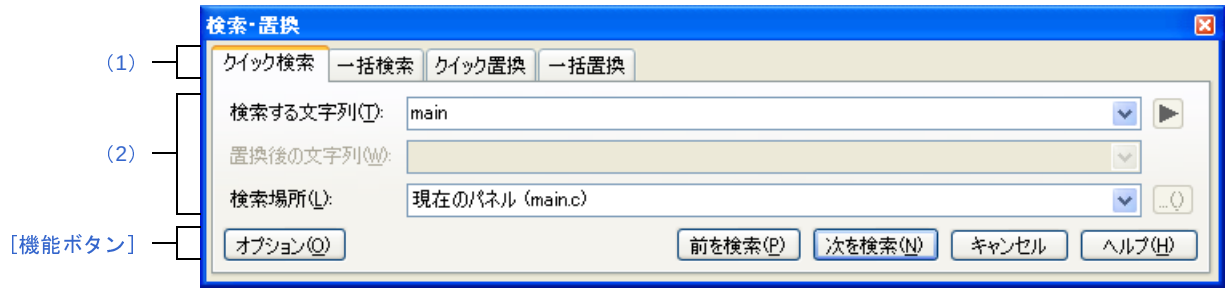
ボタン	機能
OK	指定したソース行の先頭にキャレットを移動します。
キャンセル	移動を無効とし，このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

検索・置換 ダイアログ

指定した文字列の検索，および置換を行います。

備考　［検索する文字列］エリアの ▶ ボタンは，エディタ パネルから本ダイアログをオープンした場合のみ表示されます。

図 A—32　検索・置換 ダイアログ



ここでは，以下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］

- ［編集］メニュー→［検索 ...］，または［編集］メニュー→［置換 ...］を選択

［各エリアの説明］

（1）タブ選択エリア

タブを選択することにより，検索／置換の種類が切り替わります。

本ダイアログには，以下のタブが存在します。

- ［クイック検索］タブ
- ［一括検索］タブ
- ［クイック置換］タブ
- ［一括置換］タブ

（2）検索／置換条件設定エリア

検索／置換する際の詳細条件を設定します。

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。

[機能ボタン]

検索／置換を実行する際の機能ボタンです。

各ボタンの機能についての詳細は，該当するタブの項を参照してください。

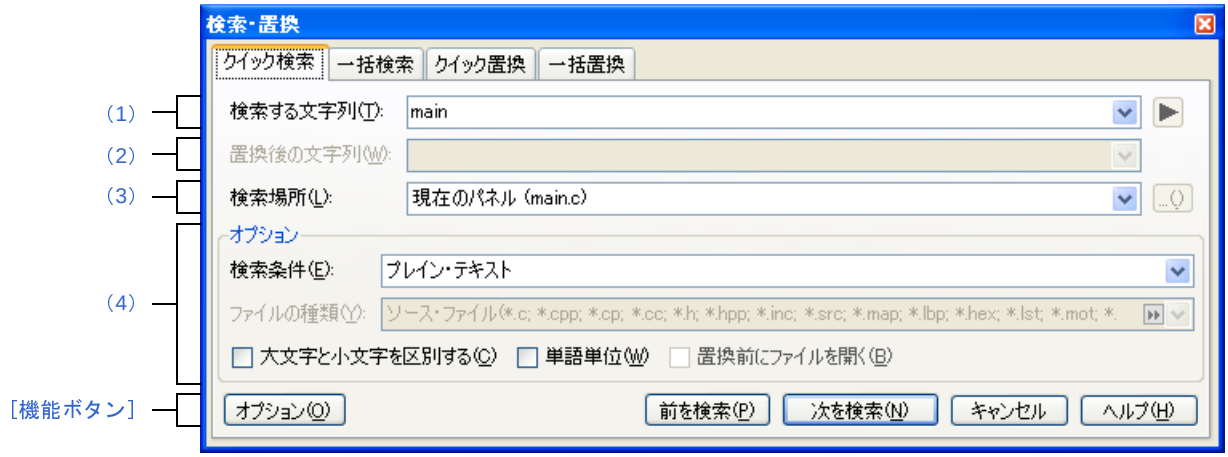
［クイック検索］タブ

指定した文字列を検索し，検索位置を選択状態にしてキャレットを移動します。

備考 1. 出力 パネル／エディタ パネル以外のパネルから検索・置換 ダイアログが呼び出された場合，本タブを選択することができません。

2. ［検索する文字列］エリアの ▶ ボタンは，エディタ パネルから本ダイアログをオープンした場合のみ表示されます。

図 A—33　検索・置換 ダイアログ：［クイック検索］タブ



ここでは，以下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］

- ［編集］メニュー→［検索 ...］，または［編集］メニュー→［置換 ...］を選択

［各エリアの説明］

(1)［検索する文字列］エリア

検索する文字列を指定します。

テキスト・ボックスに直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

なお，文字列を選択した状態のパネルから本ダイアログをオープンした場合，選択している文字列がデフォルトで表示されます。

また，エディタ パネルから本ダイアログをオープンした場合では，キャレット位置の単語（変数／関数）をデフォルトで表示します。

ボタンを使用することにより，以下のワイルド・カードを選択できます。エディタ パネルから本ダイアログをオープンし，［検索条件］で“ワイルド・カード”が選択されている場合に有効となります。

- # 任意の 1 桁の数字
- * 0 以上の任意の 1 文字
- ? 任意の 1 文字
- [!] 文字セットに含まれない任意の 1 文字
- [] 文字セットに含まれる任意の 1 文字

また，ボタンを使用することにより，以下の正規表現を選択できます。エディタ パネルから本ダイアログをオープンし，［検索条件］で“正規表現”が選択されている場合に有効となります。

- $ 行の末尾
- ( ) グループ キャプチャ
- * 0 以上
- + 1 以上
- . 任意の 1 文字
- [] 文字セットに含まれる任意の 1 文字
- [^] 文字セットに含まれない任意の 1 文字
- ¥ エスケープ文字
- ¥b 単語の境界
- ¥n 改行
- ¥s 空白
- ^ 行の先頭
- | または

備考　正規表現についての詳細は，「付録 E　正規表現の構文」を参照してください。

(2)［置換後の文字列］エリア

本項目は無効となります。

(3)［検索場所］エリア

検索する場所を指定します。

以下のいずれかをドロップダウン・リストより選択します。

項目	動作
選択範囲	最後にフォーカスのあった検索可能なパネルで選択されている範囲を検索します。 ただし，エディタ パネルから本ダイアログをオープンした場合，最後にフォーカスのあったパネルで文字列が選択されていない場合，またはそのパネルが検索不可能な場合は，本項目は無効となります。

項目	動作
現在のパネル（*パネル名*）	最後にフォーカスのあった検索可能なパネル内で検索します。 ただし，最後にフォーカスのあったパネルが検索不可能な場合，またはそのパネルが存在しない場合は，本項目は無効となります。

備考　入力履歴項目として，最大 10 個までドロップダウン・リストに記録されます。

(4)［オプション］エリア

検索条件として以下のオプションを指定することができます。

なお，本エリアは，［オプション］ボタンをクリックすることにより表示されます（デフォルトでは表示されません）。

(a)［検索条件］

以下のいずれかをドロップダウン・リストより選択します。

項目	動作
プレイン・テキスト	［検索する文字列］エリアに指定した文字列をそのまま検索します。
ワイルド・カード	［検索する文字列］エリアに指定したワイルド・カードを使用して，文字列を検索します。
正規表現[注]	［検索する文字列］エリアに指定した正規表現を使用して，文字列を検索します。

注　本項目は，エディタ パネルから本ダイアログをオープンした場合のみ有効となります。

(b)［ファイルの種類］

本項目は無効となります。

(c)［大文字と小文字を区別する］

☑	指定した文字列の大文字と小文字を区別して検索します。
☐	指定した文字列の大文字と小文字を区別せず検索します（デフォルト）。

(d)［単語単位］

☑	指定した文字列を単語単位で完全に一致する箇所のみ検索します。
☐	指定した文字列を含む箇所をすべて検索します（デフォルト）。

(e)［置換前にファイルを開く］

本項目は無効となります。

[機能ボタン]

ボタン	機能
オプション	本タブ上の［オプション］エリアの表示／非表示を切り替えます。
前を検索	現在のキャレット位置から文頭方向に，指定した検索条件で指定した文字列を検索します。検索結果箇所を選択状態にし，キャレットを移動します注。
次を検索	現在のキャレット位置から文末方向に，指定した検索条件で指定した文字列を検索します。検索結果箇所を選択状態にし，キャレットを移動します注。
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。

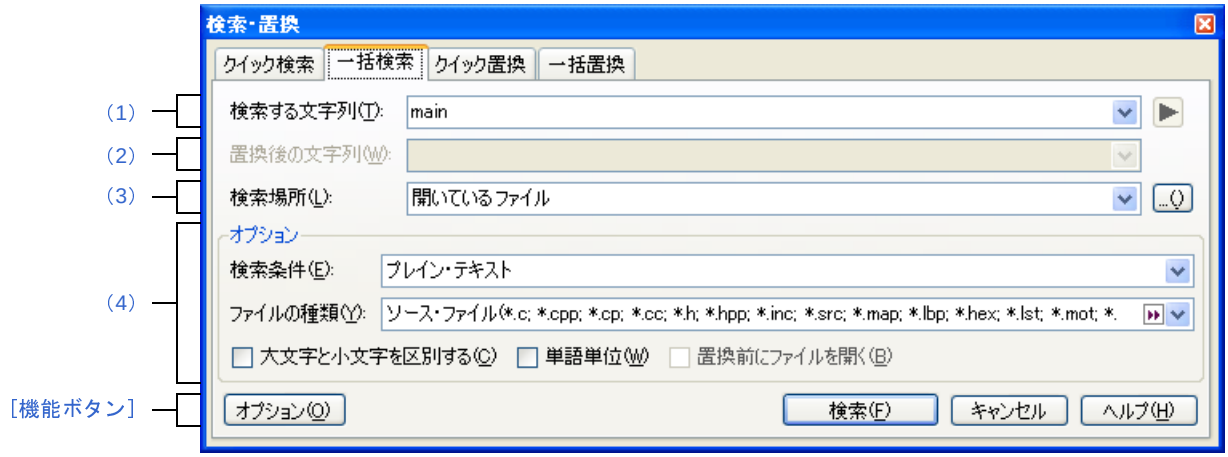
注　指定した文字列が見つからなかった場合，メイン・ウインドウのステータスバー上に“*指定文字列*は見つかりませんでした。”と表示されます。

［一括検索］タブ

指定した文字列を一括検索し，検索結果を出力 パネルに一覧表示します。
出力 パネルにおいて，検索結果の項目をダブルクリックすることにより，該当箇所へジャンプします。

備考　［検索する文字列］エリアの ▶ ボタンは，エディタ パネルから本ダイアログをオープンした場合のみ表示されます。

図 A—34　検索・置換 ダイアログ：［一括検索］タブ



ここでは，以下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］

- ［編集］メニュー→［検索 ...］，または［編集］メニュー→［置換 ...］を選択

［各エリアの説明］

(1)［検索する文字列］エリア

検索する文字列を指定します。

テキスト・ボックスに直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

なお，文字列を選択した状態のパネルから本ダイアログをオープンした場合，選択している文字列がデフォルトで表示されます。

また，エディタ パネルから本ダイアログをオープンした場合では，キャレット位置の単語（変数／関数）をデフォルトで表示します。

ボタンを使用することにより，以下のワイルド・カードを選択できます。エディタ パネルから本ダイアログをオープンし，［検索条件］で“ワイルド・カード”が選択されている場合に有効となります。

- # 任意の 1 桁の数字
- * 0 以上の任意の 1 文字
- ? 任意の 1 文字
- [!] 文字セットに含まれない任意の 1 文字
- [] 文字セットに含まれる任意の 1 文字

また，ボタンを使用することにより，以下の正規表現を選択できます。エディタ パネルから本ダイアログをオープンし，［検索条件］で“正規表現”が選択されている場合に有効となります。

- $ 行の末尾
- ( ) グループ キャプチャ
- * 0 以上
- + 1 以上
- . 任意の 1 文字
- [] 文字セットに含まれる任意の 1 文字
- [^] 文字セットに含まれない任意の 1 文字
- ¥ エスケープ文字
- ¥b 単語の境界
- ¥n 改行
- ¥s 空白
- ^ 行の先頭
- | または

備考　正規表現についての詳細は，「付録 E　正規表現の構文」を参照してください。

(2)［置換後の文字列］エリア

本項目は無効となります。

(3)［検索場所］エリア

検索する場所を指定します。

以下のいずれかをドロップダウン・リストより選択するか，キーボードより直接入力します（最大履歴数：10 個）。

項目	動作
現在のパネル（*パネル名*）	現在のエディタ パネル内で検索します。
開いているファイル	現在オープンしているすべてのエディタ パネル内で検索します。 エディタ パネルでファイルを 1 つもオープンしていない場合は，本項目は無効となります。

項目	動作
アクティブ・プロジェクト	アクティブ・プロジェクトに含まれているテキスト・ファイル内で検索します。 なお，［ファイルの種類］が指定されている場合は，そのファイルのみを検索対象とします。 ただし，現在のプロジェクトが存在しない場合は，本項目は無効となります。
メイン・プロジェクトおよびサブプロジェクト内	メイン・プロジェクト，またはサブプロジェクトに含まれているテキスト・ファイル内で検索します。 なお，［ファイルの種類］が指定されている場合は，そのファイルのみを検索対象とします。 ただし，現在のプロジェクトが存在しない場合は，本項目は無効となります。
指定されたフォルダ名	直接入力（最大指定文字数：259 文字）によりパス指定されたフォルダ（相対パスの場合はプロジェクト・フォルダ相対とします），または本エリア内の［...］ボタンのクリックによってオープンするフォルダの参照 ダイアログにより指定したフォルダ以下のテキスト・ファイル内で検索します。フォルダが指定されていない場合は，プロジェクト・フォルダ名がデフォルトで“()”内に表示されます（プロジェクトが存在しない場合は，現在のユーザのドキュメント・フォルダを対象とします）。 なお，［ファイルの種類］が指定されている場合は，そのファイルのみを検索対象とします。

備考　入力履歴項目として，最大 10 個までドロップダウン・リストに記録されます。

(4)［オプション］エリア

本エリアは，［オプション］ボタンをクリックすることにより表示されます（デフォルトでは表示されません）。

検索条件として以下のオプションを指定することができます。

(a)［検索条件］

以下のいずれかをドロップダウン・リストより選択します。

項目	動作
プレイン・テキスト	［検索する文字列］エリアに指定した文字列をそのまま検索します。
ワイルド・カード	［検索する文字列］エリアに指定したワイルド・カードを使用して，文字列を検索します。
正規表現[注]	［検索する文字列］エリアに指定した正規表現を使用して，文字列を検索します。

注　本項目は，エディタ パネルにフォーカスがある場合のみ有効となります。

(b)［ファイルの種類］

検索対象とするファイルの種類を指定します。

以下のいずれかをドロップダウン・リストより選択します。

項目	動作
ソース・ファイル (*拡張子群*[注])	検索対象のファイルをソース・ファイル限定とします。
*.txt	検索対象のファイルをテキスト・ファイル限定とします。
.	すべてのファイルを検索対象とします。

注　プロジェクト・ツリー パネルに登録しているソース・ファイルを判別して拡張子を表示します。

なお，ここでは，テキスト・ボックスにファイル名を直接入力することにより，ファイル名を限定して検索対象を指定することもできます（最大指定文字数：1024 文字）。
この場合，ワイルド・カード“*”を使用することができ，また複数のファイル名をセミコロン“;”で区切って指定することも可能です。

備考　入力履歴項目として，最大 10 個までドロップダウン・リストに記録されます。

(c)［大文字と小文字を区別する］

☑	指定した文字列の大文字と小文字を区別して検索します。
☐	指定した文字列の大文字と小文字を区別せず検索します（デフォルト）。

(d)［単語単位］

☑	指定した文字列を単語単位で完全に一致する箇所のみ検索します。
☐	指定した文字列を含む箇所をすべて検索します（デフォルト）。

(e)［置換前にファイルを開く］

本項目は無効となります。

［機能ボタン］

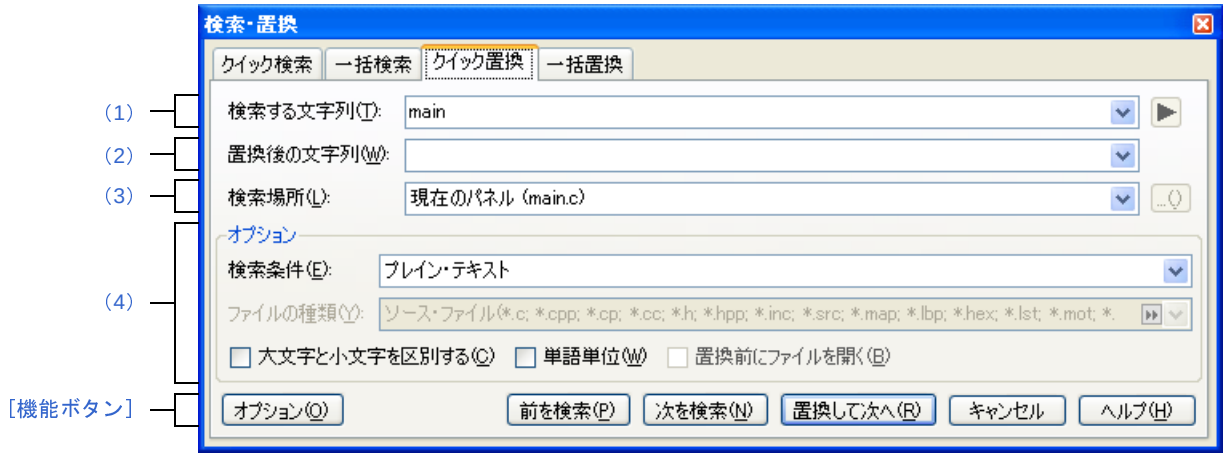
ボタン	機能
オプション	本タブ上の［オプション］エリアの表示／非表示を切り替えます。
検索	指定した検索条件で指定した文字列を一括検索し，検索結果を出力 パネルに一覧表示します。
キャンセル	設定を無効とし，本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

［クイック置換］タブ

指定した文字列を検索したのち，置換文字列に置換します。

備考　エディタ パネル以外のパネルから検索・置換 ダイアログが呼び出された場合，本タブを選択することができません。

図 A—35　検索・置換 ダイアログ：［クイック置換］タブ



ここでは，以下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］

- ［編集］メニュー→［検索 ...］，または［編集］メニュー→［置換 ...］を選択

［各エリアの説明］

(1)［検索する文字列］エリア

検索する文字列を指定します。

テキスト・ボックスに直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

なお，文字列を選択した状態のパネルから本ダイアログをオープンした場合，選択している文字列がデフォルトで表示されます。

また，キャレット位置の単語（変数／関数）をデフォルトで表示します。

▶ボタンを使用することにより，以下のワイルド・カードを選択できます。[検索条件] で“ワイルド・カード”が選択されている場合に有効となります。

- # 任意の 1 桁の数字
- * 0 以上の任意の 1 文字
- ? 任意の 1 文字
- [!] 文字セットに含まれない任意の 1 文字
- [] 文字セットに含まれる任意の 1 文字

また，▶ボタンを使用することにより，以下の正規表現を選択できます。[検索条件] で“正規表現”が選択されている場合に有効となります。

- $ 行の末尾
- ( ) グループ キャプチャ
- * 0 以上
- + 1 以上
- . 任意の 1 文字
- [] 文字セットに含まれる任意の 1 文字
- [^] 文字セットに含まれない任意の 1 文字
- ¥ エスケープ文字
- ¥b 単語の境界
- ¥n 改行
- ¥s 空白
- ^ 行の先頭
- | または

備考　正規表現についての詳細は，「付録 E　正規表現の構文」を参照してください。

(2) [置換後の文字列] エリア

置換する文字列を指定します。

テキスト・ボックスに直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

(3) [検索場所] エリア

ドロップダウン・リストには，以下を表示します。

項目	動作
現在のパネル（*パネル名*）	最後にフォーカスのあった検索可能なパネル内で検索します。 ただし，最後にフォーカスのあったパネルが検索不可能な場合，またはそのパネルが存在しない場合は，本項目は無効となります。

(4)［オプション］エリア

本エリアは，［オプション］ボタンをクリックすることにより表示されます（デフォルトでは表示されません）。検索条件として以下のオプションを指定することができます。

(a)［検索条件］

以下のいずれかをドロップダウン・リストより選択します。

項目	動作
プレイン・テキスト	［検索する文字列］エリアに指定した文字列をそのまま検索します。
ワイルド・カード	［検索する文字列］エリアに指定したワイルド・カードを使用して，文字列を検索します。
正規表現	［検索する文字列］エリアに指定した正規表現を使用して，文字列を検索します。

(b)［ファイルの種類］

本項目は無効となります。

(c)［大文字と小文字を区別する］

☑	指定した文字列の大文字と小文字を区別して検索します。
☐	指定した文字列の大文字と小文字を区別せず検索します（デフォルト）。

(d)［単語単位］

☑	指定した文字列を単語単位で完全に一致する箇所のみ検索します。
☐	指定した文字列を含む箇所をすべて検索します（デフォルト）。

(e)［置換前にファイルを開く］

本項目は無効となります。

［機能ボタン］

ボタン	機能
オプション	本タブ上の［オプション］エリアの表示／非表示を切り替えます。
前を検索	現在のキャレット位置から文頭方向に，指定した検索条件で指定した文字列を検索します。検索結果箇所を選択状態にし，キャレットを移動します注。
次を検索	現在のキャレット位置から文末方向に，指定した検索条件で指定した文字列を検索します。検索結果箇所を選択状態にし，キャレットを移動します注。
置換して次へ （置換して前へ）	選択している文字列を指定した置換文字列に置換したのち，次（前）の候補を検索して選択状態にします注。
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。

注　指定した文字列が見つからなかった場合，メイン・ウインドウのステータスバー上に“*指定文字列*は見つかりませんでした。”と表示されます。

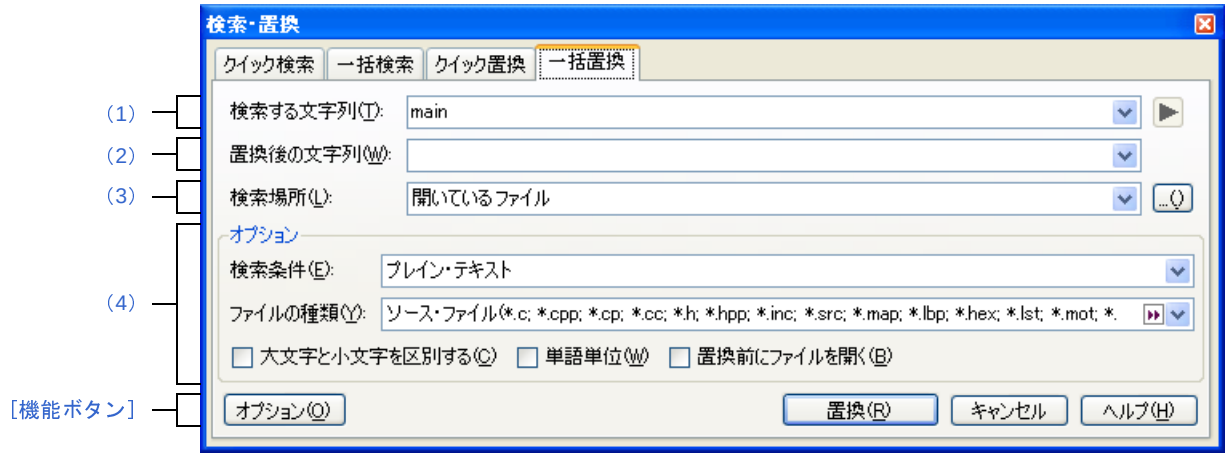
［一括置換］タブ

指定した文字列を一括検索したのち，置換文字に一括置換します。

備考 1. エディタ パネル以外のパネルから検索・置換 ダイアログが呼び出された場合，本タブを選択することができません。

2. ［検索する文字列］エリアの ボタンは，エディタ パネルから本ダイアログをオープンした場合のみ表示されます。

図 A—36　検索・置換 ダイアログ：［一括置換］タブ



ここでは，以下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］

- ［編集］メニュー→［検索 ...］，または［編集］メニュー→［置換 ...］を選択

［各エリアの説明］

（1）［検索する文字列］エリア

検索する文字列を指定します。

テキスト・ボックスに直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

なお，文字列を選択した状態のパネルから本ダイアログをオープンした場合，選択している文字列がデフォルトで表示されます。

また，エディタ パネルから本ダイアログをオープンした場合では，キャレット位置の単語（変数／関数）をデフォルトで表示します。

ボタンを使用することにより，以下のワイルド・カードを選択できます。エディタ パネルから本ダイアログをオープンし，［検索条件］で“ワイルド・カード”が選択されている場合に有効となります。

- # 任意の 1 桁の数字
- * 0 以上の任意の 1 文字
- ? 任意の 1 文字
- [!] 文字セットに含まれない任意の 1 文字
- [] 文字セットに含まれる任意の 1 文字

また，ボタンを使用することにより，以下の正規表現を選択できます。エディタ パネルから本ダイアログをオープンし，［検索条件］で“正規表現”が選択されている場合に有効となります。

- $ 行の末尾
- ( ) グループ キャプチャ
- * 0 以上
- + 1 以上
- . 任意の 1 文字
- [] 文字セットに含まれる任意の 1 文字
- [^] 文字セットに含まれない任意の 1 文字
- ¥ エスケープ文字
- ¥b 単語の境界
- ¥n 改行
- ¥s 空白
- ^ 行の先頭
- | または

備考　正規表現についての詳細は，「付録 E　正規表現の構文」を参照してください。

(2)［置換後の文字列］エリア

置換する文字列を指定します。

テキスト・ボックスに直接入力するか（最大指定文字数：1024 文字），またはドロップダウン・リストより入力履歴項目を選択します（最大履歴数：10 個）。

(3)［検索場所］エリア

検索する場所を指定します。

以下のいずれかをドロップダウン・リストより選択するか，キーボードより直接入力します（最大履歴数：10 個）。

項目	動作
現在のパネル（*パネル名*）	現在のエディタ パネル内で検索します。

項目	動作
開いているファイル	現在オープンしているすべてのエディタ パネル内で検索します。 エディタ パネルでファイルを1つもオープンしていない場合は，本項目は無効となります。
アクティブ・プロジェクト	アクティブ・プロジェクトに含まれているテキスト・ファイル内で検索します。 なお，［ファイルの種類］が指定されている場合は，そのファイルのみを検索対象とします。 ただし，現在のプロジェクトが存在しない場合は，本項目は無効となります。
メイン・プロジェクトおよびサブプロジェクト内	メイン・プロジェクト，またはサブプロジェクトに含まれているテキスト・ファイル内で検索します。 なお，［ファイルの種類］が指定されている場合は，そのファイルのみを検索対象とします。 ただし，現在のプロジェクトが存在しない場合は，本項目は無効となります。
指定されたフォルダ名	直接入力（最大指定文字数：259文字）によりパス指定されたフォルダ（相対パスの場合はプロジェクト・フォルダ相対とします），または本エリア内の［...］ボタンのクリックによってオープンするフォルダの参照 ダイアログにより指定したフォルダ以下のテキスト・ファイル内で検索します。フォルダが指定されていない場合は，プロジェクト・フォルダ名がデフォルトで“()”内に表示されます（プロジェクトが存在しない場合は，現在のユーザのドキュメント・フォルダを対象とします）。 なお，［ファイルの種類］が指定されている場合は，そのファイルのみを検索対象とします。

備考　入力履歴項目として，最大10個までドロップダウン・リストに記録されます。

(4)［オプション］エリア

本エリアは，［オプション］ボタンをクリックすることにより表示されます（デフォルトでは表示されません）。

検索条件として以下のオプションを指定することができます。

(a)［検索条件］

以下のいずれかをドロップダウン・リストより選択します。

項目	動作
プレイン・テキスト	［検索する文字列］エリアに指定した文字列をそのまま検索します。
ワイルド・カード	［検索する文字列］エリアに指定したワイルド・カードを使用して，文字列を検索します。
正規表現[注]	［検索する文字列］エリアに指定した正規表現を使用して，文字列を検索します。

注　本項目は，エディタ パネルにフォーカスがある場合のみ有効となります。

(b)［ファイルの種類］

検索対象とするファイルの種類を指定します。

以下のいずれかをドロップダウン・リストより選択します。

項目	動作
ソース・ファイル (*拡張子群*[注])	検索対象のファイルをソース・ファイル限定とします。
*.txt	検索対象のファイルをテキスト・ファイル限定とします。
.	すべてのファイルを検索対象とします。

注　プロジェクト・ツリー パネルに登録しているソース・ファイルを判別して拡張子を表示します。

なお，ここでは，テキスト・ボックスにファイル名を直接入力することにより，ファイル名を限定して検索対象を指定することもできます（最大指定文字数：1024 文字）。

この場合，ワイルド・カード“*”を使用することができ，また複数のファイル名をセミコロン“;”で区切って指定することも可能です。

備考　入力履歴項目として，最大 10 個までドロップダウン・リストに記録されます。

(c)［大文字と小文字を区別する］

☑	指定した文字列の大文字と小文字を区別して検索します。
☐	指定した文字列の大文字と小文字を区別せず検索します（デフォルト）。

(d)［単語単位］

☑	指定した文字列を単語単位で完全に一致する箇所のみ検索します。
☐	指定した文字列を含む箇所をすべて検索します（デフォルト）。

(e)［置換前にファイルを開く］

☑	文字列を検索／置換するファイルをエディタ パネルで開いてから置換します。
☐	文字列を検索／置換するファイルを開かずに置換します（デフォルト）。

［機能ボタン］

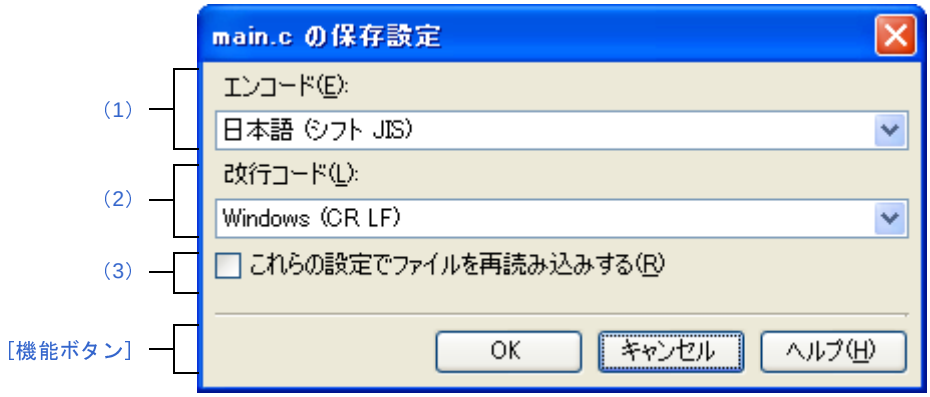
ボタン	機能
オプション	本タブ上の［オプション］エリアの表示／非表示を切り替えます。
置換	指定した検索条件で指定した文字列を一括検索し，該当箇所を指定した置換文字列で一括置換します。
キャンセル	設定を無効とし，本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

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—37　ファイルの保存設定 ダイアログ



ここでは，以下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）。

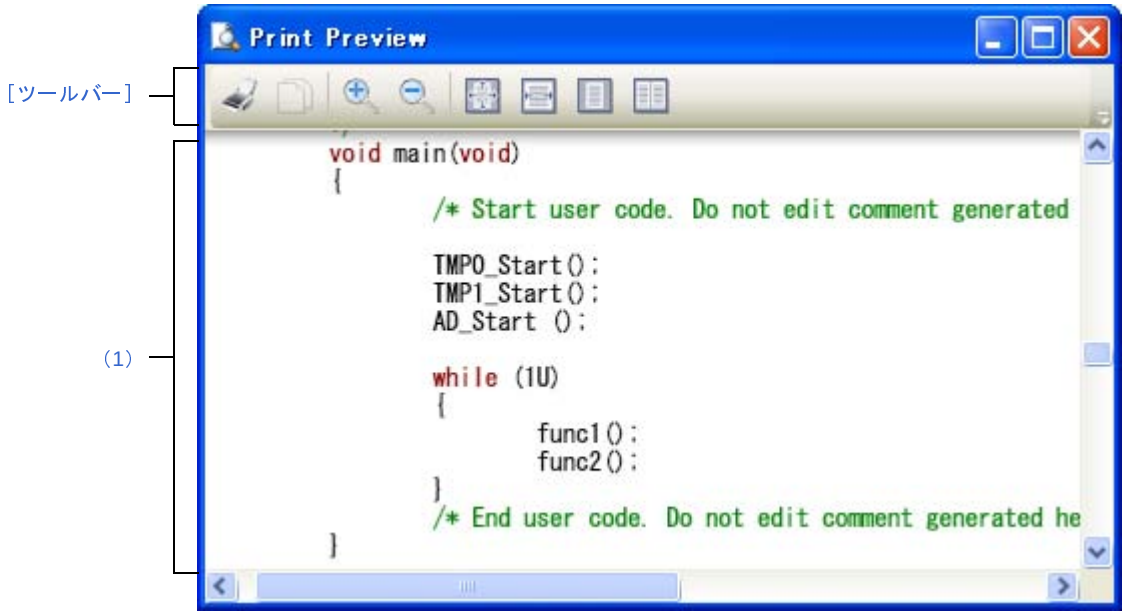
［機能ボタン］

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。

Print Preview ウインドウ

印刷をする前に，ソース・ファイルのプレビューを行います。

図 A—38　Print Preview ウインドウ



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］
- ［ツールバー］
- ［コンテキスト・メニュー］

［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→［印刷プレビュー］を選択

［各エリアの説明］

(1) プレビュー エリア

印刷イメージをプレビュー表示します。

[ツールバー]

	印刷プレビュー表示しているアクティブなエディタ パネルの内容を印刷するために，Windows で用意されている，印刷 ダイアログをオープンします。
	選択範囲をクリップボードにコピーします。
	表示サイズを拡大します。
	表示サイズを縮小します。
	100% の倍率で表示します（デフォルト）。
	ページ幅で表示します。
	1 ページ全体を表示します。
	見開き 2 ページを表示します。

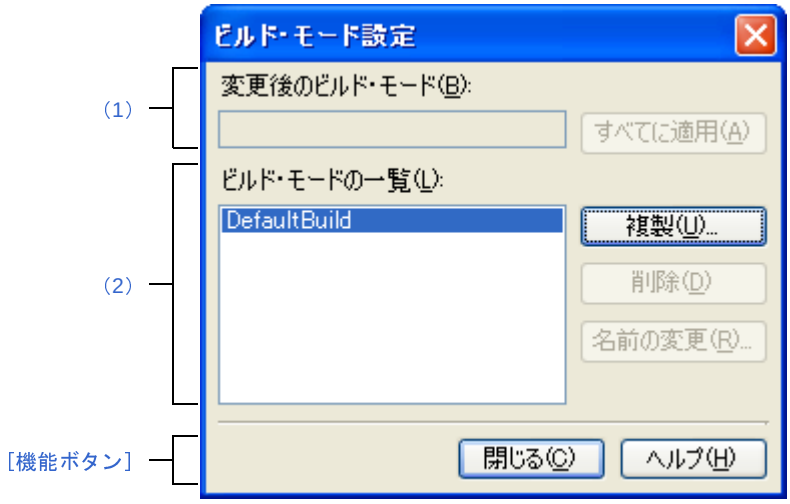
[コンテキスト・メニュー]

ズームの拡大	表示サイズを拡大します。
ズームの縮小	表示サイズを縮小します。

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—39　ビルド・モード設定 ダイアログ



ここでは，以下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。

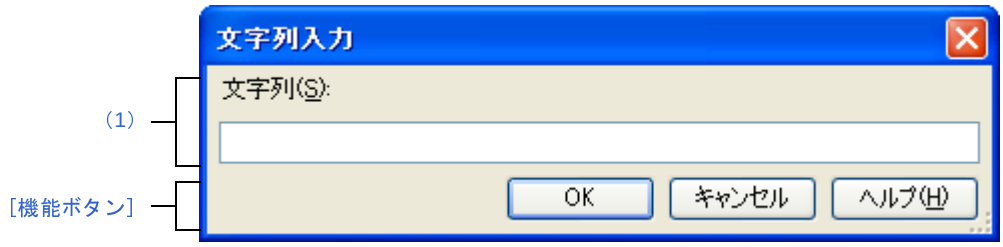
[機能ボタン]

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。

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—40　文字列入力 ダイアログ



ここでは，以下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・モード設定 ダイアログにおいて，［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

［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。

[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。

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—41　バッチ・ビルド ダイアログ

ここでは，以下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・ボックスが未チェックの状態です。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］

- 対象プロジェクトがデバッグ専用プロジェクト以外の場合

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。

- 対象プロジェクトがデバッグ専用プロジェクトの場合

本項目は無効です。

[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。

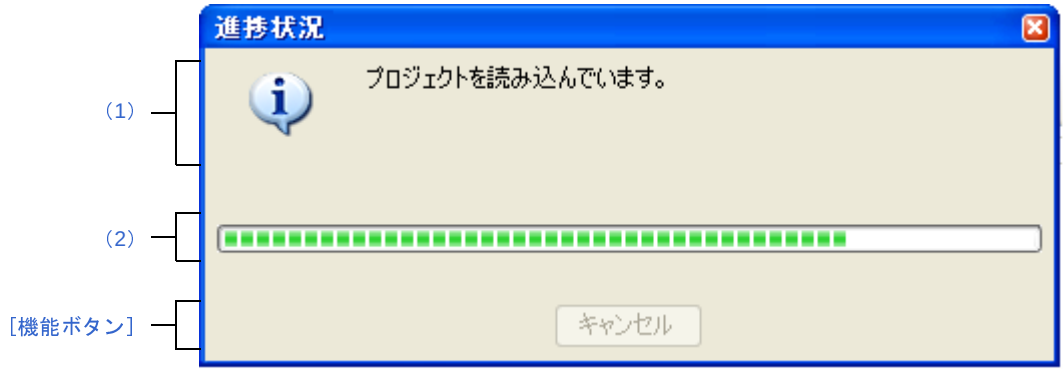
ボタン	機能
リビルド	本ダイアログをクローズし，選択しているプロジェクトをそのビルド・モードでリビルドします。 リビルドの実行結果は，出力 パネルに表示します。 リビルド完了後，ビルド・モードは本ダイアログをオープンする前の設定に戻ります。 なお，本ボタンは，プロジェクトを選択していない場合は無効となります。
クリーン	本ダイアログをクローズし，選択しているプロジェクトのそのビルド・モードでビルドしたファイルを削除します。 クリーンの実行結果は，出力 パネルに表示します。 クリーン完了後，ビルド・モードは本ダイアログをオープンする前の設定に戻ります。 なお，本ボタンは，プロジェクトを選択して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，その進捗状況を表示します。

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—42　処理中表示 ダイアログ



ここでは，以下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。

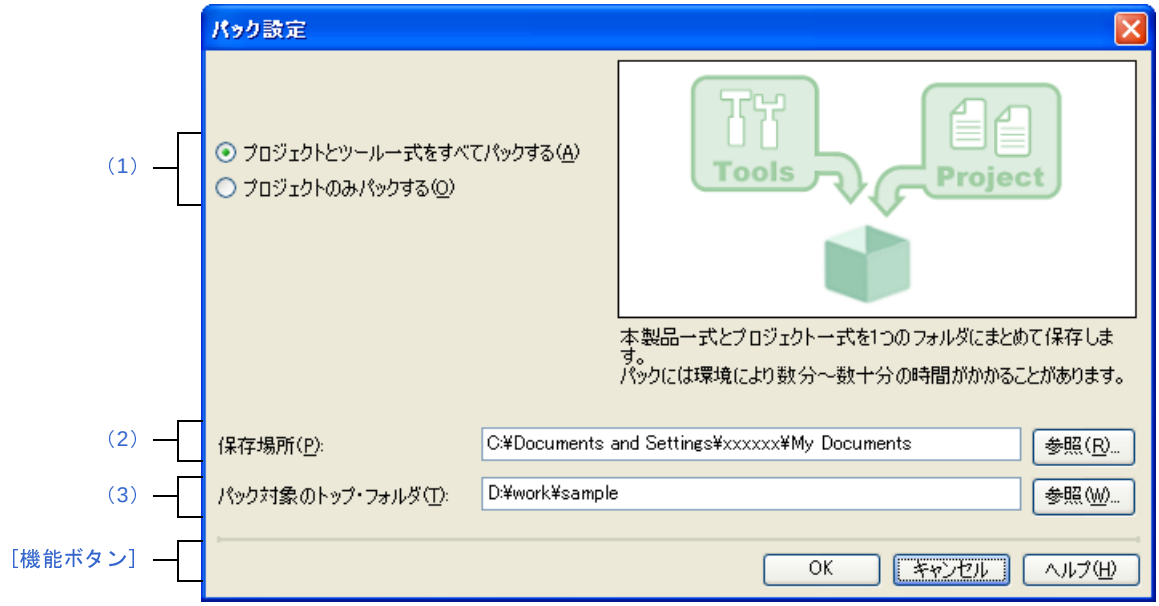
［機能ボタン］

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，このボタンは無効となります。

パック設定 ダイアログ

プロジェクトと本製品をパックして保存する際の設定を行います。

図 A—43　パック設定 ダイアログ



ここでは，以下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，プロジェクト・ノードを選択したのち，コンテキスト・メニュー→［プロジェクトと開発ツールをパックして保存 ...］を選択

［各エリアの説明］

(1) パック方法選択エリア

パック方法を選択します。

プロジェクトとツール一式をすべてパックする	プロジェクトと本製品をパックします（デフォルト）。
プロジェクトのみパックする	プロジェクトのみをパックします。

(2)［保存場所］エリア

プロジェクトと本製品の保存先を指定します。

絶対パスで直接入力，または［参照 ...］ボタンを選択し，フォルダの参照 ダイアログにおいて選択します。

(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・ボックスにパスを表示します。

備考 1. ［保存場所］エリアで指定可能な文字数は，247 文字までです。入力内容が正しくない場合，以下のメッセージがツールチップ表示されます。

メッセージ	説明
フォルダ名が長すぎます。247 文字以内にしてください。	フォルダ名が 247 文字を越えています。
フォルダ名が不正です。文字 (:, *, ?, ", <, >, |) は使用できません。	不正なパスを含むフォルダ名が指定されました。ファイル名，およびフォルダ名に文字（:, *, ?, ", <, >, |）は使用できません。
プロジェクト・フォルダ以下のフォルダは指定できません。	プロジェクト・フォルダ以下のフォルダが指定されました。

2. パス名を表示しきれない場合は，⏩が表示されます。⏩にマウス・カーソルをあわせると，絶対パスをポップアップ表示します。

(3)［パック対象のトップ・フォルダ］エリア

パック対象のトップ・フォルダを指定します。

絶対パスで直接入力，または［参照 ...］ボタンを選択し，フォルダの参照 ダイアログにおいて選択します。

(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・ボックスにパスを表示します。

備考 1. ［パック対象のトップ・フォルダ］エリアで指定可能な文字数は，247 文字までです。入力内容が正しくない場合，以下のメッセージがツールチップ表示されます。

メッセージ	説明
フォルダ名が長すぎます。247 文字以内にしてください。	フォルダ名が 247 文字を越えています。
フォルダ名が不正です。文字 (:, *, ?, ", <, >, |) は使用できません。	不正なパスを含むフォルダ名が指定されました。ファイル名，およびフォルダ名に文字（:, *, ?, ", <, >, |）は使用できません。
プロジェクト・フォルダを含む必要があります。	プロジェクト・フォルダを含まないフォルダが指定されました。

2. パス名を表示しきれない場合は，▶▶が表示されます。▶▶にマウス・カーソルをあわせると，絶対パスをポップアップ表示します。

[機能ボタン]

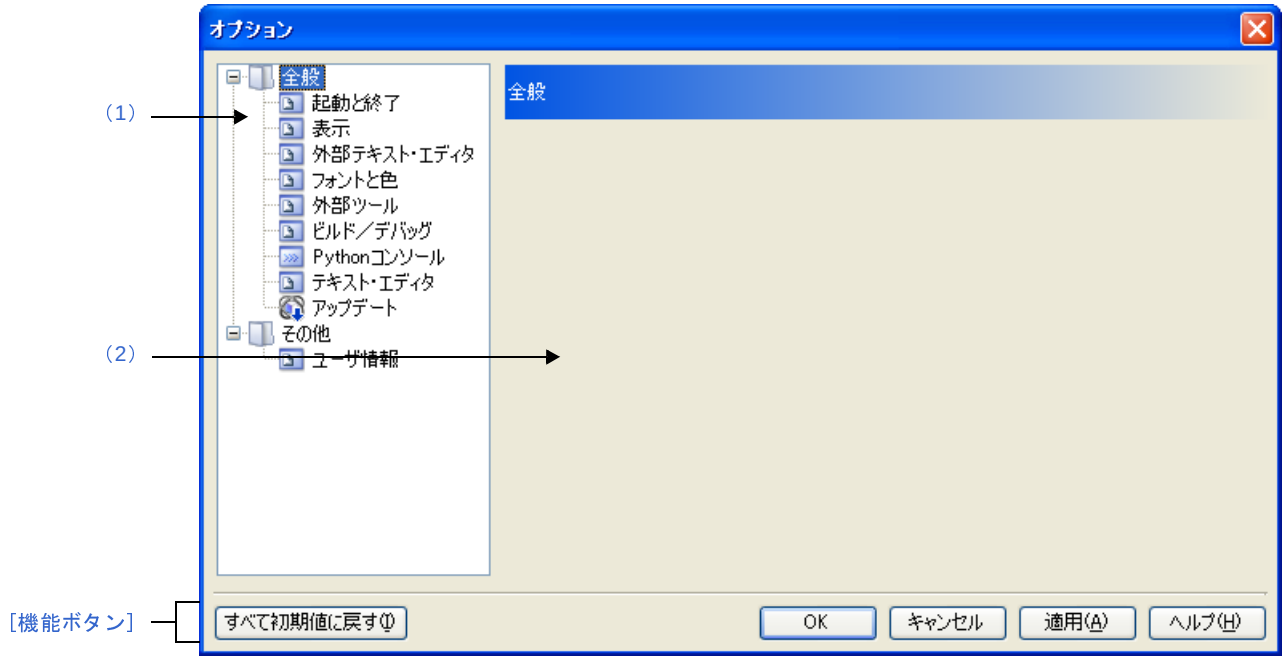
ボタン	機能
OK	本ダイアログをクローズし，設定した条件でプロジェクトと本製品をパックして保存します。
キャンセル	設定した条件を破棄し，本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

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されます。

図 A—44　オプション ダイアログ



ここでは，以下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

設定したい項目を以下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。

カテゴリ	設定内容
［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。

（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。

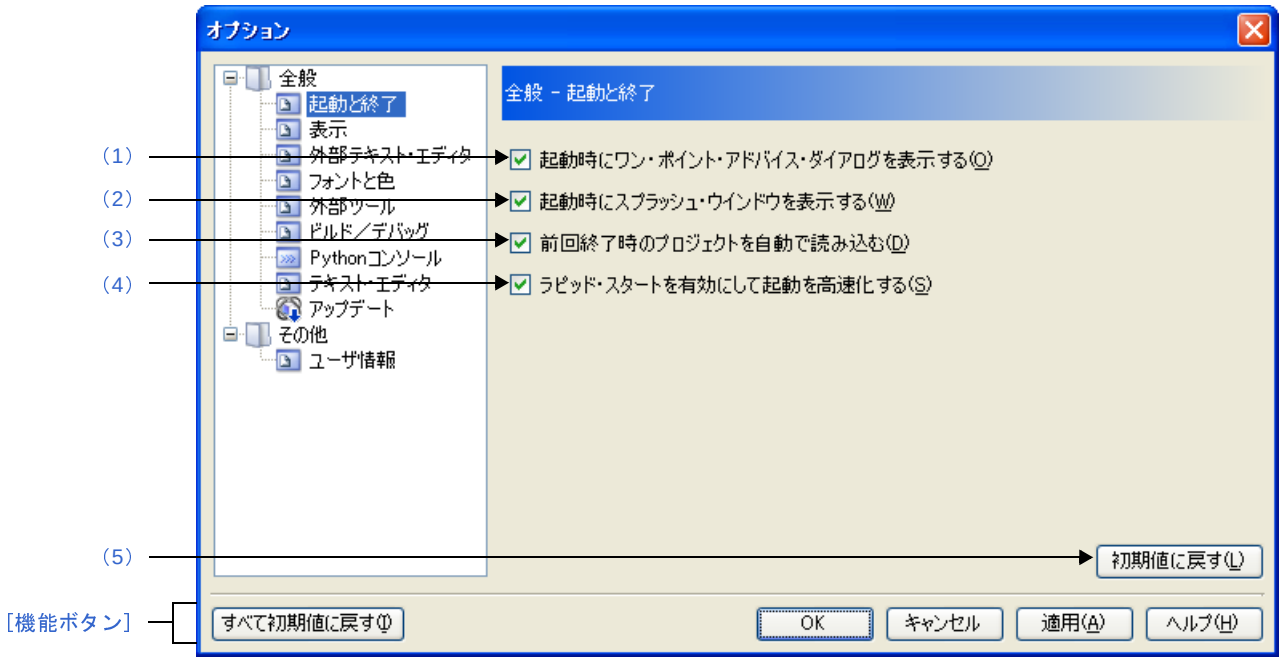
［機能ボタン］

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 - 起動と終了］カテゴリ

全般に関わる設定のうち，起動時／終了時に関連した設定を行います。

図 A—45　オプション ダイアログ（［全般 - 起動と終了］カテゴリ）



ここでは，以下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)［起動時にワンポイント・アドバイス・ダイアログを表示する］

☑	起動時に，ワン・ポイント・アドバイス ダイアログを表示します（デフォルト）。
☐	起動時に，ワン・ポイント・アドバイス ダイアログを表示しません。

(2)［起動時にスプラッシュ・ウインドウを表示する］

☑	起動時に，スプラッシュ ウインドウを表示します（デフォルト）。
☐	起動時に，スプラッシュ ウインドウを表示しません。

(3)［前回終了時のプロジェクトを自動で読み込む］

☑	起動時に，前回終了時のプロジェクトを自動で読み込みます（デフォルト）。
☐	起動時に，前回終了時のプロジェクトを読み込みません。

(4)［ラピッド・スタートを有効にして起動を高速化する］

☑	ラピッド・スタートを有効にし，本製品の起動を高速化します（デフォルト）。
☐	ラピッド・スタートを無効にします。

備考　ラピッド・スタートの詳細については，「2. 5　CubeSuite+ を起動する」を参照してください。

注意　本項目は，インストーラを使用して本製品をインストールした場合のみ有効となります。
パックしたものを使用する場合，本項目は無効となります。

(5) ボタン・エリア

初期値に戻す	現在表示している項目の指定をすべてデフォルトに戻します。

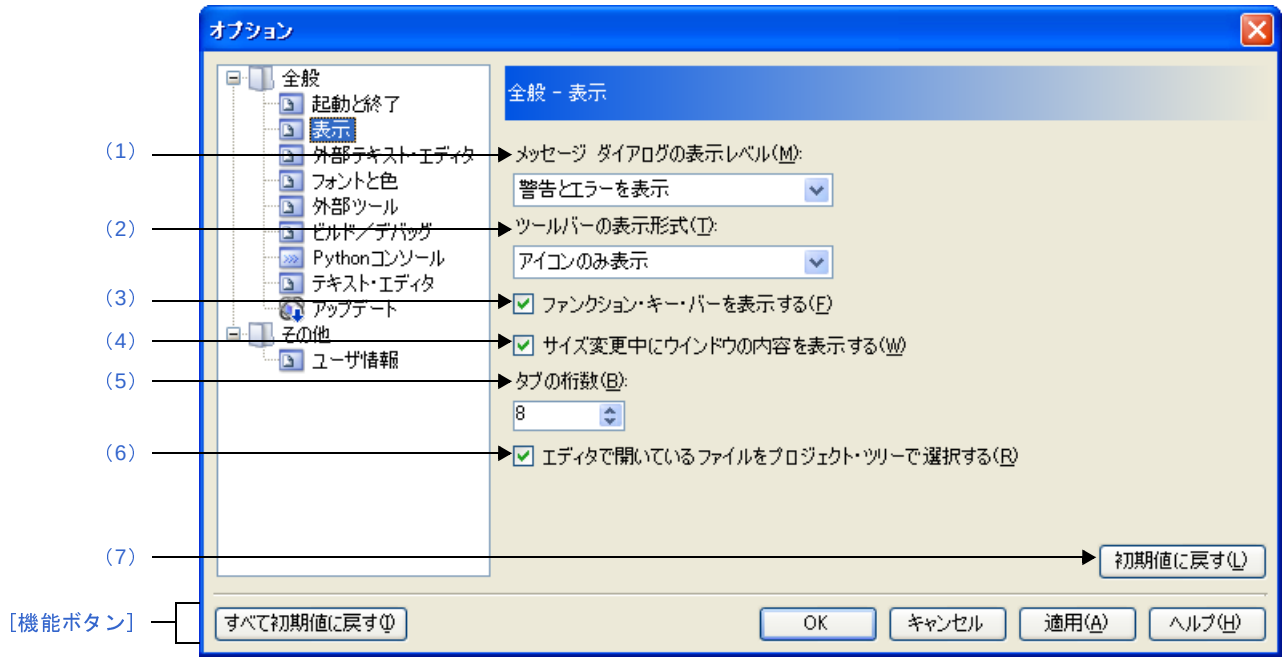
［機能ボタン］

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。

［全般 - 表示］カテゴリ

全般に関わる設定のうち，表示に関連した設定を行います。

図 A—46　オプション ダイアログ（［全般 - 表示］カテゴリ）



ここでは，以下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)［メッセージ ダイアログの表示レベル］

メッセージ ダイアログの表示レベルを，以下の中からドロップダウン・リストにより選択します。

なお，ここでの設定に関わらず，すべてのメッセージは，出力 パネルに表示されます。

情報、警告、エラーを表示	すべてのメッセージをメッセージ ダイアログをオープンして表示します。
警告とエラーを表示	警告メッセージとエラー・メッセージをメッセージ ダイアログをオープンして表示します（デフォルト）。
エラーのみを表示	エラー・メッセージのみをメッセージ ダイアログをオープンして表示します。
致命的なエラーのみを表示	致命的なメッセージのみをメッセージ ダイアログをオープンして表示します。

(2)［ツールバーの表示形式］

各パネルにおけるツールバーの表示形式を，以下の中からドロップダウン・リストにより選択します。

アイコンのみ表示	アイコンのみ表示します（デフォルト）。
アイコンとラベルを表示	アイコンとラベル（文字列）の両方を表示します。

(3)［ファンクション・キー・バーを表示する］

☑	メイン・ウインドウにファンクション・キー・バーを表示します（デフォルト）。
☐	メイン・ウインドウにファンクション・キー・バーを表示しません。

(4)［サイズ変更中にウインドウの内容を表示する］

☑	未表示だった領域の表示を行いながら，サイズの変更をします。
☐	未表示だった領域の表示を行わず，サイズを変更します（デフォルト）。

(5)［タブの桁数］

タブの桁数を指定します。

キーボードより 1 ～ 16 の数値を直接入力するか（半角数字のみ），または ボタンにより数値を選択することで指定します。デフォルトでは 8 が指定されています。

(6)［エディタで開いているファイルをプロジェクト・ツリーで選択する］

☑	エディタ パネルをアクティブにした際に，現在開いているファイルをプロジェクト・ツリー パネル上で選択状態にします（デフォルト）。
☐	エディタ パネルをアクティブにしても，現在開いているファイルをプロジェクト・ツリー パネル上で選択状態にしません。

(7) ボタン・エリア

初期値に戻す	現在表示している項目の指定をすべてデフォルトに戻します。

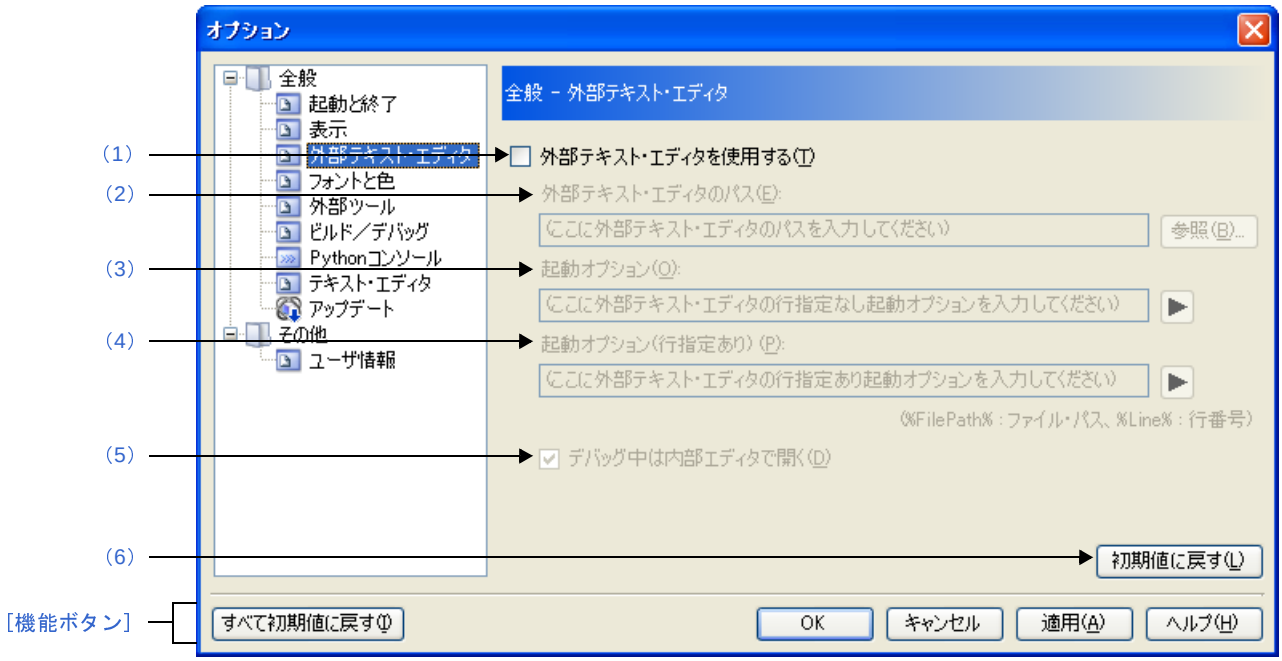
[機能ボタン]

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。

［全般 - 外部テキスト・エディタ］カテゴリ

全般に関わる設定のうち，外部テキスト・エディタに関連した設定を行います。

図 A—47　オプション ダイアログ（［全般 - 外部テキスト・エディタ］カテゴリ）



ここでは，以下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)［外部テキスト・エディタを使用する］

☑	テキスト・エディタとして，外部テキスト・エディタを使用します。
☐	テキスト・エディタとして，エディタ パネルを使用します（デフォルト）。

(2) [外部テキスト・エディタのパス]

使用する外部テキスト・エディタを，絶対パス付きの実行ファイル名でキーボードより直接入力するか（最大指定文字数：259 文字），または［参照 ...］ボタンをクリックしてオープンする，外部テキスト・エディタの選択 ダイアログにより実行ファイル名を選択することにより指定します。

本項目は，［外部テキスト・エディタを使用する］をチェックした場合のみ有効となります。

(3) [起動オプション]

使用する外部テキスト・エディタの起動オプション（行指定なし）を指定します（最大指定文字数：256 文字）。

本項目は，［外部テキスト・エディタを使用する］をチェックした場合のみ有効となります。

▶ボタンから，以下のプレースホルダを指定することも可能です。

なお，複数指定する場合は，半角スペースで区切ります。

%FileName%	表示対象となるファイルのファイル名を外部テキスト・エディタに渡します。
%FilePath%	表示対象となるファイルの絶対パスを外部テキスト・エディタに渡します。
%FileRelativePath%	表示対象となるファイルのプロジェクト・フォルダからの相対パスを外部テキスト・エディタに渡します。
%Line%	キャレット位置（行番号）を外部テキスト・エディタに渡します。

(4) [起動オプション (行指定あり)]

使用する外部テキスト・エディタの起動オプション（行指定あり）を指定します（最大指定文字数：256 文字）。

本項目は，［外部テキスト・エディタを使用する］をチェックした場合のみ有効となります。

▶ボタンから，以下のプレースホルダを指定することも可能です。

なお，複数指定する場合は，半角スペースで区切ります。

%FileName%	表示対象となるファイルのファイル名を外部テキスト・エディタに渡します。
%FilePath%	表示対象となるファイルの絶対パスを外部テキスト・エディタに渡します。
%FileRelativePath%	表示対象となるファイルのプロジェクト・フォルダからの相対パスを外部テキスト・エディタに渡します。
%Line%	キャレット位置（行番号）を外部テキスト・エディタに渡します。

備考　外部テキスト・エディタとして，秀丸エディタ，MIFES for Windows を登録する際の設定例を以下に示します。

図 A—48　外部テキスト・エディタ（秀丸エディタ）の設定例

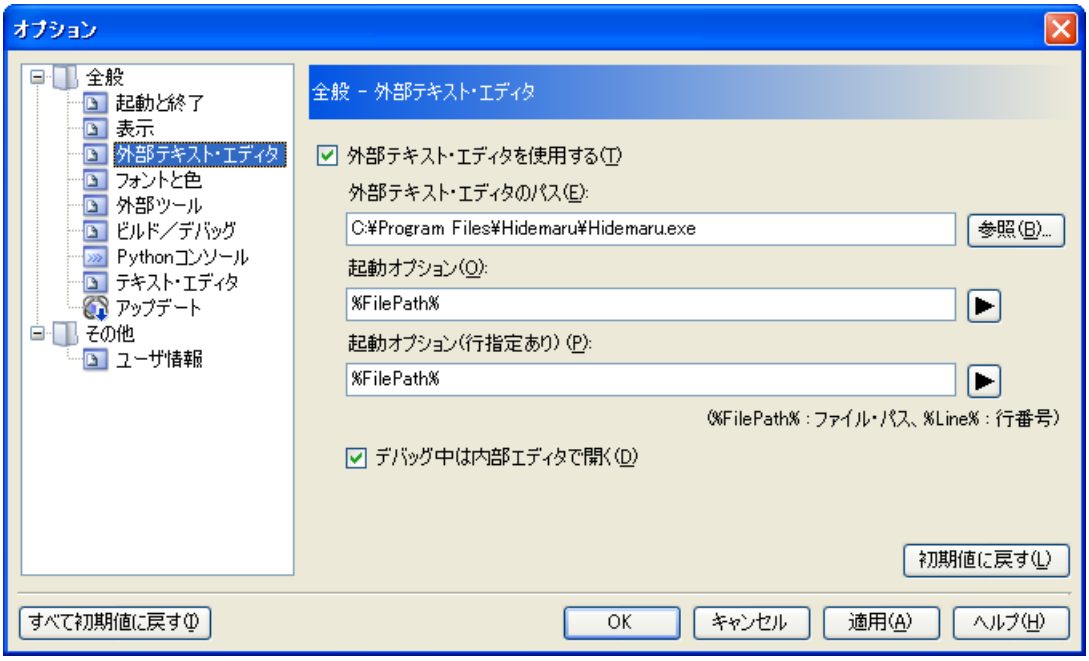
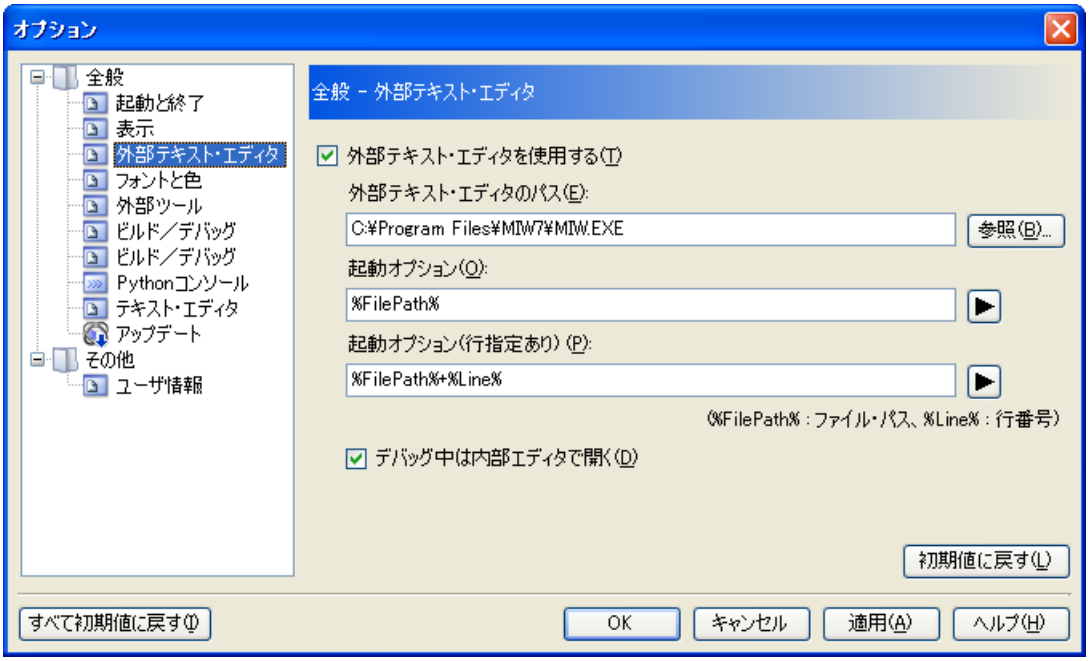


図 A—49　外部テキスト・エディタ（MIFES for Windows Ver.7.0）の設定例



(5) ［デバッグ中は内部エディタで開く］

☑	デバッグ・ツール接続中は外部テキスト・エディタの起動を抑止し，表示対象となるファイルを内部エディタでオープンします（デフォルト）。
☐	表示対象となるファイルを常に外部テキスト・エディタでオープンします。

本項目は，［外部テキスト・エディタを使用する］をチェックした場合のみ有効となります。

(6) ボタン・エリア

初期値に戻す	現在表示している項目の指定をすべてデフォルトに戻します。

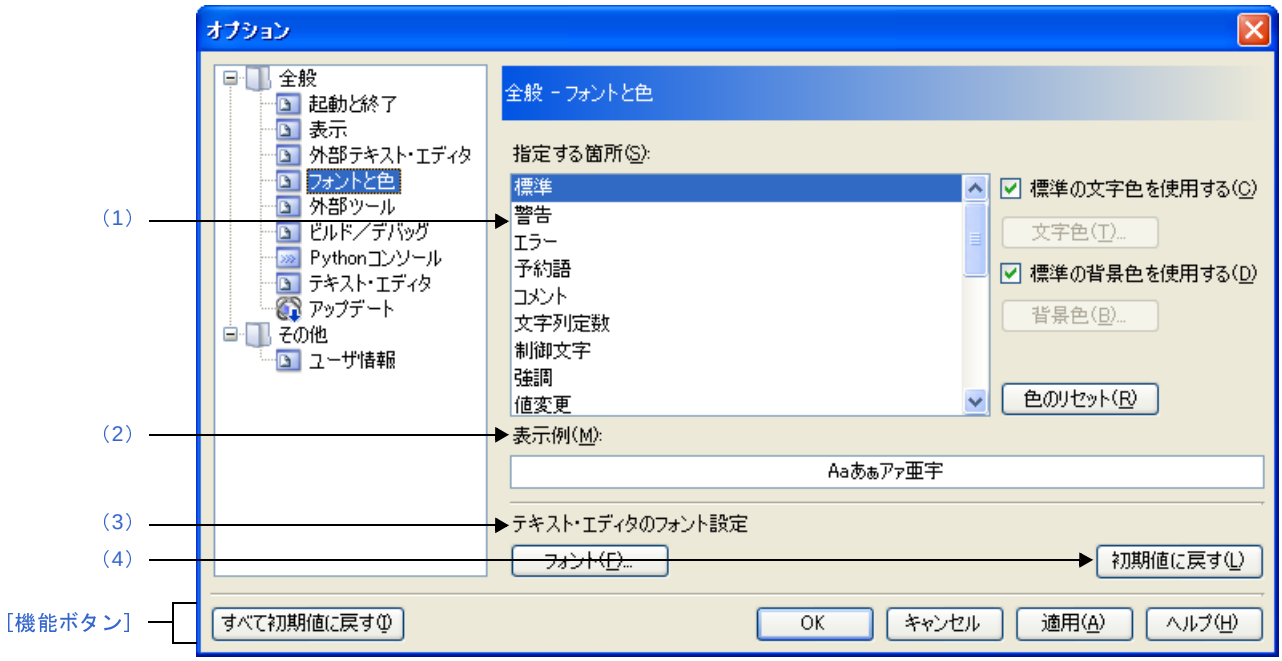
［機能ボタン］

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。

［全般 - フォントと色］カテゴリ

全般に関わる設定のうち，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。

図 A—50　オプション ダイアログ（［全般 - フォントと色］カテゴリ）



ここでは，以下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）色の設定エリア

色の設定を行うエリアです。

（a）［指定する箇所］エリア

表示色を指定したい箇所をリスト一覧から選択します。

表示されるリスト一覧の項目と，デフォルトで設定されている色の関係は以下のとおりです。

項目	表示例			説明
標準注	AaBbCc	文字色	黒	すべてのウインドウ／パネルにおいて，標準となる表示色
		背景色	白	
警告	AaBbCc	文字色	青	出力 パネルにおいて，警告メッセージの表示色，およびプロジェクト・ツリー パネルにおける“警告あり”ファイル名の表示色
		背景色	標準色	
エラー	AaBbCc	文字色	赤	出力 パネルにおいて，エラー・メッセージの表示色，およびプロジェクト・ツリー パネルにおける“エラーあり”ファイル名の表示色
		背景色	薄グレー	
予約語	AaBbCc	文字色	茶	エディタ パネルにおいて，使用するコンパイラ／アセンブラの予約語の表示色
		背景色	標準色	
コメント	AaBbCc	文字色	緑	エディタ パネルにおいて，コメント部（C ソース・ファイルの場合，“/* ～ */”）の表示色
		背景色	標準色	
文字列定数	AaBbCc	文字色	グレー	エディタ パネルにおいて，文字列定数の表示色
		背景色	標準色	
制御文字	AaBbCc	文字色	青緑	出力 パネルにおいて，制御文字の表示色
		背景色	標準色	
強調	AaBbCc	文字色	白	プラグイン製品などにおいて，強調箇所の表示色
		背景色	赤紫	
値変更	AaBbCc	文字色	薄茶	メモリ パネル／ CPU レジスタ パネル／ローカル変数パネル／ SFR パネル／ウォッチ パネルにおいて，プログラムの実行により値が変更した箇所の表示色
		背景色	クリーム	
値編集	AaBbCc	文字色	青	メモリ パネル／ CPU レジスタ パネル／ローカル変数パネル／ SFR パネル／ウォッチ パネルにおいて，ユーザが強制的に値を変更した箇所の表示色
		背景色	標準色	
PC 位置	AaBbCc	文字色	黒	エディタ パネルにおいて，カレント PC 位置のある行の表示色
		背景色	山吹	
ブレークポイント	AaBbCc	文字色	黒	エディタ パネルにおいて，ブレークポイントが設定されている行の表示色
		背景色	サーモンピンク	
リアルタイム更新中	AaBbCc	文字色	ピンク	メモリ パネル／ウォッチ パネルにおいて，リアルタイム表示更新に設定されている領域の表示色
		背景色	標準色	
リード／フェッチ	AaBbCc	文字色	標準色	トレース パネル【IECUBE】【IECUBE2】【シミュレータ】において，リード，またはフェッチされた箇所の表示色
		背景色	薄緑	
ライト	AaBbCc	文字色	標準色	トレース パネル【IECUBE】【IECUBE2】【シミュレータ】において，ライトされた箇所の表示色
		背景色	オレンジ	
リード＆ライト	AaBbCc	文字色	標標準色	トレース パネル【IECUBE】【IECUBE2】【シミュレータ】において，リードとライトされた箇所の表示色
		背景色	薄青	

項目	表示例			説明
ロスト	AaBbCc	文字色	白	メモリ パネルにおいて，デバッグ・ツールから取得した値が正しくない箇所の表示色
		背景色	ライトグレー	
カバレッジ 100%	AaBbCc	文字色	標準色	エディタ パネル／逆アセンブル パネルにおいて，コード・カバレッジ率 100% の行の表示色
		背景色	ライトグリーン	
カバレッジ 1 ～ 99%	AaBbCc	文字色	標準色	エディタ パネル／逆アセンブル パネルにおいて，コード・カバレッジ率 1 ～ 99% の行の表示色
		背景色	ライトピンク	
カバレッジ 0%	AaBbCc	文字色	標準色	エディタ パネル／逆アセンブル パネルにおいて，コード・カバレッジ率 0%（未実行）の行の表示色
		背景色	ライトグレー	
無効	AaBbCc	文字色	グレー	メモリ パネルにおいて，メモリ・マッピングされていない領域，およびプロジェクト・ツリー パネル上で実際存在しないファイル名の表示色
		背景色	標準色	

注　［標準］の文字色／背景色は，使用するホスト・マシンにおける Windows の設定に依存します。ここでは，Windows のデフォルト設定である“文字色：黒”，“背景色：白”を表記しています。

(b)［標準の文字色を使用する］

☑	［指定する箇所］エリアで選択している項目を，標準の文字色を使用して表示します。
☐	［指定する箇所］エリアで選択している項目の文字色を，任意に指定します。 ［文字色 ...］ボタンが有効となります。

(c)［標準の背景色を使用する］

☑	［指定する箇所］エリアで選択している項目を，標準の背景色を使用して表示します。
☐	［指定する箇所］エリアで選択している項目の背景色を，任意に指定します。 ［背景色 ...］ボタンが有効となります。

(d) ボタン

ボタン	機能
文字色 ...	色の設定 ダイアログがオープンし，［指定する箇所］エリアで選択している項目の文字色を指定します。 ただし，［標準の文字色を使用する］をチェックしている場合は，無効となります。
背景色 ...	色の設定 ダイアログがオープンし，［指定する箇所］エリアで選択している項目の背景色を指定します。 ただし，［標準の背景色を使用する］をチェックしている場合は，無効となります。

ボタン	機能
色のリセット	［指定する箇所］エリアで選択している項目の色情報をリセットし，デフォルトの設定に戻します。

図 A—51　色の設定 ダイアログ

(2)［表示例］エリア

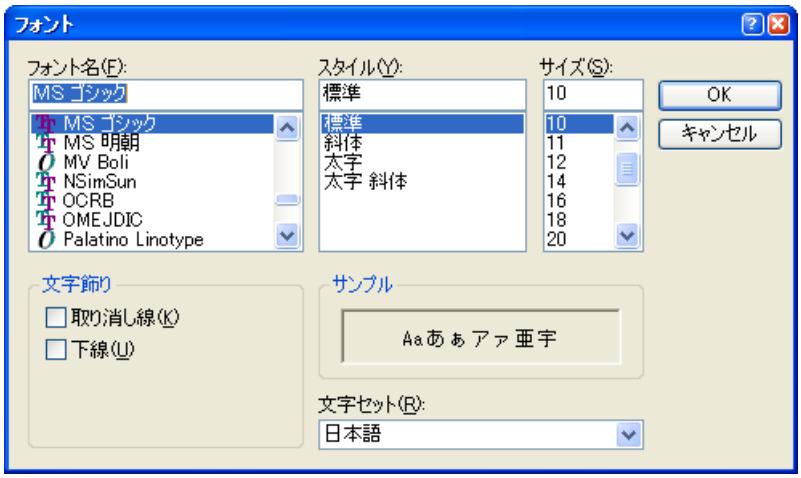
色の設定エリア，および［テキスト・エディタのフォント設定］エリアで指定した色とフォントの表示例を示します。

デフォルトでは，文字列“Aa あぁアァ亜宇”を表示しますが，テキスト・ボックスに任意の文字列を直接入力することができます。

(3)［テキスト・エディタのフォント設定］エリア

［フォント ...］ボタンをクリックすることにより，以下のフォント ダイアログをオープンし，使用するテキスト・エディタで使用するフォントを設定します。

図 A—52　フォント ダイアログ



(4) ボタン・エリア

初期値に戻す	現在表示している項目の指定をすべてデフォルトに戻します。

[機能ボタン]

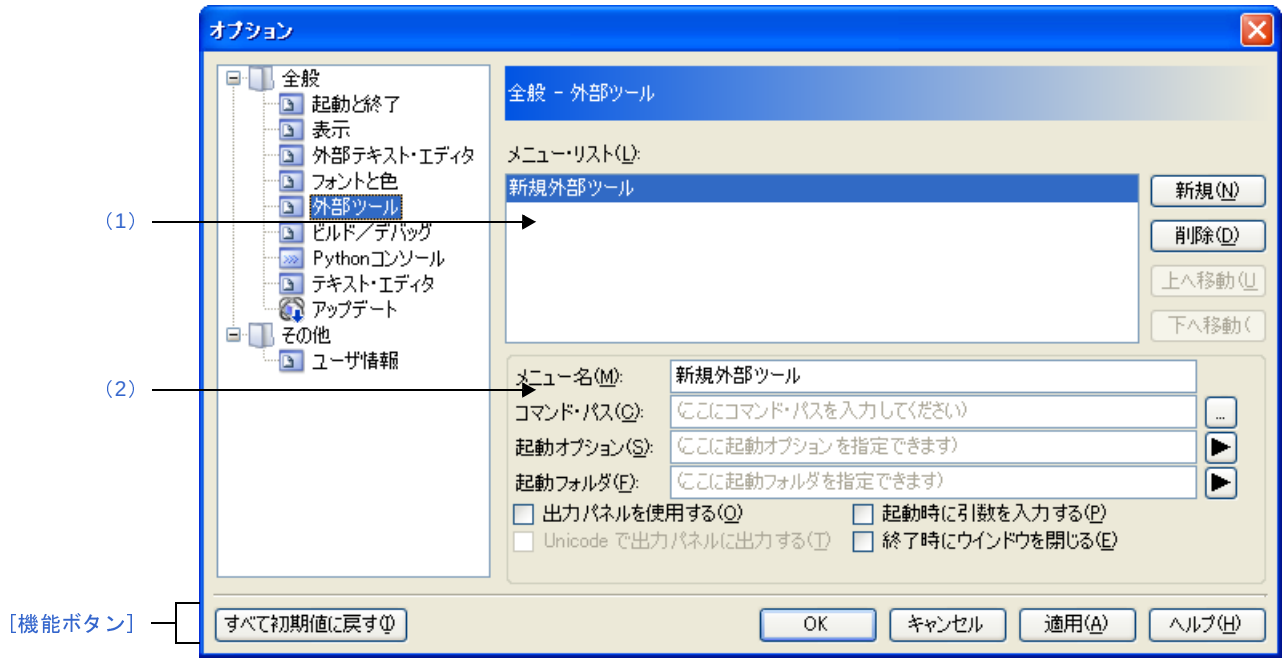
ボタン	機能
すべて初期値に戻す	本ダイアログのすべての設定項目をデフォルトの状態に戻します。 ただし，[全般 - 外部ツール] カテゴリでは，新規登録した内容の削除は行いません。
OK	変更した設定内容を適用し，本ダイアログをクローズします。
キャンセル	変更した設定内容を無効とし，本ダイアログをクローズします。
適用	変更した設定内容を適用します（本ダイアログをクローズしません）。
ヘルプ	本ダイアログのヘルプを表示します。

［全般 - 外部ツール］カテゴリ

全般に関わる設定のうち，CubeSuite+ から直接起動を可能とする外部ツールの登録，およびその際の外部ツールへの設定を行います。

ここで外部ツールを登録することにより，［ツール］メニューから直接起動することができます。

図 A—53　オプション ダイアログ（［全般 - 外部ツール］カテゴリ）



ここでは，以下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）登録済み外部ツール一覧エリア

（a）［メニュー・リスト］エリア

本ダイアログで登録した外部ツール起動用のメニュー項目（［ツール］メニューに追加される項目）の一覧を表示します。

新規に外部ツールを登録する場合は，［新規］ボタンをクリックします。

［新規］ボタンをクリックすることにより，一覧の最下段に“新規外部ツール”項目が追加され，その状態で，新規登録エリアにおいて新規に追加する外部ツールの詳細情報を設定します。

外部ツールは，最大 8 個まで登録することができます。

また，すでに登録されている外部ツールの設定を変更する場合は，対象となる外部ツール名を選択し，変更したい項目の設定を編集します。

なお，ここで表示される一覧の順番は，メニューを表示する際の順番に反映されます。

(b) ボタン

新規	新規に外部ツールを登録するために，一覧の最下段に“新規外部ツール”項目を追加します。 ただし，すでに外部ツールが 8 個登録されている場合，本ボタンは無効となります。
削除	一覧で選択している外部ツールを登録から削除します。
上へ移動	一覧で選択している外部ツールを 1 つ上に移動します。
下へ移動	一覧で選択している外部ツールを 1 つ下に移動します。

(2) 新規登録エリア

新規に外部ツールを登録する際の，詳細情報の設定を行うエリアです。

外部ツールは最大 8 個まで登録することができます。

(a)［メニュー名］

登録する外部ツール起動用のメニュー名を指定します（最大指定文字数：100 文字）。

ここで指定したメニュー名が，外部ツールの起動用として［ツール］メニューに追加されます。

メニュー名の文字列入力が完了すると（フォーカスが外れると），自動的に一覧の“新規外部ツール”部の文字列に反映されます。

(b)［コマンド・パス］

登録する外部ツールを，絶対パス付きの実行ファイル名でキーボードより直接入力するか（最大指定文字数：259 文字），または［...］ボタンをクリックしてオープンする，プログラムを選択 ダイアログにより実行ファイル名を選択することにより指定します（選択した実行ファイルの存在するフォルダの絶対パスが［起動フォルダ］に自動的に設定されます）。

(c)［起動オプション］

登録する外部ツールの起動オプションを指定します（最大指定文字数：256 文字）。

ボタンから，以下のプレースホルダを指定することも可能です。

なお，複数指定する場合は，半角スペースで区切ります。

%FileName%	現在，プロジェクト・ツリー パネルで選択しているファイルのファイル名を外部ツールに渡します。
%FilePath%	現在，プロジェクト・ツリー パネルで選択しているファイルの絶対パスを外部ツールに渡します。

%FileRelativePath%	現在，プロジェクト・ツリー パネルで選択しているファイルのプロジェクト・フォルダからの絶対パスを外部ツールに渡します。
%Line%	現在，プロジェクト・ツリー パネルで選択しているファイルをエディタ パネルで編集している場合，キャレットのある行を外部ツールに渡します（エディタ パネルにフォーカスがない場合は“1”を返します）。
%OutputFile%	現在，プロジェクト・ツリー パネルで選択している項目のプロジェクトをビルドした際に出力するモジュールのファイル名（絶対パス付き）を外部ツールに渡します（非選択の場合は何も渡しません）。

(d)［起動フォルダ］

登録する外部ツールを起動する際のフォルダを絶対パスで指定します（最大指定文字数：247 文字）。

ボタンから，以下のプレースホルダを指定することも可能です。

なお，複数指定する場合は，半角スペースで区切ります。

%ActiveProjectDir%	現在，プロジェクト・ツリー パネルで選択している項目のアクティブ・プロジェクト・フォルダの絶対パスを外部ツールに渡します（アクティブ・プロジェクトが存在しない場合は何も渡しません）。
%FileDir%	現在，プロジェクト・ツリー パネルで選択しているファイルが存在するフォルダの絶対パスを外部ツールに渡します。
%MainProjectDir%	現在，プロジェクト・ツリー パネルで選択している項目のメイン・プロジェクト・フォルダの絶対パスを外部ツールに渡します（メイン・プロジェクトが存在しない場合は何も渡しません）。
%MicomToolPath%	本製品のインストール・フォルダの絶対パスを外部ツールに渡します。
%OutputDir%	現在，プロジェクト・ツリー パネルで選択している項目のプロジェクトをビルドした際に出力するモジュールのフォルダ名（絶対パス付き）を渡します（非選択の場合は何も渡しません）。
%ProjectDir%	現在，プロジェクト・ツリー パネルで選択している項目のプロジェクト・フォルダの絶対パスを外部ツールに渡します。
%TempDir%	テンポラリ・フォルダの絶対パスを外部ツールに渡します。
%WinDir%	Windows システム・フォルダの絶対パスを外部ツールに渡します。

(e)［出力 パネルを使用する］

☑	登録する外部ツールの標準出力と標準エラー出力に出力されたメッセージを，出力 パネルに表示します。 なお，メッセージは，出力 パネルの［*ツール名*の出力メッセージ］タブ上に出力されます（“*ツール名*”には［メニュー名］で指定した文字列が反映されます）。
☐	登録する外部ツールの標準出力と標準エラー出力に出力されたメッセージを，Windows のコマンド・プロンプトを使用して出力します（デフォルト）。

(f)［起動時に引数を入力する］

☑	文字列入力 ダイアログがオープンし，登録する外部ツール起動時の引数を指定することができます。
☐	登録する外部ツール起動時に引数を指定しません（デフォルト）。

(g)［Unicode で出力 パネルに出力する］

本項目は，［出力 パネルを使用する］をチェックしている場合にのみ有効となります。

☑	登録する外部ツールの標準出力と標準エラー出力に出力されたメッセージの文字コードを，Unicode として出力 パネルに表示します。
☐	登録する外部ツールの標準出力と標準エラー出力に出力されたメッセージの文字コードを，ASCII として出力 パネルに表示します（デフォルト）。

(h)［終了時にウインドウを閉じる］

本項目は，［出力 パネルを使用する］をチェックしている場合は無効となります。

☑	登録する外部ツールが終了する際に，Windows のコマンド・プロンプトをクローズします。
☐	登録する外部ツールが終了する際に，Windows のコマンド・プロンプトをオープンしたままの状態にします（デフォルト）。

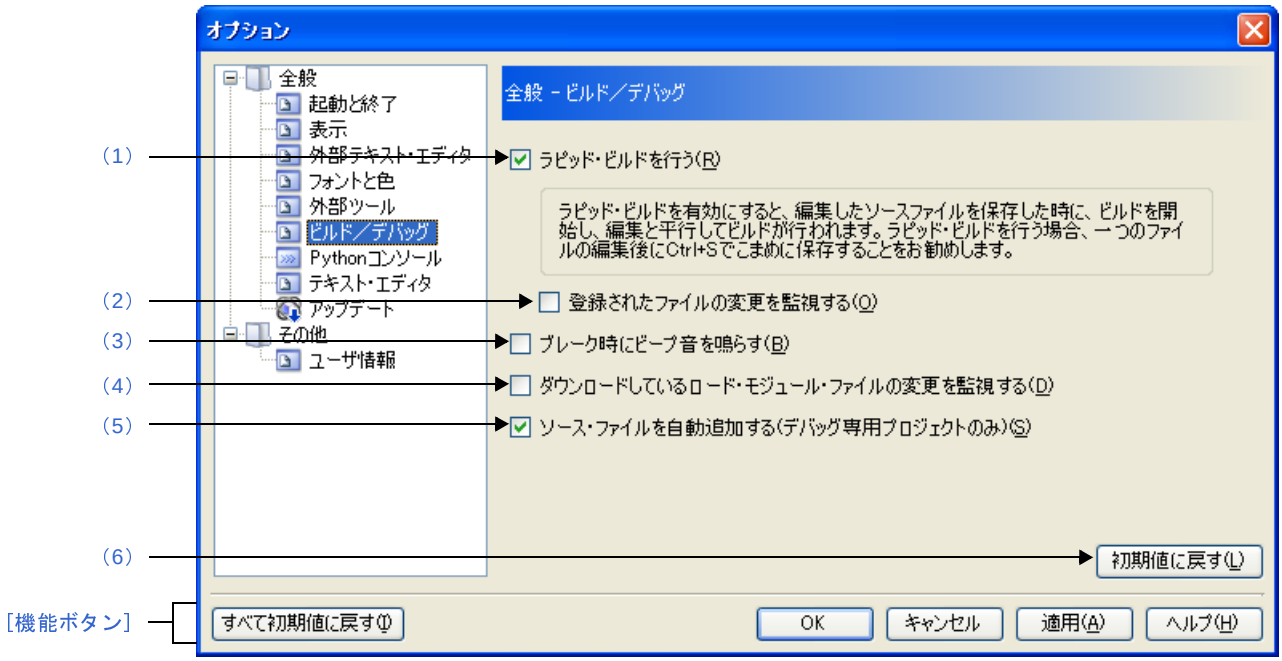
［機能ボタン］

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—54　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，以下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。

ラピッド・ビルド機能についての詳細は，以下を参照してください。

- デバッグ専用プロジェクト以外を対象とする場合：　「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」
- デバッグ専用プロジェクトを対象とする場合：　「F. 5. 4　ラピッド・ビルドを実行する」

(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され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／保存されたときに，ラピッド・ビルドを開始します。
☐	プロジェクトに登録され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／保存されたときに，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) ［ブレーク時にビープ音を鳴らす］

☑	プログラムの実行が，ブレーク・イベント（ハードウエア・ブレーク／ソフトウエア・ブレーク）により停止した際，ビープ音を鳴らします。
☐	プログラムの実行が，ブレーク・イベント（ハードウエア・ブレーク／ソフトウエア・ブレーク）により停止した際，ビープ音を鳴らしません（デフォルト）。

(4) ［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）。

(5)［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。

(6) 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。

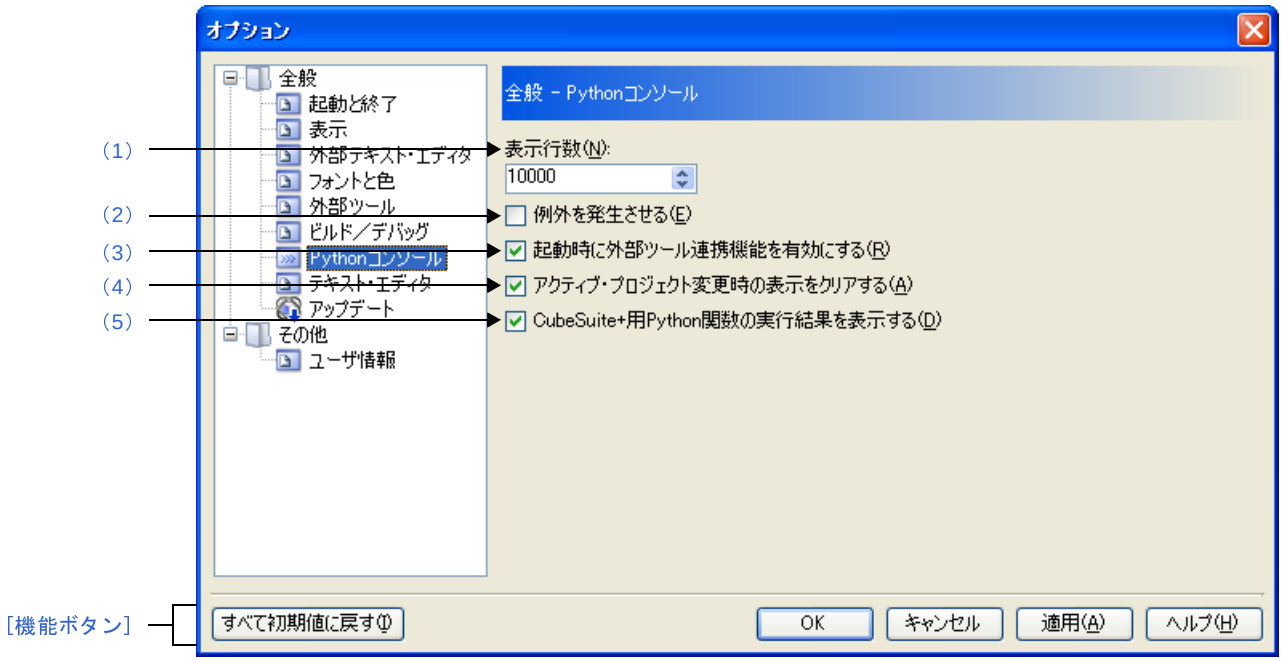
［機能ボタン］

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。

［全般 - Python コンソール］カテゴリ

全般にかかわる設定のうち，Python コンソールに関連した設定を行います。

図 A—55　オプション ダイアログ（［全般 - Python コンソール］カテゴリ）



ここでは，以下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）［表示桁数］

Python コンソールの表示桁数を指定します。

キーボードより 5000 ～ 100000 の数値を直接入力するか（半角数字のみ），または ボタンにより数値を選択することで指定します。デフォルトでは 10000 が指定されています。

備考　本項目は，CubeSuite+ Python プロパティ「common.ViewLine」でも設定／参照することができます。

(2)［例外を発生させる］

☑	Python 関数の実行時に例外を発生させます。
☐	Python 関数の実行時に例外を発生させません（デフォルト）。

備考　本項目は，CubeSuite+ Python プロパティ「common.ThrowExcept」でも設定／参照することができます。

(3)［起動時に外部ツール連携機能を有効にする］

☑	Python コンソールの起動時に外部ツールと連携する機能を有効にします（デフォルト）。
☐	Python コンソールの起動時に外部ツールと連携する機能を無効にします。

備考　本項目は，CubeSuite+ Python プロパティ「common.UsedRemoting」でも設定／参照することができます。

(4)［アクティブ・プロジェクト変更時の表示をクリアする］

☑	アクティブ・プロジェクト変更時に Python コンソールの表示をクリアします（デフォルト）。
☐	アクティブ・プロジェクト変更時に Python コンソールの表示をクリアしません。

備考　本項目は，CubeSuite+ Python プロパティ「common.ConsoleClear」でも設定／参照することができます。

(5)［CubeSuite+ 用 Python 関数の実行結果を表示する］

☑	CubeSuite+ 用 Python 関数の実行結果を Python コンソールに表示します（デフォルト）。
☐	CubeSuite+ 用 Python 関数の実行結果を Python コンソールに表示しません。

備考　本項目は，CubeSuite+ Python プロパティ「common.ViewOutput」でも設定／参照することができます。

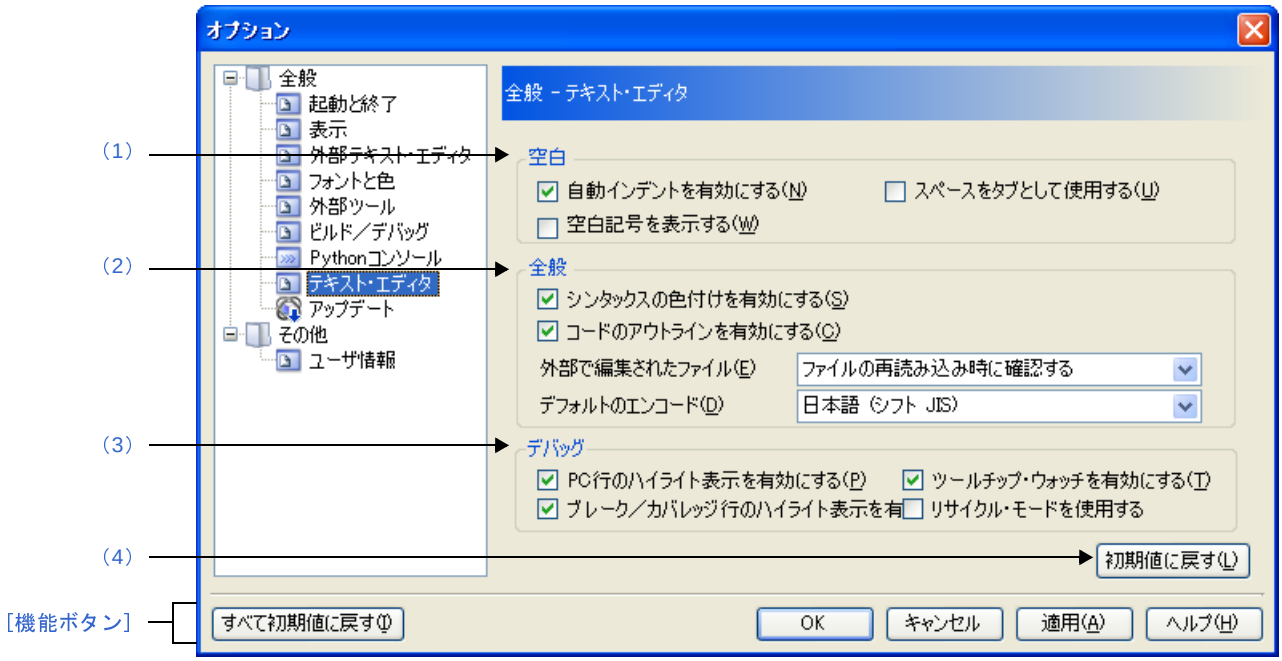
［機能ボタン］

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。

［全般 - テキスト・エディタ］カテゴリ

全般に関わる設定のうち，テキスト・エディタに関連した設定を行います。

図 A—56　オプション ダイアログ（［全般 - テキスト・エディタ］カテゴリ）



ここでは，以下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)［空白］

テキスト・エディタにおいて，空白に関連した設定を行います。

(a)［自動インデントを有効にする］

☑	テキスト・エディタにおいて，入力したコード（関数内／外やループなど）に応じて，自動的にインデントを挿入します（デフォルト）。
☐	テキスト・エディタにおいて，インデントを挿入しません。

(b)［スペースをタブとして使用する］

☑	テキスト・エディタにおいて，［Tab］キーを押下した場合，タブの代わりにタブ・サイズに指定した数のスペースを入力します。
☐	テキスト・エディタにおいて，［Tab］キーを押下した場合，タブを入力します（デフォルト）。

(c)［空白記号を表示する］

☑	テキスト・エディタにおいて，空白記号（タブ／スペース）を表示します。
☐	テキスト・エディタにおいて，空白記号（タブ／スペース）を表示しません（デフォルト）。

(2)［全般］

テキスト・エディタにおいて，全般的な設定を行います。

(a)［シンタックスの色付けを有効にする］

☑	サポートしているファイル・タイプのテキスト・エディタにおいて，キーワードの色付けを有効にします（デフォルト）。
☐	テキスト・エディタにおいて，キーワードの色付けを行いません。

(b)［コードのアウトラインを有効にする］

本項目は，［シンタックスの色付けを有効にする］をチェックした場合のみ有効となります。

☑	テキスト・エディタにおいて，コードのアウトライン機能を有効にします。 アウトライン機能をサポートしているファイル・タイプに対して，コード・ブロックの展開と折りたたみをできるようにします（デフォルト）。
☐	テキスト・エディタにおいて，アウトライン機能を使用しません。

(c)［外部で編集されたファイル］

CubeSuite+ の外部で編集されたソース・ファイルに対する操作を指定します。

ファイルの再読み込み時に確認する	テキスト・エディタで開いているファイルが CubeSuite+ の外部で編集された場合，そのファイルを再読み込み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（デフォルト）。 メッセージ ダイアログで［はい］をクリックすると，テキスト・エディタで開いているファイルをディスクから再読み込みします。現在の編集内容はすべて失われます。 メッセージ ダイアログで［いいえ］をクリックすると，そのファイルへの外部の編集は無視し，テキスト・エディタで開いているファイルの編集内容は変更しません。
常にファイルを再読み込みする	外部で編集されたファイルを自動的に再読み込みします。
ファイルを再読み込みしない	外部の編集は無視し，ファイルを再読み込みしません。

(d)［デフォルトのエンコード］

エディタ パネル内で新規ファイルを作成する場合に必要なデフォルトのファイル・エンコーディング用のオプションです。また，開こうとしているファイルのエンコーディングが検出できないタイプまたは現在エディタ パネルでサポートされていないタイプである場合にも，このオプションを使用すると便利です。

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，次の順番で表示されます。

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されません。

- *現在の OS の既定のエンコード名*（デフォルト）
- *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）*

(3)［デバッグ］

テキスト・エディタにおいて，デバッグに関連した設定を行います。

(a)［PC 行のハイライト表示を有効にする］

☑	テキスト・エディタにおいて，カレント PC 位置の行をハイライト表示します（デフォルト）。
☐	テキスト・エディタにおいて，カレント PC 位置の行をハイライト表示しません。

(b)［ツールチップ・ウォッチを有効にする］

☑	テキスト・エディタにおいて，変数の上にマウス・カーソルを重ねることにより，ツールチップに変数の値を表示します（デフォルト）。
☐	テキスト・エディタにおいて，ツールチップに変数の値を表示しません。

(c)［ブレーク／カバレッジ行のハイライト表示を有効にする］

☑	テキスト・エディタにおいて，ブレーク／カバレッジ行をハイライト表示します（デフォルト）。
☐	テキスト・エディタにおいて，ブレーク／カバレッジ行をハイライト表示しません。

(4) ボタン・エリア

初期値に戻す	現在表示している項目の指定をすべてデフォルトに戻します。

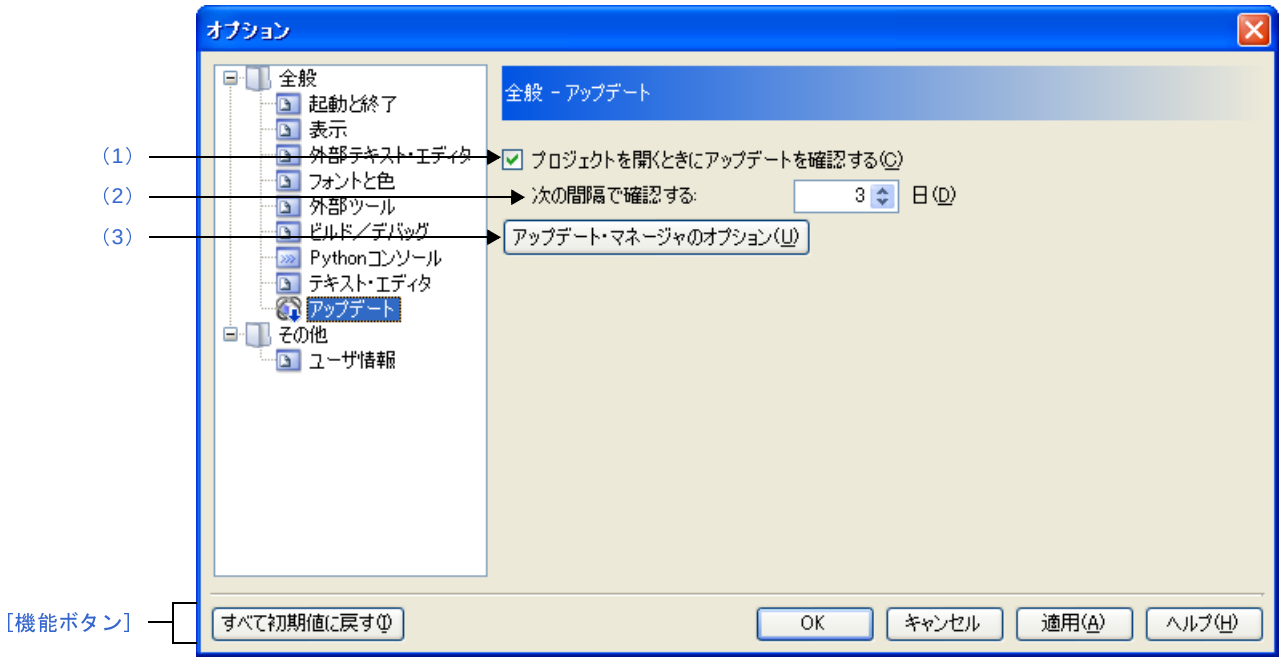
[機能ボタン]

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。

［全般 - アップデート］カテゴリ

全般にかかわる設定のうち，アップデートに関連した設定を行います。

図 A—57　オプション ダイアログ（［全般 - アップデート］カテゴリ）



ここでは，以下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)［プロジェクトを開くときにアップデートを確認する］

☑	プロジェクトを開くタイミングで自動的にアップデートを確認します。
☐	プロジェクトを開くタイミングで自動的にアップデートを確認しません。

(2)［次の間隔で確認する］

アップデートの確認を行う間隔を指定します。

キーボードより 0 ～ 99 の数値を直接入力するか（半角数字のみ），または ボタンにより数値を選択することで指定します。

(3)［アップデート・マネージャのオプション］ボタン

アップデート・マネージャのオプション ダイアログを表示します。

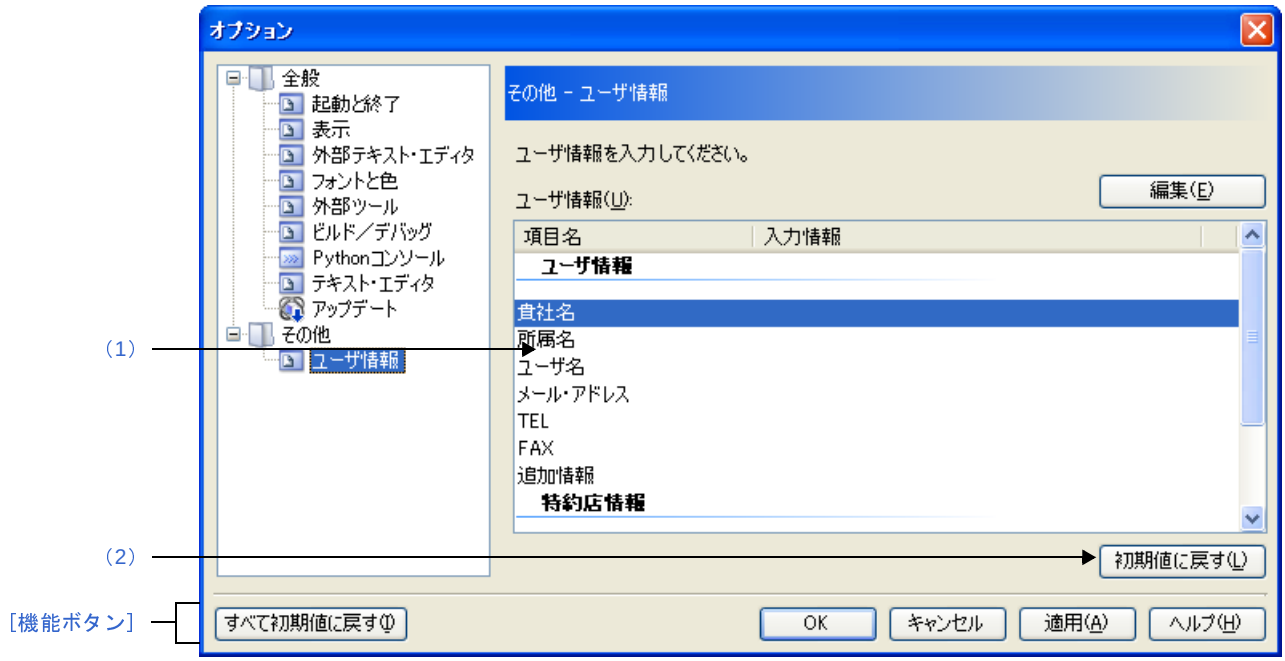
［機能ボタン］

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。

［その他 - ユーザ情報］カテゴリ

その他の設定のうち，ユーザ情報に関連した設定を行います。

図 A—58　オプション ダイアログ（［その他 - ユーザ情報］カテゴリ）



ここでは，以下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）情報エリア

（a）［ユーザ情報］エリア

ユーザ情報となる項目を一覧で表示します。

［入力情報］欄の内容は編集可能です。編集する場合は，編集したい項目名を選択したのち，［編集］ボタンをクリックし，テキスト・ボックスに直接入力します（最大指定文字数：256 文字）。

（b）ボタン

編集	一覧で選択しいる［入力情報］項目の内容を，テキスト・ボックスに直接入力することにより編集します。 一覧において何も選択していない場合，本ボタンは無効となります。

（2）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］

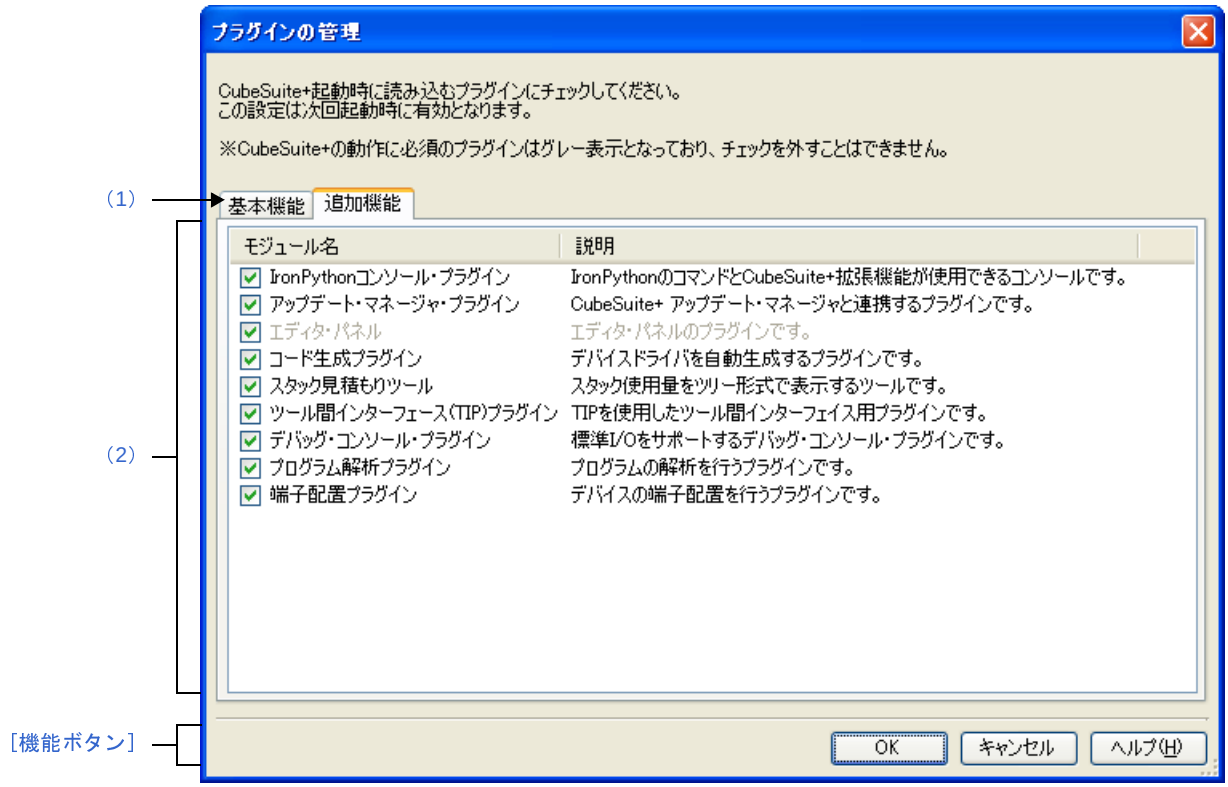
ボタン	機能
すべて初期値に戻す	本ダイアログのすべての設定項目をデフォルトの状態に戻します。 ただし，［全般 - 外部ツール］カテゴリでは，新規登録した内容の削除は行いません。
OK	変更した設定内容を適用し，本ダイアログをクローズします。
キャンセル	変更した設定内容を無効とし，本ダイアログをクローズします。
適用	変更した設定内容を適用します（本ダイアログをクローズしません）。
ヘルプ	本ダイアログのヘルプを表示します。

プラグインの管理 ダイアログ

本製品の起動時に読み込むプラグインを設定します。

注意　本ダイアログにおける設定は，本製品の次回起動時に有効となります。

図 A—59　プラグインの管理 ダイアログ



ここでは，以下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→［プラグインの管理 ...］を選択

[各エリアの説明]

(1) タブ選択エリア

タブを選択することにより，プラグイン一覧エリアに表示するプラグインが切り替わります。

本ダイアログには，以下のタブが存在します。

- ［基本機能］タブ
- ［追加機能］タブ

(2) プラグイン一覧エリア

本製品の起動時に読み込むプラグインを設定します。

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。

[機能ボタン]

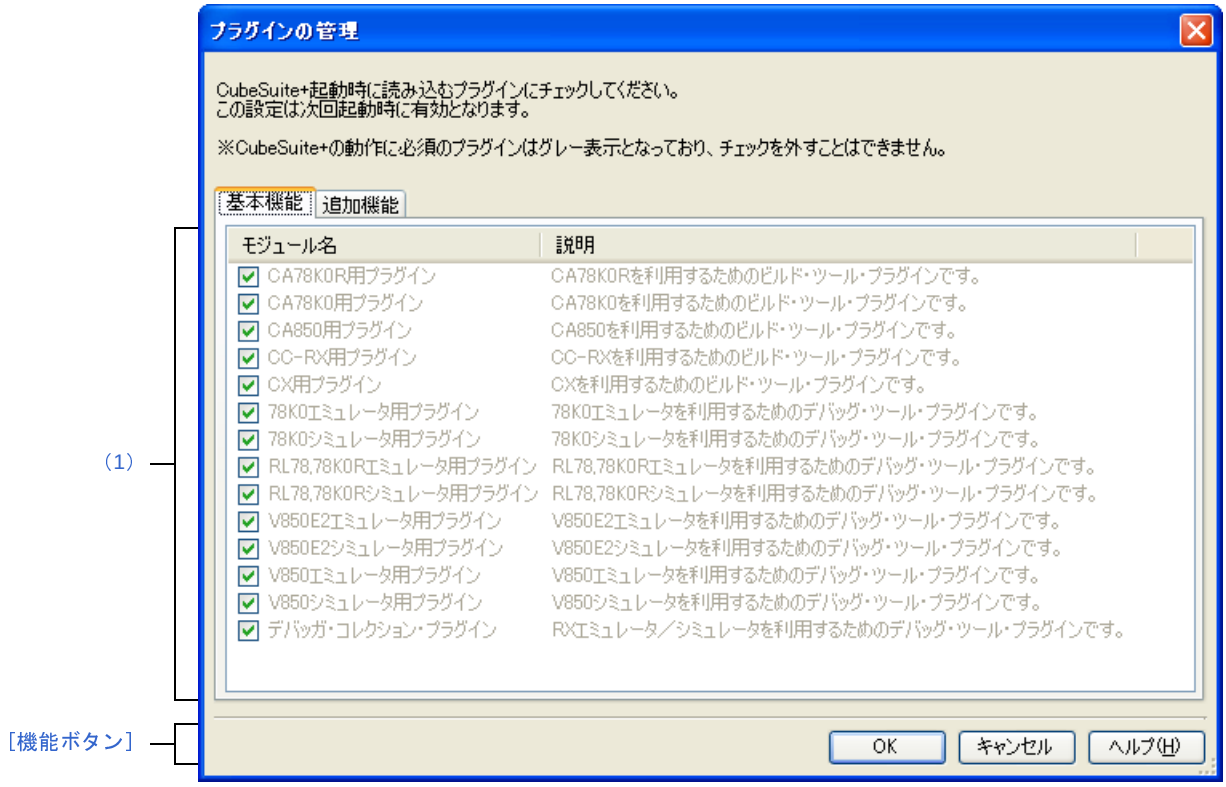
ボタン	機能
OK	変更した設定内容を適用し，本ダイアログをクローズします。
キャンセル	変更した設定内容を破棄し，本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

［基本機能］タブ

本製品の起動時に読み込むビルド・ツール，およびデバッグ・ツールのプラグインを表示します。

注意　ビルド・ツール，およびデバッグ・ツールのプラグインは読み込み対象から外すことができないため，グレー表示となります。

図 A—60　プラグインの管理 ダイアログ：［基本機能］タブ



ここでは，以下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→［プラグインの管理 ...］を選択

[各エリアの説明]

(1) プラグイン一覧エリア

本製品にインストールされているビルド・ツール，およびデバッグ・ツールのプラグインを以下のリスト形式で表示します。

モジュール名	プラグインのモジュール名を表示します。
説明	プラグインの簡易説明を表示します。

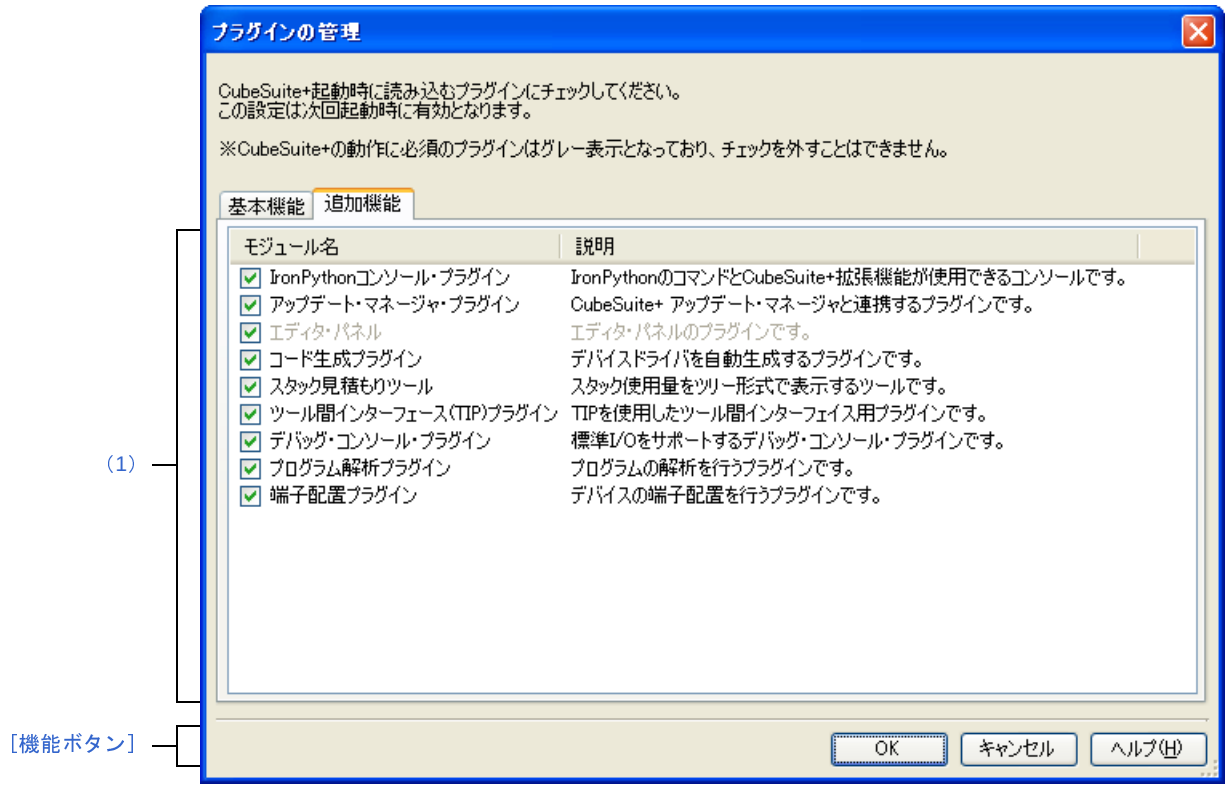
[機能ボタン]

ボタン	機能
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破棄し，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。

［追加機能］タブ

本製品の起動時に読み込むビルド・ツール，およびデバッグ・ツール以外のプラグイン（本製品の Plugins フォルダに存在するプラグイン）を設定します。

図 A—61　プラグインの管理 ダイアログ：［追加機能］タブ



ここでは，以下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→［プラグインの管理 ...］を選択

[各エリアの説明]

(1) プラグイン一覧エリア

本製品にインストールされているビルド・ツール，およびデバッグ・ツール以外のプラグイン（本製品の Plugins フォルダに存在するプラグイン）を以下のリスト形式で表示します。

モジュール名	プラグインのモジュール名を表示します。 本製品の起動時に読み込むプラグインをチェック・ボックスにより選択します。
説明	プラグインの簡易説明を表示します。

備考 1. エディタ パネルのプラグインは読み込み対象から外すことができないため，グレー表示となります。

2. 本製品をコマンドラインから起動する際に /np オプションで指定した Plugins フォルダ以下のプラグインは，本エリアには表示しません。また，/npall オプションを指定した場合は Plugins フォルダ以下のプラグインを読み込まないため，本エリアに何も表示しません。
/np，および /npall オプションについての詳細は，「2. 11　CubeSuite+ をコマンドラインで操作する」を参照してください。

[機能ボタン]

ボタン	機能
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破棄し，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。

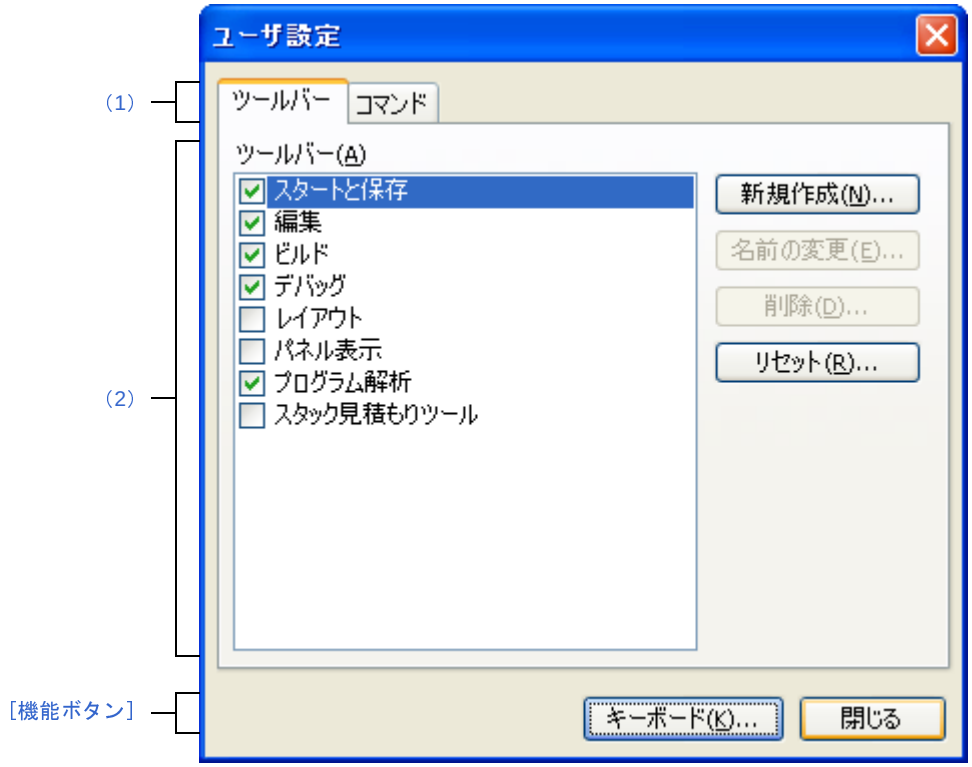
ユーザ設定 ダイアログ

メイン・ウインドウのツールバー，およびメニューバーをカスタマイズします。

なお，本ダイアログをオープン中は，メイン・ウインドウに表示されているツールバーの各ボタン／メニューバーの各メニュー項目を直接任意の箇所にドラッグ・アンド・ドロップすることにより，自由に順序の並び替え／削除などを行うことができます。

注意　本ダイアログをオープン中は，メイン・ウインドウ上のツールバー／メニューバーの機能を使用することができません。
本ダイアログをクローズしたのち，通常どおりの機能を使用することができます。

図 A—62　ユーザ設定 ダイアログ



ここでは，以下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→［ユーザ設定 ...］を選択

[各エリアの説明]

(1) タブ選択エリア

タブを選択することにより，カスタマイズする対象が切り替わります。

本ダイアログには，以下のタブが存在します。

- ［ツールバー］タブ
- ［コマンド］タブ

(2) カスタマイズ・エリア

カスタマイズする際の詳細条件を設定します。

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。

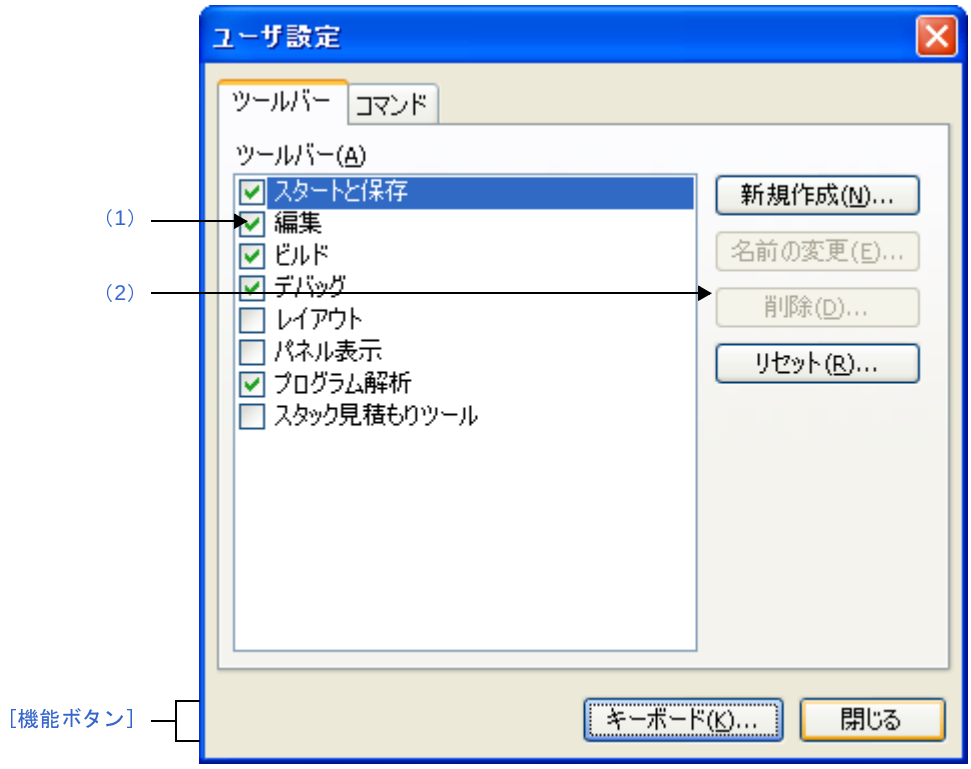
[機能ボタン]

ボタン	機能
キーボード ...	カスタマイズした項目のキーボード割り当てを行うためのキーボードのユーザ設定 ダイアログオープンします。
閉じる	ツールバー／メニューバーのカスタマイズ設定を無効とし，本ダイアログをクローズします。

［ツールバー］タブ

ツールバーの表示／非表示の設定，名称の変更，および新規ツールバーの作成を行います。

図 A—63　ユーザ設定 ダイアログ：［ツールバー］タブ



ここでは，以下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→［ユーザ設定 ...］を選択

［各エリアの説明］

(1)［ツールバー］エリア

登録されているツールバー名を種別ごとに一覧表示します。

また，現在，メイン・ウインドウ上に表示しているツールバー名にはチェック・マークが表示されます。本チェックを外すことにより，非表示に設定されます。

(2) ボタン・エリア

新規作成 ...	新規ツールバーを作成するための新しいツール ダイアログをオープンします。 新規に作成したツールバー名は，チェックが付与された状態で［ツールバー］エリアの一覧に追加されます。 なお，新規に作成したツールバーには何もボタンが設定されません。新規ツールバーへのボタンの登録は，［コマンド］タブ上で行います。
名前の変更 ...	現在選択しているツールバーの名称を変更するためのツールバーの名前の変更 ダイアログをオープンします。 ただし，ユーザにより作成されたツールバー以外を選択している場合は，無効となります。
削除 ...	現在選択しているツールバーを削除します。 ただし，ユーザにより作成されたツールバー以外を選択している場合は，無効となります。
リセット ...	現在選択しているツールバーの変更をすべて破棄し，デフォルトの状態に戻します。 ユーザにより作成されたツールバーを選択している場合は，何もボタンが設定されていない状態に戻ります。

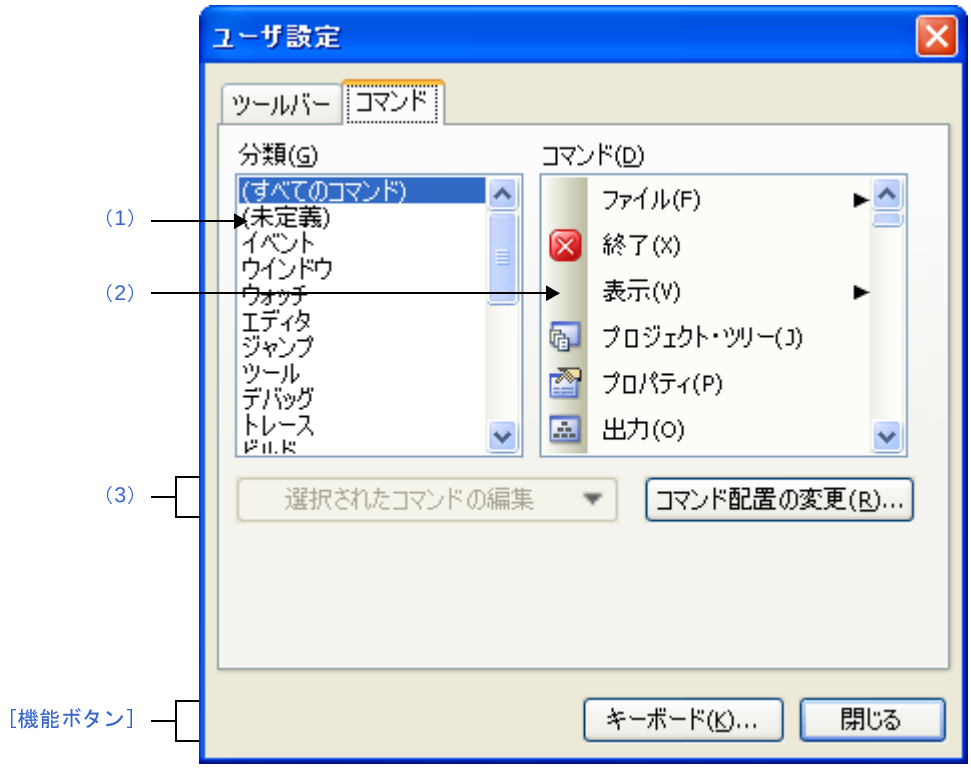
[機能ボタン]

ボタン	機能
キーボード ...	カスタマイズした項目のキーボード割り当てを行うためのキーボードのユーザ設定 ダイアログオープンします。
閉じる	ツールバー／メニューバーのカスタマイズ設定を無効とし，本ダイアログをクローズします。

［コマンド］タブ

ツールバー，およびメニューバーに含める項目のカスタマイズを行います。

図 A—64　ユーザ設定 ダイアログ：［コマンド］タブ



ここでは，以下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→［ユーザ設定 ...］を選択

［各エリアの説明］

(1)［分類］エリア

CubeSuite+ が提供するコマンドのカテゴリを一覧表示します。

(2)［コマンド］エリア

［分類］エリアで選択されているカテゴリに属するコマンド名とそのアイコン（存在する場合のみ）を一覧表示します。

［分類］エリアで“(すべてのコマンド)”を選択している場合は，CubeSuite+ が提供するすべてのコマンド名とそのアイコン（存在する場合のみ）を表示します。

ツールバー／メニューバーにコマンドを追加する場合は，本エリア内のコマンド名を実際のメイン・ウインドウ上のツールバー／メニューバー上にドラッグ・アンド・ドロップします。

(3) ボタン・エリア

<table>
<tr><td rowspan="10">選択したボタンの編集</td><td colspan="2">現在選択しているメイン・ウインドウ上のメニュー項目／ツールバーのボタン表示を編集するために，以下のメニューを表示します。</td></tr>
<tr><td>メニュー名</td><td>説明</td></tr>
<tr><td>リセット</td><td>現在選択しているメニュー項目／ボタンをリセットします。</td></tr>
<tr><td>削除</td><td>現在選択しているメニュー項目／ボタンを削除します。</td></tr>
<tr><td>名前</td><td>現在選択しているメニュー項目／ボタンの名称を表示します。</td></tr>
<tr><td>既存のスタイル</td><td>チェックすると，現在選択しているメニュー項目／ボタンの表示スタイルをデフォルトの状態に戻します（デフォルト）。</td></tr>
<tr><td>テキストのみ表示</td><td>チェックすると，現在選択しているメニュー項目／ボタンをテキストのみの表示にします（アイコンは表示しません）。</td></tr>
<tr><td>テキストのみ表示（メニューのとき）</td><td>メニュー項目を選択している場合にのみ有効となります。
チェックすると，現在選択しているメニュー項目／ボタンをテキストのみの表示にします（アイコンは表示しません）。</td></tr>
<tr><td>イメージとテキストを表示</td><td>チェックすると，現在選択しているメニュー項目／ボタンをアイコンとテキストの両方で表示にします。</td></tr>
<tr><td>グループの始まり</td><td>現在選択しているメニュー項目／ボタンの直前に，セパレータを挿入します。</td></tr>
<tr><td>最近使ったツール</td><td>今般ではサポートしていません。</td></tr>
<tr><td>コマンド配置の変更 ...</td><td colspan="2">メイン・ウインドウ上のメニュー項目／ツールバーのボタンの配置（追加／削除を含む）を変更するするためのコマンドの配置の変更 ダイアログをオープンします。</td></tr>
</table>

［機能ボタン］

ボタン	機能
キーボード ...	カスタマイズした項目のキーボード割り当てを行うためのキーボードのユーザ設定 ダイアログオープンします。
閉じる	ツールバー／メニューバーのカスタマイズ設定を無効とし，本ダイアログをクローズします。

新しいツール ダイアログ

メイン・ウインドウに表示する“新規ツールバー”を作成します。

図 A—65　新しいツール ダイアログ

ここでは，以下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］

- ユーザ設定 ダイアログの［ツールバー］タブにおいて，［新規作成 ...］ボタンをクリック

［各エリアの説明］

(1)［ツールバー名］

新規に作成するツールバーの名称をキーボードより直接入力します。

デフォルトでは，“UltraToolbar1”が指定されます。

(2)［配置］

新規に作成したツールバーの配置場所を以下のドロップダウン・リストにより選択します。

なお，ここで指定した配置場所は，新規に作成完了した直後に表示される位置を意味します（ツールバーを直接ドラッグ＆ドロップ操作することにより，配置位置は自由に変更することができます）。

上端にドッキング	メイン・ウインドウの上端に表示します（デフォルト）。
下端にドッキング	メイン・ウインドウの下端に表示します。
左端にドッキング	メイン・ウインドウの左端に表示します。
右端にドッキング	メイン・ウインドウの右端に表示します。
フローティング	メイン・ウインドウ上で固定されずに表示します。

[機能ボタン]

ボタン	機能
OK	指定した内容で新規にツールバーを作成し，本ダイアログをクローズします。 新規に作成されたツールバー名は，チェックが付与された状態でユーザ設定 ダイアログの［ツールバー］タブ上の一覧に追加されます。
キャンセル	新規ツールバーの作成を中止し，本ダイアログをクローズします。

ツールバーの名前の変更 ダイアログ

ユーザが作成したツールバーの名称を編集します。

図 A—66　ツールバーの名前の変更 ダイアログ

ここでは，以下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］

- ユーザ設定 ダイアログの［ツールバー］タブにおいて，ユーザが作成したツールバー名を選択したのち，［名前の変更 ...］ボタンをクリック

［各エリアの説明］

(1)［ツールバー名］

ツールバーの名称をキーボードより直接編集します。

デフォルトでは，選択しているツールバー名が表示されます。

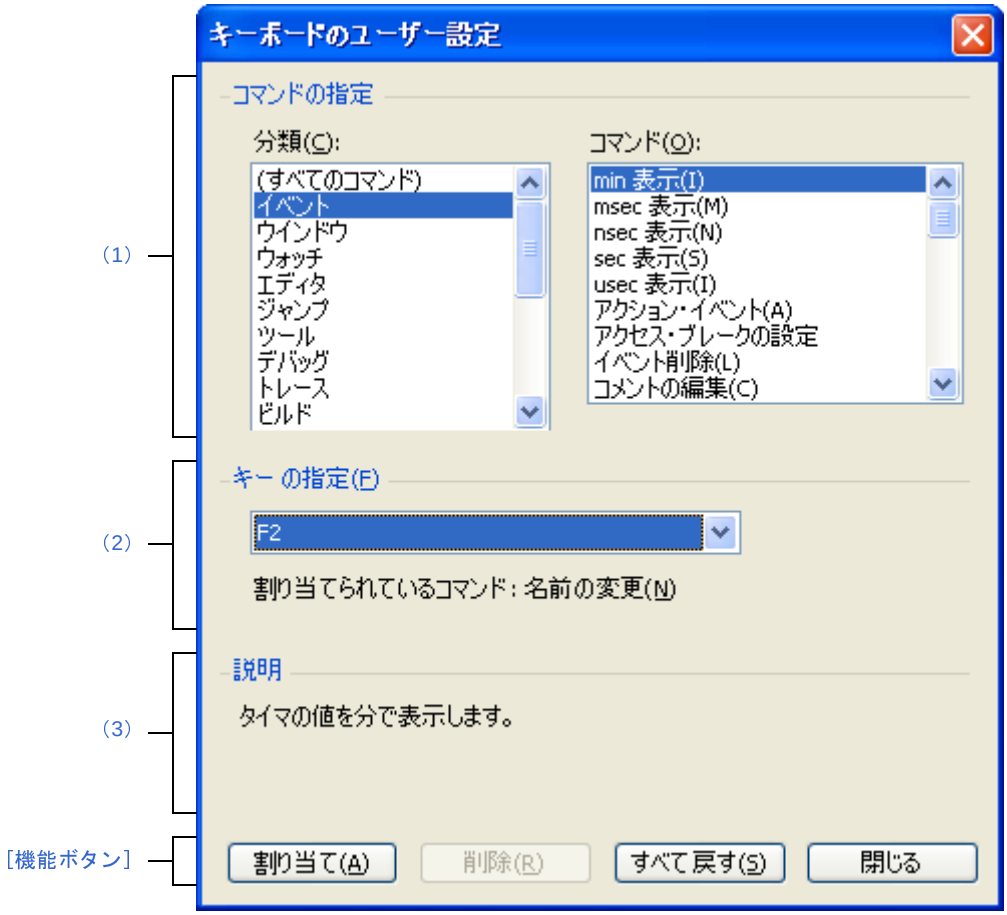
［機能ボタン］

ボタン	機能
OK	選択しているツールバーを指定した名称に変更し，本ダイアログをクローズします。
キャンセル	ツールバーの名称の編集を中止し，本ダイアログをクローズします。

キーボードのユーザ設定 ダイアログ

各コマンドに対して，ショートカット・キーを割り当てます。

図 A—67　キーボードのユーザ設定 ダイアログ



ここでは，以下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］

- ユーザ設定 ダイアログの［ツールバー］タブにおいて，［キーボード ...］ボタンをクリック

[各エリアの説明]

(1) [コマンドの指定] エリア

(a) [分類]

CubeSuite+ が提供するコマンドのカテゴリを一覧表示します。

(b) [コマンド]

[分類] で選択されているカテゴリに属するコマンド名とそのアイコン（存在する場合のみ）を一覧表示します。

[分類] で“(すべてのコマンド)”を選択している場合は，CubeSuite+ が提供するすべてのコマンド名とそのアイコン（存在する場合のみ）を表示します。

(2) [キーの指定] エリア

現在，[コマンド] で選択されているコマンドに割り当てられているショートカット・キーをデフォルトで表示します（キーの割り当てがない場合は“なし”を表示）。

割り当てるショートカット・キーを変更する場合は，以下のドロップダウン・リストの中から任意のキーを選択したのち，[割り当て] ボタンをクリックします。

なし	Shift + F1 ～ F12	Ctrl + Shift + 0 ～ 9	Alt + Right
Insert	Ctrl + Insert	Ctrl + Shift + A ～ Z	Alt + Down
Delete	Ctrl + Delete	Ctrl + Shift + F1 ～ F12	Alt + 0 ～ 9
F1 ～ F12	Ctrl + 0 ～ 9	Alt + Backspace	Alt + F1 ～ F12
Shift + Insert	Ctrl + A ～ Z	Alt + Left	
Shift + Delete	Ctrl + F1 ～ F12	Alt + Up	

[割り当てられているコマンド] には，現在，[キーの指定] エリアで選択されているショートカット・キーに割り当てられているコマンド名を表示します（キーの割り当てがない場合は“なし”を表示）。

(3) [説明] エリア

現在，[コマンド] で選択されているコマンドの機能説明をポップアップ表示します。

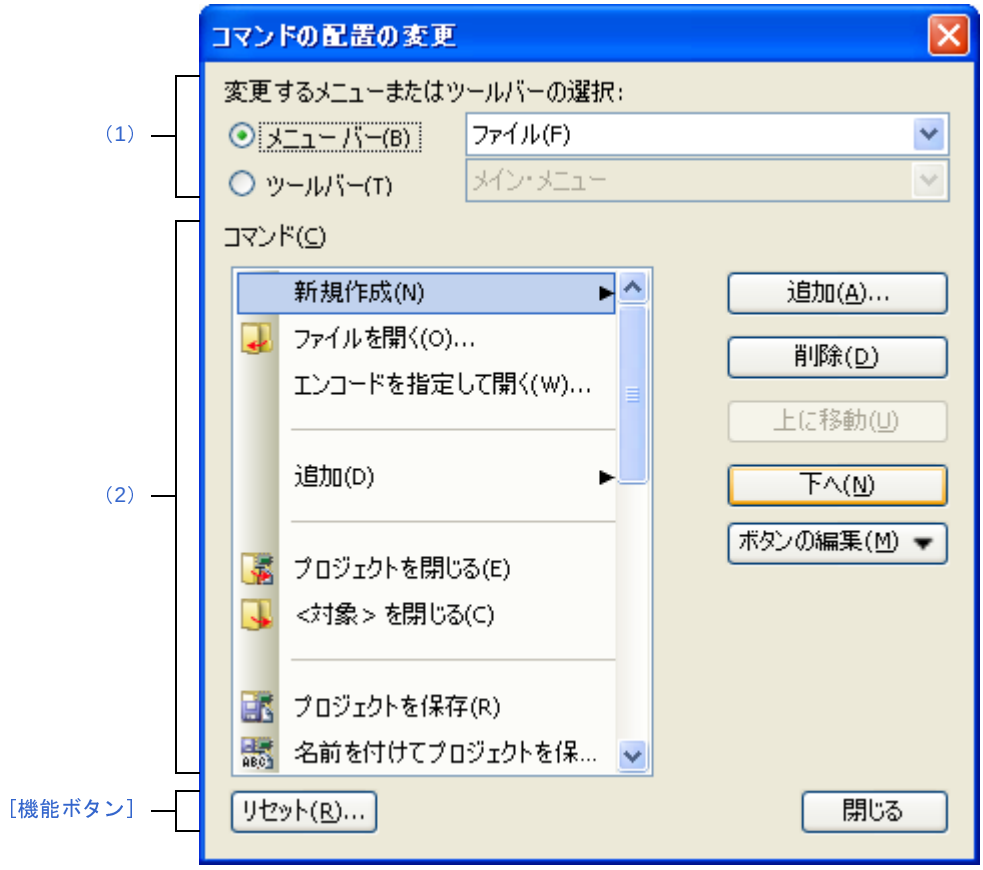
[機能ボタン]

ボタン	機能
割り当て	現在，［コマンド］で選択されているコマンドに対し，［キーの指定］エリアで選択されているキーをショートカット・キーとして割り当てます。 ただし，［キーの指定］エリアで選択されているキーが，他のコマンドのキーとしてすでに割り当てられている場合は，本ボタンは無効となります。
削除	現在，［コマンド］で選択されているコマンドに割り当てられているショートカット・キーの割り当てを解除します（［キーの指定］エリアのドロップダウン・リストでは“なし”が表示されます）。 ただし，［コマンド］で選択されているコマンドにキーの割り当てがない場合は，本ボタンは無効となります。
すべて戻す	現在設定されているショートカット・キーの割り当てをすべてデフォルトの状態に戻します。
閉じる	ショートカット・キーの割り当てを中止し，本ダイアログをクローズします。

コマンドの配置の変更 ダイアログ

メイン・ウインドウのメニュー項目／ボタン配置を変更（追加／削除を含む）します。

図 A—68　コマンドの配置の変更 ダイアログ



ここでは，以下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］

- ユーザ設定 ダイアログの［コマンド］タブにおいて，［コマンド配置の変更 ...］ボタンをクリック

[各エリアの説明]

(1) [変更するメニューまたはツールバーの選択] エリア

配置を変更したい項目を指定します。

オプション・ボタンにより，メニュー項目を変更する場合は［メニュー］を，ツールバーのボタンを変更する場合は［ツールバー］を選択したのち，変更の対象となるカテゴリをドロップダウン・リストにより選択します。

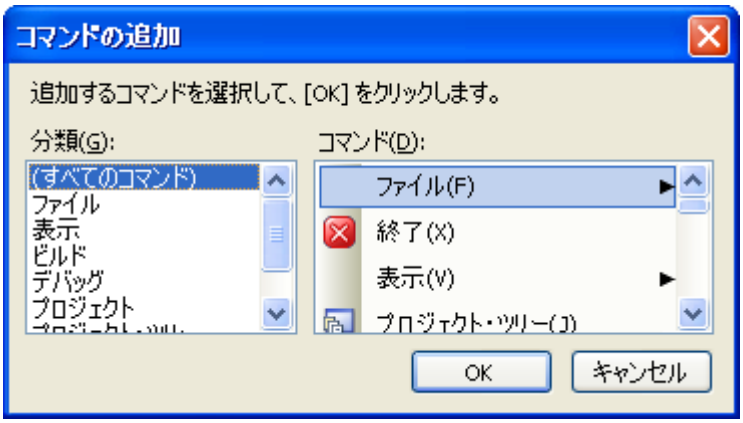
(2) [コマンド] エリア

［変更するメニューまたはツールバーの選択］エリアで選択しているカテゴリのうち，現在メイン・ウインドウ上で表示されるコマンドの一覧を表示します。

各コマンドの配置の変更は，本エリアの以下のボタンにより行います。

<table>
<tr><td>追加 ...</td><td colspan="2">コマンドの追加 ダイアログがオープンし，現在本エリアで選択しているコマンドの直上に追加するコマンドを選択します。
コマンドの追加 ダイアログでは，追加可能なコマンドがカテゴリ別にすべて表示されます。
［分類］エリアでコマンドのカテゴリを選択したのち，［コマンド］エリアより追加したいコマンドを選択したのち，［OK］ボタンをクリックしてください（［キャンセル］ボタンでコマンドの追加をキャンセルし，本ダイアログをクローズします）。</td></tr>
<tr><td>削除</td><td colspan="2">現在本エリアで選択しているコマンドを削除します。</td></tr>
<tr><td>上に移動</td><td colspan="2">現在本エリアで選択しているコマンドの配置を 1 つ上に移動します。</td></tr>
<tr><td>下へ</td><td colspan="2">現在本エリアで選択しているコマンドの配置を 1 つ下に移動します。</td></tr>
<tr><td rowspan="9">ボタンの編集</td><td colspan="2">現在本エリアで選択しているコマンドのメニュー項目／ボタン表示を編集するために，以下のメニューを表示します。</td></tr>
<tr><td>メニュー名</td><td>説明</td></tr>
<tr><td>リセット</td><td>現在選択しているメニュー項目／ボタンをリセットします。</td></tr>
<tr><td>削除</td><td>現在選択しているメニュー項目／ボタンを削除します。</td></tr>
<tr><td>名前： 新規作成</td><td>現在選択しているメニュー項目／ボタンの名称を表示します。</td></tr>
<tr><td>既存のスタイル</td><td>チェックすると，現在選択しているメニュー項目／ボタンの表示スタイルをデフォルトの状態に戻します（デフォルト）。</td></tr>
<tr><td>テキストのみ表示</td><td>チェックすると，現在選択しているメニュー項目／ボタンをテキストのみの表示にします（アイコンは表示しません）。</td></tr>
<tr><td>テキストのみ表示
（メニューのとき）</td><td>メニュー項目を選択している場合にのみ有効となります。
チェックすると，現在選択しているメニュー項目／ボタンをテキストのみの表示にします（アイコンは表示しません）。</td></tr>
<tr><td>イメージとテキストを表示</td><td>チェックすると，現在選択しているメニュー項目／ボタンをアイコンとテキストの両方で表示にします。</td></tr>
<tr><td>グループの始まり</td><td>現在選択しているメニュー項目／ボタンの直前に，セパレータを挿入します。</td></tr>
<tr><td>最近使ったツール</td><td>今般ではサポートしていません。</td></tr>
</table>

図 A—69　コマンドの追加 ダイアログ



[機能ボタン]

ボタン	機能
リセット ...	現在，［変更するメニューまたはツールバーの選択］エリアで選択しているカテゴリのメニュー項目／ツールバーのボタンの配置をすべてデフォルトに戻します。
閉じる	コマンドの配置の変更を終了し，本ダイアログをクローズします。

バージョン情報 ダイアログ

CubeSuite+，および各プラグイン製品のバージョン情報を表示します。

図 A—70　バージョン情報 ダイアログ

ここでは，以下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］

- ［ヘルプ］メニュー→［バージョン情報 ...］を選択

［各エリアの説明］

(1) 製品名エリア

CubeSuite+ のバージョンを表示します。

(2)［製品ライセンス］エリア

ライセンス・キーに対応するライセンス情報を表示します。

(3) インストールされている製品エリア

CubeSuite+ で使用している各モジュールの名称とそのバージョンを以下のリスト形式で表示します。

モジュール名	モジュール名を表示します。
バージョン	モジュールのバージョンを表示します。
説明	モジュールの簡易説明を表示します。

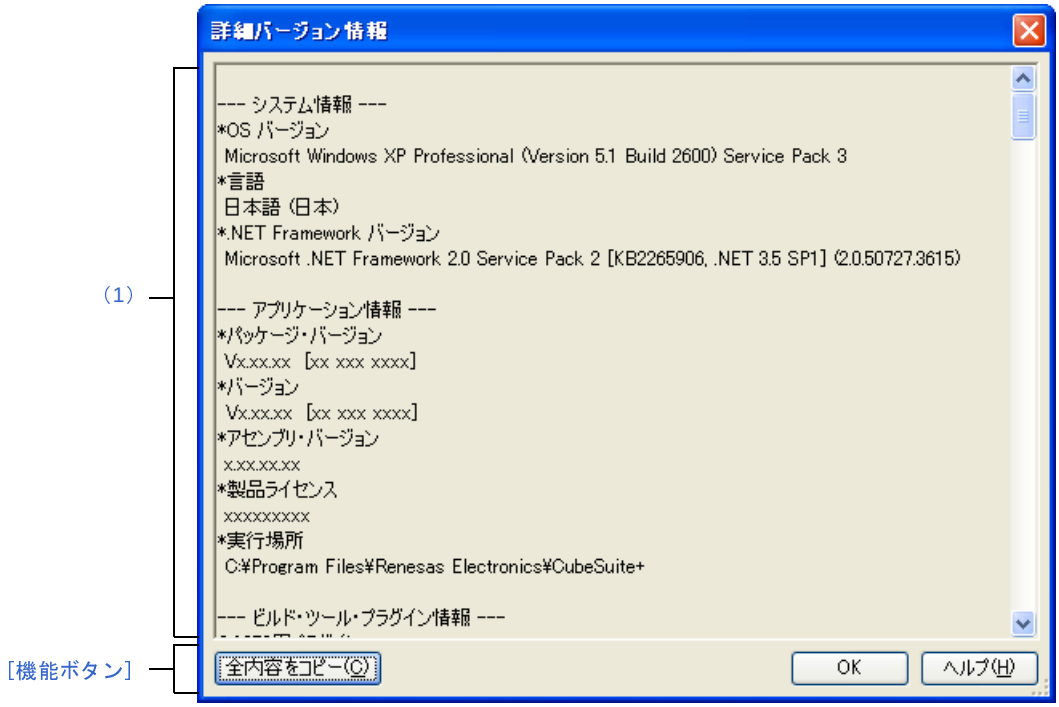
[機能ボタン]

ボタン	機能
全内容をコピー	表示している文字列をすべてクリップ・ボードにコピーします。
OK	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。

詳細バージョン情報 ダイアログ

本製品の詳細バージョンや現在のプロジェクトの情報を表示します。

図 A—71　詳細バージョン情報 ダイアログ



ここでは，以下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（詳細バージョン情報 ダイアログ専用部分）］
- ［コンテキスト・メニュー］
- ［機能ボタン］

［オープン方法］

- ［ヘルプ］メニュー→［詳細バージョン情報 ...］を選択
- メッセージ ダイアログにおいて，［詳細バージョン情報 ...］ボタンをクリック

[各エリアの説明]

(1) 詳細バージョン情報エリア

本製品の詳細バージョンや現在のプロジェクトの情報を表示します。

表示する情報は以下のとおりです。

項目名		表示内容
システム情報	OS バージョン	Windows の情報
	言語	
	.NET Franework バージョン	
アプリケーション情報	パッケージ・バージョン	本製品の情報
	バージョン	
	アセンブリ・バージョン	
	拠点情報注 1	
	製品ライセンス	
	追加情報注 1	
	実行場所	
プラグイン情報注 2	バージョン注 3	使用中の各プラグインの情報
	アセンブリ・バージョン注 3	
	DLL ファイル名注 3	
	URL 注 1	
現在のプロジェクト情報注 4	メイン・プロジェクト情報（サブプロジェクト情報）	プロジェクト・ファイルの絶対パス（メイン・プロジェクトの場合は，*.mtpj ファイルのファイ名を含む絶対パス，サブプロジェクトの場合は，*.mtsp ファイルのファイル名を含む絶対パス）を表示
	マイクロコントローラ情報	設定されているマイクロコントローラ情報の以下の項目 - デバイス名 - デバイス・ファイル名：バージョン一覧
	ビルド・ツール情報	設定されているビルド・ツール情報の以下の項目注 5 - ビルド・ツール名 - ビルド・ツール・プラグインのバージョン - 使用するコンパイラ・パッケージのバージョン注 1
	デバッグ・ツール情報	設定されているデバッグ・ツール・プラグイン情報の以下の項目（デバッグ・ツールが未設定の場合は非表示） - 接続しているデバッグ・ツール名 - デバッグ・ツール・プラグインのバージョン - デバッグ・ツール制御プログラムのバージョン
発生したエラー情報注 6	発生したエラーの詳細情報	

注 1.　情報が存在しない場合は表示しません。

2. 読み込まれなかったプラグインのモジュール名の末尾には“(-)”を付加します。
   読み込まれているが機能していないプラグインのモジュール名の末尾には“(!)”を付加します。
3. 読み込まれなかったプラグインは“-”と表示します。
4. プロジェクトを開いている場合にのみ表示します（サブプロジェクトが存在する場合は，並べて列挙します）。
5. デバッグ専用プロジェクトが対象の場合は表示しません。
6. メッセージ ダイアログ上の［詳細バージョン情報 ...］ボタンにより，本ダイアログをオープンした場合にのみ表示します。

[[編集] メニュー（詳細バージョン情報 ダイアログ専用部分）]

コピー	詳細バージョン情報エリアで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	詳細バージョン情報エリアの表示文字列をすべて選択状態にします。

[コンテキスト・メニュー]

コピー	詳細バージョン情報エリアで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	詳細バージョン情報エリアの表示文字列をすべて選択状態にします。

[機能ボタン]

ボタン	機能
全内容をコピー	詳細バージョン情報エリアの表示文字列をすべてクリップ・ボードにコピーします。
OK	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。

ワン・ポイント・アドバイス ダイアログ

CubeSuite+ を使用する際のワン・ポイント・アドバイスを表示します。

図 A—72　ワン・ポイント・アドバイス ダイアログ

ここでは，以下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］

- ［ヘルプ］メニュー→［ワン・ポイント・アドバイス ...］を選択
- ［起動時に表示しない］をチェックしていない場合，CubeSuite+ 起動時に自動的にオープン

［各エリアの説明］

(1) アドバイス表示エリア

CubeSuite+ を使用する際のワン・ポイント・アドバイスをランダムに表示します（編集不可）。

(2) ページ切り替えエリア

本ダイアログで現在表示しているページ番号と，ページ切り替え用の以下のボタンを表示します。

ボタン	機能
前へ	現在表示しているページの 1 つ前のページを表示します。 ただし，1 ページ目を表示している場合は無効となります。
次へ	現在表示しているページの次のページを表示します。 ただし，最終ページを表示している場合は無効となります。

(3)［起動時に表示しない］

この設定は，使用中のユーザの設定として保存されます。

☑	起動時，本ダイアログをオープンしません。
☐	起動時，メイン・ウインドウが表示された際に，本ダイアログを自動的にオープンします（デフォルト）。

備考　この設定は，オプション ダイアログ の ［全般 - 起動と終了］カテゴリでも設定ができます。

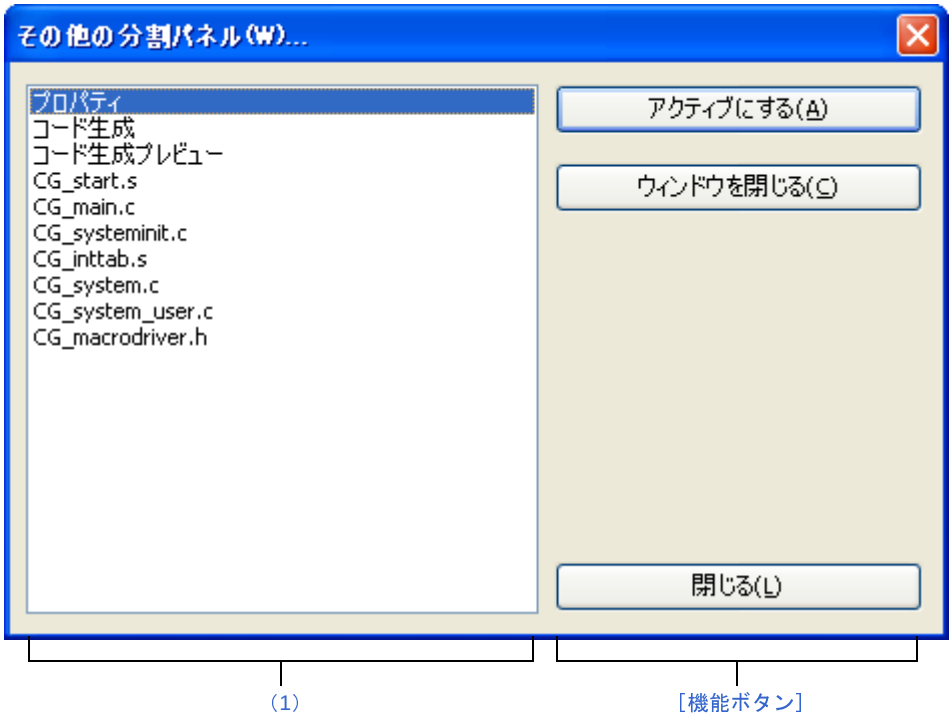
［機能ボタン］

ボタン	機能
OK	本ダイアログをクローズします。
ヘルプ	本ダイアログのヘルプを表示します。

その他の分割パネル ... ダイアログ

分割パネルとしてメイン・ウインドウに表示しているパネルの内，アクティブにするパネル，もしくは閉じるパネルを選択します。

図 A—73　その他の分割パネル ... ダイアログ



ここでは，以下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］

- ［ウインドウ］メニュー→［その他の分割パネル ...］を選択

備考　［その他の分割パネル ...］は，分割パネルを 10 個以上開いている場合のみ表示されます。

［各エリアの説明］

(1) 分割パネル選択エリア

分割パネルを選択することにより，対象が切り替わります。

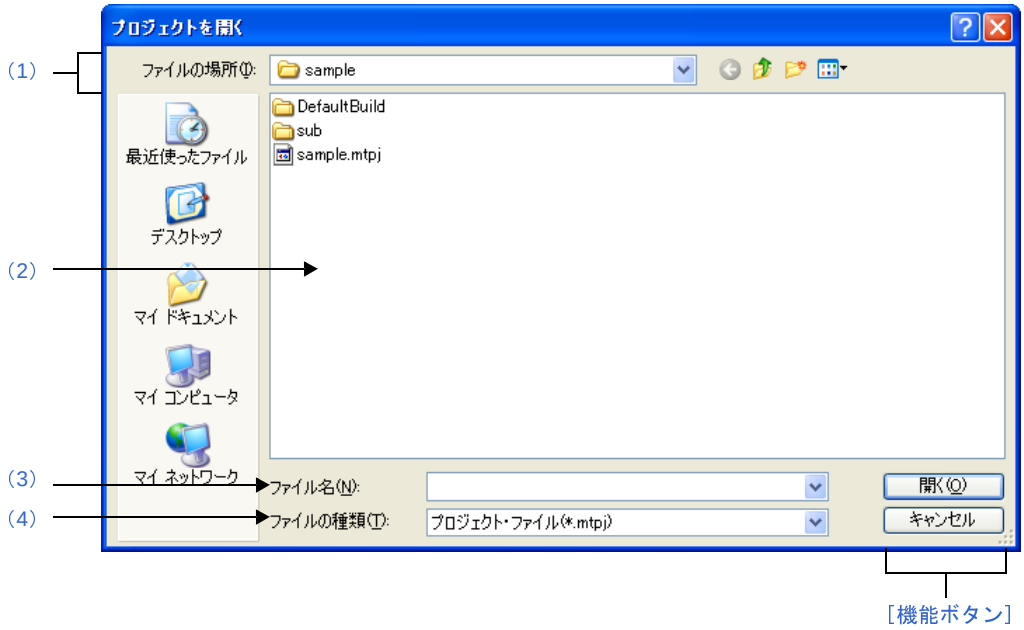
[機能ボタン]

ボタン	機能
アクティブにする	分割パネル選択エリアで選択している分割パネルをアクティブにし，本ダイアログをクローズします。
ウインドウを閉じる	分割パネル選択エリアで選択している分割パネルをクローズします。
閉じる	本ダイアログをクローズします。

プロジェクトを開く ダイアログ

既存のプロジェクトを開く，または新しいプロジェクトの作成時に流用元のプロジェクトを指定するために，プロジェクト・ファイルを選択します。

図 A—74　プロジェクトを開く ダイアログ



ここでは，以下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] メニュー→ [プロジェクトを開く ...] を選択
- プロジェクト作成 ダイアログにおいて，プロジェクト・ファイル・エリア内の [流用元のプロジェクト] の [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

開きたいプロジェクトのプロジェクト・ファイルが存在するフォルダを選択します。

はじめて流用する場合は“C:¥Documents and Settings¥*ユーザ名*¥My Documents”，2 回目以降は前回選択したフォルダが，デフォルトで選択されます。

(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

開きたいプロジェクトのプロジェクト・ファイル名を指定します。

(4)［ファイルの種類］エリア

開きたいプロジェクト・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。

(a)［プロジェクト］メニューからオープンした場合

プロジェクト・ファイル (*.mtpj)	プロジェクト・ファイル
CubeSuite 用プロジェクト・ファイル (*.cspj)	CubeSuite 用プロジェクト・ファイル
HEW 用ワークスペース・ファイル (*.hws)	HEW 用ワークスペース・ファイル
HEW 用プロジェクト・ファイル (*.hwp)	HEW 用プロジェクト・ファイル
PM+ 用ワークスペース・ファイル (*.prw)	PM+ 用ワークスペース・ファイル
PM+ 用プロジェクト・ファイル (*.prj)	PM+ 用プロジェクト・ファイル

(b) スタート パネルの［既存のプロジェクトを開く］エリアからオープンした場合

プロジェクト・ファイル (*.mtpj)	プロジェクト・ファイル

(c) スタート パネルの［CubeSuite ／ High-performance Embedded Workshop ／ PM+ のプロジェクトを開く］エリアからオープンした場合

CubeSuite 用プロジェクト・ファイル (*.cspj)	CubeSuite 用プロジェクト・ファイル
HEW 用ワークスペース・ファイル (*.hws)	HEW 用ワークスペース・ファイル
HEW 用プロジェクト・ファイル (*.hwp)	HEW 用プロジェクト・ファイル
PM+ 用ワークスペース・ファイル (*.prw)	PM+ 用ワークスペース・ファイル
PM+ 用プロジェクト・ファイル (*.prj)	PM+ 用プロジェクト・ファイル

(d) プロジェクト作成 ダイアログからオープンした場合

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

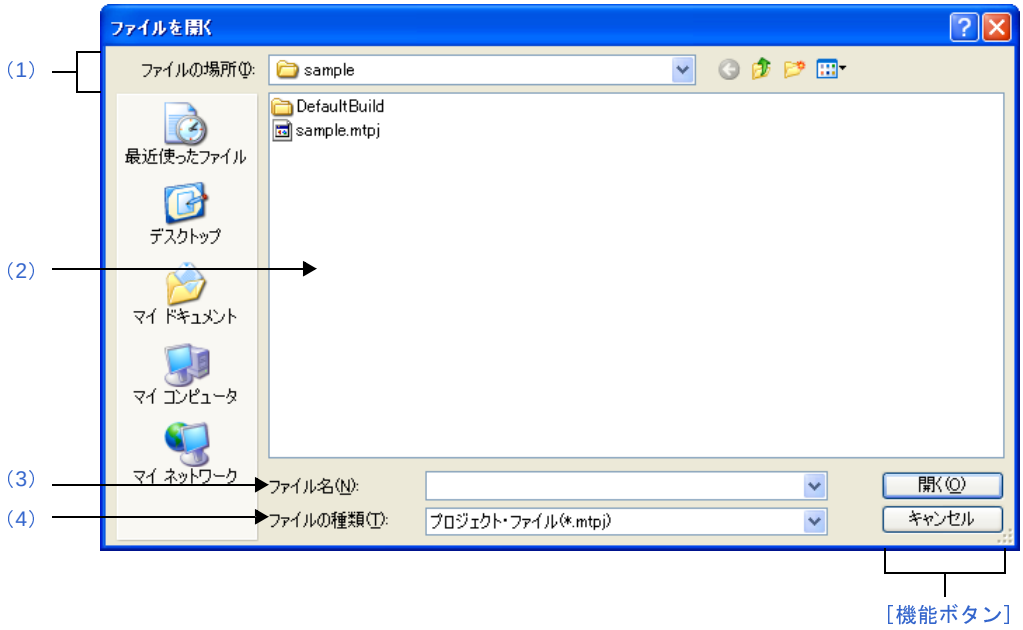
[機能ボタン]

ボタン	機能
開く	- ［プロジェクト］メニュー（CubeSuite+ のプロジェクト），またはスタート パネルの［既存のプロジェクトを開く］エリアからオープンした場合 指定したプロジェクト・ファイルを開きます。 - ［プロジェクト］メニュー（CubeSuite/High-performance Embedded Workshop/PM+ のプロジェクト），またはスタート パネルの［CubeSuite ／ High-performance Embedded Workshop ／ PM+ のプロジェクトを開く］エリアからオープンした場合 プロジェクト変換設定 ダイアログをオープンします。 - プロジェクト作成 ダイアログからオープンした場合 呼び出し元のダイアログにおいて，プロジェクト・ファイル・エリア内の［流用元のプロジェクト］にプロジェクト・ファイルを指定します。
キャンセル	本ダイアログをクローズします。

ファイルを開く ダイアログ

オープンするファイルの選択を行います。

図 A—75　ファイルを開く 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→ [ファイルを開く ...]，または [エンコードを指定して開く ...] を選択

[各エリアの説明]

(1) [ファイルの場所] エリア

オープンするファイルが存在するフォルダを選択します。

初回は “C:¥Documents and Settings¥*ユーザ名*¥My Documents”，2 回目以降は前回選択したフォルダが，デフォルトで選択されます。

(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

オープン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

オープンするファイルの種類（ファイル・タイプ）を選択します。

(a) プロジェクトを開いていない場合

すべてのファイル (*.*)	すべての形式
プロジェクト・ファイル (*.mtpj)	プロジェクト・ファイル
CubeSuite 用プロジェクト・ファイル (*.cspj)	CubeSuite 用プロジェクト・ファイル
HEW 用ワークスペース・ファイル (*.hws)	HEW 用ワークスペース・ファイル
HEW 用プロジェクト・ファイル (*.hwp)	HEW 用プロジェクト・ファイル
PM+ 用ワークスペース・ファイル (*.prw)	PM+ 用ワークスペース・ファイル
PM+ 用プロジェクト・ファイル (*.prj)	PM+ 用プロジェクト・ファイル
テキスト・ファイル (*.txt)	テキスト形式

(b) デバッグ専用プロジェクト以外の場合

すべてのファイル (*.*)	すべての形式
プロジェクト・ファイル (*.mtpj)	プロジェクト・ファイル
CubeSuite 用プロジェクト・ファイル (*.cspj)	CubeSuite 用プロジェクト・ファイル
HEW 用ワークスペース・ファイル (*.hws)	HEW 用ワークスペース・ファイル
HEW 用プロジェクト・ファイル (*.hwp)	HEW 用プロジェクト・ファイル
PM+ 用ワークスペース・ファイル (*.prw)	PM+ 用ワークスペース・ファイル
PM+ 用プロジェクト・ファイル (*.prj)	PM+ 用プロジェクト・ファイル
C ソース・ファイル (*.c)	C ソース・ファイル
C++ ソース・ファイル (*.cpp; *.cc; *.cp)【CC-RX】	C++ ソース・ファイル
ヘッダ・ファイル (*.h; *.hpp; *.inc)【CC-RX】	ヘッダ・ファイル
ヘッダ・ファイル (*.h; *.inc)【CA850】【CX】【CA78K0R】【CA78K0】	ヘッダ・ファイル
アセンブル・ファイル（*.src）【CC-RX】	アセンブラ・ソース・ファイル
アセンブル・ファイル (*.s)【CA850】	アセンブラ・ソース・ファイル
アセンブル・ファイル (*.asm; *.s)【CX】	アセンブラ・ソース・ファイル
アセンブル・ファイル (*.asm)【CA78K0R】【CA78K0】	アセンブラ・ソース・ファイル
リンク・ディレクティブ・ファイル (*.dir; *.dr)【CA850】【CX】	リンク・ディレクティブ・ファイル
リンク・ディレクティブ・ファイル (*.dr; *.dir)【CA78K0R】【CA78K0】	リンク・ディレクティブ・ファイル
セクション・ファイル（*.sf）【CA850】	セクション・ファイル
シンボル情報ファイル (*.sfg)【CX】	シンボル情報ファイル

変数／関数情報ファイル（*.vfi）【CA78K0R】	変数／関数情報ファイル
変数情報ファイル（*.vfi）【CA78K0】	変数情報ファイル
関数情報ファイル (*.fin) 注【CA78K0】	関数情報ファイル
マップ・ファイル (*.map; *.lbp)【CC-RX】	マップ・ファイル
マップ・ファイル (*.map)【CA850】【CX】【CA78K0R】【CA78K0】	マップ・ファイル
シンボル・テーブル・ファイル (*.sym)【CA78K0R】【CA78K0】	シンボル・テーブル・ファイル
ヘキサ・ファイル (*.hex)【CC-RX】【CA850】【CX】	ヘキサ・ファイル
ヘキサ・ファイル (*.hex; *.hxb; *.hxf)【CA78K0R】【CA78K0】	ヘキサ・ファイル
アセンブル・リスト・ファイル（*.lst）【CC-RX】	アセンブル・リスト・ファイル
S レコード・ファイル（*.mot）【CC-RX】	S レコード・ファイル
テキスト・ファイル (*.txt)	テキスト形式

注　メモリ・バンク搭載品のみ

（c）デバッグ専用プロジェクトの場合

すべてのファイル (*.*)	すべての形式
プロジェクト・ファイル (*.mtpj)	プロジェクト・ファイル
CubeSuite 用プロジェクト・ファイル (*.cspj)	CubeSuite 用プロジェクト・ファイル
HEW 用ワークスペース・ファイル (*.hws)	HEW 用ワークスペース・ファイル
HEW 用プロジェクト・ファイル (*.hwp)	HEW 用プロジェクト・ファイル
PM+ 用ワークスペース・ファイル (*.prw)	PM+ 用ワークスペース・ファイル
PM+ 用プロジェクト・ファイル (*.prj)	PM+ 用プロジェクト・ファイル
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
アセンブル・ファイル (*.asm; *.s)	アセンブラ・ソース・ファイル
テキスト・ファイル (*.txt)	テキスト形式

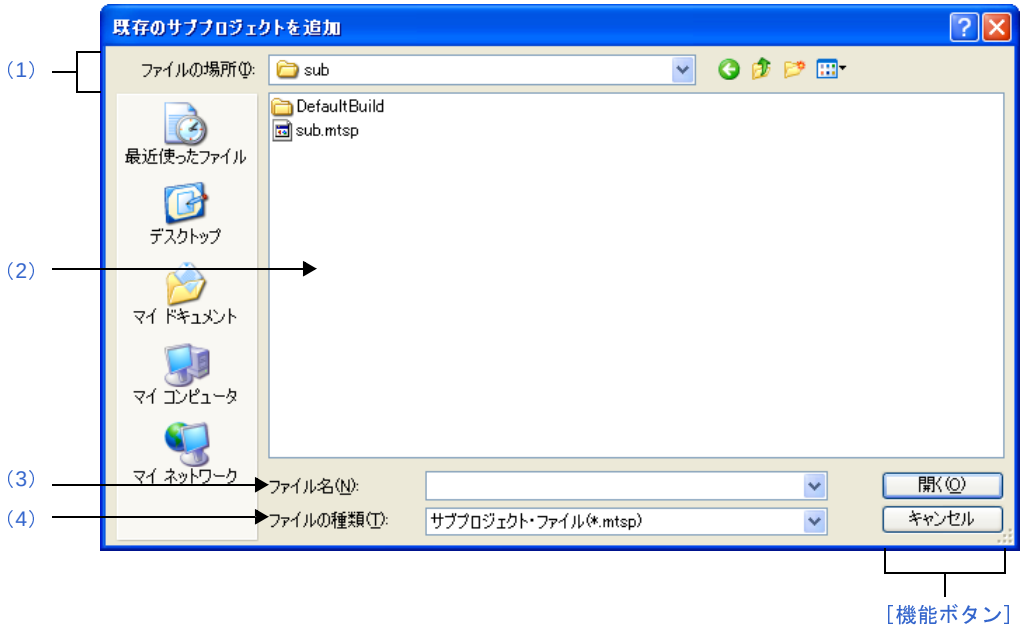
[機能ボタン]

ボタン	機能
開く	- ［ファイル］メニュー→［ファイルを開く ...］からオープンした場合 指定したファイルをオープンします。 - ［ファイル］メニュー→［エンコードを指定して開く ...］からオープンした場合 ファイル・エンコードの選択 ダイアログをオープンします。
キャンセル	本ダイアログをクローズします。

既存のサブプロジェクトを追加 ダイアログ

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，サブプロジェクト・ファイルを選択します。

図 A—76　既存のサブプロジェクトを追加 ダイアログ



ここでは，以下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，または［プロジェクト］メニュー→［追加］→［既存のサブプロジェクトを追加 ...］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］→［既存のサブプロジェクトを追加 ...］を選択

［各エリアの説明］

(1)［ファイルの場所］エリア

追加す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

追加するサブプロジェクトのサブプロジェクト・ファイル名を指定します。

(4)［ファイルの種類］エリア

以下のファイルの種類（ファイル・タイプ）が表示されます。

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

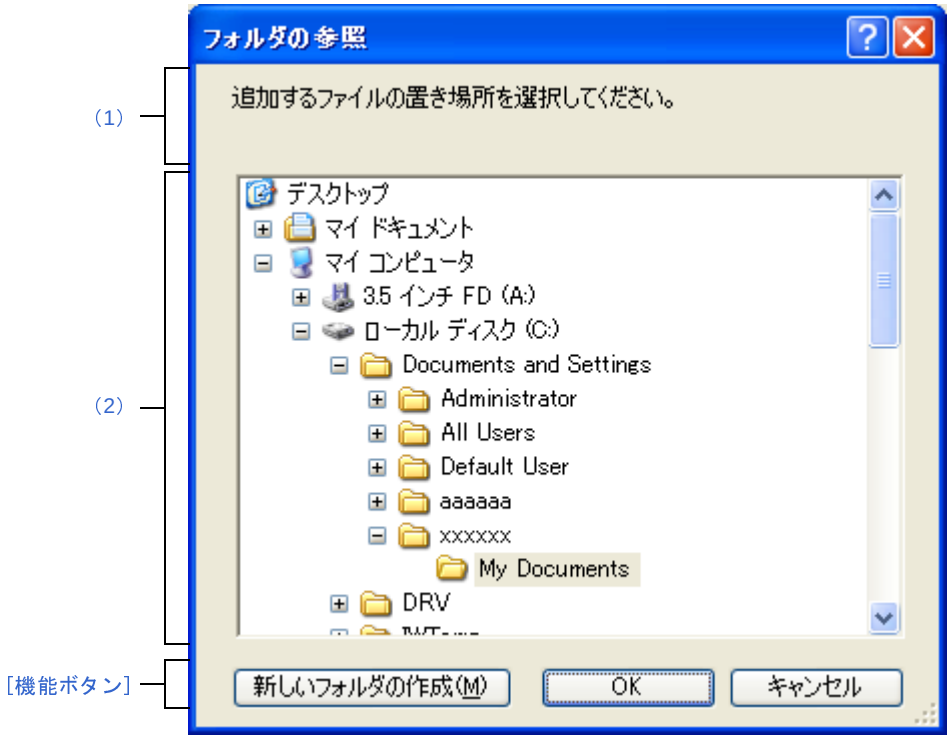
［機能ボタン］

ボタン	機能
開く	指定したサブプロジェクト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。

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，またはファイル（ソース・コード，レポート・ファイルなど）の出力先を設定します。

図 A—77　フォルダの参照 ダイアログ



ここでは，以下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］

- CubeSuite+ アップデート・マネージャ ウインドウの更新履歴 ページ→［コピー］ボタンをクリック
- ［プロジェクト］メニュー→［プロジェクトと開発ツールをパックして保存 ...］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・ノードを選択したのち，コンテキスト・メニュー→［プロジェクトと開発ツールをパックして保存 ...］を選択
- スタート パネルにおいて，［サンプル・プロジェクトを読み込む］エリアの［GO］ボタンをクリック
- プロジェクト作成 ダイアログにおいて，プロジェクト・ファイル・エリア内の［参照 ...］ボタンをクリック
- ソース・コンバート設定 ダイアログ【CX】において，［流用元プロジェクトのバックアップ］エリア内の［参照 ...］ボタンをクリック
- 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

- 検索・置換 ダイアログにおいて，［一括検索］タブの［...］ボタンをクリック，または［一括置換］タブの［...］ボタンをクリック
- パック設定 ダイアログにおいて，［保存場所］エリア内の［参照 ...］ボタンをクリック

［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，またはファイル（ソース・コード，レポート・ファイルなど）を出力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）［プロジェクト］メニュー，およびプロジェクト・ツリー パネルの場合

初回は“C:¥Documents and Settings¥*ユーザ名*¥My Documents”，2 回目以降は前回指定したフォルダが選択されます。

（b）検索・置換 ダイアログ，ソース・コンバート設定 ダイアログ，およびファイル追加 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。

（c）プロジェクト作成 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，“C:¥Documents and Settings¥*ユーザ名*¥My Documents”が選択されます。

（d）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。

（e）CubeSuite+ アップデート・マネージャ ウインドウの場合

“C:¥Documents and Settings¥*ユーザ名*¥デスクトップ”が選択されます。

［機能ボタン］

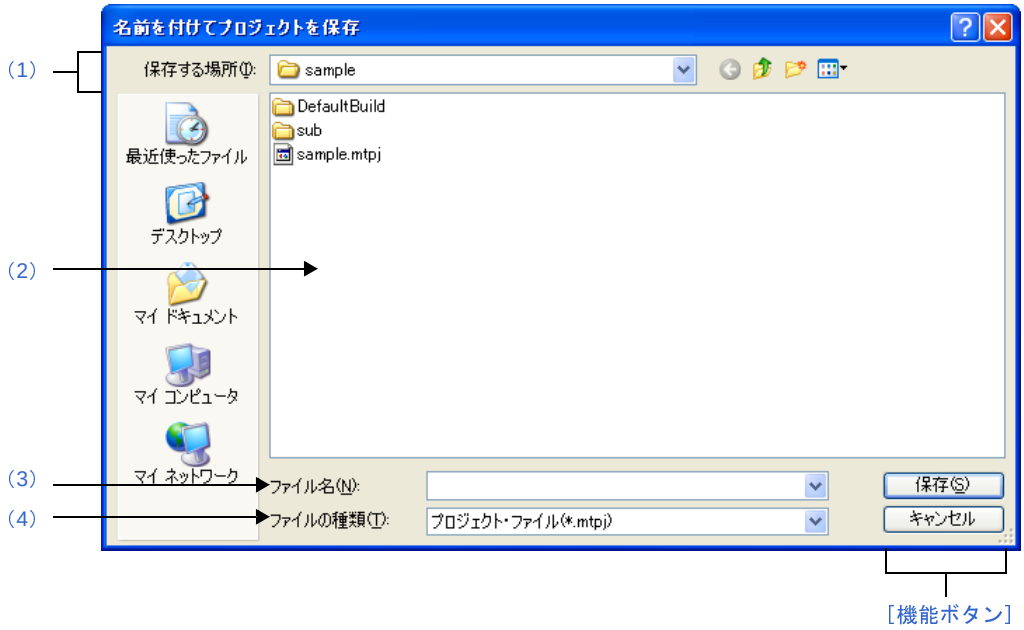
ボタン	機能
新しいフォルダの作成	選択したフォルダの直下に新しいフォルダを作成します。 フォルダ名は，デフォルトで“新しいフォルダ”となります。

ボタン	機能
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。 ファイルの出力先をフォルダの場所エリアで選択されたフォルダに設定します。
キャンセル	本ダイアログをクローズします。

名前を付けてプロジェクトを保存 ダイアログ

プロジェクト・ファイルを別名で保存します。

図 A—78　名前を付けてプロジェクトを保存 ダイアログ



ここでは，以下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，または［プロジェクト］メニュー→［名前を付けてプロジェクトを保存 ...］を選択

［各エリアの説明］

(1)［保存する場所］エリア

プロジェクト・ファイルを保存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

［保存する場所］，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

保存する際のプロジェクト・ファイル名を指定します。

(4)［ファイルの種類］エリア

以下のファイルの種類（ファイル・タイプ）が表示されます。

プロジェクト・ファイル (*.mtpj)	プロジェクト・ファイル

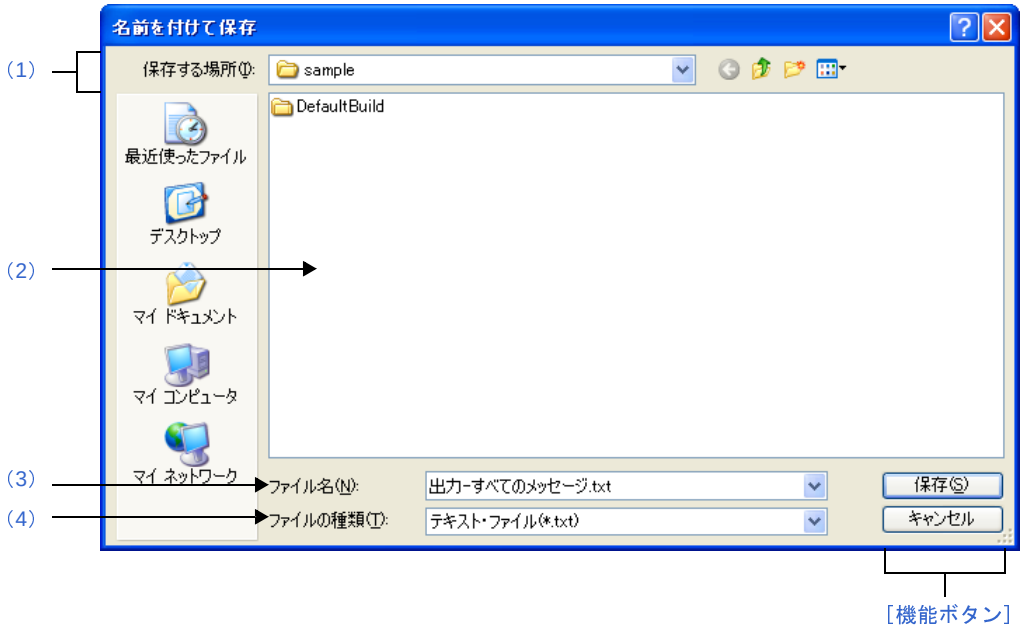
［機能ボタン］

ボタン	機能
保存	指定したファイル名でプロジェクト・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。

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名前を付けてファイルに保存します。

図 A—79　名前を付けて保存 ダイアログ



ここでは，以下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
- Python コンソール 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 Python コンソールを保存 ...］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。

(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

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

編集中のファイルの種類に依存して，以下のファイルの種類（ファイル・タイプ）が表示されます。

C ソース・ファイル (*.c)	C ソース・ファイル
C++ ソース・ファイル (*.cpp; *.cc; *.cp)【CC-RX】	C++ ソース・ファイル
ヘッダ・ファイル (*.h; *.hpp; *.inc)【CC-RX】	ヘッダ・ファイル
ヘッダ・ファイル (*.h; *.inc)【CA850】【CX】【CA78K0R】【CA78K0】	ヘッダ・ファイル
アセンブル・ファイル (*.src)【CC-RX】	アセンブラ・ソース・ファイル
アセンブル・ファイル (*.s)【CA850】【CX】	アセンブラ・ソース・ファイル
アセンブル・ファイル (*.asm)【CX】【CA78K0R】【CA78K0】	アセンブラ・ソース・ファイル
アセンブル・ファイル (*.asm; *.s) 注1	アセンブラ・ソース・ファイル
リンク・ディレクティブ・ファイル (*.dir; *.dr)【CA850】【CX】	リンク・ディレクティブ・ファイル
リンク・ディレクティブ・ファイル (*.dr; *.dir)【CA78K0R】【CA78K0】	リンク・ディレクティブ・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
セクション・ファイル (*.sf)【CA850】	セクション・ファイル
シンボル情報ファイル (*.sfg)【CX】	シンボル情報ファイル
変数／関数情報ファイル (*.vfi)【CA78K0R】	変数／関数情報ファイル
変数情報ファイル (*.vfi)【CA78K0】	変数情報ファイル
関数情報ファイル (*.fin) 注2【CA78K0】	関数情報ファイル
マップ・ファイル (*.map)【CA850】【CX】【CA78K0R】【CA78K0】	マップ・ファイル
マップ・ファイル (*.map; *.lbp)【CC-RX】	マップ・ファイル
シンボル・テーブル・ファイル (*.sym)【CA78K0R】【CA78K0】	シンボル・テーブル・ファイル
ヘキサ・ファイル (*.hex)【CC-RX】【CA850】【CX】	ヘキサ・ファイル
ヘキサ・ファイル (*.hex; *.hxb; *.hxf)【CA78K0R】【CA78K0】	ヘキサ・ファイル
アセンブル・リスト・ファイル (*.lst)【CC-RX】	アセンブル・リスト・ファイル
S レコード・ファイル (*.mot)【CC-RX】	S レコード・ファイル
クロス・リファレンス・ファイル (*.cref)【CC-RX】	クロス・リファレンス・ファイル
リンク・サブコマンド・ファイル (*.clnk)【CC-RX】	リンク・サブコマンド・ファイル

テキスト・ファイル (*.txt)	テキスト形式

注 1. デバッグ専用プロジェクトのみ

2. メモリ・バンク搭載品のみ

(b) 出力 パネルの場合

以下のファイルの種類（ファイル・タイプ）が表示されます。

テキスト形式でのみ保存することができます。

テキスト・ファイル (*.txt)	テキスト形式

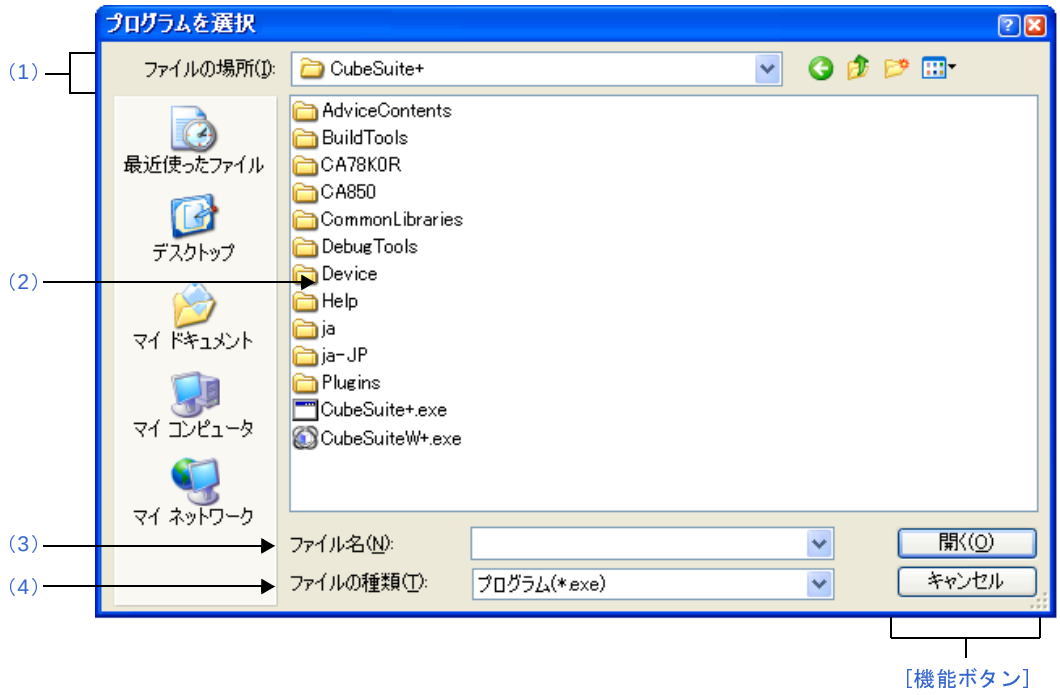
[機能ボタン]

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。

プログラムを選択 ダイアログ

外部ツールの実行ファイルを選択します。

図 A—80　プログラムを選択 ダイアログ



ここでは，以下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]

- オプション ダイアログの［全般 - 外部ツール］カテゴリにおいて，新規登録エリアの［...］ボタンをクリック

[各エリアの説明]

(1)［ファイルの場所］

登録する外部ツールの実行ファイルが存在する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］

登録する外部ツールの実行ファイル名を指定します。

(4)［ファイルの種類］

登録する外部ツール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

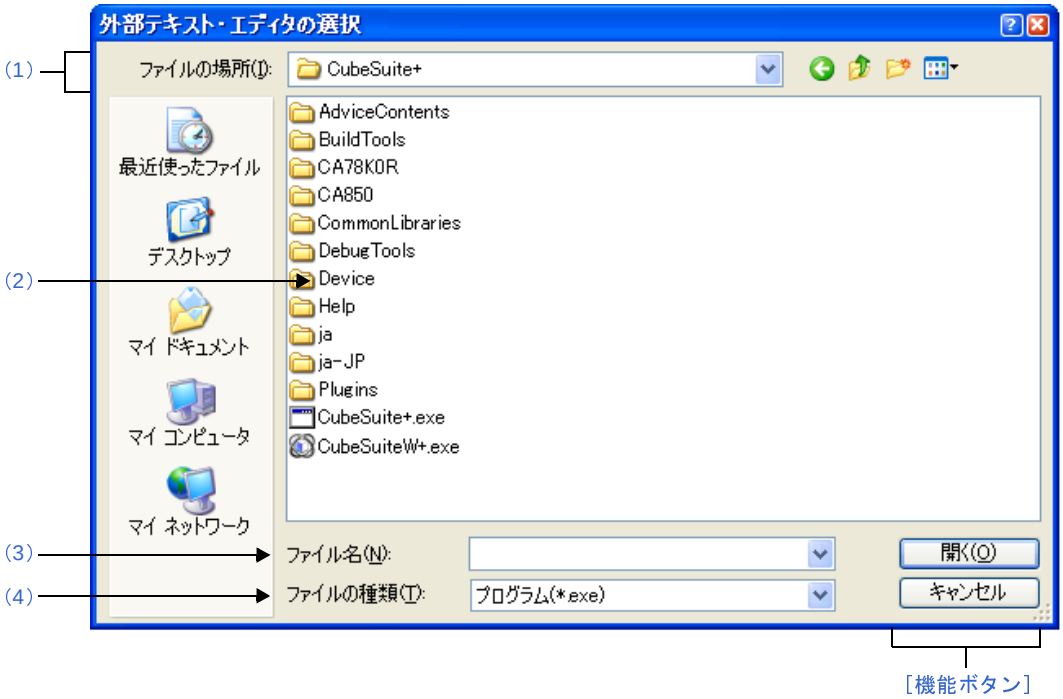
［機能ボタン］

ボタン	機能
開く	指定した実行ファイルをオプション ダイアログに指定します。
キャンセル	本ダイアログをクローズします。

外部テキスト・エディタの選択 ダイアログ

外部テキスト・エディタの実行ファイルを選択します。

図 A—81　外部テキスト・エディタの選択 ダイアログ



ここでは，以下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]

- オプション ダイアログの［全般 - テキスト・エディタ］カテゴリにおいて，［外部テキスト・エディタ］エリアの［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］

登録する外部テキスト・エディタの実行ファイルが存在する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］

外部テキスト・エディタの実行ファイル名を指定します。

(4)［ファイルの種類］

登録する外部テキスト・エディタ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	指定した実行ファイルをオプション ダイアログに指定します。
キャンセル	本ダイアログをクローズします。

Python コンソール パネル

IronPython を利用して，コマンド入力方式で CubeSuite+，およびデバッグ・ツールを操作します。

図 A—82　Python コンソール パネル

ここでは，以下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（Python コンソール 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→［Python コンソール］の選択

［各エリアの説明］

（1）入出力エリア

IronPython 関数や制御文，および CubeSuite+ Python 関数を入力，実行します。

また，関数の実行結果やエラーの表示も行います。

IronPython 関数の結果を表示する場合は print 文を使用してください。

[[ファイル］メニュー（Python コンソール パネル専用部分）]

Python コンソール パネル専用の［ファイル］メニューは以下のとおりです（その他の項目は共通です）。

Python コンソール を保存	現在パネル上に表示されている内容を，前回保存したテキスト・ファイル（*.txt）に保存します。 なお，起動後にはじめてこの項目を選択した場合は，［名前を付けて Python コンソールを保存 ...］の選択と同等の動作となります。
名前を付けて Python コンソール を保存 ...	現在パネル上に表示されている内容を，指定したテキスト・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。

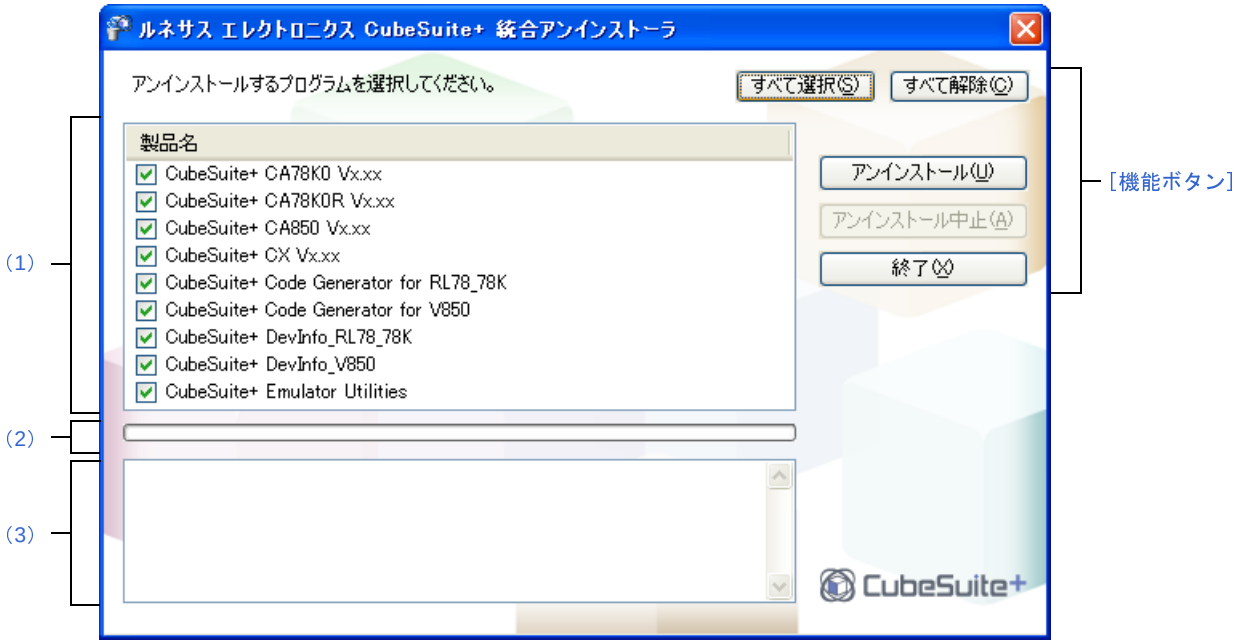
[コンテキスト・メニュー]

切り取り	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・ボードの内容をキャレット位置に挿入します。
すべて選択	このパネルに表示しているすべての文字列を選択状態にします。
強制停止	実行中のコマンドを強制停止します。
クリア	出力結果をすべてクリアします。
Python を初期化	Python を初期化します。

統合アンインストーラ ウインドウ

インストールされている CubeSuite+ 製品のアンインストールを一括で指定します。

図 A—83　統合アンインストーラ ウインドウ



ここでは，以下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]

- Windows の［スタート］→［すべてのプログラム］→［Renesas Electronics CubeSuite+］→［統合アンインストーラ］を選択

[各エリアの説明]

(1) ツール選択エリア

インストールされている CubeSuite+ 製品が表示されるので，アンインストール対象のツールを選択しチェックを入れます。

(2) 進行状況グラフ・エリア

アンインストーラの進行状況がプログレスバーで表示されます。

(3) 進行状況詳細表示エリア

アンインストール対象のツール毎に，アンインストール実行中，および完了の情報が表示されます。

[機能ボタン]

ボタン	機能
すべて選択	すべてのチェック・ボックスにチェックを入れます。
選択解除	チェック・ボックスのすべてのチェックをはずします。
アンインストール	選択されたツールのアンインストールを開始します。
アンインストール中止	アンインストールを中断します。
終了	本ウインドウを閉じます。

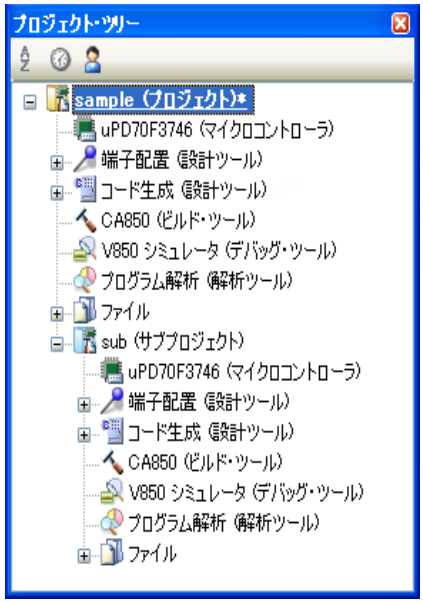
付録 B　用語説明

ここでは，CubeSuite+ のマニュアルで使用される用語について説明します。

(1) ノード

プロジェクト・ツリー パネルなどで表示されるようなツリー構造で表現されるもので，線で結ばれたツリーの先に記述されたひとつひとつの項目のことです。

図 B—1　プロジェクト・ツリー パネル



(2) リスト・コントロール

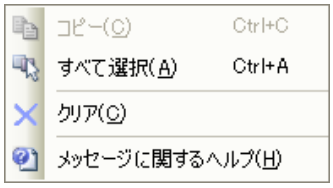
各パネルの設定エリアにおいて，使用しない のように表示されるものです。

部分をクリックすると，選択可能な項目のリストが表示されます。

(3) コンテキスト・メニュー

各ウインドウ，各種アイコンなど，画面上のオブジェクト上でマウスを右クリックしたときに出てくるメニューのことです。その対象について現在利用可能な操作の一覧が表示されます。

図 B—2　コンテキスト・メニュー例

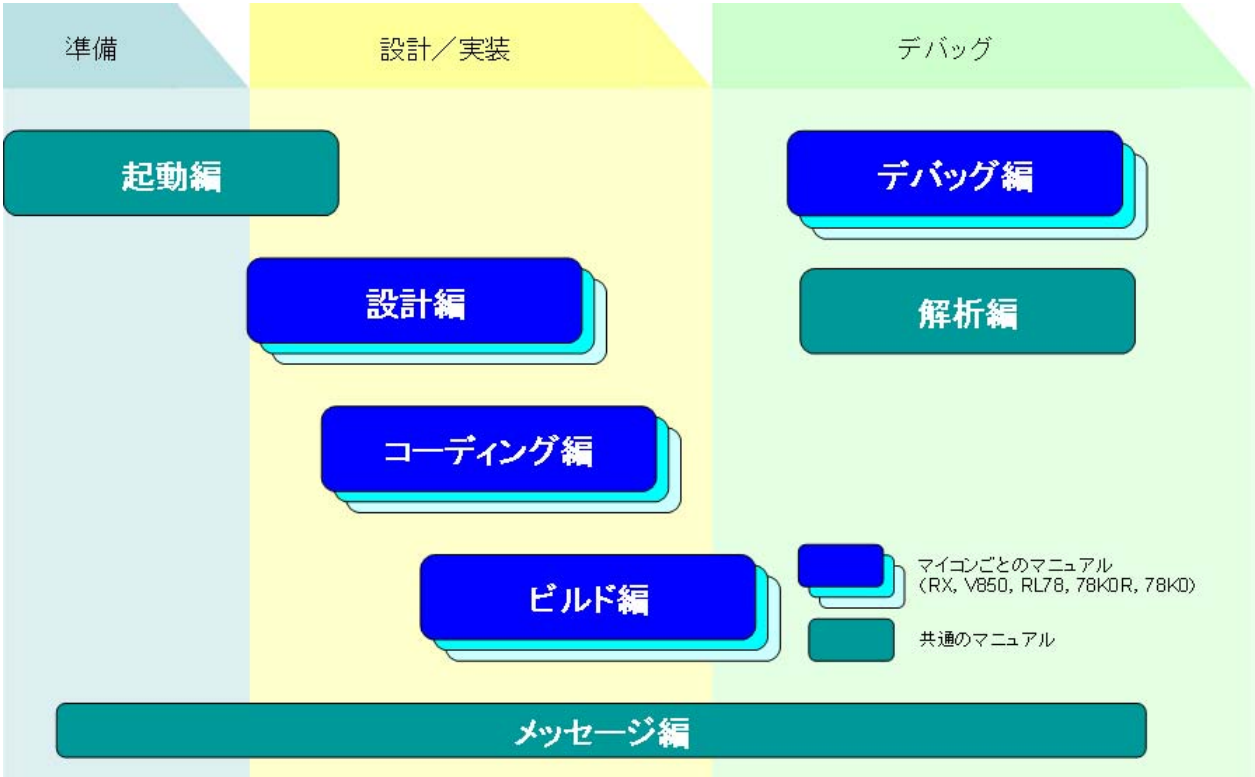


付録 C　マニュアル構成

ここでは，マイクロコントローラ（RX，V850，RL78，78K0R，78K0）用の統合開発環境である CubeSuite+ のマニュアルについて説明します。

マニュアルの構成は，ソフトウエアの開発フェーズに合わせて，以下のようになっています。

図 C—1　マニュアル一覧



備考 1.　インストールした製品により参照できるマニュアルは異なります。

2.　起動編では，外部ビルド・ツールを使用する場合を対象に，“設計／実装”の工程に関わるビルド関連の記載を含みます。

（1）起動編

CubeSuite+ の全体概要について説明しています。

また，インストール，アップデート，ライセンスの設定などについて説明しています。

CubeSuite+ の起動からプロジェクトの作成までの動作を説明しています。

（2）設計編

設計ツール（端子配置，およびコード生成）について説明しています。

(3) コーディング編

CubeSuite+ に含まれている C コンパイラ，アセンブラなどのコマンドの役割，および機能について説明しています。C コンパイラ，アセンブラを使用した開発において必要となる情報，および効果的なソース記述のノウハウを提供しています。

(4) ビルド編

ビルド・ツールについて説明しています。ビルド・ツールは，CubeSuite+ が提供しているコンポーネントで構成されており，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，ヘキサ・ファイル，ライブラリ・ファイルのいずれかを，目的に応じて生成することができます。

(5) デバッグ編

デバッグ・ツールの接続設定や実行制御について説明しています。また，接続しているデバッグ・ツールを使用して，プログラムのデバッグを行う場合の情報を提供しています。

(6) 解析編

解析ツールについて説明しています。解析ツールは，ソース・プログラム，およびプログラム実行時の情報を解析し，関数／変数情報を提供します。

(7) メッセージ編

出力メッセージの説明をしています。

付録 D　入力規約

ここでは，入力規約について説明します。

D. 1　入力規約

以下に，CubeSuite+ が提供するパネル／ダイアログの各種情報を設定する際の入力規約を示します。

（1）文字セット

文字を入力する際に許可している文字セットは，以下のとおりです。

表 D—1　文字セットの一覧

文字セット	概要
ASCII	半角のアルファベット（英字），半角の数字，半角の記号
Shift-JIS	全角のアルファベット（英字），全角の数字，全角の記号，全角のひらがな，全角のカタカナ，全角の漢字，および半角のカタカナ
EUC-JP	全角のアルファベット（英字），全角の数字，全角の記号，全角のひらがな，全角のカタカナ，全角の漢字，および半角のカタカナ
UTF-8	全角のアルファベット（英字），全角の数字，全角の記号，全角のひらがな，全角のカタカナ，全角の漢字（中国語を含む），および半角のカタカナ

（2）数値

数値を入力する際に許可している進数は，以下のとおりです。

表 D—2　進数の一覧

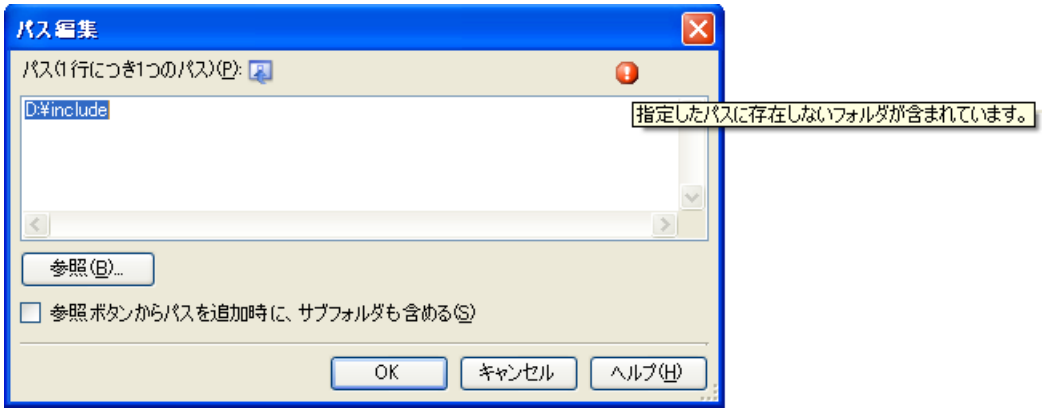
進数表記	概要
10 進数	1 ～ 9 の数字で始まり 0 ～ 9 の数字が続く数値，および 0
16 進数	0x で始まり 0 ～ 9 の数字，および a ～ f の英字が続く数値 （英字の大文字／小文字については，不問）

D. 2　入力不備箇所に対するアイコン表示

CubeSuite+ が提供する一部のパネル／ダイアログでは，不正な文字列が入力された際，および入力が必須な箇所に値が未入力の場合に，設定すべき情報として誤っていることを示すアイコンを該当箇所に表示し，入力の不備を警告します。

備考　アイコン上にマウス・カーソルを移動した際には，入力すべき文字列に関する情報（入力の不備を解消するためのヒント）がポップアップ表示されます。

図 D—1　入力不備箇所に対するアイコン表示



付録 E　正規表現の構文

ここでは，検索・置換 ダイアログで使用する正規表現についての詳細を説明します。
CubeSuite+ では，Microsoft .NET の正規表現の構文に基づいた正規表現をサポートします。

備考　空白文字はすべての正規表現の文字列で無視されます。また，¥s を使うと，検索文字列に空白文字を指定することができます。

E. 1　文字のエスケープ

以下の表に，正規表現で使用できるエスケープ文字とシーケンスを示します。

表 E—1　文字のエスケープ一覧

エスケープ文字	説明
（通常の文字）	. $ ^ { [(|) * + ? ¥以外の文字は，文字自体に一致します。
¥a	ベル（アラート）の¥u0007 と一致します。
¥t	タブの¥u0009 と一致します。
¥r	キャリッジ・リターンの¥u000D と一致します。
¥v	垂直タブの¥u000B と一致します。
¥f	フォーム・フィードの¥u000C と一致します。
¥n	改行文字の¥u000A と一致します。
¥e	エスケープ文字の¥u001B と一致します。
¥040	3 桁までの 8 進数で表される ASCII 文字と一致します。文字¥040 は，空白を表します。
¥x20	2 桁までの 16 進数で表される ASCII 文字と一致します。
¥u0020	4 桁までの 16 進数で表される Unicode 文字と一致します。
¥	エスケープ文字として認識されない文字が後ろに付いている場合は，その文字と一致します。たとえば，¥* は¥x2A と同じです。

E. 2　文字クラス

以下の表に，文字と一致する構文を示します。

表 E—2　文字クラス一覧

文字クラス	説明
.	¥n を除く任意の文字に一致します。文字クラスの中にある場合，ピリオド文字として扱われます。
[aeiou]	指定された文字セットに含まれる任意の文字と一致します。

文字クラス	説明
[^aeiou]	指定された文字セットに含まれない任意の文字と一致します。
[0-9a-fA-F]	ハイフン（-）を使って連続した文字の範囲を指定できます。
¥p{name}	name で指定された Unicode 一般カテゴリ内の任意の文字に一致します（例：Ll，Nd，Z）。Unicode 一般カテゴリについての詳細は，「E. 3　サポートされている Unicode 一般カテゴリ」を参照してください。
¥w	文字，数字，アンダースコアを含む，単語に使用される任意の文字と一致します。
¥W	単語に使用される文字以外の任意の文字と一致します。
¥s	空白文字と一致します。
¥S	空白以外の文字と一致します。
¥d	10 進数字と一致します。
¥D	数字以外と一致します。
[.¥w¥s]	¥w や¥s などの，エスケープされた組み込み文字クラスは，文字クラスの中で使用できます。この例では，任意のピリオド，単語，または空白文字と一致します。

E. 3　サポートされている Unicode 一般カテゴリ

以下の表に，サポートされている Unicode 一般カテゴリを示します。これらのカテゴリは，¥p および¥P 文字クラスとあわせて使用できます。文字クラスについての詳細は，「E. 2　文字クラス」を参照してください。

表 E—3　サポートされている Unicode 一般カテゴリ一覧

Unicode 一般カテゴリ	説明
Lu	Letter, Uppercase（文字，大文字）
Ll	Letter, Lowercase（文字，小文字）
Lt	Letter, Titlecase（文字，タイトル文字）
Lm	Letter, Modifier（文字，修飾）
Lo	Letter, Other（文字，その他）
Mn	Mark, Nonspacing（結合文字，幅なし）
Mc	Mark, Spacing Combining（結合文字，幅あり）
Me	Mark, Enclosing（結合文字，囲み）
Nd	Number, Decimal Digit（数字，10 進数字）
Nl	Number, Letter（数字，文字）
No	Number, Other（数字，その他）
Pc	Punctuation, Connector（句読点，接続）
Pd	Punctuation, Dash（句読点，ダッシュ）
Ps	Punctuation, Open（句読点，開き）
Pe	Punctuation, Close（句読点，閉じ）
Pi	Punctuation, Initial quote（句読点，開始引用符）
Pf	Punctuation, Final quote（句読点，終了引用符）

Unicode 一般カテゴリ	説明
Po	Punctuation, Other（句読点，その他）
Sm	Symbol, Math（記号，数学）
Sc	Symbol, Currency（記号，通貨）
Sk	Symbol, Modifier（記号，修飾）
So	Symbol, Other（記号，その他）
Zs	Separator, Space（区切り，空白）
Zs	Separator, Line（区切り，行）
Zl	Separator, Paragraph（区切り，段落）
Zp	Other, Control（区切り，制御）
Cc	Other, Control（区切り，制御）
Cf	Other, Format（その他，書式）
Cs	Other, Surrogate（その他，サロゲート）
Co	Other, Private Use（その他，プライベート用途）
Cn	Other, Not Assigned（その他，未割り当て）

以下の表に示す Unicode 文字カテゴリのセットを表す追加のカテゴリも用意されています。

表 E—4　Unicode 文字カテゴリのセット一覧

カテゴリ	説明
C	（すべての制御文字）Cc，Cf，Cs，Co，および Cn
L	（すべての文字）Lu，Ll，Lt，Lm，および Lo
M	（すべての分音記号）Mm，Mc，および Me
N	（すべての数字）Nd，Nl，および No
P	（すべての句読点）Pc，Pd，Ps，Pe，Pi，Pf，および Po
S	（すべての記号）Sm，Sc，Sk，および So
Z	（すべての区切り）Zs，Zl，および Zp

E. 4　量指定子

量指定子は，オプションの量データを正規表現に追加します。量指定子は，その直前の文字，グループ，または文字クラスに適用されます。

以下の表に，一致する量に影響するメタ文字を示します。

表 E—5　量指定子一覧

量指定子	説明
*	0 回以上の一致を指定します（¥ w*，(abc)* など）。{0,} と等価です。
+	1 回以上の一致を指定します（¥ w+，(abc)+ など）。{1,} と等価です。

量指定子	説明
?	0 回または 1 回の一致を指定します（¥w?，(abc)? など）。{0,1} と等価です。
{n}	ちょうど n 回の一致を指定します（(pizza){2} など）。
{n,}	n 回以上の一致を指定します（(abc){2,} など）。
{n,m}	n 回以上 m 回以下の一致を指定します。

E. 5　アトミック・ゼロ幅アサーション

以下の表に，アトミック・ゼロ幅アサーションを示します。以下の表に示されているメタ文字は，後方の文字列を読み込んだり，複数の文字と一致したりはしません。メタ文字は単に，文字列内の現在の位置での一致が成立するかどうかを制御します。これらは，単純に文字列での現在の位置によって，一致するかしないかが決まります。

表 E—6　アトミック・ゼロ幅アサーション一覧

アサーション	説明
^	文書の先頭または行の先頭で一致する必要があることを指定します。 たとえば，^#region は，行の先頭で一致した文字列 #region だけを返します。
$	文字列の末尾，文字列の末尾の¥n の前，または行の末尾で一致する必要があることを指定します。
¥A	文書の先頭で一致する必要があることを指定します。
¥z	文書の末尾で一致する必要があることを指定します。
¥b	¥w（英数字）と¥W（英数字以外）の境界位置で一致する必要があることを指定します。
¥B	¥b 境界以外で一致する必要があることを指定します。

E. 6　グループ化構成体

以下の表に，グループ化構成体を示します。グループ化構成体は，部分式のグループをキャプチャします。また，修飾の先と後をキャプチャなしで読むことで，正規表現の効率性を高めます。

表 E—7　グループ化構成体一覧

グループ化構成体	説明
( )	検索／置換操作で一致した部分文字列をキャプチャします。
(?=)	ゼロ幅の正の先読みアサーションです。部分式がこの位置の右で一致した場合にだけ照合を継続します。たとえば，(_?=¥w) は，アンダースコアの後に単語が続いている場合に一致し，単語とは一致しません。
(?!)	ゼロ幅の負の先読みアサーションです。部分式がこの位置の右で一致しない場合にだけ照合を継続します。たとえば，¥b(?!un)\w+¥b は，先頭が un でない単語と一致します。
(?<=)	ゼロ幅の正の後読みアサーションです。部分式がこの位置の左で一致した場合にだけ照合を継続します。たとえば，(?<=19)99 は，19 に続く 99 のインスタンスと一致します。
(?<!)	ゼロ幅の負の後読みアサーションです。部分式がこの位置の左で一致しない場合にだけ照合を継続します。

E. 7　置換構成体

置換構成体は，検索／置換のパターンでのみ，使用できます。エスケープ文字と置換構成体は，置換パターンで認識される特殊な構成体です。

以下の表に，名前付き置換パターンおよび番号付き置換パターンを定義する方法を示します。

表 E—8　置換構成体一覧

文字	説明
$1	グループ番号 1（10 進数）と一致した最後の部分文字列に置換されます。2 番目のグループは 2（$2）となり，以降同様です。 たとえば，置換パターン a*$1b の場合，文字列 a* の後に，最初にキャプチャしたグループが一致する部分文字列，さらに文字列 b（指定されている場合）が続くような文字列を挿入します。
$0	一致した部分のコピーに置換されます。
$&	一致した部分のコピーに置換されます。
$$	単一の $ リテラルに置換されます。

備考 1.　* 文字は，置換パターンではメタ文字として認識されません。

2.　$ パターンは，正規表現一致パターンでは認識されません。正規表現では，$ は文字列の末尾を示します。

E. 8　その他の構成体

以下の表に，その他の正規表現の構成体を示します。

表 E—9　その他の構成体一覧

構成体	説明
" "	固定文字列を囲みます。
{ }	語彙マクロを呼び出します。WordMacro（¥ w と似ています）を使うと，{WordMacro} のように表されます。
(?#)	正規表現に挿入するインライン・コメントです。コメントは，最初の右かっこ文字で終了します。
|	|（縦棒）で区切られた単語のいずれかに一致する構造体です。たとえば，cat|dog|tiger のように指定します。一致した単語のうち，最も左側にあるものが優先されます。

付録 F　外部ビルド・ツールの使用

ここでは，CubeSuite+ が提供するビルド・ツール（CC-RX/CA850/CX/CA78K0R/CA78K0）以外のビルド・ツールにより生成されたファイルを対象にデバッグを行う場合のプロジェクトの作成方法について説明します。

F. 1　概　　要

CubeSuite+ では，各プロジェクトごとに CubeSuite+ が提供するビルド・ツール（CC-RX/CA850/CX/CA78K0R/CA78K0）により出力されるロード・モジュール・ファイル／ヘキサ・ファイルが，デバッグ対象のダウンロード・ファイルとして自動的に決定されます。

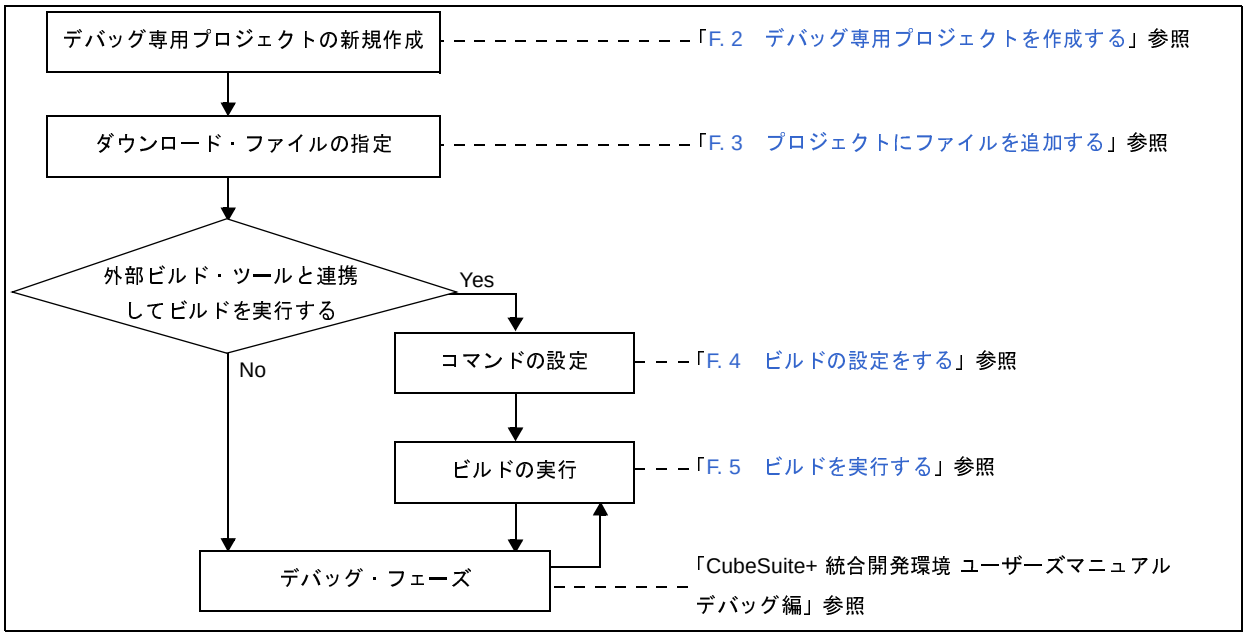
したがって，外部ビルド・ツール（CubeSuite+ が提供するビルド・ツール以外のコンパイラ／アセンブラなど）により作成されたロード・モジュール・ファイル／ヘキサ・ファイルをダウンロード・ファイルとして扱いデバッグを行うためには，専用のプロジェクト（以降，“デバッグ専用プロジェクト”と呼びます）を作成する必要があります。

このデバッグ専用プロジェクトを作成することにより，CubeSuite+ が提供するビルド・ツールを介することなくデバッグを行うことができます。

また，デバッグ専用プロジェクトでは，ユーザの実行環境に即したビルド処理を実行させるコマンド（make など）を設定することができ，このコマンドを CubeSuite+ から実行させることにより，外部ビルド・ツールと連携したビルド処理を行うことができます。

デバッグ専用プロジェクトを使用した操作の流れを以下に示します。

図 F—1　デバッグ専用プロジェクトを使用した操作の流れ



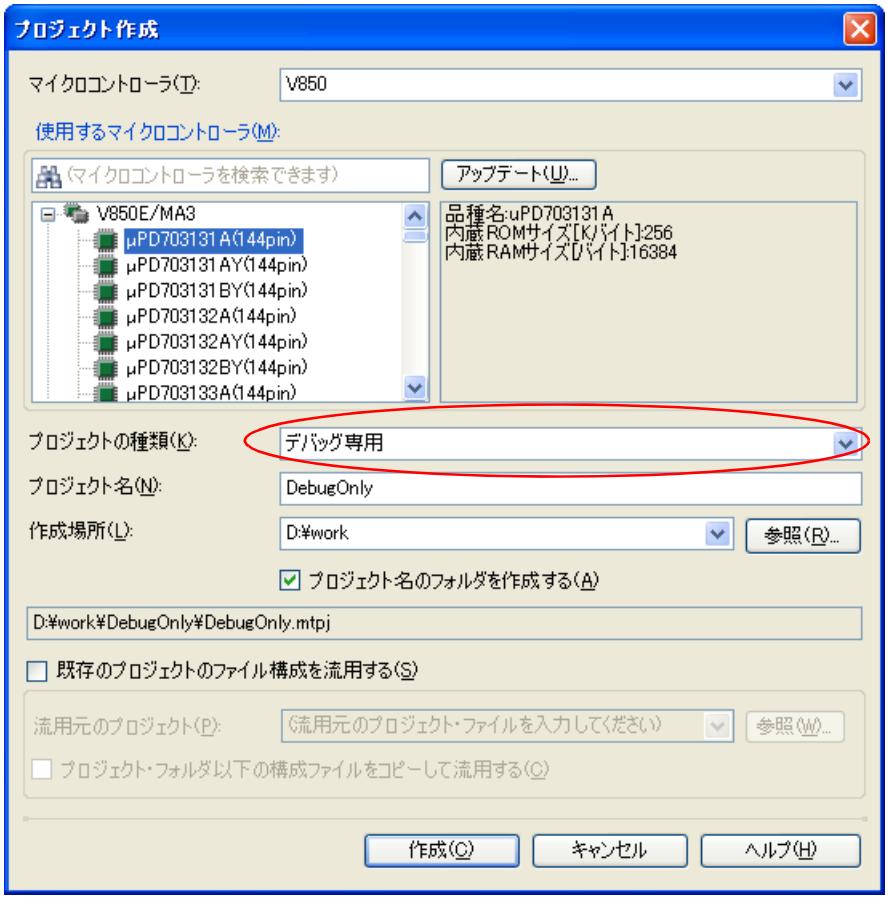
注意　デバッグ専用プロジェクトを対象に，設計ツール（コード生成），および解析ツールを使用することはできません。

F. 2　デバッグ専用プロジェクトを作成する

デバッグ専用プロジェクトの作成は，「2. 6. 2　新しいプロジェクトを作成する」，または「2. 6. 3　新しいサブプロジェクトを追加する」で示したプロジェクトの作成方法と同様に，［プロジェクト］メニュー→［新しいプロジェクトを作成 ...］，またはプロジェクト・ツリーでプロジェクト・ノードを選択したのちコンテキスト・メニューの［追加］→［新しいサブプロジェクトを追加 ...］を選択することでオープンするプロジェクト作成 ダイアログで行います。

ただし，デバッグ専用プロジェクトを作成するためには，プロジェクト作成 ダイアログ上の［プロジェクトの種類］において，［デバッグ専用］を選択します。

図 F—2　プロジェクト作成 ダイアログ（デバッグ専用プロジェクト作成時）

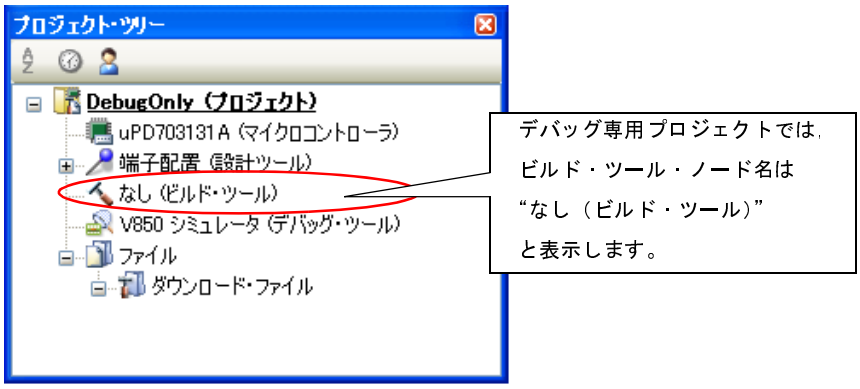


すべての項目の設定[注]を完了したのち，［作成］ボタンをクリックすると，［作成場所］で指定した場所にデバッグ専用プロジェクトのプロジェクト・ファイルを作成し，プロジェクト・ツリー パネルに作成したデバッグ専用プロジェクトの構成がツリー表示されます。

注　各項目の設定方法についての詳細は，以下を参照してください。

- 「2. 6. 2　新しいプロジェクトを作成する」参照
- 「2. 6. 3　新しいサブプロジェクトを追加する」参照

図 F—3　プロジェクト・ツリー パネル（デバッグ専用プロジェクト作成後）



F. 3　プロジェクトにファイルを追加する

作成したデバッグ専用プロジェクトに，デバッグ対象とするダウンロード・ファイルを追加します。

また，ダウンロードしたプログラムをソース・レベルでデバッグを行うために，ダウンロード・ファイルの作成元となったＣソース・ファイルをプロジェクトに追加します。

ここでは，デバッグ専用プロジェクトに，これらのファイルを追加する方法を説明します。

F. 3. 1　ダウンロード・ファイルを追加する

デバッグ専用プロジェクトに，デバッグ対象とするダウンロード・ファイルとして，ロード・モジュール・ファイルを追加します。

注意　ソース・レベル・デバッグ（ソース・レベル単位のステップ実行など）を行うためには，シンボル情報を持つロード・モジュール・ファイルを追加する必要があります。

備考 1.　指定したロード・モジュール・ファイルは，自動的にデバッグ・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］タブに反映されます。
なお，ロード・モジュール・ファイルの種類やダウンロードの際の各種オプションの設定，およびここで追加したロード・モジュール・ファイルに加え，ヘキサ・ファイル／バイナリ・ファイルもダウンロードする場合の設定は，この［ダウンロード・ファイル設定］タブ上で行ってください（ダウンロ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 デバッグ編」を参照してください）。

2.　複数のロード・モジュール・ファイルを追加することができます。【V850】

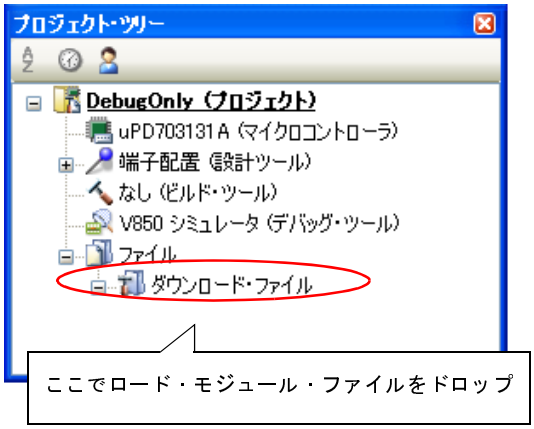
ロード・モジュール・ファイルを追加するには，以下の方法があります。

- 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

エクスプローラなどからロード・モジュール・ファイルをドラッグし，プロジェクト・ツリーのダウンロード・ファイル・ノード上でドロップしてください。

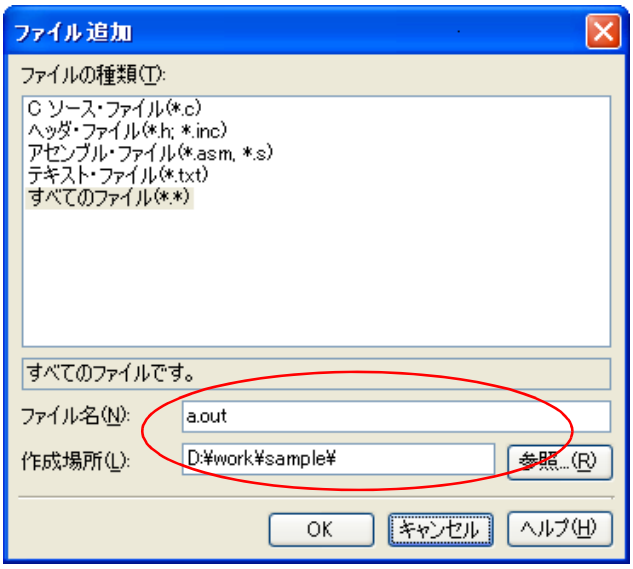
図 F—4　プロジェクト・ツリー パネル（ロード・モジュール・ファイルの追加）



(2) 空のファイルを作成して追加する場合

プロジェクト・ツリーのダウンロード・ファイル・ノードを選択し，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。

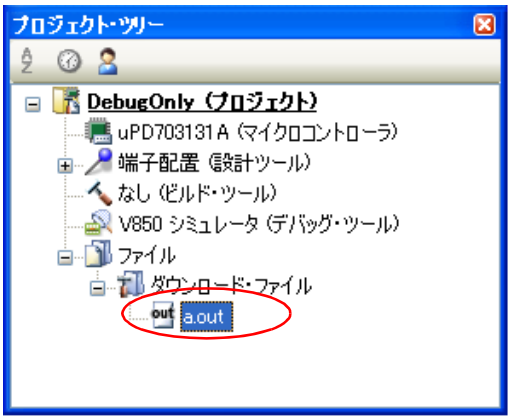
図 F—5　ファイル追加 ダイアログ（ロード・モジュール・ファイルの追加）



ダイアログ上で，ロード・モジュール・ファイルとして新しく作成するファイル名と作成場所を指定し，［OK］ボタンをクリックしてください。

ロード・モジュール・ファイルの追加完了後のプロジェクト・ツリーは，以下のようになります。

図 F—6　プロジェクト・ツリー パネル（ロード・モジュール・ファイル a.out 追加後）



F. 3. 2　C ソース・ファイル／その他のファイルを追加する

デバッグ専用プロジェクトに，ダウンロードするロード・モジュール・ファイルの作成元となった C ソース・ファイル，およびその他のファイルを追加します。

C ソース・ファイルを追加することにより，ソース・レベル・デバッグ（ソース・レベル単位のステップ実行など）を行うことができます。

また，必要に応じて，C ソース・ファイル以外のファイルを追加することもできます。

注意　ソース・レベル・デバッグを行うためには，ダウンロード・ファイルとして，シンボル情報を持つロード・モジュール・ファイルを追加する必要があります（「F. 3. 1　ダウンロード・ファイルを追加する」参照）。

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，デバッグ・ツールにダウンロードすると，ダウンロード・ファイルから取得したソース・ファイル情報を元に，プロジェクト・ツリーにソース・ファイルが自動的に追加されます。

注意 1.　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。

2.　プロジェクト・ツリーにソース・ファイルを自動追加する／しないは，オプション ダイアログの［全般 - ビルド／デバッグ］カテゴリで設定することができます。

上記の方法で追加できないファイルがある場合は，以下の方法で追加してください。

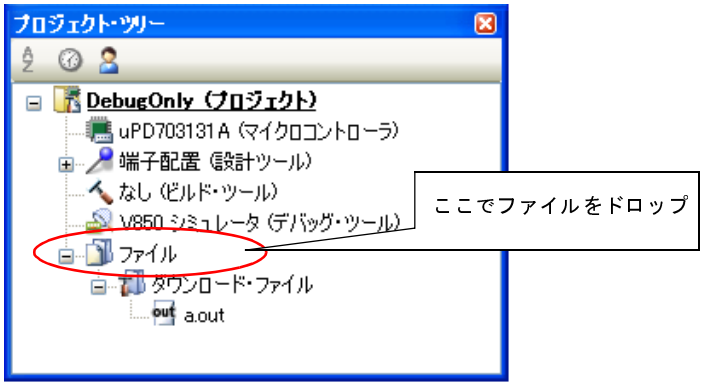
- 既存のファイルを追加する場合
- 空のファイルを作成して追加する場合

(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（複数選択可），プロジェクト・ツリーのファイル・ノード上でドロップしてください。

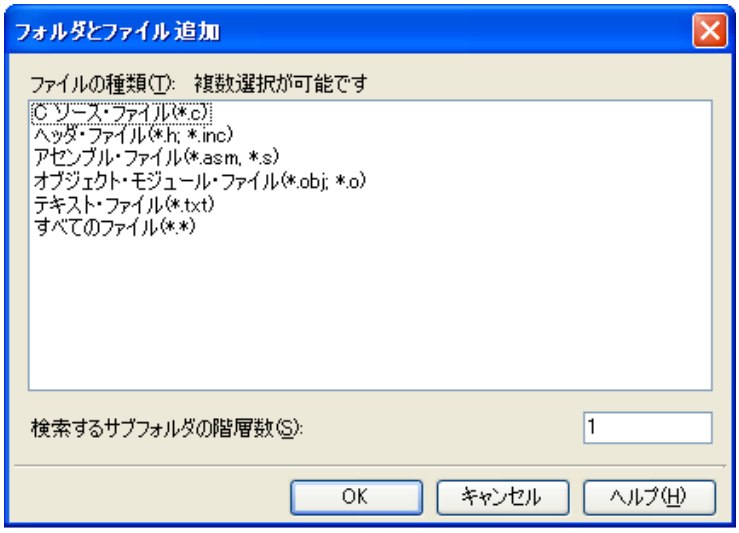
図 F—7　プロジェクト・ツリー パネル（ファイルの追加）



(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（複数選択可），プロジェクト・ツリーのファイル・ノード上でドロップすると，フォルダとファイル追加 ダイアログがオープンします。

図 F—8　フォルダとファイル追加 ダイアログ



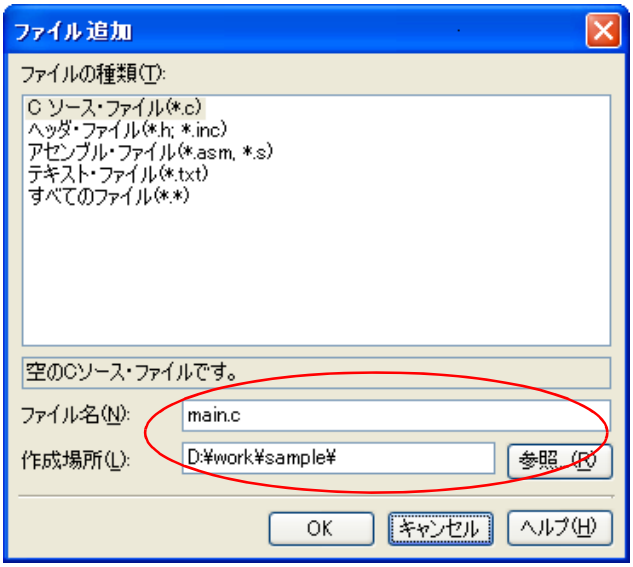
ダイアログ上で，デバッグ専用プロジェクトに追加するファイルの種類を選択し，追加するサブフォルダの階層数を指定したのち，［OK］ボタンをクリックしてください。

なお，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。

(2) 空のファイルを作成して追加する場合

プロジェクト・ツリーのファイル・ノードを選択し，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 F—9　ファイル追加 ダイアログ（ファイルの追加）



ダイアログ上で，新しく作成するファイル名と作成場所を指定し，［OK］ボタンをクリックしてください。

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。

なお，ファイル・ノード以下におけるファイルの追加位置は，プロジェクト・ツリー パネル上の［ツールバー］の現在の設定に依存します。

図 F—10　プロジェクト・ツリー パネル（ファイル main.c 追加後）

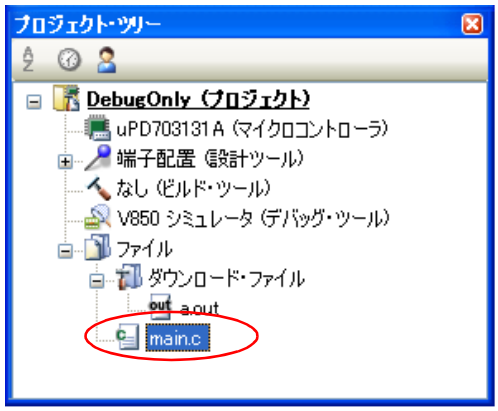
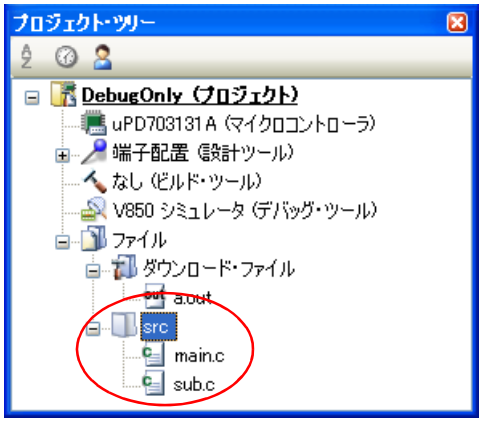


図 F—11　プロジェクト・ツリー パネル（フォルダ src 追加後）



追加したファイル名をプロジェクト・ツリー上 でダブルクリックすることによりエディタ パネルがオープンし，その内容を直接確認／編集することができます。

なお，エディタ パネルでオープン可能なファイルは以下のとおりです。

- C ソース・ファイル（*.c）
- C++ ソース・ファイル（*.cpp，*.cc，*.cp）【CC-RX】
- ヘッダ・ファイル（*.h，*.inc）
- ヘッダ・ファイル（*.hpp）【CC-RX】
- アセンブラ・ソース・ファイル（*.src）【CC-RX】
- アセンブラ・ソース・ファイル（*.s）【CA850】【CX】
- アセンブラ・ソース・ファイル（*.asm）【CX】【CA78K0R】【CA78K0】
- リンク・ディレクティブ・ファイル（*.dir，*.dr）【CA850】【CX】【CA78K0R】【CA78K0】
- リンク順指定ファイル（*.mtls）
- セクション・ファイル（*.sf）【CA850】
- シンボル情報ファイル（*.sfg）【CX】
- 変数／関数情報ファイル（*.vfi）【CA78K0R】
- 変数情報ファイル（*.vfi）【CA78K0】
- 関数情報ファイル（*.fin）【CA78K0】
- マップ・ファイル（*.map）
- マップ・ファイル（*.lbp）【CC-RX】
- シンボル・テーブル・ファイル（*.sym）【CA78K0R】【CA78K0】
- ヘキサ・ファイル（*.hex）
- ヘキサ・ファイル（*.hxb，*.hxf）【CA78K0R】【CA78K0】
- アセンブル・リスト・ファイル（*.lst）【CC-RX】
- S レコード・ファイル（*.mot）【CC-RX】
- クロス・リファレンス・ファイル（*.cref）【CC-RX】
- リンク・サブコマンド・ファイル（*.clnk）【CC-RX】
- テキスト・ファイル（*.txt）

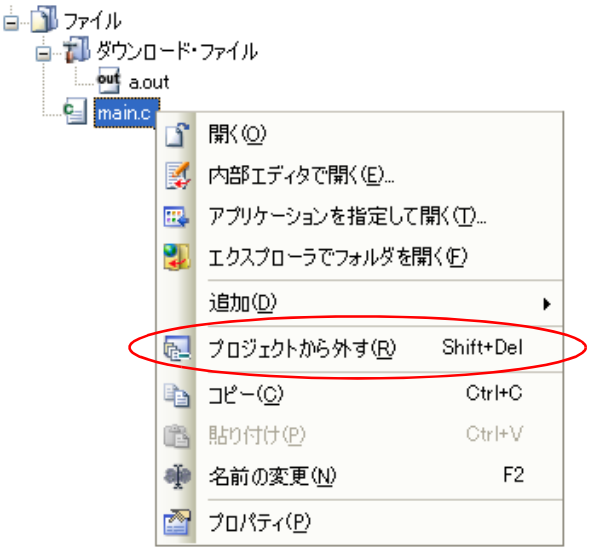
備考 1. ファイルをドラッグし，エディタ パネルにドロップすることにより，上記以外のファイルもエディタ パネルでオープンすることができます。

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。

F. 3. 3　追加したファイルをプロジェクトから外す

上記の操作により追加したファイルをデバッグ専用プロジェクトから外すには，プロジェクト・ツリー で対象のファイルを選択したのち，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 F—12　［プロジェクトから外す］項目



F. 4　ビルドの設定をする

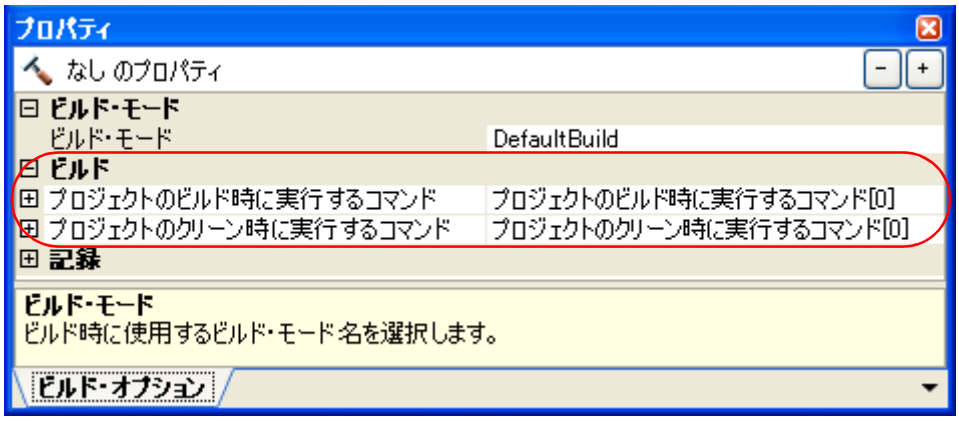
外部ビルド・ツール（CubeSuite+ が提供するビルド・ツール以外のコンパイラ／アセンブラなど）と連携してビルドを実行するための設定を行います。

F. 4. 1　コマンドを設定する

外部ビルド・ツールとの連携は，CubeSuite+ におけるビルド関連操作（「F. 5　ビルドを実行する」参照）を行った際に，ここで設定するコマンドを実行することにより実現します。

コマンドの設定は，プロジェクト・ツリーで対象デバッグ専用プロジェクト（メイン・プロジェクト，またはサブプロジェクト）のビルド・ツール・ノードを選択したのち，プロパティ パネルの［ビルド・オプション］タブ上の［ビルド］カテゴリで行います。

図 F—13　プロパティ パネル：［ビルド・オプション］タブ（［ビルド］カテゴリ）］



以下に，コマンドの設定方法を示します。

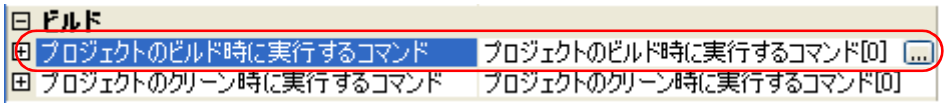
- ビルド実行時のコマンドの設定
- クリーン実行時のコマンドの設定

（1）ビルド実行時のコマンドの設定

ビルド時（F. 5. 1　ビルドを実行する」参照）に実行するコマンドを，［プロジェクトのビルド時に実行するコマンド］プロパティで設定します。

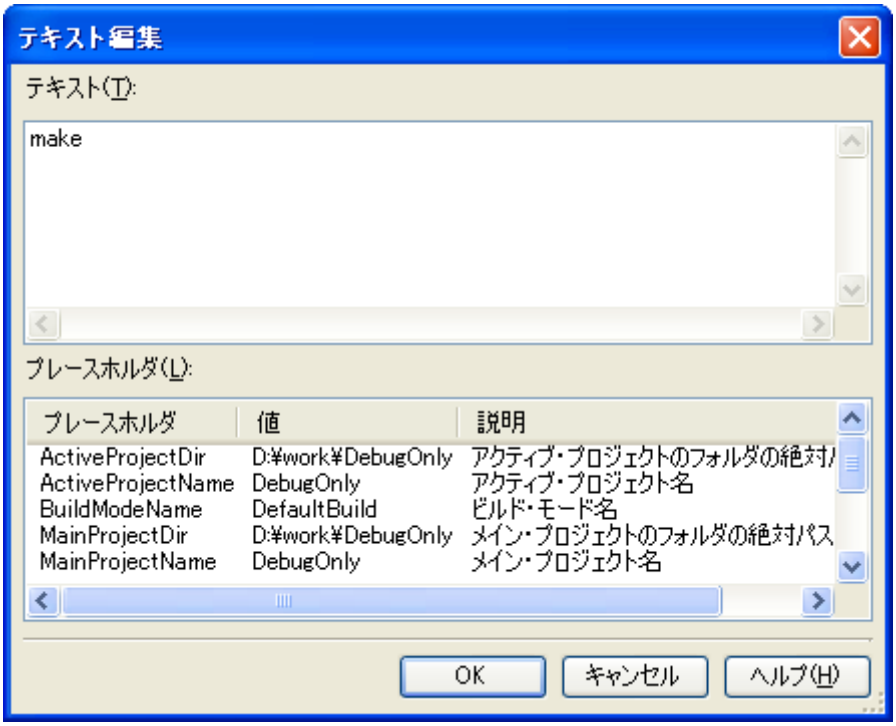
ここでは，必要となるビルド処理を実行させるコマンド（make など）を設定してください。

図 F—14　［プロジェクトのビルド時に実行するコマンド］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 F—15　テキスト編集 ダイアログ



［テキスト］に実行するコマンドを 1 行に 1 つずつ入力します。

コマンドは，1 行に 1023 文字まで，64 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。

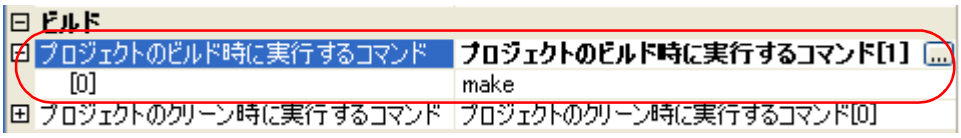
［プレースホルダ］において行をダブルクリックすると，プレースホルダが［テキスト］に反映されます。

2.　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，ビルド時に 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。

［OK］ボタンをクリックすると，入力したコマンドをサブプロパティとして表示します。

図 F—16　［プロジェクトのビルド時に実行するコマンド］プロパティ（コマンド設定後）

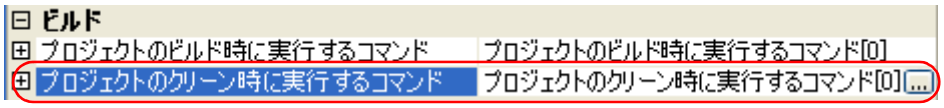


(2) クリーン実行時のコマンドの設定

クリーン時（「F. 5. 3　クリーンを実行する」参照）に実行するコマンドを，［プロジェクトのクリーン時に実行するコマンド］プロパティで設定します。

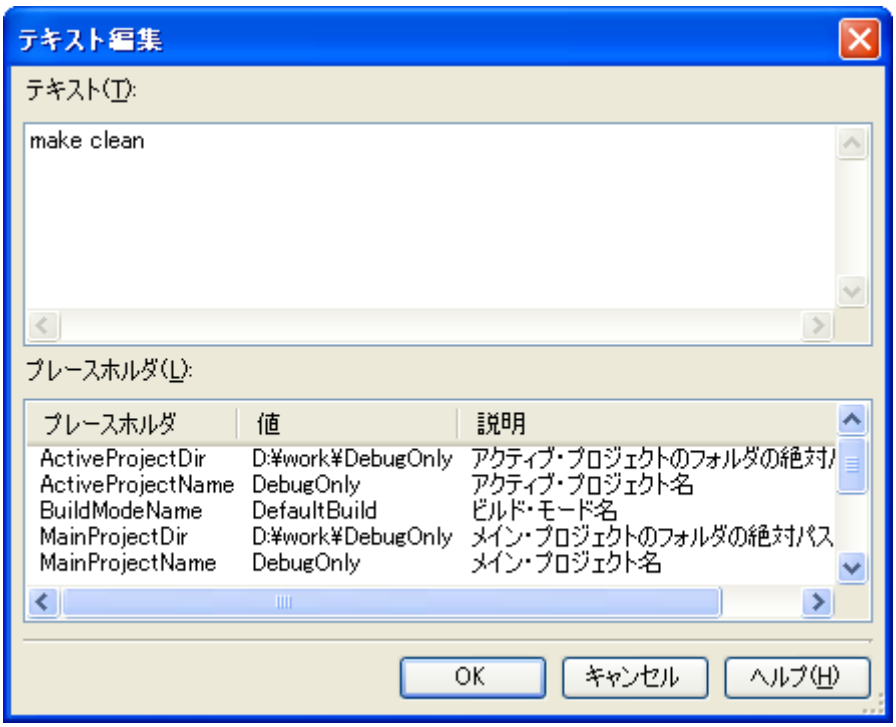
ここでは，ビルドの実行により出力された中間ファイル／生成ファイルなどを削除するコマンドを設定してください。

図 F—17　［プロジェクトのクリーン時に実行するコマンド］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 F—18　テキスト編集 ダイアログ



［テキスト］に実行するコマンドを 1 行に 1 つずつ入力します。

コマンドは，1 行に 1023 文字まで，64 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。

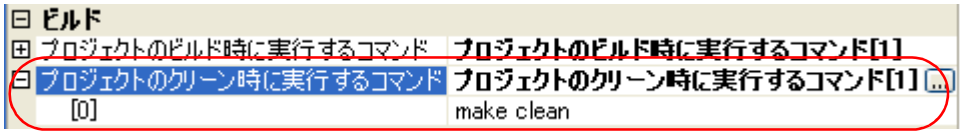
［プレースホルダ］において行をダブルクリックすると，プレースホルダが［テキスト］に反映されます。

2.　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，クリーン時に 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。

［OK］ボタンをクリックすると，入力したコマンドをサブプロパティとして表示します。

図 F—19　［プロジェクトのクリーン時に実行するコマンド］プロパティ（コマンド設定後）



なお，設定したコマンドの変更は，［...］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。

F. 4. 2　ビルド・モードを設定する

ビルドの目的に応じて，ビルド時に実行するコマンドを一括して変更することができます。

「F. 4. 1　コマンドを設定する」で設定したコマンド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，コマンドの設定を毎回変更する必要がなくなります（ビルド・モードごとに，コマンドを設定することができます）。

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・モードの作成方法，および変更方法を示します。

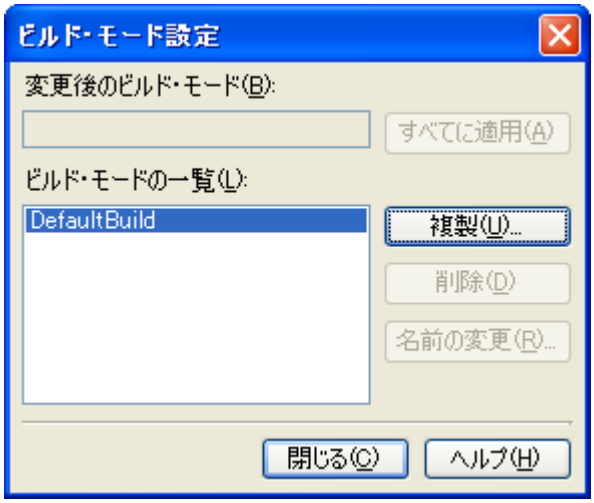
- ビルド・モードの新規作成
- ビルド・モードの変更

(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。

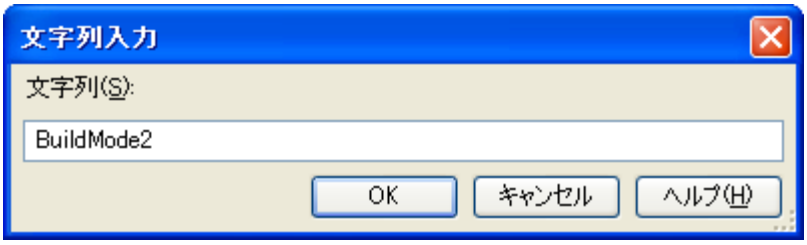
［ビルド］メニュー→［ビルド・モードの設定 ...］を選択すると，ビルド・モード設定 ダイアログがオープンします。

図 F—20　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。

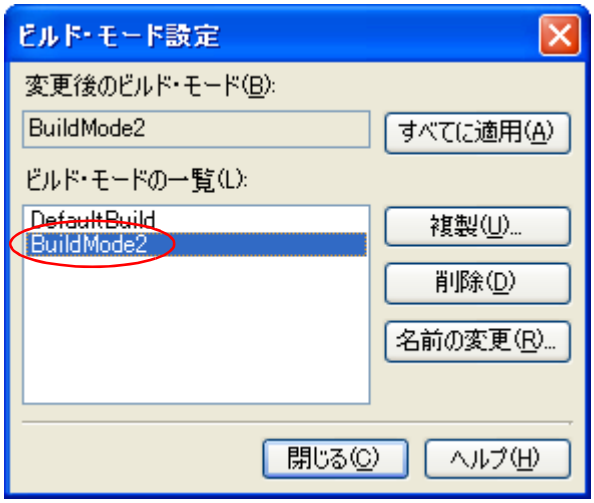
図 F—21　文字列入力 ダイアログ



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。

現在，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（デバッグ専用プロジェクト以外のプロジェクトを含む）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 F—22　ビルド・モード設定 ダイアログ（ビルド・モード追加後）



備考 1.　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.　［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。

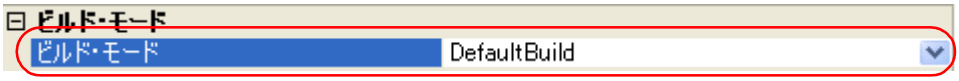
(2) ビルド・モードの変更

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。

(a) 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

対象デバッグ専用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［ビルド・オプション］タブ上の［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。

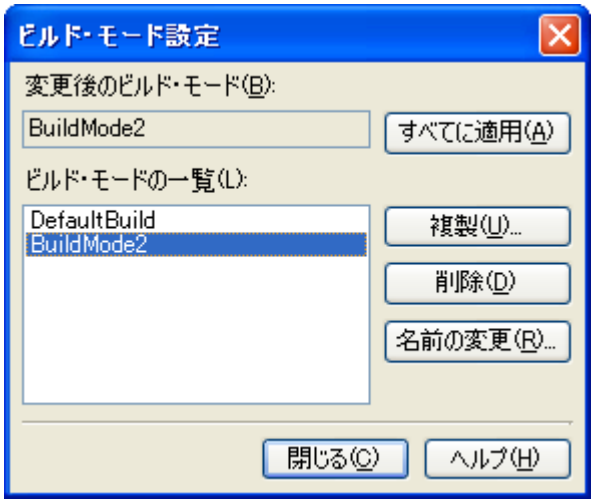
図 F—23　［ビルド・モード］プロパティ



(b) 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 F—24　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（デバッグ専用プロジェクト以外のプロジェクトを含む）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サブプロジェクトについては，該当サブプロジェクトの“DefaultBuild”の内容を複製し，選択したビルド・モードを新規に作成します（“DefaultBuild”の内容で，選択したビルド・モードを追加します）。

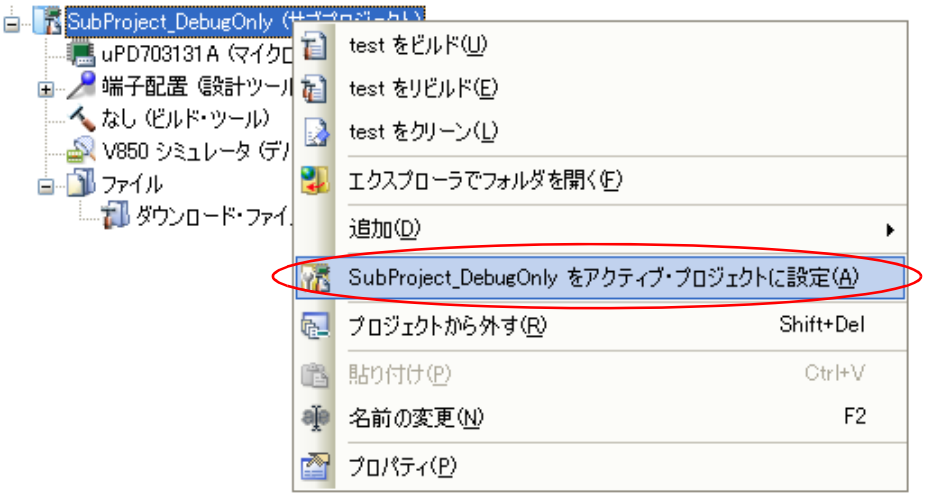
F. 4. 3　ビルド対象プロジェクトを設定する

特定のデバッグ専用プロジェクト（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。

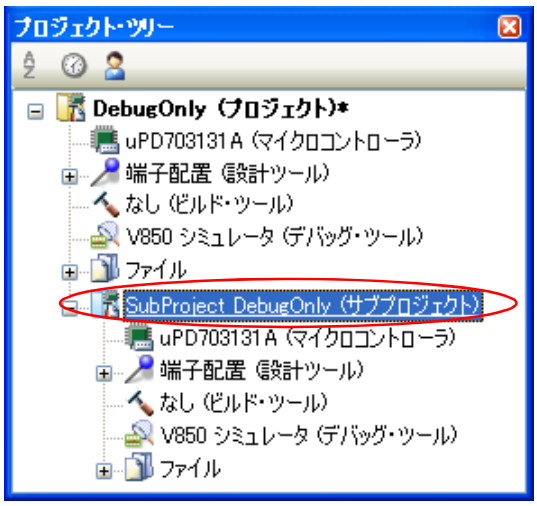
アクティブ・プロジェクトを設定するには，プロジェクト・ツリーでアクティブ・プロジェクトに設定するメイン・プロジェクト・ノード，またはサブプロジェクト・ノード選択し，コンテキスト・メニューの［*選択しているプロジェクト*をアクティブ・プロジェクトに設定］を選択してください。

図 F—25　［*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・ノードには下線を付加します。

図 F—26　アクティブ・プロジェクト



備考 1. プロジェクトの新規作成直後は，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。

F. 5　ビルドを実行する

ここでは，デバッグ専用プロジェクトに対する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

デバッグ専用プロジェクトにおいて，外部ビルド・ツールと連携可能なビルド関連の操作には，以下の種類があります。

表 F—1　ビルドの種類

種類	説明
ビルド	プロパティ パネルの［ビルド］カテゴリの［プロジェクトのビルド時に実行するコマンド］プロパティで設定したコマンドを実行します。 更新したファイルのみビルドを実行することができます注。 →「F. 5. 1　ビルドを実行する」参照
リビルド	プロパティ パネルの［ビルド］カテゴリの［プロジェクトのクリーン時に実行するコマンド］プロパティで設定したコマンドを実行したのち，同カテゴリの［プロジェクトのビルド時に実行するコマンド］プロパティで設定したコマンドを実行します。 ビルド対象のすべてのファイルのビルドを実行することができます注。 →「F. 5. 2　リビルドを実行する」参照
クリーン	プロパティ パネルの［ビルド］カテゴリの［プロジェクトのクリーン時に実行するコマンド］プロパティで設定したコマンドを実行します。 ビルドの実行により出力された中間ファイル，生成ファイルなどをすべて削除することができます注。 →「F. 5. 3　クリーンを実行する」参照
ラピッド・ビルド	ファイルの編集／保存と平行してビルドを実行します。 →「F. 5. 4　ラピッド・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。 →「F. 5. 5　バッチ・ビルドを実行する」参照

注　プロパティ パネルにおいて，適切なコマンドが設定されている必要があります（「F. 4. 1　コマンドを設定する」参照）。

(2) 実行結果の表示

ビルドの実行結果（外部ビルド・ツールからの標準出力，および標準エラー出力）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 F—27　ビルドの実行結果（ビルド／リビルド／バッチ・ビルドの場合）

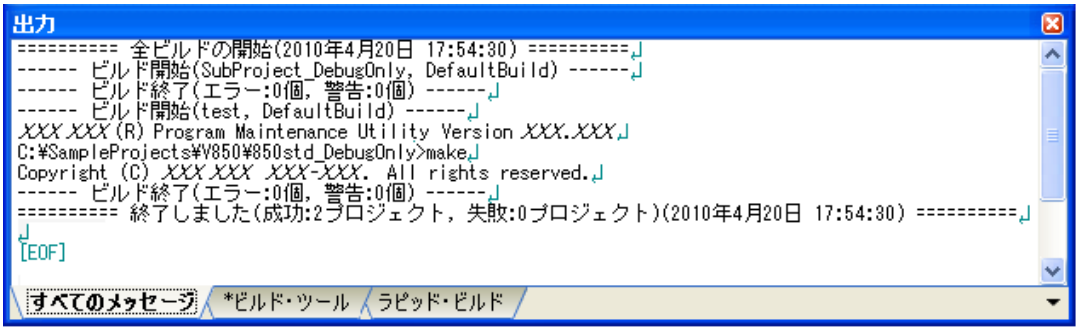


図 F—28　ビルドの実行結果（ラピッド・ビルドの場合）

備考　［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。

F. 5. 1　ビルドを実行する

プロパティ パネルの［ビルド］カテゴリの［プロジェクトのビルド時に実行するコマンド］プロパティで設定したコマンド（「(1) ビルド実行時のコマンドの設定」参照）を実行し，更新したファイルのみを対象としたビルドを行います。

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（「F. 4. 3　ビルド対象プロジェクトを設定する」参照）に設定しているコマンドを実行することができます。

備考　ビルドを実行する際，エディタ パネルで編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。

(1) プロジェクト全体のビルドを実行する場合

プロジェクトに属するすべてのデバッグ専用プロジェクトに設定しているコマンドを実行します。

ツールバーの ボタンをクリックしてください。

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（サブプロジェクトはドラッグ・アンド・ドロップ操作により表示順を変更することができます）。

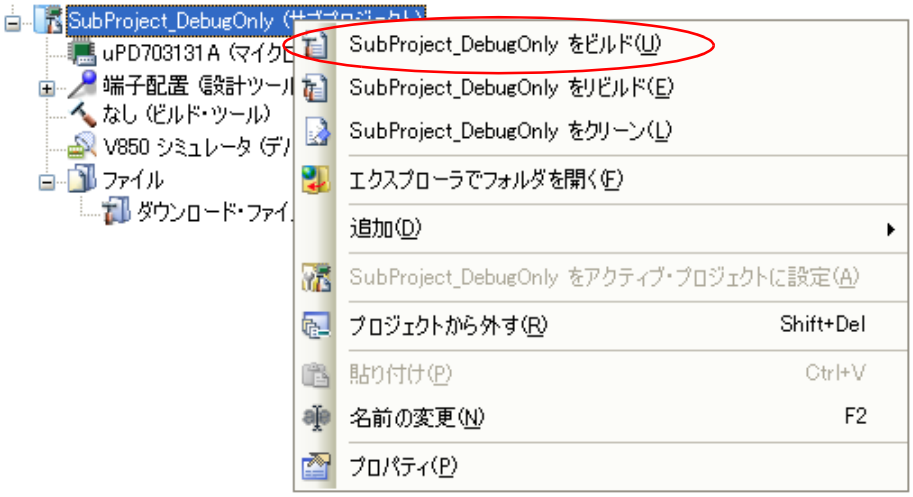
2.　プロジェクトに，デバッグ専用プロジェクト以外のプロジェクトが含まれている場合，該当プロジェクトに対しては，CubeSuite+ が提供するビルド・ツールによるビルドを実行します（ビルド

機能についての詳細は，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください）。

（2）アクティブ・プロジェクトのビルドを実行する場合

対象デバッグ専用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 F—29　［*アクティブ・プロジェクトをビルド*］項目



F. 5. 2　リビルドを実行する

プロパティ パネルの［ビルド］カテゴリの［プロジェクトのクリーン時に実行するコマンド］プロパティで設定したコマンド（「（2）クリーン実行時のコマンドの設定」参照）を実行したのち，同カテゴリの［プロジェクトのビルド時に実行するコマンド］プロパティで設定したコマンド（「（1）ビルド実行時のコマンドの設定」参照）を実行し，ビルド対象のすべてのファイルのビルドを行います。

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（「F. 4. 3　ビルド対象プロジェクトを設定する」参照）に設定しているコマンドを実行することができます。

備考　リビルドを実行する際，エディタ パネルで編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。

（1）プロジェクト全体のリビルドを実行する場合

プロジェクトに属するすべてのデバッグ専用プロジェクトに設定しているコマンドを実行します。

ツールバーの ボタンをクリックしてください。

備考 1.　リビルドの実行は，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リビルドを行います（サブプロジェクトはドラッグ・アンド・ドロップ操作により表示順を変更することができます）。

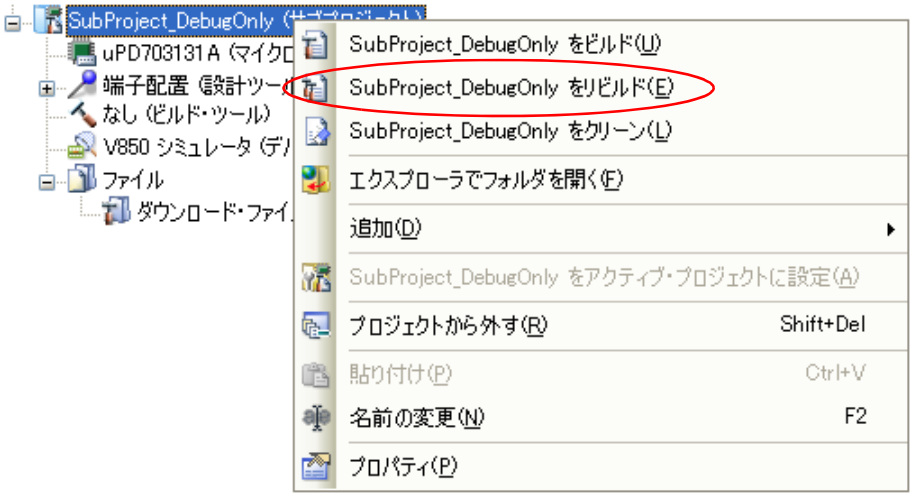
2.　プロジェクトに，デバッグ専用プロジェクト以外のプロジェクトが含まれている場合，該当プロジェクトに対しては，CubeSuite+ が提供するビルド・ツールによるリビルドを実行します（リビ

ルド機能についての詳細は，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください）。

（2）アクティブ・プロジェクトのリビルドを実行する場合

対象デバッグ専用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 F—30　［*アクティブ・プロジェクト*をリビルド］項目



F. 5. 3　クリーンを実行する

プロパティ パネルの［ビルド］カテゴリの［プロジェクトのクリーン時に実行するコマンド］プロパティで設定したコマンド（「（2）クリーン実行時のコマンドの設定」参照）を実行し，ビルドの実行により出力した中間ファイル，生成ファイルを削除します。

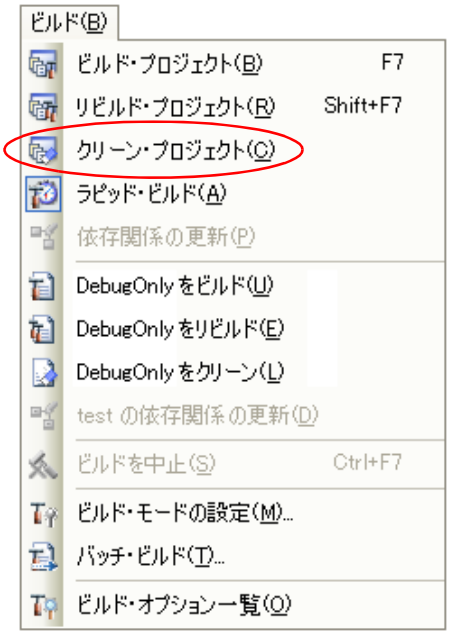
プロジェクト全体（メイン・プロジェクト，およびサブプロジェクト），またはアクティブ・プロジェクト（「F. 4. 3　ビルド対象プロジェクトを設定する」参照）に設定しているコマンドを実行することができます。

（1）プロジェクト全体のクリーンを実行する場合

プロジェクトに属するすべてのデバッグ専用プロジェクトに設定しているコマンドを実行します。

［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 F—31 ［クリーン・プロジェクト］項目



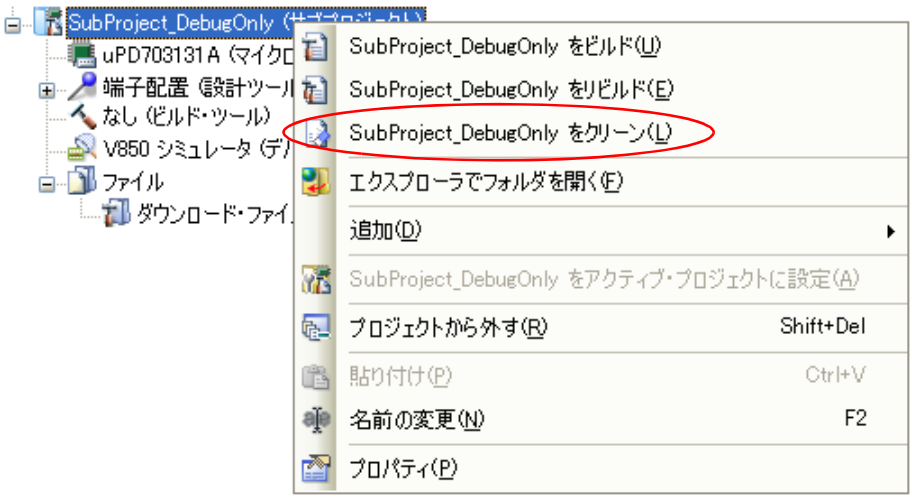
備考 1. クリーンは，サブプロジェクト，メイン・プロジェクトの順で行います。
サブプロジェクトは，プロジェクト・ツリーでの表示順にビルドを行います（サブプロジェクトはドラッグ・アンド・ドロップ操作により表示順を変更することができます）。

2. プロジェクトに，デバッグ専用プロジェクト以外のプロジェクトが含まれている場合，該当プロジェクトに対しては，CubeSuite+ が提供するビルド・ツールによるクリーンを実行します（クリーン機能についての詳細は，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください）。

(2) アクティブ・プロジェクトのクリーンを実行する場合

対象デバッグ専用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 F—32 ［*アクティブ・プロジェクトをクリーン*］項目



F. 5. 4　ラピッド・ビルドを実行する

ラピット・ビルドとは，以下のタイミングで，ビルド（「F. 5. 1　ビルドを実行する」参照）の実行を自動で開始する機能です。

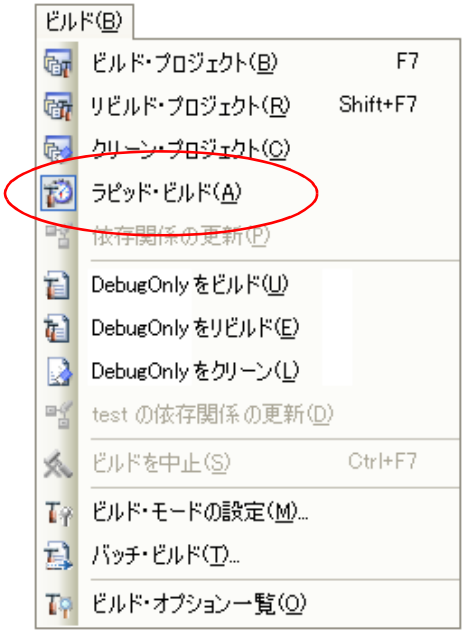
- デバッグ専用プロジェクトに追加している C ソース・ファイル／アセンブラ・ソース・ファイル／ヘッダ・ファイルを更新したとき
- デバッグ専用プロジェクトに，C ソース・ファイル／アセンブラ・ソース・ファイル／ヘッダ・ファイルを追加，または削除したとき
- デバッグ専用プロジェクトのプロパティ パネルにおいて，ビルド・ツールのいずれかのプロパティを変更したとき

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。

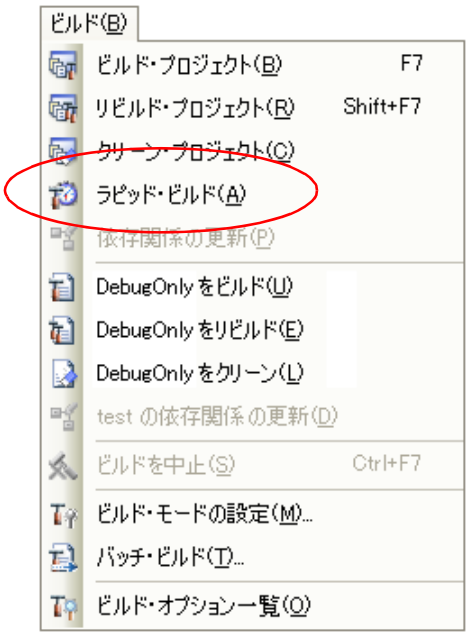
ラピッド・ビルドの有効／無効は，［ビルド］メニュー→［ラピッド・ビルド］の選択により切り替えます。デフォルトでは，有効となっています。

図 F—33　［ラピッド・ビルド］項目

【ラピッド・ビルドが有効の場合】



【ラピッド・ビルドが無効の場合】



注意　この機能は，ソース・ファイルの編集をエディタ パネルで行った場合のみ有効です。

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。

F. 5. 5　バッチ・ビルドを実行する

バッチ・ビルドとは，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／クリーンを行う機能です。

［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 F—34　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（デバッグ専用プロジェクトを対象としている場合は無効の項目です）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 1. ビルド／リビルド／クリーンについての詳細は，それぞれ以下を参照してください。

- ビルド：　「F. 5. 1　ビルドを実行する」参照
- リビルド：　「F. 5. 2　リビルドを実行する」参照
- クリーン：　「F. 5. 3　クリーンを実行する」参照

2. 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。

3. バッチ・ビルドを実行する際，エディタ パネルで編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。

4. デバッグ専用プロジェクト以外のプロジェクトを選択した場合，該当プロジェクトに対しては，CubeSuite+ が提供するビルド・ツールによるビルド／リビルド／クリーンを実行します（ビルド／リビルド／クリーン機能についての詳細は，「CubeSuite+ 統合開発環境 ユーザーズマニュアル ビルド編」を参照してください）。

F. 5. 6　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。

備考　ビルド／リビルド／バッチ・ビルドについての詳細は，それぞれ以下を参照してください。

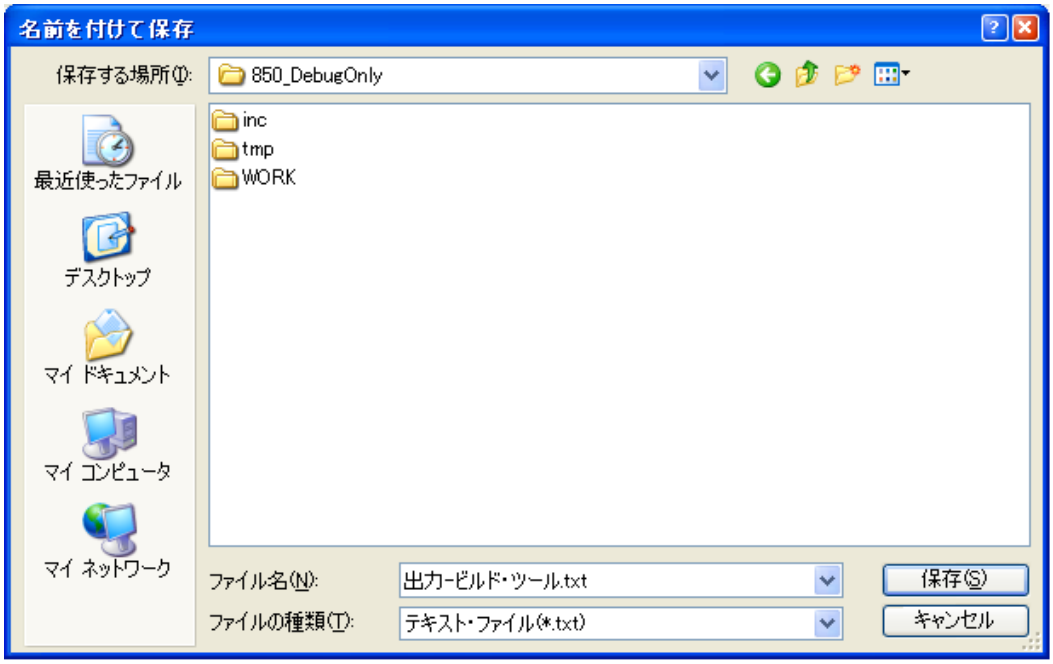
- ビルド：　「F. 5. 1　ビルドを実行する」参照
- リビルド：　「F. 5. 2　リビルドを実行する」参照
- バッチ・ビルド：「F. 5. 5　バッチ・ビルドを実行する」参照

F. 5. 7　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（外部ビルド・ツールからの標準出力，および標準エラー出力）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 F—35　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。

付録 G　Python コンソール /Python 関数

ここでは，CubeSuite+ が提供する Python コンソール，および Python 関数について説明します。

G. 1　概　　要

Python コンソール・プラグインは，IronPython 言語を用いたコンソール・ツールです。

IronPython 言語でサポートされている関数や制御文に加えて，CubeSuite+ を制御するために追加された CubeSuite+ Python 関数も使用することができます。

以下に，CubeSuite+ が提供する機能を示します。

- Python コンソール パネル上で，IronPython 関数や制御文，および CubeSuite+ Python 関数（「G. 3 CubeSuite+ Python 関数／クラス／プロパティ」参照）を実行することができます（「2. 10　Python 関数を実行する」参照）。
- CubeSuite+ をコマンドラインから起動する際に，スクリプト・ファイルを指定して実行することができます（「2. 11　CubeSuite+ をコマンドラインで操作する」参照）。
- プロジェクト・ファイルの読み込み時に，あらかじめ用意しておいたスクリプトを実行することができます（「G. 2　関連ファイル」参照）。

G. 2　関連ファイル

以下に，CubeSuite+ Python 関数の関連ファイルを示します。

- *プロジェクト・ファイル名* .py

  プロジェクト・ファイルと同じフォルダにプロジェクト・ファイルと同じ名前で拡張子が“py”のファイルが存在する場合，プロジェクト・ファイルの読み込み時に自動的に読み込んで実行します。

- *ダウンロード・ファイル名* .py

  ダウンロード・ファイルと同じフォルダにダウンロード・ファイルと同じ名前で拡張子が“py”のファイルが存在する場合，ダウンロードした後に自動的に読み込んで実行します。

G. 3　CubeSuite+ Python 関数／クラス／プロパティ

ここでは，CubeSuite+ Python 関数，クラス，プロパティについて説明します。

CubeSuite+ Python 関数には，以下の規約があります。

- 引数にデフォルト値が存在する場合，【指定形式】の引数は“*引数名 = デフォルト値*”と表記されています。
  引数は，値のみを指定することも可能です。

  例　【指定形式】が function(*arg1*, *arg2* = 1, *arg3* = True) の場合，*arg1* はデフォルト値なし，*arg2* のデフォルト値は 1，*arg3* のデフォルト値は True となります。
  引数は，function(*arg1*, *arg2*, *arg3*) のように指定することが可能です。

- デフォルト値が存在する引数は，省略することができます。
  ただし，引数が判別可能な場合に限ります。

  例　【指定形式】が function(*arg1*, *arg2* = 1, *arg3* = True) の場合

```
>>>function("main")                    …function("main", 1, True) とみなします
>>>function("main", 2)                 …function("main", 2, True) とみなします
>>>function("main", arg3 = False)      …function("main", 1, False) とみなします
>>>function("main", False)             …arg2 と arg3 が判別できないため NG
```

- 引数は，“*引数名 = 値*”のように指定することにより，指定順を変更することができます。

  例　【指定形式】が function(*arg1*, *arg2* = 1, *arg3* = True) の場合

```
>>>function(arg3 = False, arg1 = "main", arg2 = 3)      …OK
>>>function(False, "main", 3)   …arg1 = False, arg2 = "main", arg3 = 3 とみなすため NG
```

- 引数にフォルダやファイルへのパスを記載する場合は注意が必要です。
  IronPython では，¥（バックスラッシュ）を制御文字として認識します。例えば，先頭が t で始まるフォルダ名やファイル名の場合，¥t で TAB 文字と認識してしまいます。これを回避するには次のように記載します。

  例 1.　パス指定の前に r を記載すると，IronPython は "" の中がパスと認識します。

```
r"C:¥test¥test.py"
```

  2.　¥（バックスラッシュ）ではなく /（スラッシュ）を使用します。

```
"C:/test/test.py"
```

本ドキュメントでは，/（スラッシュ）を使用して記載しています。

G. 3. 1　CubeSuite+ Python 関数（共通）

以下に，CubeSuite+ Python 関数（共通）の一覧を示します。

表 G—1　CubeSuite+ Python 関数（共通）

関数名	機能概要
CubeSuiteExit	CubeSuite+ を終了します。
Help	CubeSuite+ Python 関数のヘルプを表示します。
Hook	フック関数を登録します。
Save	編集中のすべてのファイル，およびプロジェクトを保存します。
Source	スクリプト・ファイルを実行します。

CubeSuiteExit

CubeSuite+ を終了します。

[指定形式]

```
CubeSuiteExit()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- CubeSuite+ を終了します。

注意　プロジェクト・ファイルに変更があった場合でも，編集中のファイルは保存しません。
保存する必要がある場合は，Save 関数を使用してください。

[使用例]

```
>>>CubeSuiteExit()
```

Help

CubeSuite+ Python 関数のヘルプを表示します。

[指定形式]

```
Help()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- CubeSuite+ 統合ヘルプを起動して，CubeSuite+ Python 関数のヘルプを表示します。

[使用例]

```
>>>Help()
```

Hook

フック関数，またはコールバック関数を登録します。

[指定形式]

```
Hook(scriptFile)
```

[引数]

引数	説明
scriptFile	フック関数，またはコールバック関数が定義されたスクリプト・ファイルを指定します。

[戻り値]

なし

[詳細説明]

- *scriptFile* を読み込んで，スクリプト・ファイルに宣言されているフック関数，またはコールバック関数を登録します。
  フック関数，コールバック関数以外の関数が宣言されていても，問題はありません。
  なお，フック関数，コールバック関数は，スクリプト・ファイルの終了時に登録します。
- フック関数が宣言されている場合は，CubeSuite+ のイベント発生後に呼び出されます。
- フック関数の種類を以下に示します。
  なお，フック関数には引数はありません。

フック関数	イベント
BeforeBuild	ビルド開始前
BeforeDownload	ダウンロード前
AfterDownload	ダウンロード後
AfterCpuReset	CPU リセット後
BeforeCpuRun	実行開始前
AfterCpuStop	ブレーク後

例　スクリプト・ファイルの記述例

```
def BeforeDownload():
  # ダウンロード前に行いたい処理を追記
```

- フック関数は，以下の操作で初期化されます。
  - プロジェクト・ファイルの読み込み
  - プロジェクト・ファイルの新規作成
  - アクティブ・プロジェクトの変更
  - デバッグ・ツールの切り替え
- コールバック関数が宣言されている場合は，CubeSuite+ のイベント発生後に呼び出されます。
- コールバック関数名は，“pythonConsoleCallback”です。

なお，コールバック関数の引数は，コールバック要因を示します。

引数の値	コールバック要因
10	イベント登録後
11	イベント削除後
12	実行の開始前
13	ブレーク後
14	CPU リセット後
18	デバッグ・ツールのプロパティ変更後
19	ダウンロード後
20	メモリ，またはレジスタ変更後
30	ビルド前
63	XRunBreak で指定した時間経過後

例　スクリプト・ファイルの記述例

```
def pythonConsoleCallback(Id):
  if Id == 63:
    # XRunBreak で指定した時間経過後に行いたい処理を追記 r
```

注意 1.　コールバック関数内では以下の関数を実行しないでください。

debugger.Reset 関数

debugger.Run 関数

debugger.Breakpoint 関数

2.　コールバック関数内で，別の条件の debugger.XRunBreak.Set することはできません。

以下のような指定は行わないでください。

```
def pythonConsoleCallback(Id):
  if Id = 63:
    debugger.XRunBreak.Delete()
    debugger.XRunBreak.Set(1, TimeType.Ms, True)
```

［使用例］

```
>>>Hook("E:/TestFile/TestScript/testScriptFile2.py")
```

Save

編集中のすべてのファイル，およびプロジェクトを保存します。

[指定形式]

```
Save()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- 編集中のすべてのファイル，およびプロジェクトを保存します。

[使用例]

```
>>>Save()
>>>
```

Source

スクリプト・ファイルを実行します。

[指定形式]

```
Source(scriptFile)
```

[引数]

引数	説明
scriptFile	実行するスクリプト・ファイルを指定します。

[戻り値]

なし

[詳細説明]

- *scriptFile* で指定したスクリプト・ファイルを実行します。
- 本関数は，IronPython の execfile と同様の動作を行います。

[使用例]

```
>>>Source("../../testScriptFile2.py")
>>>Source("E:/TestFile/TestScript/testScriptFile.py")
>>>
```

G. 3. 2　CubeSuite+ Python 関数（プロジェクト用）

以下に，CubeSuite+ Python 関数（プロジェクト用）の一覧を示します。

表 G—2　CubeSuite+ Python 関数（プロジェクト用）

関数名	機能概要
project.Change	アクティブ・プロジェクトを変更します。
project.Close	プロジェクトを閉じます。
project.File.Add	アクティブ・プロジェクトにファイルを追加します。
project.File.Exists	アクティブ・プロジェクトにファイルが存在するかどうかを確認します。
project.File.Information	アクティブ・プロジェクトに登録されているファイルの一覧を表示します。
project.File.Remove	アクティブ・プロジェクトからファイルを外します。
project.Information	プロジェクト・ファイルの一覧を表示します。
project.Open	プロジェクトを開きます。

project.Change

アクティブ・プロジェクトを変更します。

[指定形式]

```
project.Change(projectName)
```

[引数]

引数	説明
projectName	変更するプロジェクト・ファイル名，またはサブプロジェクト名をフルパスで指定します。

[戻り値]

アクティブ・プロジェクトを変更するのに成功した場合　：True

アクティブ・プロジェクトを変更するのに失敗した場合　：False

[詳細説明]

- *projectName* に指定したプロジェクトをアクティブ・プロジェクトに変更します。
- *projectName* に指定するプロジェクト・ファイルは，現在開いているプロジェクトに含まれている必要があります。

[使用例]

```
>>>project.Change("C:/project/sample/sub1/subproject.mtpj")
>>>
```

project.Close

プロジェクトを閉じます。

[指定形式]

```
project.Close(save = False)
```

[引数]

引数	説明
save	編集中のすべてのファイル，およびプロジェクトを保存するかどうかを指定します。 True ：編集中のすべてのファイル，およびプロジェクトを保存します。 False ：編集中のすべてのファイル，およびプロジェクトを保存しません（デフォルト）。

[戻り値]

プロジェクトを閉じるのに成功した場合　：True

プロジェクトを閉じるのに失敗した場合　：False

[詳細説明]

- 現在開いているプロジェクトを閉じます。
- *save* に“True”を指定した場合は，編集中のすべてのファイル，およびプロジェクトを保存します。

[使用例]

```
>>>project.Close()
>>>
```

project.File.Add

アクティブ・プロジェクトにファイルを追加します。

[指定形式]

```
project.File.Add(fileName, category="")
```

[引数]

引数	説明
fileName	アクティブ・プロジェクトに追加するファイルをフルパスで指定します。
category	ファイルの追加先カテゴリを指定します（デフォルト：カテゴリを指定しません）。 複数階層を指定する場合は，“[“*one*”, “*two*”]”の形式で指定します。

[戻り値]

アクティブ・プロジェクトへのファイルの追加に成功した場合　：True

アクティブ・プロジェクトへのファイルの追加に失敗した場合　：False

[詳細説明]

- *fileName* に指定したファイルをアクティブ・プロジェクトに追加します。
- *category* を指定した場合，指定したカテゴリの下にファイルを追加します。
  指定したカテゴリが存在しない場合は，新規に作成します。

[使用例]

```
>>>project.File.Add("C:/project/sample/src/test.c", "test")
True
>>>project.File.Add("C:/project/sample/src/test.c", ["test", "src"])
True
```

project.File.Exists

アクティブ・プロジェクトにファイルが存在するかどうかを確認します。

[指定形式]

```
project.File.Exists(fileName)
```

[引数]

引数	説明
fileName	アクティブ・プロジェクトに存在するかどうかを確認するファイルをフルパスで指定します。

[戻り値]

指定したファイルがアクティブ・プロジェクトに存在する場合　：True
指定したファイルがアクティブ・プロジェクトに存在しない場合　：False

[詳細説明]

- *fileName* に指定したファイルがアクティブ・プロジェクトに存在するかどうかを確認します。

[使用例]

```
>>>project.File.Exists("C:/project/sample/src/test.c")
True
>>>
```

project.File.Information

アクティブ・プロジェクトに登録されているファイルの一覧を表示します。

[指定形式]

```
project.File.Information()
```

[引数]

なし

[戻り値]

アクティブ・プロジェクトに登録されているファイル名のリスト

[詳細説明]

- アクティブ・プロジェクトに登録されているファイルの一覧を表示します。

[使用例]

```
>>>project.File.Information()
file1.c
file2.c
file3.c
>>>
```

project.File.Remove

アクティブ・プロジェクトからファイルを外します。

[指定形式]

```
project.File.Remove(fileName)
```

[引数]

引数	説明
fileName	アクティブ・プロジェクトから外すファイルをフルパスで指定します。

[戻り値]

アクティブ・プロジェクトからファイルを外すのに成功した場合　：True

アクティブ・プロジェクトからファイルを外すのに失敗した場合　：False

[詳細説明]

- *fileName* に指定したファイルをアクティブ・プロジェクトから外します。
- ファイルの削除は行いません。

[使用例]

```
>>>project.File.Remove("C:/project/sample/src/test.c")
True
>>>
```

project.Information

プロジェクト・ファイルの一覧を表示します。

[指定形式]

```
project.Information()
```

[引数]

なし

[戻り値]

プロジェクト・ファイル名のリスト

[詳細説明]

- 開いているプロジェクトに含まれているメイン・プロジェクト，およびサブプロジェクトのプロジェクト・ファイルの一覧を表示します。

[使用例]

```
>>>project.Information()
C:¥project¥sample¥test.mtpj
C:¥project¥sample¥sub1¥sub1project.mtsp
C:¥project¥sample¥sub2¥sub2project.mtsp
>>>
```

project.Open

プロジェクトを開きます。

[指定形式]

```
project.Open(fileName, save = False)
```

[引数]

引数	説明
fileName	プロジェクト・ファイルを指定します。
save	他のプロジェクトを開いていた場合，開いていたプロジェクトを閉じる際に，編集中のすべてのファイル，およびプロジェクトを保存するかどうかを指定します。 True ：編集中のすべてのファイル，およびプロジェクトを保存します。 False ：編集中のすべてのファイル，およびプロジェクトを保存しません（デフォルト）。

[戻り値]

プロジェクトを開くのに成功した場合　：True

プロジェクトを開くのに失敗した場合　：False

[詳細説明]

- *fileName* で指定したプロジェクトを開きます。
- 他のプロジェクトを開いていた場合は，開いていたプロジェクトを閉じます。
  save に“True”を指定した場合は，開いていたプロジェクトの編集中のすべてのファイル，およびプロジェクトを保存します。
- 他のプロジェクトを開いていない場合は，*save* の指定は無視します。

[使用例]

```
>>>project.Open("C:/test/test.mtpj")
>>>
```

G. 3. 3　CubeSuite+ Python 関数（ビルド・ツール用）

以下に，CubeSuite+ Python 関数（ビルド・ツール用）の一覧を示します。

表 G—3　CubeSuite+ Python 関数（ビルド・ツール用）

関数名	機能概要
build.All	プロジェクトのビルドを行います。
build.ChangeBuildMode	ビルド・モードの変更を行います。
build.Clean	プロジェクトのクリーンを行います。
build.File	指定したファイルのビルドを行います。
build.Update	ビルド・ツールの依存関係の更新を行います。

build.All

プロジェクトのビルドを行います。

[指定形式]

```
build.All(rebuild = False, waitBuild = True)
```

[引数]

引数	説明
rebuild	プロジェクトをリビルドするかどうかを指定します。 True ：プロジェクトをリビルドします。 False ：プロジェクトをビルドします（デフォルト）。
waitBuild	ビルドが完了するまで待つかどうかを指定します。 True ：ビルドが完了するまで待ちます（デフォルト）。 False ：ビルドが完了するのを待たずに，プロンプトを返します。

[戻り値]

- *waitBuild* に“True”を指定した場合
  ビルドが正常に完了した場合　：True
  ビルドが失敗，またはキャンセルされた場合　：False

- *waitBuild* に“False”を指定した場合
  ビルドの実行開始に成功した場合　：True
  ビルドの実行開始に失敗した場合　：False

[詳細説明]

- プロジェクトのビルドを行います。
  プロジェクトに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。
- *rebuild* に“True”を指定した場合は，プロジェクトをリビルドします。
- *waitBuild* に“False”を指定した場合は，ビルドが完了するのを待たずに，プロンプトを返します。
- ビルドの成否にかかわらず，ビルドの完了後に build.BuildCompleted イベントが発行されます。

[使用例]

```
>>>build.All()
True
>>>
```

build.ChangeBuildMode

ビルド・モードの変更を行います。

[指定形式]

```
build.ChangeBuildMode(buildmode)
```

[引数]

引数	説明 dd
buildmode	変更するビルド・モードを文字列で指定します。

[戻り値]

なし

[詳細説明]

- メイン・プロジェクト，およびサブプロジェクトのビルド・モードを *buildmode* で指定したビルド・モードに変更します。
- *buildmode* が存在しないプロジェクトの場合は，“DefaultBuild” を元にビルド・モードを作成して変更します。

[使用例]

```
>>>build.ChangeBuildMode("test_release")
>>>
```

build.Clean

プロジェクトのクリーンを行います。

[指定形式]

```
build.Clean(all = False)
```

[引数]

引数	説明 dd
all	サブプロジェクトも含めてプロジェクトのクリーンを行うかどうかを指定します。 True ：サブプロジェクトを含めたすべてのプロジェクトのクリーンを行います。 False ：アクティブ・プロジェクトのみクリーンを行います（デフォルト）。

[戻り値]

クリーンが正常に完了した場合　：True

クリーンが失敗した場合　：False

[詳細説明]

- プロジェクトのクリーンを行います（ビルド時の生成物を削除します）。
- *all* に“True”を指定した場合は，サブプロジェクトのクリーンも行います。

[使用例]

```
>>>build.Clean()
True
>>>
```

build.File

指定したファイルのビルドを行います。

[指定形式]

```
build.File(fileName, rebuild = False, waitBuild = True)
```

[引数]

引数	説明
fileName	ビルドするファイルを指定します。
rebuild	指定したファイルをリビルドするかどうかを指定します。 True ：指定したファイルをリビルドします。 False ：指定したファイルをビルドします（デフォルト）。
waitBuild	ビルドが完了するまで待つかどうかを指定します。 True ：ビルドが完了するまで待ちます（デフォルト）。 False ：ビルドが完了するのを待たずに，プロンプトを返します。

[戻り値]

- *waitBuild* に“True”を指定した場合

  ビルドが成功した場合　：True

  ビルドが失敗した場合　：False

- *waitBuild* に“False”を指定した場合

  ビルドの実行開始に成功した場合　：True

  ビルドの実行開始に失敗した場合　：False

[詳細説明]

- *fileName* で指定したファイルのビルドを行います。
- *rebuild* に“True”を指定した場合は，指定したファイルをリビルドします。
- *waitBuild* に“False”を指定した場合は，ビルドが完了するのを待たずに，プロンプトを返します。
- ビルドの完了後に build.BuildCompleted イベントが発行されます。

[使用例]

```
>>>build.File("C:/test/test.c")
True
>>>
```

build.Update

ビルド・ツールの依存関係の更新を行います。

[指定形式]

```
build.Update()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- ビルド時のファイルの依存関係を更新します。

[使用例]

```
>>>build.Update()
>>>
```

G. 3. 4　CubeSuite+ Python 関数（デバッグ・ツール用）

以下に，CubeSuite+ Python 関数（デバッグ・ツール用）の一覧を示します。

表 G—4　CubeSuite+ Python 関数（デバッグ・ツール用）

関数名	機能概要
debugger.Address	アドレス式を評価します。
debugger.Assemble.Disassemble	逆アセンブルを行います。
debugger.Assemble.LineAssemble	ライン・アセンブルを行います。
debugger.Breakpoint.Delete	ブレークポイントを削除します。
debugger.Breakpoint.Disable	ブレークポイントの設定を無効にします。
debugger.Breakpoint.Enable	ブレークポイントの設定を有効にします。
debugger.Breakpoint.Information	ブレークポイントの情報を表示します。
debugger.Breakpoint.Set	ブレークポイントを設定します。
debugger.Connect	デバッグ・ツールに接続します。
debugger.DebugTool.Change	デバッグ・ツールを変更します。
debugger.DebugTool.GetType	デバッグ・ツールの情報を表示します。
debugger.Disconnect	デバッグ・ツールから切断します。
debugger.Download.Binary	バイナリ・ファイルをダウンロードします。
debugger.Download.Binary64Kb	64KB 以内用形式でバイナリ・ファイルをダウンロードします。
debugger.Download.BinaryBank	メモリ・バンク用形式でバイナリ・ファイルをダウンロードします。
debugger.Download.Coverage	カバレッジ・データをダウンロードします。
debugger.Download.Hex	ヘキサ・ファイルをダウンロードします。
debugger.Download.Hex64Kb	64KB 以内用形式でヘキサ・ファイルをダウンロードします。
debugger.Download.HexBank	メモリ・バンク用形式でヘキサ・ファイルをダウンロードします。
debugger.Download.HexIdTag	ID タグ付きヘキサ・ファイルをダウンロードします。
debugger.Download.Information	ダウンロード情報を表示します。
debugger.Download.LoadModule	ロード・モジュールをダウンロードします。
debugger.Erase	フラッシュ・メモリを消去します。
debugger.GetBreakStatus	ブレーク要因を表示します。
debugger.GetCpuStatus	現在の CPU の状態を表示します。
debugger.GetIeStatus	現在の IE の状態を表示します。
debugger.GetPC	PC 値を表示します。
debugger.Go	プログラムを継続して実行します。
debugger.Ie.GetValue/ debugger.Ie.SetValue	IE レジスタ，または DCU レジスタを設定／参照します。
debugger.IsConnected	デバッグ・ツールの接続状態を確認します。
debugger.IsRunning	デバッグ・ツールの実行状態を確認します。

関数名	機能概要
debugger.Jump.File/ debugger.Jump.Address	各種パネルを表示します。
debugger.Map.Clear	マッピング設定をクリアします。
debugger.Map.Information	マップ情報を表示します。
debugger.Map.Set	メモリ・マッピングの設定を行います。
debugger.Memory.Copy	メモリをコピーします。
debugger.Memory.Fill	メモリを補填します。
debugger.Memory.Read	メモリを参照します。
debugger.Memory.Write	メモリに書き込みます。
debugger.Next	プロシージャ・ステップ実行を行います。
debugger.Register.GetValue	レジスタ，I/O レジスタ，SFR を参照します。
debugger.Register.SetValue	レジスタ，I/O レジスタ，SFR に値を設定します。
debugger.Reset	CPU をリセットします。
debugger.ReturnOut	現在の関数を呼び出したプログラムに戻るまで実行します。
debugger.Run	プログラムをリセット後に実行します。
debugger.Step	ステップ実行を行います。
debugger.Stop	デバッグ・ツールの実行を停止します。
debugger.Upload.Binary	メモリ・データをバイナリ形式で保存します。
debugger.Upload.Coverage	カバレッジ・データを保存します。
debugger.Upload.Intel	メモリ・データをインテル形式で保存します。
debugger.Upload.IntelIdTag	メモリ・データを ID タグ付きインテル形式で保存します。
debugger.Upload.Motorola	メモリ・データをモトローラ形式で保存します。
debugger.Upload.MotorolaIdTag	メモリ・データを ID タグ付きモトローラ形式で保存します。
debugger.Upload.Tektronix	メモリ・データをテクトロニクス形式で保存します。
debugger.Upload.TektronixIdTag	メモリ・データを ID タグ付きテクトロニクス形式で保存します。
debugger.Watch.GetValue	変数値を参照します。
debugger.Watch.SetValue	変数値を設定します。
debugger.Where	スタックのバック・トレースを表示します。
debugger.Whereami	ロケーションを表示します。
debugger.XCoverage.Clear	カバレッジ・メモリをクリアします。
debugger.XCoverage.GetCoverage	カバレッジを取得します。
debugger.XRunBreak.Delete	XRunBreak の設定情報を削除します。
debugger.XRunBreak.Refer	XRunBreak の設定情報を表示します。
debugger.XRunBreak.Set	XRunBreak の情報を設定します。
debugger.XTime	Go-Break 間の時間情報を表示します。
debugger.XTrace.Clear	トレース・メモリをクリアします。
debugger.XTrace.Dump	トレース・データをダンプします。

debugger.Address

アドレス式を評価します。

[指定形式]

```
debugger.Address(expression)
```

[引数]

引数	説明
expression	アドレス式を指定します。

[戻り値]

変換したアドレス（数値）

[詳細説明]

- *expression* で指定したアドレス式をアドレスに変換します。

注意　CubeSuite+.exe の起動オプションでスクリプトを指定して実行する場合，デバッグ・ツールと接続するまでシンボル変換機能は使用できません。
つまり，本関数は使用できませんので，接続後に実行してください。

[使用例]

```
>>>debugger.Address("main")
0x4088
>>>debugger.Address("main + 1")
0x4089
>>>
```

debugger.Assemble.Disassemble

逆アセンブルを行います。

[指定形式]

```
debugger.Assemble.Disassemble(address, number = 1, code = True)
```

[引数]

引数	説明
address	逆アセンブルを開始するアドレスを指定します。
number	表示行数を指定します（デフォルト：1）。
code	命令コードを表示するかどうかを指定します。 True　：命令コードを表示します（デフォルト）。 False　：命令コードを表示しません。

[戻り値]

逆アセンブル結果のリスト

[詳細説明]

- *address* で指定したアドレスから逆アセンブルします。
- *number* を指定した場合は，指定した数分の行を表示します。
- *code* に“False”を指定した場合は，命令コードを表示しません。
- *address* に“.”を指定した場合は，直前の逆アセンブルの続きのアドレスを指定したと解釈します。

[使用例]

```
>>>debugger.Assemble.Disassemble("main")
0x00004088  F545    br _TestInit+0x8e
>>>debugger.Assemble.Disassemble("main", 2)
0x00004088  F545    br _TestInit+0x8e
0x0000408A  0A5A    mov 0xa, r11
>>>debugger.Assemble.Disassemble("main", 5, False)
0x00004088  br _TestInit+0x8e
0x0000408A  mov 0xa, r11
0x0000408C  movea 0x19, r0, r13
0x00004090  mov r13, r12
0x00004092  movhi 0xffff, gp, r1
>>>
```

debugger.Assemble.LineAssemble

ライン・アセンブルを行います。

[指定形式]

```
debugger.Assemble.LineAssemble(address, code)
```

[引数]

引数	説明
address	アセンブルを開始するアドレスを指定します。
code	アセンブルする文字列を指定します。

[戻り値]

なし

[詳細説明]

- *code* で指定した文字列を *address* で指定したアドレスからアセンブルします。
- *address* に“.”を指定した場合は，直前のアセンブルの続きのアドレスを指定したと解釈します。

[使用例]

```
>>>debugger.Assemble.Disassemble("main")
0x00004088  F545    br _TestInit+0x8e
>>>debugger.Assemble.Disassemble(".")
0x0000408A  0A5A    mov 0xa, r11
>>>debugger.Assemble.LineAssemble("main", "mov r13, r12")
>>>debugger.Assemble.Disassemble("main", 1, False)
0x00004088  mov r13, r12
>>>
```

debugger.Breakpoint.Delete

ブレークポイントを削除します。

[指定形式]

```
debugger.Breakpoint.Delete(breakNumber = "")
```

[引数]

引数	説明
breakNumber	削除するブレーク・イベント番号を指定します。

[戻り値]

なし

[詳細説明]

- *breakNumber* で指定したブレーク・イベントを削除します。
- *breakNumber* を指定しない場合は，すべてのブレーク・イベント番号のブレークを削除します。

[使用例]

```
>>>debugger.Breakpoint.Enable(1)
>>>debugger.Breakpoint.Disable(1)
>>>debugger.Breakpoint.Delete(1)
>>>
```

debugger.Breakpoint.Disable

ブレークポイントの設定を無効にします。

[指定形式]

```
debugger.Breakpoint.Disable(breakNumber = "")
```

[引数]

引数	説明
breakNumber	無効にするブレーク・イベント番号を指定します。

[戻り値]

なし

[詳細説明]

- *breakNumber* で指定したブレーク・イベントを無効にします。
- *breakNumber* を指定しない場合は，すべてのブレーク・イベント番号のブレークを無効にします。

[使用例]

```
>>>debugger.Breakpoint.Enable(1)
>>>debugger.Breakpoint.Disable(1)
>>>debugger.Breakpoint.Delete(1)
>>>
```

debugger.Breakpoint.Enable

ブレークポイントの設定を有効にします。

[指定形式]

```
debugger.Breakpoint.Enable(breakNumber = "")
```

[引数]

引数	説明
breakNumber	有効にするブレーク・イベント番号を指定します。

[戻り値]

なし

[詳細説明]

- *breakNumber* で指定したブレーク・イベントを有効にします。
- *breakNumber* を指定しない場合は，すべてのブレーク・イベント番号のブレークを有効にします。

[使用例]

```
>>>debugger.Breakpoint.Enable(1)
>>>debugger.Breakpoint.Disable(1)
>>>debugger.Breakpoint.Delete(1)
>>>
```

debugger.Breakpoint.Information

ブレークポイントの情報を表示します。

[指定形式]

```
debugger.Breakpoint.Information()
```

[引数]

なし

[戻り値]

ブレークポイント情報のリスト

[詳細説明]

- 設定されているブレークポイントの情報を，以下の形式で表示します。
  ブレーク名は，“Python ブレーク *xxxx*”（*xxxx*：4 桁の数字）となります。

```
ブレーク・イベント番号  ブレーク名  状態  アドレス・ロケーション
```

[使用例]

```
>>>debugger.Breakpoint.Information()
   1 Python ブレーク 0001 Enable  0x000002dc
   2 ブレーク 0001 Enable  test1.c#_sub1
   3 Python ブレーク 0002 Enable  0x000002ec
   4 ブレーク 0002 Enable  test1.c#_sub1+10
>>>
```

debugger.Breakpoint.Set

ブレークポイントを設定します。

[指定形式]

```
debugger.Breakpoint.Set(BreakCondition)
```

[引数]

引数	説明
BreakCondition	ブレーク条件を指定します。 ブレーク条件の作成については，「BreakCondition」を参照してください。

[戻り値]

設定したブレーク・イベント番号（数値）

[詳細説明]

- *BreakCondition* で指定されている内容に従って，ブレークポイントを設定します。

[使用例]

```
>>>Condition = BreakCondition()
>>>Condition.Address = "main"
>>>breakNumber = debugger.Breakpoint.Set(Condition)
>>>print breakNumber
1
>>>debugger.Breakpoint.Information()
   1 Python ブレーク 0001 Enable  0x000002dc
```

debugger.Connect

デバッグ・ツールに接続します。

[指定形式]

```
debugger.Connect()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- デバッグ・ツールに接続します。

[使用例]

```
>>>debugger.Connect()
>>>
```

debugger.DebugTool.Change

デバッグ・ツールを変更します。

[指定形式]

```
debugger.DebugTool.Change(debugTool)
```

[引数]

引数	説明	
debugTool	変更するデバッグ・ツールを指定します。 指定可能なデバッグ・ツールを以下に示します。	
	種類	説明
	DebugTool.Simulator	シミュレータ
	DebugTool.Minicube	MINICUBE
	DebugTool.Minicube2	MINICUBE2（シリアル接続）
	DebugTool.Minicube2Jtag	MINICUBE2（JTAG 接続）
	DebugTool.Iecube	IECUBE
	DebugTool.Iecube2	IECUBE2
	DebugTool.E1Jtag	E1（JTAG 接続）
	DebugTool.E1Serial	E1（シリアル接続）
	DebugTool.E20Jtag	E20（JTAG 接続）
	DebugTool.E20Serial	E20（シリアル接続）
	DebugTool.EZ_Emulator	EZ Emulator

[戻り値]

なし

[詳細説明]

- *DebugTool* で指定したデバッグ・ツールに変更します。
  ただし，変更可能なデバッグ・ツールは，使用するデバイスによって異なります。変更可能なデバッグ・ツールは，プロジェクト・ツリーで［デバッグ・ツール］を選択し，コンテキストメニューの［使用するデバッグ・ツール］で確認してください。

注意　選択エミュレータできないエミュレータも指定できてしまいます。
CubeSuite+ のデバッグ・ツールで選択できるエミュレータのみ指定してください。

［使用例］

```
>>>debugger.DebugTool.Change(DebugTool.Simulator)
>>>
```

debugger.DebugTool.GetType

デバッグ・ツールの情報を表示します。

[指定形式]

```
debugger.DebugTool.GetType()
```

[引数]

なし

[戻り値]

デバッグ・ツールの種類の定数

デバッグ・ツールの種類の定数	説明
Simulator	シミュレータ
Minicube	MINICUBE
Minicube2	MINICUBE2（シリアル接続）
Minicube2Jtag	MINICUBE2（JTAG 接続）
Iecube	IECUBE
Iecube2	IECUBE2
E1Jtag	E1（JTAG 接続）
E1Serial	E1（シリアル接続）
E20Jtag	E20（JTAG 接続）
E20Serial	E20（シリアル接続）
EZ_Emulator	EZ Emulator

[詳細説明]

- デバッグ・ツールの情報を表示します。

[使用例]

```
>>>debugType = debugger.DebugTool.GetType()
Minicube2
>>>if debugType != DebugTool.Simulator:
... debugger.DebugTool.Change(DebugTool.Simulator)
...
>>>
```

debugger.Disconnect

デバッグ・ツールから切断します。

[指定形式]

```
debugger.Disconnect()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- デバッグ・ツールから切断します。

[使用例]

```
>>>debugger.Disconnect()
```

debugger.Download.Binary

バイナリ・ファイルをダウンロードします。

[指定形式]

```
debugger.Download.Binary(fileName, address, append = False, flashErase = False)
```

[引数]

引数	説明
fileName	ダウンロード・ファイルを指定します。
address	ダウンロード開始アドレスを指定します。
append	追加ダウンロードするかどうかを指定します。 True ：追加ダウンロードします。 False ：上書きダウンロードします（デフォルト）。
flashErase	ダウンロード前にフラッシュ・メモリを初期化するかどうかを指定します。 True ：ダウンロード前にフラッシュ・メモリを初期化します。 False ：ダウンロード前にフラッシュ・メモリを初期化しません（デフォルト）。

注意　2 つ以上の引数を指定する場合は，3 つの引数を指定する必要があります。
***fileName*，*address* のみを指定することはできません。**

[戻り値]

なし

[詳細説明]

- バイナリ形式のデータをダウンロードします。

[使用例]

```
>>>debugger.Download.Binary("C:/test/testModule.bin",  0x1000, False)
>>>debugger.Download.Binary("C:/test/testModule2.bin", 0x2000, True)
>>>
```

debugger.Download.Binary64Kb

64KB 以内用形式でバイナリ・ファイルをダウンロードします。

[指定形式]

```
debugger.Download.Binary64Kb(fileName, address, append = False, flashErase = False)
```

[引数]

引数	説明
fileName	ダウンロード・ファイルを指定します。
address	ダウンロード開始アドレスを指定します。
append	追加ダウンロードするかどうかを指定します。 True ：追加ダウンロードします。 False ：上書きダウンロードします（デフォルト）。
flashErase	ダウンロード前にフラッシュ・メモリを初期化するかどうかを指定します。 True ：ダウンロード前にフラッシュ・メモリを初期化します。 False ：ダウンロード前にフラッシュ・メモリを初期化しません（デフォルト）。

注意　2 つ以上の引数を指定する場合は，3 つの引数を指定する必要があります。
***fileName*，*address* のみを指定することはできません。**

[戻り値]

なし

[詳細説明]

- メモリ・バンク使用時に，64KB 以内用形式でバイナリ・ファイルをダウンロードします。

[使用例]

```
>>>debugger.Download.Binary64Kb("C:/test/testModule.bin",  0x1000, False)
>>>debugger.Download.Binary64Kb("C:/test/testModule2.bin", 0x2000, True)
>>>
```

debugger.Download.BinaryBank

メモリ・バンク用形式でバイナリ・ファイルをダウンロードします。

[指定形式]

```
debugger.Download.BinaryBank(fileName, address, append = False, flashErase = False)
```

[引数]

引数	説明
fileName	ダウンロード・ファイルを指定します。
address	ダウンロード開始アドレスを指定します。
append	追加ダウンロードするかどうかを指定します。 True　：追加ダウンロードします。 False：上書きダウンロードします（デフォルト）。
flashErase	ダウンロード前にフラッシュ・メモリを初期化するかどうかを指定します。 True　：ダウンロード前にフラッシュ・メモリを初期化します。 False：ダウンロード前にフラッシュ・メモリを初期化しません（デフォルト）。

注意　2 つ以上の引数を指定する場合は，3 つの引数を指定する必要があります。
***fileName*，*address* のみを指定することはできません。**

[戻り値]

なし

[詳細説明]

- メモリ・バンク使用時に，メモリ・バンク用形式でバイナリ・ファイルをダウンロードします。

[使用例]

```
>>>debugger.Download.BinaryBank("C:/test/testModule.bin",  0x1000, False)
>>>debugger.Download.BinaryBank("C:/test/testModule2.bin", 0x2000, True)
>>>
```

debugger.Download.Coverage

カバレッジ・データをダウンロードします。【IECUBE】【IECUBE2】【シミュレータ】

[指定形式]

```
debugger.Download.Coverage(fileName)
```

[引数]

引数	説明
fileName	カバレッジ・データ・ファイルを指定します。

[戻り値]

なし

[詳細説明]

- カバレッジ・データをダウンロードします。

[使用例]

```
>>>debugger.Download.Coverage("C:/test/testModule.csrcv")
>>>
```

debugger.Download.Hex

ヘキサ・ファイルをダウンロードします。

[指定形式]

```
debugger.Download.Hex(fileName, offset = 0, append = False, flashErase = False)
```

[引数]

引数	説明
fileName	ダウンロード・ファイルを指定します。
offset	オフセットを指定します（デフォルト：0）。
append	追加ダウンロードするかどうかを指定します。 True　：追加ダウンロードします。 False：上書きダウンロードします（デフォルト）。
flashErase	ダウンロード前にフラッシュ・メモリを初期化するかどうかを指定します。 True　：ダウンロード前にフラッシュ・メモリを初期化します。 False：ダウンロード前にフラッシュ・メモリを初期化しません（デフォルト）。

注意　2つ以上の引数を指定する場合は，3つの引数を指定する必要があります。
***fileName*，*offset* のみを指定することはできません。**

[戻り値]

なし

[詳細説明]

- ヘキサ形式のデータをダウンロードします。

[使用例]

```
>>>debugger.Download.Hex("C:/test/testModule.hex")
>>>
```

debugger.Download.Hex64Kb

64KB 以内用形式でヘキサ・ファイルをダウンロードします。

[指定形式]

```
debugger.Download.Hex64Kb(fileName, offset = 0, append = False, flashErase = False)
```

[引数]

引数	説明
fileName	ダウンロード・ファイルを指定します。
offset	オフセットを指定します（デフォルト：0）。
append	追加ダウンロードするかどうかを指定します。 True　：追加ダウンロードします。 False：上書きダウンロードします（デフォルト）。
flashErase	ダウンロード前にフラッシュ・メモリを初期化するかどうかを指定します。 True　：ダウンロード前にフラッシュ・メモリを初期化します。 False：ダウンロード前にフラッシュ・メモリを初期化しません（デフォルト）。

注意　2 つ以上の引数を指定する場合は，3 つの引数を指定する必要があります。
***fileName*，*offset* のみを指定することはできません。**

[戻り値]

なし

[詳細説明]

- メモリ・バンク使用時に，64KB 以内用形式でヘキサ・ファイルをダウンロードします。

[使用例]

```
>>>debugger.Download.Hex64Kb("C:/test/testModule.hex")
>>>
```

debugger.Download.HexBank

メモリ・バンク用形式でヘキサ・ファイルをダウンロードします。

[指定形式]

```
debugger.Download.HexBank(fileName, offset = 0, append = False, flashErase = False)
```

[引数]

引数	説明
fileName	ダウンロード・ファイルを指定します。
offset	オフセットを指定します（デフォルト：0）。
append	追加ダウンロードするかどうかを指定します。 True　：追加ダウンロードします。 False：上書きダウンロードします（デフォルト）。
flashErase	ダウンロード前にフラッシュ・メモリを初期化するかどうかを指定します。 True　：ダウンロード前にフラッシュ・メモリを初期化します。 False：ダウンロード前にフラッシュ・メモリを初期化しません（デフォルト）。

注意　2 つ以上の引数を指定する場合は，3 つの引数を指定する必要があります。
***fileName*，*offset* のみを指定することはできません。**

[戻り値]

なし

[詳細説明]

- メモリ・バンク使用時に，メモリ・バンク用形式でヘキサ・ファイルをダウンロードします。

[使用例]

```
>>>debugger.Download.HexBank("C:/test/testModule.hex")
>>>debugger.Download.HexBank("C:/test/testModule2.hex", 0x1000, True)
>>>
```

debugger.Download.HexIdTag

ID タグ付きヘキサ・ファイルをダウンロードします。

[指定形式]

```
debugger.Download.HexIdTag(fileName, offset = 0, append = False, flashErase = False)
```

[引数]

引数	説明
fileName	ダウンロード・ファイルを指定します。
offset	オフセットを指定します（デフォルト：0）。
append	追加ダウンロードするかどうかを指定します。 True　：追加ダウンロードします。 False：上書きダウンロードします（デフォルト）。
flashErase	ダウンロード前にフラッシュ・メモリを初期化するかどうかを指定します。 True　：ダウンロード前にフラッシュ・メモリを初期化します。 False：ダウンロード前にフラッシュ・メモリを初期化しません（デフォルト）。

注意　2 つ以上の引数を指定する場合は，3 つの引数を指定する必要があります。
***fileName*，*offset* のみを指定することはできません。**

[戻り値]

なし

[詳細説明]

- ID タグ付きヘキサ・ファイルをダウンロードします。

[使用例]

```
>>>debugger.Download.HexIdTag("C:/test/testModule.hex")
>>>debugger.Download.HexIdTag("C:/test/testModule2.hex", 0x1000, True)
>>>
```

debugger.Download.Information

ダウンロード情報を表示します。

[指定形式]

```
debugger.Download.Information()
```

[引数]

なし

[戻り値]

ダウンロード情報のリスト

[詳細説明]

- ダウンロード情報を，以下の形式で表示します。

ダウンロード番号： ダウンロード・ファイル名

[使用例]

```
>>>debugger.Download.Information()
   1: DefaultBuild¥test.lmf
```

debugger.Download.LoadModule

ロード・モジュールをダウンロードします。

[指定形式]

```
debugger.Download.LoadModule(fileName = "", downloadOption = DownloadOption.Both, append =
False, flashErase = False)
```

[引数]

<table>
<tr><th>引数</th><th colspan="2">説明</th></tr>
<tr><td>fileName</td><td colspan="2">ダウンロード・ファイルを指定します。</td></tr>
<tr><td rowspan="5">downloadOption</td><td colspan="2">オプションを指定します。
指定可能なオプションを以下に示します。</td></tr>
<tr><td>種類</td><td>説明</td></tr>
<tr><td>DownloadOption.NoSymbol</td><td>シンボル情報を読み込みません。</td></tr>
<tr><td>DownloadOption.SymbolOnly</td><td>シンボル情報のみ読み込みます。</td></tr>
<tr><td>DownloadOption.Both</td><td>シンボル情報とオブジェクト情報の両方を読み込みます（デフォルト）。</td></tr>
<tr><td>append</td><td colspan="2">追加ダウンロードするかどうかを指定します。
True ：追加ダウンロードします。
False ：上書きダウンロードします（デフォルト）。</td></tr>
<tr><td>flashErase</td><td colspan="2">ダウンロード前にフラッシュ・メモリを初期化するかどうかを指定します。
True ：ダウンロード前にフラッシュ・メモリを初期化します。
False ：ダウンロード前にフラッシュ・メモリを初期化しません（デフォルト）。</td></tr>
</table>

[戻り値]

なし

[詳細説明]

- ロード・モジュールをダウンロードします。
- *fileName* を指定しない場合は，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］タブに指定されているファイルをダウンロードします。
- *downloadOption* を指定した場合，指定した内容に従って処理を行います。

［使用例］

```
>>>debugger.Download.LoadModule("C:/test/testModule.lmf")
>>>debugger.Download.LoadModule("C:/test/testModule2.lmf", DownloadOption.SymbolOnly, True)
>>>
```

debugger.Erase

フラッシュ・メモリを消去します。

[指定形式]

```
debugger.Erase(eraseOption = EraseOption.Code)
```

[引数]

引数	説明	
eraseOption	オプションを指定します。 指定可能なオプションを以下に示します。	
	種類	説明
	EraseOption.Code	コード・フラッシュ・メモリを消去します（デフォルト）。
	EraseOption.Data	データフラッシュ・メモリを消去します。
	EraseOption.External	外部空間にあるフラッシュ・メモリを消去します。

注意　IECUBE，IECUBE2，シミュレータにはコード・フラッシュ・メモリの消去機能がないため，IECUBE，IECUBE2，シミュレータを使用する場合は，*eraseOption* の省略，および EraseOption.Code の指定はできません。

[戻り値]

なし

[詳細説明]

- *eraseOption* で指定したフラッシュ・メモリを消去します。

[使用例]

```
>>>debugger.Erase()
>>>debugger.Erase(EraseOption.External)
>>>
```

debugger.GetBreakStatus

ブレーク要因を表示します。

[指定形式]

```
debugger.GetBreakStatus()
```

[引数]

なし

[戻り値]

ブレーク要因の文字列（[詳細説明] 参照）

備考 1. BreakStatus という enum 定義の文字列部分を返します。

2. 条件判断する場合は，“BreakStatus. *文字列*”と記述してください。

[詳細説明]

- ブレーク要因を表示します。

  実行中は，“None”になります。

ブレーク要因の文字列	説明	78K0			RL78,78K0R			V850			
		Iecube	Minicube2 注 1	Simulator	Iecube	Minicube2 注 1	Simulator	Iecube	Minicube 注 2	Minicube2 注 1	Simulator
None	ブレークしていない	○	○	－	○	○	－	○	○	○	－
Manual	強制ブレーク	○	○	○	○	○	○	○	○	○	○
Event	イベントによるブレーク	○	○	○	○	○	○	○	○	○	○
Software	ソフトウエア・ブレーク	○	○	－	○	○	－	○	○	○	－
TraceFull	トレース・フルによるブレーク	○	－	○	○	－	○	○	－	－	○
TraceDelay	トレース・ディレイによるブレーク	○	－	－	○	－	－	－	－	－	－
NonMap	ノンマップ・エリアをアクセス	○	－	○	○	－	○	○	－	－	○
WriteProtect	ライト・プロテクト領域に対してライト	○	－	○	○	－	○	○	－	－	○
ReadProtect	リード・プロテクト領域からリード	○	－	－	－	－	－	－	－	－	－

ブレーク要因の文字列	説明	78K0			RL78,78K0R			V850			
		Iecube	Minicube2 注1	Simulator	Iecube	Minicube2 注1	Simulator	Iecube	Minicube 注2	Minicube2 注1	Simulator
SfrIllegal	SFR に対してイリーガルなアクセス	○	－	－	－	－	－	－	－	－	－
SfrReadProtect	リード禁止の SFR からリード	○	－	－	○	－	－	－	－	－	－
SfrWriteProtect	ライト禁止の SFR に対してライト	○	－	－	○	－	－	－	－	－	－
IorIllegal	周辺 I/O レジスタに対してイリーガルなアクセス（アドレス付き）	－	－	－	－	－	－	○	－	－	－
StackOverflow	スタック・オーバーフローによるブレーク	○	－	－	○	－	－	－	－	－	－
StackUnderflow	スタック・アンダーフローによるブレーク	○	－	－	○	－	－	－	－	－	－
UninitializeStackPointer	スタック・ポインタ初期化忘れによるブレーク	○	－	－	○	－	－	－	－	－	－
UninitializeMemoryRead	初期化していないメモリをリードした	○	－	－	○	－	－	－	－	－	－
TimerOver	実行時間オーバーを検出した	○	－	－	○	－	－	○	－	－	－
UnspecifiedIllegal	周辺チップ機能に関するユーザ・プログラムの不正動作が発生	○	－	－	○	－	－	－	－	－	－
ImsIxsIllegal	IMS，IXS レジスタ不正書き込みによるブレーク	○	－	－	－	－	－	－	－	－	－
BeforeExecution	実行前ブレーク	○	－	－	○	－	－	－	－	－	－
SecurityProtect	セキュリティ保護領域に対してアクセス	－	－	－	－	－	－	－	－	－	－
FlashMacroService	フラッシュ・マクロ・サービス中	－	－	－	－	－	－	－	○	○	－
RetryOver	RETRY 回数オーバ・ブレーク	○	－	－	－	－	－	－	－	－	－
FlashIllegal	フラッシュ・イリーガル・ブレーク	○	－	－	○	－	－	－	－	－	－
Peripheral	周辺からのブレーク	○	－	－	○	－	－	－	－	－	－
WordMissAlignAccess	奇数番地に対するワード・アクセスを行った	－	－	－	○	－	○	－	－	－	－
Temporary	テンポラリ・ブレーク	○	○	○	○	○	○	○	○	○	○
Escape	エスケープ・ブレークによるブレーク	－	－	－	－	－	－	○	○	○	－

ブレーク要因の文字列	説明	78K0			RL78,78K0R			V850			
		Iecube	Minicube2 注1	Simulator	Iecube	Minicube2 注1	Simulator	Iecube	Minicube 注2	Minicube2 注1	Simulator
Fetch	ガード領域，フェッチ禁止領域をフェッチした	○	－	－	○	－	－	－	－	－	－
IRamWriteProtect	IRAM ガード領域の書き込み（アドレス付き）注3	－	－	－	－	－	－	○	－	－	－
IllegalOpcodeTrap	不正命令例外発生によるブレーク	－	－	－	－	－	－	○	△ 注6	－	－
Step	ステップ実行・ブレーク注4	○	○	○	○	○	○	－	－	－	○
FetchGuard	フェッチガード・ブレーク注4	○	－	－	○	－	－	－	－	－	－
TraceStop	トレース・ストップ注4	○	－	－	○	－	－	－	－	－	－
ExecutionFails	実行しようとして失敗注5	○	○	－	○	○	－	○	○	○	－

注 1.　Minicube2，E1Serial，E20Serial のすべてに該当します。

2.　Minicube，E1Jtag，E20Jtag，Minicube2Jtag のすべてに該当します。

3.　ブレーク時に IRAM ガード領域のベリファイ・チェックを行い，値が書き換わっていた場合です（該当アドレスが複数ある場合は，最初のアドレスのみ表示します）。

4.　トレース時のみのブレーク要因です。

5.　ブレーク時のみのブレーク要因です。

6.　V850-MINICUBE で ET コア系デバイス（ME2 等）で，実行後イベントを使用した場合は表示しません。

ブレーク要因の文字列	説明	RX		V850E2			
		E1Jtag, E1Serial E20Jtag, E20Serial	Simulator	Iecube2	Minicube 注2	Minicube2 注1	Simulator
None	ブレークしていない	○	－	○	○	○	－
Manual	強制ブレーク	○	○	○	○	○	○
Event	イベントによるブレーク	○	○	○	○	○	○
Software	ソフトウエア・ブレーク	○	－	○	○	○	－
TraceFull	トレース・フルによるブレーク	○	○	○	－	－	○
NonMap	ノンマップ・エリアをアクセス	－	－	－	－	－	○
WriteProtect	ライト・プロテクト領域に対してライト	－	－	－	－	－	○
TimerOver	実行時間オーバーを検出した	－	－	○	○	－	－
FlashMacroService	フラッシュ・マクロ・サービス中	－	－	○	○	○	－
Temporary	テンポラリ・ブレーク	○	○	○	○	○	○
IllegalOpcodeTrap	不正命令例外発生によるブレーク	－	－	○	○	－	－

ブレーク要因の文字列	説明	RX		V850E2			
		E1Jtag, E1Serial E20Jtag, E20Serial	Simulator	Iecube2	Minicube 注2	Minicube2 注1	Simulator
Step	ステップ実行・ブレーク注3	○	－	－	－	－	○
ExecutionFails	実行しようとして失敗注4	○	－	○	○	○	－
WaitInstruction	WAIT 命令実行によるブレーク	－	○	－	－	－	－
UndefinedInstructionException	未定義命令例外発生によるブレーク	－	○	－	－	－	－
PrivilegeInstructionException	特権命令例外発生によるブレーク	－	○	－	－	－	－
AccessException	アクセス例外発生によるブレーク	－	○	－	－	－	－
FloatingPointException	浮動小数点例外発生によるブレーク	－	○	－	－	－	－
InterruptException	割り込み発生によるブレーク	－	○	－	－	－	－
IntInstructionException	INT 命令例外発生によるブレーク	－	○	－	－	－	－
BrkInstructionException	BRK 命令例外発生によるブレーク	－	○	－	－	－	－
IOFunctionSimulationBreak	周辺機能シミュレーションによるブレーク	－	○	－	－	－	－
IllegalMemoryAccessBreak	不正なメモリ・アクセスによるブレーク	－	○	－	－	－	－
StreamIoError	ストリーム入出力エラーによるブレーク	－	○	－	－	－	－
CoverageMemoryAllocationFailure	カバレッジ・メモリの確保に失敗	－	○	－	－	－	－
TraceMemoryAllocationFailure	トレース・メモリの確保に失敗	－	○	－	－	－	－

注 1. Minicube2，E1Serial，E20Serial のすべてに該当します。

2. Minicube，E1Jtag，E20Jtag，Minicube2Jtag のすべてに該当します。

3. トレース時のみのブレーク要因です。

4. ブレーク時のみのブレーク要因です。

［使用例］

```
>>>debugger.GetBreakStatus()
Temporary
>>>a = debugger.GetBreakStatus()
Temporary
>>>print a
Temporary
>>>if (debugger.GetBreakStatus() == BreakStatus.Temporary):
... print " テンポラリ・ブレークしました "
...
Temporary
テンポラリ・ブレークしました
>>>
```

debugger.GetCpuStatus

現在の CPU の状態を表示します。

[指定形式]

```
debugger.GetCpuStatus()
```

[引数]

なし

[戻り値]

現在の CPU の状態（文字列）

CPU の状態	説明
Hold	パス・ホールド中
HoldStopIdle	HOLD/SOFT STOP/HARD STOP/IDLE モード中
PowOff	ターゲットに電源が供給されていない
Reset	リセット状態中
Standby	スタンバイ・モード中
Stop	ストップ・モード中
StopIdle	SOFT STOP/HARD STOP/IDLE モード中
Wait	ウエイト状態中
Halt	Halt 中
None	該当なし

[詳細説明]

- 現在の CPU の状態を表示します。

[使用例]

```
>>>debugger.GetCpuStatus()
Stop
>>>
```

debugger.GetIeStatus

現在の IE の状態を表示します。

[指定形式]

```
debugger.GetIeStatus()
```

[引数]

なし

[戻り値]

現在の IE の状態（文字列）

IE の状態	説明
Break	ブレーク中
Coverage	カバレッジ動作中
Timer	タイマ動作中
Tracer	トレース動作中
Step	ステップ実行中
Run	ユーザ・プログラム実行中
RunOrStep	ユーザ・プログラム実行中，またはステップ実行中

注意　PM+ のワークスペースを CubeSuite+ のプロジェクトに変換した場合，メイン・プロジェクトにはデバッグ・ツールがありません。そのため，メイン・プロジェクトがアクティブ・プロジェクトである場合は，“None”を返します。
また，デバッグ・ツールと接続前の場合も，“None”を返します。

[詳細説明]

- 現在の IE の状態を表示します。

[使用例]

```
>>>debugger.GetIeStatus()
Run
>>>
```

debugger.GetPC

PC 値を表示します。

[指定形式]

```
debugger.GetPC()
```

[引数]

なし

[戻り値]

PC 値（数値）

[詳細説明]

- PC 値を表示します。

[使用例]

```
>>>debugger.GetPC()
0x92B0
```

debugger.Go

プログラムを継続して実行します。

[指定形式]

```
debugger.Go(goOption = GoOption.Normal)
```

[引数]

引数	説明	
goOption	オプションを指定します。 指定可能なオプションを以下に示します。	
	種類	説明
	GoOption.IgnoreBreak	ブレークポイントを無視した実行を行います。
	GoOption.WaitBreak	プログラムが停止するまで待機します。
	GoOption.Normal	ブレークポイントは有効で，プログラムが停止するまで待機しません（デフォルト）。

[戻り値]

なし

[詳細説明]

- プログラムを継続して実行します。
- *goOption* を指定した場合，指定した内容に従って処理を行います。

[使用例]

```
>>>debugger.Go()
>>>debugger.Go(GoOption.WaitBreak)
>>>
```

debugger.Ie.GetValue/ debugger.Ie.SetValue

IE レジスタ，または DCU レジスタを設定／参照します。

[指定形式]

```
debugger.Ie.GetValue(ieType, address)
debugger.Ie.SetValue(ieType, address, value)
```

[引数]

引数	説明	
ieType	レジスタを指定します。 指定可能なレジスタを以下に示します。	
	種類	説明
	IeType.Reg	IE レジスタ【78K0】【RL78】【78K0R】【IECUBE【V850】】【IECUBE2【V850】】
	IeType.Dcu	DCU レジスタ【IECUBE【V850】】
address	参照／設定アドレスを指定します。	
value	設定値を指定します。	

[戻り値]

debugger.Ie.GetValue はレジスタ値（数値）

debugger.Ie.SetValue はなし

[詳細説明]

- debugger.Ie.GetValue は，*address* で指定したレジスタ値を表示します。
  レジスタの種類は *ieType* で指定します。
- debugger.Ie.SetValue は，*address* で指定したレジスタに *value* を書き込みます。
  レジスタの種類は *ieType* で指定します。

備考　DCU レジスタの参照を行うと，レジスタ値は 0 にリセットされます。

［使用例］

```
>>>debugger.Ie.GetValue(IeType.Reg, 0x100)
0x12
>>>debugger.Ie.SetValue(IeType.Reg, 0x100, 0x10)
>>>debugger.Ie.GetValue(IeType.Reg, 0x100)
0x10
>>>
```

debugger.IsConnected

デバッグ・ツールの接続状態を確認します。

[指定形式]

```
debugger.IsConnected()
```

[引数]

なし

[戻り値]

デバッグ・ツールに接続している場合　：True
デバッグ・ツールに接続していない場合　：False

[詳細説明]

- デバッグ・ツールの接続状態を確認します。

[使用例]

```
>>>if debugger.IsConnected() == True :
...  print "OK"
...
OK
>>>
```

debugger.IsRunning

ユーザ・プログラムの実行状態を確認します。

[指定形式]

```
debugger.IsRunning()
```

[引数]

なし

[戻り値]

ユーザ・プログラムを実行している場合　: True
ユーザ・プログラムを実行していない場合: False

[詳細説明]

- ユーザ・プログラムの実行状態を確認します。

[使用例]

```
>>>if debugger.IsRunning() == True :
...  print "OK"
...
OK
>>>
```

debugger.Jump.File/ debugger.Jump.Address

各種パネルを表示します。

[指定形式]

```
debugger.Jump.File(fileName, lineNumber = 1)
debugger.Jump.Address(jumpType, adddress = 0)
```

[引数]

引数	説明	
fileName	表示するファイル名を指定します。	
lineNumber	表示する行を指定します（デフォルト：1）。	
jumpType	表示するパネルの種類を指定します。 指定可能なパネルの種類を以下に示します。	
	種類	説明
	JumpType.Source	エディタ パネル
	JumpType.Assemble	逆アセンブル パネル
	JumpType.Memory	メモリ パネル
address	表示するアドレスを指定します（デフォルト：0）。	

[戻り値]

なし

[詳細説明]

- debugger.Jump.File は，*fileName* で指定したファイルをエディタ パネルで表示します。
  lineNumber を指定した場合，*fileName* で指定したファイルの *lineNumber* で指定した行が表示されます。
- debugger.Jump.Address は，*jumpType* で指定したパネルを表示します。
  address を指定した場合，指定した *address* に該当する部分を表示します。

[使用例]

```
>>>debugger.Jump.File("C:/test/testJump.c")
>>>debugger.Jump.File("C:/test/testJump.h", 25)
>>>debugger.Jump.Address(JumpType.Memory, 0x2000)
>>>
```

debugger.Map.Clear

マッピング設定をクリアします。

[指定形式]

```
debugger.Map.Clear()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- マッピング設定をクリアします。

[使用例]

```
>>>debugger.Map.Clear()
>>>
```

debugger.Map.Information

マップ情報を表示します。

[指定形式]

```
debugger.Map.Information()
```

[引数]

なし

[戻り値]

マップ情報のリスト

[詳細説明]

- マップ情報を，以下の形式で表示します。

番号： 開始アドレス 終了アドレス アクセス・サイズ メモリ種別

[使用例]

```
>>>debugger.Map.Information()
1: 0x00000000 0x0005FFFF 32 ( 内蔵 ROM 領域 )
2: 0x00060000 0x03FF6FFF 8 ( ノン・マップ領域 )
3: 0x03FF7000 0x03FFEFFF 32 ( 内蔵 RAM 領域 )
4: 0x03FFF000 0x03FFFFFF 8 (SFR)
>>>
```

debugger.Map.Set

メモリ・マッピングの設定を行います。

[指定形式]

```
debugger.Map.Set(mapType, address1, address2, accessSize = 8, cs = "")
```

[引数]

<table>
<tr><th>引数</th><th colspan="2">説明</th></tr>
<tr><td>mapType</td><td colspan="2">メモリ種別を指定します。
指定可能なメモリ種別を以下に示します。</td></tr>
<tr><td></td><td>種類</td><td>説明</td></tr>
<tr><td></td><td>MapType.EmulationRom</td><td>エミュレーション ROM 領域</td></tr>
<tr><td></td><td>MapType.EmulationRam</td><td>エミュレーション RAM 領域</td></tr>
<tr><td></td><td>MapType.Target</td><td>ターゲット・メモリ領域</td></tr>
<tr><td></td><td>MapType.TargetRom</td><td>ターゲット ROM 領域</td></tr>
<tr><td></td><td>MapType.Stack</td><td>スタック領域</td></tr>
<tr><td></td><td>MapType.Protect</td><td>I/O プロテクト領域</td></tr>
<tr><td>address1</td><td colspan="2">マップ開始アドレスを指定します。</td></tr>
<tr><td>address2</td><td colspan="2">マップ終了アドレスを指定します。</td></tr>
<tr><td>accessSize</td><td colspan="2">アクセス・サイズ（単位：ビット）を指定します（デフォルト：8）。
V850 の場合は，8，16，32 のいずれかを指定します。
78K0R【IECUBE】の場合は，8，16 のどちらかを指定します。</td></tr>
<tr><td>cs</td><td colspan="2">チップ・セレクトを指定します（デフォルト：指定なし）。
IECUBE【V850E1】でエミュレーション・メモリ（代替 ROM/RAM) をマッピングする場合は，cs0，cs1，cs2，cs3，cs4，cs5，cs6，cs7 のいずれかのチップ・セレクトを文字列で指定します。
ただし，V850ES シリーズの品種の場合は，チップ・セレクトの割り当てが固定，またはチップ・セレクト機能がないので，省略することができます。
チップ・セレクトを指定する場合は，accessSize を省略することはできません。</td></tr>
</table>

[戻り値]

なし

[詳細説明]

- *mapType* で指定したメモリ種別で，メモリ・マッピングの設定を行います。

［使用例］

```
>>>debugger.Map.Set(MapType.EmulationRom, 0x100000, 0x10ffff)
>>>
```

debugger.Memory.Copy

メモリをコピーします。

[指定形式]

```
debugger.Memory.Copy(address1, address2, address3)
```

[引数]

引数	説明
address1	コピー元の開始アドレスを指定します。
address2	コピー元の終了アドレスを指定します。
address3	コピー先のアドレスを指定します。

[戻り値]

なし

[詳細説明]

- *address1* から *address2* までの間を *address3* にコピーします。

[使用例]

```
>>>debugger.Memory.Copy(0x1000, 0x2000, 0x3000)
>>>
```

debugger.Memory.Fill

メモリを補填します。

[指定形式]

```
debugger.Memory.Fill(address1, address2, value, memoryOption = MemoryOption.Byte)
```

[引数]

引数	説明	
address1	補填開始アドレスを指定します。	
address2	補填終了アドレスを指定します。	
value	補填する値を指定します。	
memoryOption	補填する単位を指定します。 指定可能な単位を以下に示します。	
	種類	説明
	MemoryOption.Byte	バイト単位（8 ビット）（デフォルト）
	MemoryOption.HalfWord	ハーフ・ワード単位（16 ビット）【V850】
	MemoryOption.Word	ワード単位（RL78,78K：16 ビット，V850：32 ビット）

[戻り値]

なし

[詳細説明]

- *address1* から *address2* までの間を *value* で補填します。
- *memoryOption* を指定した場合，指定した内容に従って補填します。

[使用例]

```
>>>debugger.Memory.Fill(0x1000, 0x2000, 0xFF)
>>>debugger.Memory.Fill(0x2000, 0x3000, 0x0A, MemoryOption.Word)
>>>
```

debugger.Memory.Read

メモリを参照します。

[指定形式]

```
debugger.Memory.Read(address, memoryOption = MemoryOption.Byte)
```

[引数]

引数	説明	
address	参照するアドレスを指定します。	
memoryOption	表示する単位を指定します。 指定可能な単位を以下に示します。	
	種類	説明
	MemoryOption.Byte	バイト単位（8 ビット）（デフォルト）
	MemoryOption.HalfWord	ハーフ・ワード単位（16 ビット）【V850】
	MemoryOption.Word	ワード単位（RL78,78K：16 ビット，V850：32 ビット）

[戻り値]

参照したメモリ値（数値）

[詳細説明]

- *address* で指定したアドレスのメモリ値を，*memoryOption* に従って 16 進数で表示します。

[使用例]

```
>>>debugger.Memory.Read(0x100)
0x10
>>>value = debugger.Memory.Read(0x100)
0x10
>>>print value
16
>>>debugger.Memory.Read(0x100, MemoryOption.HalfWord)
0x0010
>>>
```

debugger.Memory.Write

メモリに書き込みます。

[指定形式]

```
debugger.Memory.Write(address, value, memoryOption = MemoryOption.Byte)
```

[引数]

引数	説明	
address	設定するアドレスを指定します。	
value	設定する値を指定します。	
memoryOption	設定する単位を指定します。 指定可能な単位を以下に示します。	
	種類	説明
	MemoryOption.Byte	バイト単位（8 ビット）（デフォルト）
	MemoryOption.HalfWord	ハーフ・ワード単位（16 ビット）【V850】
	MemoryOption.Word	ワード単位（RL78,78K：16 ビット，V850：32 ビット）

[戻り値]

なし

[詳細説明]

- *address* で指定したアドレスに，*memoryOption* に従って *value* を設定します。

[使用例]

```
>>>debugger.Memory.Read(0x100)
0x10
>>>debugger.Memory.Write(0x100, 0xFF)
>>>debugger.Memory.Read(0x100)
0xFF
>>>debugger.Memory.Write(0x100, 0xFE, MemoryOption.HalfWord)
>>>
```

debugger.Next

プロシージャ・ステップ実行を行います。

[指定形式]

```
debugger.Next(nextOption = NextOption.Source)
```

[引数]

引数	説明	
nextOption	実行する単位を指定します。 指定可能な単位を以下に示します。	
	種類	説明
	NextOption.Source	ソースの行単位（デフォルト）
	NextOption.Instruction	命令単位

[戻り値]

なし

[詳細説明]

- プロシージャ・ステップ実行を行います。
  関数呼び出しを行っている場合は，関数実行後に停止します。

[使用例]

```
>>>debugger.Next()
>>>debugger.Next(NextOption.Instruction)
>>>
```

debugger.Register.GetValue

レジスタ，I/O レジスタ，SFR を参照します。

[指定形式]

```
debugger.Register.GetValue(regName)
```

[引数]

引数	説明
regName	参照するレジスタ名を指定します。

[戻り値]

レジスタ値（文字列）

[詳細説明]

- *regName* で指定したレジスタ値を表示します。

[使用例]

```
>>>debugger.Register.GetValue("pc")
0x100
>>>debugger.Register.GetValue("A:RB1")
0x20
>>>debugger.Register.SetValue("pc", 0x200)
>>>debugger.Register.GetValue("pc")
0x200
>>>
```

debugger.Register.SetValue

レジスタ，I/O レジスタ，SFR に値を設定します。

[指定形式]

```
debugger.Register.SetValue(regName, value)
```

[引数]

引数	説明
regName	設定するレジスタ名を指定します。
value	設定する値を指定します。

[戻り値]

なし

[詳細説明]

- *regName* で指定したレジスタに *value* で指定した値を設定します。

[使用例]

```
>>>debugger.Register.GetValue("pc")
0x100
>>>debugger.Register.GetValue("A:RB1")
0x20
>>>debugger.Register.SetValue("pc", 0x200)
>>>debugger.Register.GetValue("pc")
0x200
>>>
```

debugger.Reset

CPU をリセットします。

[指定形式]

```
debugger.Reset()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- CPU をリセットします。

[使用例]

```
>>>debugger.Reset()
>>>
```

debugger.ReturnOut

現在の関数を呼び出したプログラムに戻るまで実行します。

[指定形式]

```
debugger.ReturnOut()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- 現在の関数を呼び出したプログラムに戻るまで実行します。

[使用例]

```
>>>debugger.ReturnOut()
>>>
```

debugger.Run

プログラムをリセット後に実行します。

[指定形式]

```
debugger.Run(runOption = RunOption.Normal)
```

[引数]

引数	説明	
runOption	オプションを指定します。 指定可能なオプションを以下に示します。	
	種類	説明
	RunOption.WaitBreak	プログラムが停止するまで待機します。
	RunOption.Normal	ブレークポイントは有効で，プログラムが停止するまで待機しません（デフォルト）。

[戻り値]

なし

[詳細説明]

- プログラムをリセット後に実行します。

  runOption に RunOption.WaitBreak を指定した場合，プログラムの停止まで待機します。

[使用例]

```
>>>debugger.Run()
>>>debugger.Run(RunOption.WaitBreak)
```

debugger.Step

ステップ実行を行います。

[指定形式]

```
debugger.Step(stepOption = StepOption.Source)
```

[引数]

<table>
<tr><th>引数</th><th colspan="2">説明</th></tr>
<tr><td rowspan="4">stepOption</td><td colspan="2">実行する単位を指定します。
指定可能な単位を以下に示します。</td></tr>
<tr><td>種類</td><td>説明</td></tr>
<tr><td>StepOption.Source</td><td>ソースの行単位（デフォルト）</td></tr>
<tr><td>StepOption.Instruction</td><td>命令単位</td></tr>
</table>

[戻り値]

なし

[詳細説明]

- ステップ実行を行います。
  関数呼び出しを行っている場合は，関数の先頭で停止します。

[使用例]

```
>>>debugger.Step()
>>>debugger.Step(StepOption.Instruction)
```

debugger.Stop

デバッグ・ツールの実行を停止します。

[指定形式]

```
debugger.Stop()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- デバッグ・ツールの実行を停止します。
  プログラムを強制的に停止します。

[使用例]

```
>>>debugger.Stop()
>>>
```

debugger.Upload.Binary

メモリ・データをバイナリ形式で保存します。

[指定形式]

```
debugger.Upload.Binary(fileName, address1, address2, force = False)
```

[引数]

引数	説明
fileName	ファイル名を指定します。
address1	アップロード開始アドレスを指定します。
address2	アップロード終了アドレスを指定します。
force	上書きをするかどうかを指定します。 True　：上書きします。 False ：上書きしません（デフォルト）。

[戻り値]

なし

[詳細説明]

- *address1* から *address2* までのメモリ・データをバイナリ形式で保存します。

[使用例]

```
>>>debugger.Upload.Binary("C:/test/testBinary.bin", 0x1000, 0x2000, True)
>>>
```

debugger.Upload.Coverage

カバレッジ・データを保存します。【IECUBE】【IECUBE2】【シミュレータ】

[指定形式]

```
debugger.Upload.Coverage(fileName, force = False)
```

[引数]

引数	説明
fileName	ファイル名を指定します。
force	上書きをするかどうかを指定します。 True　：上書きします。 False：上書きしません（デフォルト）。

[戻り値]

なし

[詳細説明]

- カバレッジ・データをファイルに保存します。

[使用例]

```
>>>debugger.Upload.Coverage("C:/test/coverageData.csrcv")
>>>
```

debugger.Upload.Intel

メモリ・データをインテル形式で保存します。

[指定形式]

```
debugger.Upload.Intel(fileName, address1, address2, force = False)
```

[引数]

引数	説明
fileName	ファイル名を指定します。
address1	アップロード開始アドレスを指定します。
address2	アップロード終了アドレスを指定します。
force	上書きをするかどうかを指定します。 True ：上書きします。 False ：上書きしません（デフォルト）。

[戻り値]

なし

[詳細説明]

- *address1* から *address2* までのメモリ・データをインテル形式で保存します。

[使用例]

```
>>>debugger.Upload.Intel("C:/test/testIntel.hex", 0x1000, 0x2000, True)
>>>
```

debugger.Upload.IntelIdTag

メモリ・データを ID タグ付きインテル形式で保存します。

[指定形式]

```
debugger.Upload.IntelIdTag(fileName, address1, address2, force = False)
```

[引数]

引数	説明
fileName	ファイル名を指定します。
address1	アップロード開始アドレスを指定します。
address2	アップロード終了アドレスを指定します。
force	上書きをするかどうかを指定します。 True ：上書きします。 False ：上書きしません（デフォルト）。

[戻り値]

なし

[詳細説明]

- *address1* から *address2* までのメモリ・データを ID タグ付きインテル形式で保存します。

[使用例]

```
>>>debugger.Upload.IntelIdTag("C:/test/testIdTagIntel.hex", 0x1000, 0x2000, True)
>>>
```

debugger.Upload.Motorola

メモリ・データをモトローラ形式で保存します。

[指定形式]

```
debugger.Upload.Motorola(fileName, address1, address2, force = False)
```

[引数]

引数	説明
fileName	ファイル名を指定します。
address1	アップロード開始アドレスを指定します。
address2	アップロード終了アドレスを指定します。
force	上書きをするかどうかを指定します。 True　：上書きします。 False ：上書きしません（デフォルト）。

[戻り値]

なし

[詳細説明]

- *address1* から *address2* までのメモリ・データをモトローラ形式で保存します。

[使用例]

```
>>>debugger.Upload.Motorola("C:/test/testMotorola.hex", 0x1000, 0x2000, True)
>>>
```

debugger.Upload.MotorolaIdTag

メモリ・データを ID タグ付きモトローラ形式で保存します。

[指定形式]

```
debugger.Upload.MotorolaIdTag(fileName, address1, address2, force = False)
```

[引数]

引数	説明
fileName	ファイル名を指定します。
address1	アップロード開始アドレスを指定します。
address2	アップロード終了アドレスを指定します。
force	上書きをするかどうかを指定します。 True ：上書きします。 False ：上書きしません（デフォルト）。

[戻り値]

なし

[詳細説明]

- *address1* から *address2* までのメモリ・データを ID タグ付きモトローラ形式で保存します。

[使用例]

```
>>>debugger.Upload.MotorolaIdTag("C:/test/testIdTagMotorola.hex", 0x1000, 0x2000, True)
>>>
```

debugger.Upload.Tektronix

メモリ・データをテクトロニクス形式で保存します。

[指定形式]

```
debugger.Upload.Tektronix(fileName, address1, address2, force = False)
```

[引数]

引数	説明
fileName	ファイル名を指定します。
address1	アップロード開始アドレスを指定します。
address2	アップロード終了アドレスを指定します。
force	上書きをするかどうかを指定します。 True ：上書きします。 False ：上書きしません（デフォルト）。

[戻り値]

なし

[詳細説明]

- *address1* から *address2* までのメモリ・データをテクトロニクス形式で保存します。

[使用例]

```
>>>debugger.Upload.Tektronix("C:/test/testTektronix.hex", 0x1000, 0x2000, True)
>>>
```

debugger.Upload.TektronixIdTag

メモリ・データを ID タグ付きテクトロニクス形式で保存します。

[指定形式]

```
debugger.Upload.TektronixIdTag(fileName, address1, address2, force = False)
```

[引数]

引数	説明
fileName	ファイル名を指定します。
address1	アップロード開始アドレスを指定します。
address2	アップロード終了アドレスを指定します。
force	上書きをするかどうかを指定します。 True ：上書きします。 False ：上書きしません（デフォルト）。

[戻り値]

なし

[詳細説明]

- *address1* から *address2* までのメモリ・データを ID タグ付きテクトロニクス形式で保存します。

[使用例]

```
>>>debugger.Upload.TektronixIdTag("C:/test/testIdTagTektronix.hex", 0x1000, 0x2000, True)
>>>
```

debugger.Watch.GetValue

変数値を参照します。

[指定形式]

```
debugger.Watch.GetValue(variableName, encode = Encoding.Default, watchOption =
WatchOption.Auto)
```

[引数]

引数	説明	
variableName	参照する変数名，レジスタ名，I/O レジスタ名 /SFR レジスタ名を指定します。	
encode	文字列表示時のエンコードを指定します。 デフォルトでは，システムのエンコードを使用します。 エンコード名は，.NET の仕様に準拠します。 例）Encoding.utf-8，Encoding.euc-jp	
watchOption	オプションを指定します。 指定可能なオプションを以下に示します。	
	種類	説明
	WatchOption.Auto	自動判別して表示します（デフォルト）。
	WatchOption.Binary	2 進数で表示します。
	WatchOption.Octal	8 進数で表示します。
	WatchOption.Decimal	10 進数で表示します。
	WatchOption.SignedDecimal	符号あり 10 進数で表示します。
	WatchOption.UnsignedDecimal	符号なし 10 進数で表示します。
	WatchOption.Hexdecimal	16 進数で表示します。
	WatchOption.String	文字列で表示します。
	WatchOption.Sizeof	変数のサイズを 10 進数で表示します。
	WatchOption.Float	float 型で表示します。
	WatchOption.Double	double 型で表示します。

[戻り値]

表示した値を *watchOption* で指定した型で返します。

watchOption に“WatchOption.Auto”を指定した場合は，変数値にあわせた型で返します。

ただし，戻り値が double 型の場合は string 型で返します（*watchOption* に“WatchOption.Double”を指定した場合，および *watchOption* に“WatchOption.Auto”を指定して戻り値が double 型だった場合）。

[詳細説明]

- *variableName* で指定した変数値を表示します。
- *encode* を指定した場合，*encode* を使用してエンコードを行います。
- *watchOption* を指定した場合，*watchOption* に従って表示します。

[使用例]

```
>>>debugger.Watch.GetValue("testVal")
128
>>>debugger.Watch.GetValue("testVal", WatchOption.Hexdecimal)
0x80
>>>debugger.Watch.GetValue("testVal", WatchOption.Binary)
0b10000000
```

debugger.Watch.SetValue

変数値を設定します。

[指定形式]

```
debugger.Watch.SetValue(variableName, value)
```

[引数]

引数	説明
variableName	設定する変数名，レジスタ名，I/O レジスタ名 /SFR レジスタ名を指定します。
value	設定する値を指定します。

[戻り値]

なし

[詳細説明]

- *variableName* で指定した変数，レジスタ，I/O レジスタ /SFR レジスタに *value* で指定した値を設定します。

[使用例]

```
>>>debugger.Watch.GetValue("testVal")
128
>>>debugger.Watch.GetValue("testVal", WatchOption.Hexdecimal)
0x80
>>>debugger.Watch.GetValue("testVal", WatchOption.Binary)
0b10000000
>>>debugger.Watch.SetValue("testVal", 100)
>>>debugger.Watch.GetValue("testVal")
100
>>>debugger.Watch.GetValue("testVal", WatchOption.Hexdecimal)
0x64
>>>debugger.Watch.GetValue("testVal", WatchOption.Binary)
0b1100100
>>>debugger.Watch.SetValue("testVal", 0x256)
>>>debugger.Watch.GetValue("testVal", WatchOption.Hexdecimal)
0x256
```

debugger.Where

スタックのバック・トレースを表示します。

[指定形式]

```
debugger.Where()
```

[引数]

なし

[戻り値]

バック・トレースのリスト

[詳細説明]

- スタックのバック・トレースを表示します。

注意　「--- Information below might be inaccurate.」を表示した場合，それ以降の表示は信用できない可能性があります。【RL78】【78K0R】

[使用例]

```
>>>debugger.Where()
1: test2.c#sub2#13
--- Information below might be inaccurete.
2:func.c#func#34
>>>
```

debugger.Whereami

ロケーションを表示します。

[指定形式]

```
debugger.Whereami(address)
```

[引数]

引数	説明
address	ロケーション表示するアドレスを指定します。

[戻り値]

ロケーションの文字列

[詳細説明]

- *address* で指定したアドレスに対するロケーションを表示します。
- 通常は，以下の形式でロケーションを表示します。

```
ファイル名#関数名 at ファイル名#行番号
```

ただし，アドレスに対する関数，または行番号が見つからない場合は，以下の形式でロケーションを表示します。

```
at シンボル名+オフセット値
```

シンボルが見つからない場合は，以下の形式でロケーションを表示します。

```
at アドレス値
```

- *address* を省略した場合，pc 値のロケーションを表示します。

[使用例]

```
>>>debugger.Whereami()
foo.c#func at foo.c#100
>>>debugger.Whereami(0x100)
foo.c#main at foo.c#20
>>>
```

debugger.XCoverage.Clear

カバレッジ・メモリをクリアします。【IECUBE】【IECUBE2】【シミュレータ】

[指定形式]

```
debugger.XCoverage.Clear()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- カバレッジ・メモリをクリアします。

[使用例]

```
>>>debugger.XCoverageClear()
>>>
```

debugger.XCoverage.GetCoverage

カバレッジを取得します。【IECUBE】【IECUBE2】【シミュレータ】

[指定形式]

```
debugger.XCoverage.GetCoverage(funcName, progName = "", fileName = "")
```

[引数]

引数	説明
funcName	カバレッジを取得する関数名を指定します。
progName	関数が含まれているロード・モジュール名を指定します。 単数ロード・モジュールの場合は省略可能です（デフォルト）。
fileName	関数が含まれているファイル名を指定します。 グローバル関数の場合は省略可能です（デフォルト）。

注意　2 つ以上の引数を指定する場合は，3 つの引数を指定する必要があります。

[戻り値]

% を除いた値（数値）

備考　関数の実行結果には，“%”を付けて表示します。

[詳細説明]

- *funcName* で指定した関数のカバレッジを取得します。
- 複数ロード・モジュールの場合は，*progName* を指定してください。
- スタティック関数の場合は，*fileName* を指定してください。

[使用例]

```
>>>debugger.XCoverage.GetCoverage("TestInit", "C:/test/Test.out", "C:/test/Test.c")
81.50%
>>>
```

debugger.XRunBreak.Delete

XRunBreak の設定情報を削除します。【V850 シミュレータ】

[指定形式]

```
debugger.XRunBreak.Delete()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- XRunBreak の設定情報を削除します。

[使用例]

```
>>>debugger.XRunBreak.Refer()
None
>>>debugger.XRunBreak.Set(1, TimeType.S, True)
>>>debugger.XRunBreak.Refer()
1Second Periodic
>>>debugger.XRunBreak.Delete()
>>>debugger.XRunBreak.Refer()
None
```

debugger.XRunBreak.Refer

XRunBreak の設定情報を表示します。【V850 シミュレータ】

[指定形式]

```
debugger.XRunBreak.Refer()
```

[引数]

なし

[戻り値]

周期時間の数値と周期情報（TimeType）のリスト

[詳細説明]

- 設定されている XRunBreak の周期情報（周期時間 [Periodic]）を表示します。
- XRunBreak の設定が存在しない場合は，“None”を表示します。

[使用例]

```
>>>debugger.XRunBreak.Refer()
None
>>>debugger.XRunBreak.Set(1, TimeType.S, True)
>>>debugger.XRunBreak.Refer()
1Second Periodic
```

debugger.XRunBreak.Set

XRunBreak の情報を設定します。【V850 シミュレータ】

[指定形式]

```
debugger.XRunBreak.Set(time, timeType = TimeType.Ms, periodic = False)
```

[引数]

<table>
<tr><th>引数</th><th colspan="2">説明</th></tr>
<tr><td>time</td><td colspan="2">ブレーク時間を指定します。</td></tr>
<tr><td>timeType</td><td colspan="2">ブレーク時間の単位を指定します。
指定可能な単位を以下に示します。</td></tr>
<tr><td></td><td>種類</td><td>説明</td></tr>
<tr><td></td><td>TimeType.Min</td><td>分単位</td></tr>
<tr><td></td><td>TimeType.S</td><td>秒単位</td></tr>
<tr><td></td><td>TimeType.Ms</td><td>ミリ秒単位（デフォルト）</td></tr>
<tr><td></td><td>TimeType.Us</td><td>マイクロ秒単位</td></tr>
<tr><td></td><td>TimeType.Ns</td><td>ナノ秒単位</td></tr>
<tr><td>periodic</td><td colspan="2">指定時間毎にコールバックを呼び出すかどうかを指定します。
True　：指定時間毎に呼び出します。
False　：1 回のみ呼び出します（デフォルト）。</td></tr>
</table>

[戻り値]

なし

[詳細説明]

- XRunBreak の情報を設定します。
- XRunBreak のコール間隔は，シミュレータに依存します。
- 指定時間経過後に処理する Python 関数は Hook 関数で登録します。詳細は Hook 関数を参照してください。

注意　XRunBreak 情報を設定後のプログラム実行中に，

・CPU リセット

・CPU リセット後，プログラムを実行

・ブレークポイントの設定

を行う場合は，一度プラグラムを停止してから行ってください。

［使用例］

```
>>>debugger.XRunBreak.Refer()
None
>>>debugger.XRunBreak.Set(1, TimeType.S, True)
>>>debugger.XRunBreak.Refer()
1Second Periodic
```

debugger.XTime

Go-Break 間の時間情報を表示します。

[指定形式]

```
debugger.XTime()
```

[引数]

なし

[戻り値]

時間情報の文字列

[詳細説明]

- Go-Break 間の時間情報を nsec 単位で表示します。

[使用例]

```
>>>debugger.XTime()
9820214200nsec
>>>
```

debugger.XTrace.Clear

トレース・メモリをクリアします。【IECUBE】【IECUBE2】【シミュレータ】

[指定形式]

```
debugger.XTrace.Clear()
```

[引数]

なし

[戻り値]

なし

[詳細説明]

- トレース・メモリをクリアします。

[使用例]

```
>>>debugger.XTrace.Clear()
>>>
```

debugger.XTrace.Dump

トレース・データをダンプします。【IECUBE】【IECUBE2】【シミュレータ】

[指定形式]

```
debugger.XTrace.Dump(frameCount, fileName = "", append = False)
```

[引数]

引数	説明
frameCount	ダンプ数を指定します。
fileName	ダンプするファイル名を指定します（デフォルト：指定なし）。
append	トレース・データをファイルに追記するかどうかを指定します。 True　：トレース・データをファイルに追記します。 False ：トレース・データをファイルに追記しません（デフォルト）。

[戻り値]

トレース情報のリスト

[詳細説明]

- *frameCount* で指定した数分のトレース・データをダンプします。
- *fileName* を指定した場合，トレース・データをファイルに書き込みます。
- *append* に“True”を指定した場合，トレース・データをファイルに追記します。

[使用例]

```
>>>debugger.XTrace.Dump(3)
    1851  00h00min00s003ms696µs000ns  0x000003be  cmp r11, r14
    1852  00h00min00s003ms700µs000ns  0x000003c0  blt _func_static3+0x2c
    1853  00h00min00s003ms702µs000ns  0x000003c2  jarl _errfunc, lp
>>>debugger.XTrace.Dump(10, "C:/test/TestTrace.txt")
>>>
```

G. 3. 5　CubeSuite+ Python クラス

以下に，CubeSuite+ Python クラスの一覧を示します。

表 G—5　CubeSuite+ Python クラス

クラス名	機能概要
BreakCondition	ブレーク条件を作成します。
BuildCompletedEventArgs	ビルド完了時のパラメータを保持します。

BreakCondition

ブレーク条件を作成します。

[型]

```
class BreakCondition:
        Address = ""
        Data = None
        AccessSize = None
        BreakType = BreakType.Hardware
```

[変数]

<table>
<tr><th>変数</th><th colspan="2">説明</th></tr>
<tr><td>Address</td><td colspan="2">ブレークを設定するアドレスを指定します。
必ず指定してください。</td></tr>
<tr><td>Data</td><td colspan="2">データのブレーク条件を設定する数値を指定します。
“None”を指定した場合，データ条件は無視されます。</td></tr>
<tr><td>AccessSize</td><td colspan="2">アクセス・サイズ（8，16，32 のいずれか）を指定します。
“None”を指定した場合，すべてのアクセス・サイズを指定したことになります。</td></tr>
<tr><td rowspan="6">BreakType</td><td colspan="2">ブレークの種類を指定します。
指定可能なブレークの種類を以下に示します。</td></tr>
<tr><td>種類</td><td>説明</td></tr>
<tr><td>BreakType.Software</td><td>ソフトウエア・ブレーク（シミュレータ以外）</td></tr>
<tr><td>BreakType.Hardware</td><td>ハードウエア・ブレーク（デフォルト）</td></tr>
<tr><td>BreakType.Read</td><td>データ・リード・ブレーク</td></tr>
<tr><td>BreakType.Write</td><td>データ・ライト・ブレーク</td></tr>
<tr><td>BreakType.Access</td><td>データ・アクセス・ブレーク</td></tr>
</table>

[詳細説明]

- BreakCondition は class 形式になっており，ブレーク条件を変数に指定します。
  ブレーク条件を作成するには，インスタンスを生成し，生成したインスタンスに対して条件を設定します。

[使用例]

```
>>>executeBreak = BreakCondition()              …インスタンスを生成
>>>executeBreak.Address = "main"
>>>executeBreak.BreakType = BreakType.Software
>>>debugger.Breakpoint.Set(executeBreak)        …ブレークポイント設定関数の引数に指定
>>>
>>>dataBreak = BreakCondition()                 …インスタンスを生成
>>>dataBreak.Address = "chData"
>>>dataBreak.Data = 0x10
>>>dataBreak.BreakType = BreakType.Access
>>>debugger.Breakpoint.Set(dataBreak)           …ブレークポイント設定関数の引数に指定
>>>
>>>executeBreak.Address = "sub + 0x10"          …ブレーク条件を再利用
>>>debugger.Breakpoint.Set(executeBreak)        …ブレークポイント設定関数の引数に指定
>>>
```

BuildCompletedEventArgs

ビルド完了時のパラメータを保持します。

[型]

```
class BuildCompletedEventArgs:
        Error = None
        Cancelled = False
        HasBuildError = False
        HasBuildWarning = False
```

[変数]

変数	説明
Error	ビルドで例外が発生した場合，エラーの内容（System.Exception）が格納されます。
Cancelled	ビルドの実行がキャンセルされたかどうかが格納されます。
HasBuildError	ビルドでエラーが発生したかどうかが格納されます。
HasBuildWarning	ビルドでワーニングが発生したかどうかが格納されます。

[詳細説明]

- BreakCompletedEventArgs は class 形式になっており，build.BuildCompleted イベントが発生した場合のみ引数として渡されます。
  そのため，この class のインスタンスを生成することはできません。

[使用例]

```
>>>def buildCompleted(sender, e):
... print "Error = {0}".format(e.Error)
... print "BuildError = " + e.HasBuildError.ToString()
... print "BuildWarning = " + e.HasBuildWarning.ToString()
... print "BuildCancelled = " + e.Cancelled.ToString()
...
>>>build.BuildCompleted += buildCompleted   …イベントの接続
>>>build.All(True)
Error = None
BuildError = False
BuildWarning = False
BuildCancelled = False
True
```

```
>>>                                                    …例外が発生した場合，下記のように表示されます
>>>build.All(True)
Error = System.Exception: ビルド中にエラーが発生しました。(E0203001)
BuildError = False
BuildWarning = False
BuildCancelled = False
False
>>>
>>>                                                    …ビルド・エラーが発生した場合，下記のように表示されます
>>>build.All(True)
Error = None
BuildError = True
BuildWarning = False
BuildCancelled = False
False
>>>
```

G. 3. 6　CubeSuite+ Python プロパティ（共通）

以下に，CubeSuite+ Python プロパティ（共通）の一覧を示します。

表 G—6　CubeSuite+ Python プロパティ（共通）

プロパティ名	機能概要
common.ConsoleClear	アクティブ・プロジェクト変更時に Python コンソールの表示をクリアする／しないを設定／参照します。
common.Output	CubeSuite+ 用 Python 関数の戻り値，またはエラー内容を参照します。
common.ThrowExcept	Python 関数の実行時に例外を発生させる／させないを設定／参照します。
common.UsedRemoting	Python コンソールの起動時に外部ツールと連携する機能を有効にする／しないを設定／参照します。
common.Version	CubeSuite+ のバージョンを参照します。
common.ViewLine	Python コンソールの表示桁数を設定／参照します。
common.ViewOutput	CubeSuite+ 用 Python 関数の実行結果，またはエラー内容を Python コンソールに表示する／しないを設定／参照します。

common.ConsoleClear

アクティブ・プロジェクト変更時に Python コンソールの表示をクリアする／しないを設定／参照します。

[指定形式]

```
common.ConsoleClear = bool
```

[設定]

設定	説明
bool	アクティブ・プロジェクト変更時に Python コンソールの表示をクリアするかどうかを設定します。 True ：Python コンソールの表示をクリアします（デフォルト）。 False ：Python コンソールの表示をクリアしません。

[参照]

現在の設定値

[詳細説明]

- アクティブ・プロジェクト変更時に Python コンソールの表示をクリアする／しないを設定／参照します。

[使用例]

```
>>>print common.ConsoleClear
True
>>>common.ConsoleClear = False
```

common.Output

CubeSuite+ 用 Python 関数の戻り値，またはエラー内容を参照します。

[指定形式]

```
common.Output
```

[設定]

なし

[参照]

CubeSuite+ 用 Python 関数の戻り値，またはエラー・メッセージ（文字列）

注意　エラー・メッセージを参照できるのは，common.ThrowExcept で例外を発生させない（False）設定をしている場合のみです。

備考　次の CubeSuite+ 用 Python 関数が実行されるまで，参照内容を保持します。

[詳細説明]

- CubeSuite+ 用 Python 関数の戻り値，またはエラー内容を参照します。

[使用例]

```
>>>debugger.Memory.Read("data")
0x0
>>>print common.Output
0
```

common.ThrowExcept

Python 関数の実行時に例外を発生させる／させないを設定／参照します。

[指定形式]

```
common.ThrowExcept = bool
```

[設定]

設定	説明
bool	Python 関数の実行時に例外を発生させるかどうかを設定します。 True ：例外を発生させます。 False ：例外を発生させません（デフォルト）。

[参照]

現在の設定値

[詳細説明]

- Python 関数の実行時に例外を発生させる／させないを設定／参照します。
- try ～ except を使用したい場合は，*bool* に“True”を設定します。

[使用例]

```
>>>print common.ThrowExcept
False
>>>common.ThrowExcept = True
```

common.UsedRemoting

Python コンソールの起動時に外部ツールと連携する機能を有効にする／しないを設定／参照します。

[指定形式]

```
common.UsedRemoting = bool
```

[設定]

設定	説明
bool	Python コンソールの起動時に外部ツールと連携する機能を有効にするかどうかを設定します。 True ：外部ツールと連携する機能を有効にします（デフォルト）。 False ：外部ツールと連携する機能を無効にします。

[参照]

現在の設定値

[詳細説明]

- Python コンソールの起動時に外部ツールと連携する機能を有効にする／しないを設定／参照します。

[使用例]

```
>>>print common.UsedRemoting
False
>>>common.UsedRemoting = True
```

common.Version

CubeSuite+ のバージョンを参照します。

[指定形式]

```
common.Version
```

[設定]

なし

[参照]

CubeSuite+ のバージョン

[詳細説明]

- CubeSuite+ のバージョンを参照します。

[使用例]

```
>>>print common.Version
V1.02.00  [01 Apr 2012]
```

common.ViewLine

Python コンソールの表示桁数を設定／参照します。

[指定形式]

```
common.ViewLine = number
```

[設定]

設定	説明
number	Python コンソールの表示桁数を数値で設定します（デフォルト：10000）。

[参照]

現在の設定値

[詳細説明]

- Python コンソールの表示桁数を設定／参照します。

[使用例]

```
>>>print common.ViewLine
10000
>>>common.ViewLine = 20000
```

common.ViewOutput

CubeSuite+ 用 Python 関数の実行結果，またはエラー内容を Python コンソールに表示する／しないを設定／参照します。

[指定形式]

```
common.ViewOutput = bool
```

[設定]

設定	説明
bool	CubeSuite+ 用 Python 関数の実行結果，またはエラー内容を Python コンソールに表示するかどうかを設定します。 True ：Python コンソールに表示します（デフォルト）。 False ：Python コンソールに表示しません。

[参照]

現在の設定値

[詳細説明]

- CubeSuite+ 用 Python 関数の実行結果，またはエラー内容を Python コンソールに表示する／しないを設定／参照します。

[使用例]

```
>>>print common.ViewOutput
False
>>>common.ViewOutput = True
```

G. 3. 7　CubeSuite+ Python プロパティ（プロジェクト用）

以下に，CubeSuite+ Python プロパティ（プロジェクト用）の一覧を示します。

表 G—7　CubeSuite+ Python プロパティ（プロジェクト用）

プロパティ名	機能概要
project.Device	アクティブ・プロジェクトのマイクロコントローラ名を参照します。
project.IsOpen	プロジェクトを読み込んでいるかどうかを確認します。
project.Name	アクティブ・プロジェクト・ファイル名（パスなし）を参照します。
project.Nickname	アクティブ・プロジェクトのマイクロコントローラ名の愛称を参照します。
project.Path	アクティブ・プロジェクト・ファイル名（パス付き）を参照します。

project.Device

アクティブ・プロジェクトのマイクロコントローラ名を参照します。

[指定形式]

```
project.Device
```

[設定]

なし

[参照]

アクティブ・プロジェクトのマイクロコントローラ名

[詳細説明]

- アクティブ・プロジェクトのマイクロコントローラ名を参照します。

[使用例]

```
>>>print project.Device
R5F100LE
```

project.IsOpen

プロジェクトを読み込んでいるかどうかを確認します。

[指定形式]

```
project.IsOpen
```

[設定]

なし

[参照]

プロジェクトを読み込んでいる場合　：True
プロジェクトを読み込んでいない場合　：False

[詳細説明]

- プロジェクトが開いているかどうかを確認します。

[使用例]

```
>>>print project.IsOpen
True
>>>
```

project.Name

アクティブ・プロジェクト・ファイル名（パスなし）を参照します。

[指定形式]

```
project.Name
```

[設定]

なし

[参照]

アクティブ・プロジェクト・ファイル名（パスなし）

[詳細説明]

- アクティブ・プロジェクト・ファイル名（パスなし）を参照します。

[使用例]

```
>>>print project.Name
test.mtpj
```

project.Nickname

アクティブ・プロジェクトのマイクロコントローラ名の愛称を参照します。

[指定形式]

```
project.Nickname
```

[設定]

なし

[参照]

アクティブ・プロジェクトのマイクロコントローラ名の愛称

[詳細説明]

- アクティブ・プロジェクトのマイクロコントローラ名の愛称を参照します。

[使用例]

```
>>>print project.Nickname
RL78/G13 (ROM:64KB)
```

project.Path

アクティブ・プロジェクト・ファイル名（パス付き）を参照します。

[指定形式]

```
project.Path
```

[設定]

なし

[参照]

アクティブ・プロジェクト・ファイル名（パス付き）

[詳細説明]

- アクティブ・プロジェクト・ファイル名（パス付き）を参照します。

[使用例]

```
>>>print project.Path
C:¥project¥test.mtpj
```

G. 3. 8　CubeSuite+ Python プロパティ（ビルド・ツール用）

以下に，CubeSuite+ Python プロパティ（ビルド・ツール用）の一覧を示します。

表 G—8　CubeSuite+ Python プロパティ（ビルド・ツール用）

プロパティ名	機能概要
build.IsBuilding	ビルド実行中かどうかを確認します。
build.Link.LibraryFile	アクティブ・プロジェクトのライブラリ・ファイルの設定／参照を行います。

build.IsBuilding

ビルド実行中かどうかを確認します。

[指定形式]

```
build.IsBuilding
```

[設定]

なし

[参照]

ビルド実行中の場合　：True
ビルドを行っていない場合　：False

[詳細説明]

- ビルド実行中かどうかを確認します。

[使用例]

```
>>>print build.IsBuilding
False
>>>
```

build.Link.LibraryFile

アクティブ・プロジェクトのライブラリ・ファイルの設定／参照を行います。

[指定形式]

```
build.Link.LibraryFile = filelist
```

[設定]

設定	説明
filelist	アクティブ・プロジェクトのライブラリ・ファイルを文字列のリストで設定します。

[参照]

ライブラリ・ファイルのリスト

[詳細説明]

- アクティブ・プロジェクトのライブラリ・ファイルの設定／参照を行います。
- 設定を変更する場合は，参照したリストに対して追加／変更してください。

[使用例]

```
>>>lib1 = build.Link.LibraryFile        …現在の設定を参照してライブラリ・ファイルを追加
test1.lib
test2.lib
>>>lib1.append("test3.lib")
>>>build.Link.LibraryFile = lib1
>>>print build.Link.LibraryFile
test1.lib
test2.lib
test3.lib
>>>
>>>lib2 = ["test1.lib", "test2.lib"]    …複数のライブラリ・ファイルを設定
>>>build.Link.LibraryFile = lib2
>>>print build.Link.LibraryFile
test1.lib
test2.lib
```

G. 3. 9　CubeSuite+ Python プロパティ（デバッグ・ツール用）

以下に，CubeSuite+ Python プロパティ（デバッグ・ツール用）の一覧を示します。

表 G—9　CubeSuite+ Python プロパティ（デバッグ・ツール用）

プロパティ名	機能概要
debugger.Memory.NoVerify	書き込み時のベリファイ設定を切り替えます。
debugger.Opiton.Coverage/ debugger.Option.OpenBreak/ debugger.Option.ReuseCoverageData/ debugger.Option.Timer/ debugger.Option.Trace/ debugger.Option.UseTraceData	デバッグ・ツールのオプションを設定／参照します。
debugger.XTrace.Addup/ debugger.XTrace.Complement/ debugger.XTrace.Mode	トレース・オプションを設定／参照します。

debugger.Memory.NoVerify

書き込み時のベリファイ設定を切り替えます。【シミュレータ以外】

[指定形式]

```
debugger.Memory.NoVerify = noverify
```

[設定]

設定	説明
noverify	書き込み時にベリファイするかどうかを設定します。 True ：書き込み時にベリファイします。 False ：書き込み時にベリファイしません。

[参照]

設定されている値

注意　PM+ のワークスペースを CubeSuite+ のプロジェクトに変換した場合，メイン・プロジェクトにはデバッグ・ツールがありません。そのため，メイン・プロジェクトがアクティブ・プロジェクトである場合は，“None”を返します。

[詳細説明]

- 書き込み時のベリファイ設定を切り替えます。

[使用例]

```
>>>print debugger.Memory.NoVerify
False
>>>debugger. Memory.NoVerify = True
>>>print debugger. Memory.NoVerify
True
>>>
```

debugger.Opiton.Coverage/
debugger.Option.OpenBreak/
debugger.Option.ReuseCoverageData/
debugger.Option.Timer/
debugger.Option.Trace/
debugger.Option.UseTraceData

デバッグ・ツールのオプションを設定／参照します。

[指定形式]

```
debugger.Opiton.Coverage = coverage
debugger.Option.OpenBreak = openBreak
debugger.Option.ReuseCoverageData = reuseCoverageData
debugger.Option.Timer = timer
debugger.Option.Trace = trace
debugger.Option.UseTraceData = useTraceDataType
```

[設定]

設定	説明
coverage	カバレッジ機能を使用するかどうかを設定します。【IECUBE】【IECUBE2】【シミュレータ】 True ：カバレッジ機能を使用します。 False ：カバレッジ機能を使用しません。
openBreak	オープン・ブレーク機能を使用するかどうかを設定します。 True ：オープン・ブレーク機能を使用します。 False ：オープン・ブレーク機能を使用しません。
reuseCoverageData	カバレッジ結果を再利用するかどうかを設定します。 True ：カバレッジ結果を再利用します。 False ：カバレッジ結果を再利用しません。
timer	タイマ機能を使用するかどうかを設定します。 True ：タイマ機能を使用します。 False ：タイマ機能を使用しません。

<table>
<tr><th>設定</th><th colspan="2">説明</th></tr>
<tr><td>trace</td><td colspan="2">トレース機能を使用するかどうかを設定します。【IECUBE】【IECUBE2】【シミュレータ】
True　：トレース機能を使用します。
False ：トレース機能を使用しません。</td></tr>
<tr><td>useTraceDataType</td><td colspan="2">トレース・データをどの機能で使用するかを設定します。【IECUBE【V850】】【IECUBE2】
指定可能な機能を以下に示します。</td></tr>
<tr><td></td><td>種類</td><td>説明</td></tr>
<tr><td></td><td>UseTraceDataType.RRM</td><td>RRM 機能</td></tr>
<tr><td></td><td>UseTraceDataType.Trace</td><td>トレース機能</td></tr>
<tr><td></td><td>UseTraceDataType.Coverage</td><td>カバレッジ機能</td></tr>
</table>

[参照]

設定されている値

注意　PM+ のワークスペースを CubeSuite+ のプロジェクトに変換した場合，メイン・プロジェクトにはデバッグ・ツールがありません。そのため，メイン・プロジェクトがアクティブ・プロジェクトである場合は，“None”を返します。

[詳細説明]

- デバッグ・ツールのオプションを設定／参照します。

[使用例]

```
>>>print debugger.Option.UseTraceData
Trace
>>>debugger.Option.UseTraceData = UseTraceDataType.Coverage
>>>print debugger.Option.Coverage
False
>>>debugger.Option.Coverage = True
>>>print debugger.Option.Coverage
True
>>>
```

debugger.XTrace.Addup/ debugger.XTrace.Complement/ debugger.XTrace.Mode

トレース・オプションを設定／参照します。【IECUBE】【IECUBE2】【シミュレータ】

[指定形式]

```
debugger.XTrace.Addup = addup【シミュレータ】
debugger.XTrace.Complement = complement【IECUBE【V850】】【IECUBE2【V850】】
debugger.XTrace.Mode = traceMode【シミュレータ】【IECUBE】【IECUBE2】
```

[設定]

設定	説明	
addup	タイム・タグを積算するかどうかを設定します。 True　：タイム・タグを積算します。 False：タイム・タグを積算しません。	
complement	トレースを補完するかどうかを設定します。 True　：トレースを補完します。 False：トレースを補完しません。	
traceMode	トレース制御モードを設定します。 指定可能なトレース制御モードを以下に示します。	
	種類	説明
	TraceMode.FullBreak	トレース・メモリを使い切ったら，プログラムの実行とトレース・データの書き込みを停止します。
	TraceMode.FullStop	トレース・メモリを使い切ったら，トレース・データの書き込みを停止します。
	TraceMode.NonStop	トレース・メモリを使い切っても，トレース・データの上書きを続けます。

[参照]

設定されている値

注意　PM+ のワークスペースを CubeSuite+ のプロジェクトに変換した場合，メイン・プロジェクトにはデバッグ・ツールがありません。そのため，メイン・プロジェクトがアクティブ・プロジェクトである場合は，“None”を返します。

[詳細説明]

- トレース・オプションを設定／参照します。

[使用例]

```
>>>print debugger.XTrace.Addup
False
>>>debugger.XTrace.Addup = True
>>>print debugger.XTrace.Addup
True
>>>
```

G. 3. 10　CubeSuite+ Python イベント

以下に，CubeSuite+ Python イベントの一覧を示します。

表 G—10　CubeSuite+ Python イベント

イベント名	機能概要
build.BuildCompleted	ビルド実行が完了したことを通知します。

build.BuildCompleted

ビルド実行が完了したことを通知します。

[ハンドラ形式]

```
build.BuildCompleted(sender, e)
```

[ハンドラ引数]

引数	説明
sender	ビルド・イベントの発行元が渡されます。
e	ビルド実行完了時のパラメータが渡されます。

[戻り値]

なし

[詳細説明]

- ビルド実行が完了したことを通知します。

[使用例]

```
>>>def buildCompleted(sender, e):
... print "Error = {0}".format(e.Error)
... print "BuildError = " + e.HasBuildError.ToString()
... print "BuildWarning = " + e.HasBuildWarning.ToString()
... print "BuildCancelled = " + e.Cancelled.ToString()
...
>>>build.BuildCompleted += buildCompleted        …イベントの接続
>>>build.All(True)
Error = None
BuildError = False
BuildWarning = False
BuildCancelled = False
True
>>>
>>>build.File("C:/sample/src/test1.c")
Error = None
BuildError = False
BuildWarning = False
```

```
BuildCancelled = False
True
>>>
>>>
>>>build.Clean()
Error = None
BuildError = False
BuildWarning = False
BuildCancelled = False
True
>>>
```

G. 4　Python コンソールの注意事項

(1) 日本語入力に関する注意事項

Python コンソールでは日本語入力機能を有効にすることができません。日本語を入力する場合は，外部テキスト・エディタ等で作成し，コピーして貼り付けてください。

(2) プロンプト表示に関する注意事項

Python コンソールのプロンプトが >>> であるところが >>>>>> というように複数表示される場合や，>>> の後に結果が表示され，キャレットの前に >>> がない場合があります。このような状態でも継続して関数を入力することが可能です。

(3) ロード・モジュールがないプロジェクトのスクリプト実行に関する注意事項

ロード・モジュール・ファイルがないプロジェクトを使用して起動オプションでスクリプト指定した場合，または*プロジェクト・ファイル名* .py をプロジェクト・ファイルと同じフォルダに置いている場合，通常プロジェクト読み込み後に自動的にスクリプトを実行しますが，ロード・モジュール・ファイルがない場合は実行しません。

(4) 強制終了に関する注意事項

無限ループしているようなスクリプトを実行中に以下の操作を行うと，強制的に関数の実行を終了させるため，関数の実行結果がエラーになる場合があります。

- Python コンソールのコンテキスト・メニューの［強制終了］や Ctrl ＋ D で強制終了
- 複数のプロジェクトを持つプロジェクトでアクティブ・プロジェクトを変更

付録 H 索 引

【C】
CubeSuite+ Python 関数 … 353
CubeSuite+ アップデート・マネージャ ウインドウ … 100
アップデート完了 ページ … 108
アップデートの選択 ページ … 102
エラー ページ … 112
更新履歴 ページ … 109
ダウンロードとインストール ページ … 104
ダウンロードのみ ページ … 106
非表示のアップデートの復元 ページ … 111
CubeSuite+ ライセンス・マネージャ ウインドウ … 119
CubeSuite のプロジェクト … 62

【H】
HEW のプロジェクト … 65

【P】
PM+ のプロジェクト … 73
Print Preview ウインドウ … 214
Python コンソール パネル … 311

【U】
Unicode 一般カテゴリ … 321

【あ行】
新しいツール ダイアログ … 273
アップデート完了 ページ … 108
アップデート実行中 ダイアログ … 114
アップデートの確認中 ダイアログ … 113
アップデートの選択 ページ … 102
アップデート・マネージャのオプション ダイアログ … 116
アトミック・ゼロ幅アサーション … 323
［一括検索］タブ … 200
［一括置換］タブ … 208
エディタ パネル … 180
エラー ページ … 112
お気に入りメニュー … 52, 53
オプション ダイアログ … 228
［全般 - Python コンソール］カテゴリ … 251
［全般 - アップデート］カテゴリ … 257
［全般 - 外部ツール］カテゴリ … 244
［全般 - 外部テキスト･エディタ］カテゴリ … 235
［全般 - 起動と終了］カテゴリ … 230
［全般 - テキスト・エディタ］カテゴリ … 253
［全般 - 表示］カテゴリ … 232
［全般 - ビルド／デバッグ］カテゴリ … 248
［全般 - フォントと色］カテゴリ … 239
［その他 - ユーザ情報］カテゴリ … 259

【か行】
外部テキスト・エディタの選択 ダイアログ … 309
キーボードのユーザ設定 ダイアログ … 276
既存のサブプロジェクトを追加 ダイアログ … 297
［基本機能］タブ … 263
［クイック検索］タブ … 196
［クイック置換］タブ … 204
クリーン … 346
グループ化構成体 … 323
検索・置換 ダイアログ … 194
［一括検索］タブ … 200
［一括置換］タブ … 208
［クイック検索］タブ … 196
［クイック置換］タブ … 204
更新履歴 ページ … 109
［コマンド］タブ … 271
コマンドの設定 … 334
コマンドの配置の変更 ダイアログ … 279

【さ行】
サブプロジェクト … 42
［サブプロジェクト］タブ … 173
指定行へのジャンプ ダイアログ … 193
出力 パネル … 181
詳細バージョン情報 ダイアログ … 284

処理中表示 ダイアログ … 224
スタート パネル … 134
正規表現 … 320
［全般 - Python コンソール］カテゴリ … 251
［全般 - アップデート］カテゴリ … 257
［全般 - 外部ツール］カテゴリ … 244
［全般 - 外部テキスト・エディタ］カテゴリ … 235
［全般 - 起動と終了］カテゴリ … 230
［全般 - テキスト・エディタ］カテゴリ … 253
［全般 - 表示］カテゴリ … 232
［全般 - ビルド／デバッグ］カテゴリ … 248
［全般 - フォントと色］カテゴリ … 239
ソース・コンバート設定 ダイアログ … 143
その他の分割パネル ... ダイアログ … 289
［その他 - ユーザ情報］カテゴリ … 259

【た行】
ダウンロードとインストール ページ … 104
ダウンロードのみ ページ … 106
タグ・ジャンプ … 182
タスクトレイ … 118
置換構成体 … 324
［追加機能］タブ … 265
［ツールバー］タブ … 269
ツールバーの名前の変更 ダイアログ … 275
テキスト編集 ダイアログ … 189
デバッグ専用プロジェクト … 325
統合アンインストーラ ウインドウ … 313

【な行】
名前を付けてプロジェクトを保存 ダイアログ … 302
名前を付けて保存 ダイアログ … 304

【は行】
バージョン情報 ダイアログ … 282
パック設定 ダイアログ … 225
バッチ・ビルド ダイアログ … 221
バッチ・ビルド … 349
非表示のアップデートの復元 ページ … 111
ビルド … 343
［ビルド・オプション］タブ … 177
ビルドの設定 … 334
ビルド・モード … 338
ビルド・モード設定 ダイアログ … 216
ファイル・エンコードの選択 ダイアログ … 191
ファイル追加 ダイアログ … 184
ファイルの保存設定 ダイアログ … 212
ファイルを開く ダイアログ … 294
フォルダとファイル追加 ダイアログ … 187
フォルダの参照 ダイアログ … 299
プラグインの管理 ダイアログ … 261
［基本機能］タブ … 263
［追加機能］タブ … 265
プログラムを選択 ダイアログ … 307
プロジェクト … 40
プロジェクト作成ダイアログ … 136
［プロジェクト］タブ … 171
プロジェクト・ツリー パネル … 152
プロジェクトの種類 … 46
プロジェクト・ファイルの保存 … 81
プロジェクト変換設定 ダイアログ … 146
プロジェクト名の変更 … 55
プロジェクトを開く ダイアログ … 291
プロパティ パネル … 165
［サブプロジェクト］タブ … 173
［ビルド・オプション］タブ … 177
［プロジェクト］タブ … 171
［マイクロコントローラ情報］タブ … 175

【ま行】
［マイクロコントローラ情報］タブ … 175
メイン・ウインドウ … 121
文字クラス … 320
文字のエスケープ … 320
文字列入力 ダイアログ … 219

【や行】
ユーザ設定 ダイアログ … 267
［コマンド］タブ … 271
［ツールバー］タブ … 269

【ら行】
ラピッド・ビルド … 348
リビルド … 345

量指定子　…　322

【わ行】

ワン・ポイント・アドバイス ダイアログ　…　287

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.03.01	―	初版発行

CubeSuite+ V1.02.00　ユーザーズマニュアル
起動編

発行年月日　2012年3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/inquiry

© 2012 Renesas Electronics Corporation. All rights reserved.
Colophon 1.0

CubeSuite+ V1.02.00

ルネサス エレクトロニクス株式会社

R20UT0975JJ0100