

この度は、統合開発環境 CS+をご使用いただきまして、誠にありがとうございます。

この添付資料では、本製品をお使いいただく上での制限事項および注意事項等を記載しております。

ご使用の前に、必ずお読みくださいますようお願い申し上げます。

第 1 章	対象デバイスについて	2
第 2 章	ユーザーズ・マニュアルについて	3
第 3 章	アンインストール時の選択キーワード	4
第 4 章	変更点	5
第 5 章	注意事項	10
第 6 章	制限事項	25

第1章 対象デバイスについて

統合開発環境 CS+がサポートする対象デバイスに関しては、WEB サイトに掲載しています。

こちらをご覧ください。

CS+製品ページ：

<http://japan.renesas.com/cs+>

第2章 ユーザーズ・マニュアルについて

本製品に対応したユーザーズ・マニュアルは、次のようになります。本文書と合わせてお読みください。

マニュアル名	資料番号
CS+ V3.00.00 インストーラ編	R20UT3094JJ0100
CS+ V3.02.00 エディタ編	R20UT3458JJ0100
CS+ V3.02.00 Python コンソール編	R20UT3457JJ0100
CS+ V3.00.00 アップデータ編	R20UT3098JJ0100
CS+ V3.02.00 メッセージ編	R20UT3459JJ0100
CS+ V3.02.00 プロジェクト操作編	R20UT3460JJ0100
CS+ V3.02.00 解析編	R20UT3461JJ0100
CS+ V3.02.00 RH850 デバッグ・ツール編	R20UT3456JJ0100
CS+ V3.02.00 RX デバッグ・ツール編	R20UT3455JJ0100
CS+ V3.02.00 RL78 デバッグ・ツール編	R20UT3454JJ0100

第3章 アンインストール時の選択キーワード

本製品をアンインストールする場合は、2つの方法があります。

- ・統合アンインストーラを使用する(CS+自体をアンインストールする)
- ・個別にアンインストールする(本製品のみをアンインストールする)

個別にアンインストールを行なう場合、コントロールパネルの

- ・「プログラムと機能」

から、「CS+ for CC」を選択してください。

第4章 変更点

本章では、CS+の V3.01.00 から V3.02.00 の変更点について説明します。

4.1 CS+の全体の機能追加

4.1.1 プロジェクト・ファイル監視機能の追加

CS+で開いているプロジェクト・ファイルとサブ・プロジェクト・ファイルを監視する機能を追加しました。本機能を有効にすると、ファイルが更新されていた場合にはメッセージを出力し、プロジェクト・ファイル、サブ・プロジェクト・ファイルを再度読み込むかの指定が可能となります。これにより、同時に開いているプロジェクト・ファイルが保存されることがわかり、その内容を反映することが可能です。

4.1.2 プロジェクト・ファイル名変更機能の変更

プロジェクト・ツリーパネルのプロジェクト・ファイル、サブ・プロジェクト・ファイルに対して、名前を変更する機能を使用した場合の、ファイル名を変更する処理を変更しました。

（変更前）プロジェクト情報を保存するタイミングでファイルを複製する

（変更後）名前を変更したタイミングでファイル名を変更する

4.1.3 メッセージ ダイアログ機能の追加

メッセージ ダイアログの [画像を記録] ボタンを選択した場合に、それまでの操作のログを記録したファイルを保存するようにしました。

これにより、弊社でメッセージが出力するまでの間の処理を確認することができ、解決までの時間を短くできます。

4.1.4 スプラッシュウインドウ機能の追加

スプラッシュウインドウに、ワンポイントアドバイスを表示するようにしました。

これにより、起動のたびにワンポイントアドバイスダイアログを閉じる操作なく、ワンポイントアドバイスを参照することが可能となり、CS+の操作方法の習得時間の短縮できます。

4.1.5 オプション ダイアログ（ビルド）機能の機能追加

依存プロジェクトにビルド・エラーがある場合に、ビルドをスキップする機能を追加しました。

これにより、ビルド・エラーの修正を開始するまでの時間を短くできます。

4.1.6 全画面表示の機能追加

CS+を全画面表示する機能を追加しました。

これにより、メイン・ウインドウのタイトルバーを表示せず、ウインドウ状態を最大化できます。

4.1.7 CS+使用上の注意事項の解除

次の注意事項を解除しました。

- ・デバッグ・ツールへプログラムをダウンロードした場合、またはダウンロード後にエディタ・パネルを開く場合にCS+が強制終了してしまう場合があります。
- ・プロジェクトに下記のいずれかの条件に該当するソースコードが存在する場合に発生します。
 - (a) 関数内に{}で囲まれたブロックが 128 以上ネストしている。
 - (b) else if の連続が 64 以上続いている。
 - (c) "else if の連続数"を 2 倍した数と,"else if が連続している箇所内で{}で囲まれたブロックのネスト数"を足した合計値が 128 以上である。

なお、上記(b)(c)は RX ファミリー用 C/C++コンパイラパッケージ (統合開発環境つき)で C99 オプションを指定している場合にのみ発生します。

4.1.8 サポートマイクロコントローラ追加

本バージョンより、RH850 ファミリーの「RH850/D1x シリーズ」および「RH850/P1x-C シリーズ」のサポートを追加しました。

4.2 エディタ・パネルの機能追加

4.2.1 エディタ・パネルのフローティング表示機能の追加

エディタ・パネルをフローティングにする機能を追加しました。

これにより、CS+のメイン・ウインドウの外にもエディタ・パネルを表示させることが可能となり、よりお客様好みのウインドウ、パネルの配置の実現が可能です。

4.2.2 インクリメンタル検索機能の追加

エディタ・パネルに、インクリメンタル検索機能を追加しました。

これにより、文字列検索がしやすくなり、ソース・ファイルの編集やデバッグ操作時の効率向上に貢献します。

4.2.3 エディタ・パネル プラグインの扱いの改善

CS+をコマンドラインで操作する際の、/np プラグイン名 オプション、/nall オプションにて、エディタ・パネル プラグインを無効にしないように変更しました。

これにより、誤ってエディタ・パネル プラグインを無効にしてしまうことを防げます。

4.3 ビルド・ツールの機能追加

4.3.1 ビルド・モードのプロパティ設定機能の追加

ビルド・ツールのプロパティに値を設定する際、設定した値をすべてのビルド・モードに反映する機能を追加しました。

これにより、すべてのビルド・モードに対して設定を一括で指定することが可能です。

4.4 メモリ・マップ・プロファイラの機能追加

4.4.1 最適なビルド・オプションに変更する機能の追加

メモリ・マッピング情報を収集する際に、プロファイルに適したビルド・オプションに一発で変更する機能を追加しました。

4.4.2 エディタ・パネルへのジャンプ機能の追加

詳細メモリ・マッピングエリアの項目を選択している場合に、エディタ・パネル、または外部エディタにジャンプする機能を追加しました。

これにより、メモリ・マッピングを確認してソース・ファイルを参照、編集するまでの時間を短くすることが可能です。

4.4.3 メモリ・マッピングの名前変更機能の追加

メモリ・マッピング一覧に表示されている、保存したメモリ・マッピング（表示の2番目以降）の名前を変更する機能を追加しました。

これにより、複数の状態を保存する際に、ユニークな名前を付けることが可能となり、どのような状態の保存結果化であるかがわかりやすくすることが可能です。

4.4.4 ファイル保存機能の拡張

詳細メモリ・マッピングエリアをファイル保存する際に、1行目にラベル名、または、メモリ・マッピング一覧の選択項目名を出力するようにしました。

これにより、保存したファイルを参照した際に、何の情報を保存したかの判別が可能となります。

4.5 解析ツールの機能追加

4.5.1 動的解析を有効にする機能の拡張

動的解析を有効にする機能を使用すると、エミュレータ／シミュレータが持つデバッグ機能に対し、動的解析に必要な標準的な設定を行うようにしました。

これにより、動的解析の設定操作を必要最低限にすることが可能です。

4.5.2 関数一覧パネルの関数属性の拡張【RL78】

RL78 ファミリ用コンパイラ CC-RL の callt 関数に対応しました。callt 関数として定義された場合には、関数一覧パネルの属性に callt と表示されます。

これにより、callt 関数に配置された関数がわかるので、実行回数情報なども合わせて考慮することにより、効率的に callt 関数を指定しているか確認でき、callt 指定の見直しをすることが可能です。

4.5.3 関数一覧パネルの時間表示機能の追加【RH850】

トレース・データの時間情報を解析する機能により、関数一覧パネルの実行時間、実行時間（割合）、平均実行時間を表示する機能を追加しました。

これにより、プログラムのチューニングを検討する際の情報を得ることが可能です。

4.5.4 クロック数表示機能の追加【RH850】

トレース・データの時間情報を解析する機能により、時間をクロック数で表示する機能を追加しました。
これにより、時間情報を状況に応じた表示（時間秒、またはクロック数）に切り替えて、読みやすくすることが可能です。

4.6 Python コンソールの機能追加

4.6.1 Python 関数の追加

以下の Python 関数を追加しました。

関数名	機能概要
build.Stop	実行中のビルドを中止します。
debugger.DebugTool.RestoreState	デバッグ・ツールの状態を、保存したファイルの内容に復帰します。
debugger.DebugTool.SaveState	デバッグ・ツールの状態をファイルに保存します。

4.7 I/O ヘッダ生成ツールの機能追加

4.7.1 I/O ヘッダ・ファイルをカスタマイズする機能の追加【RH850】

I/O ヘッダ・ファイルに出力するモジュールを選択する機能と、I/O ヘッダ・ファイルを分割する機能を追加しました。必要なモジュールのみを選択することにより、I/O ヘッダ・ファイルのサイズを削減することが可能です。また、I/O ヘッダ・ファイルを分割することにより、1 ファイルのコンパイル時間を短縮することが可能です。

4.7.2 MISRA-C:2004 ルールに適合した I/O ヘッダ・ファイルを出力する機能の追加【RH850】【RL78】

以下の MISRA-C:2004 ルールに適合した I/O ヘッダ・ファイルを出力する機能を追加しました。

- Rule 6.3: typedefs that indicate size and signedness should be used in place of the basic types.
- Rule 6.4: Bit fields shall only be defined to be of type unsigned int or signed int. (int 型のサイズの IOR のみ)
- Rule 10.6: A "U" suffix shall be applied to all constants of unsigned type.
- Rule 18.4: Unions shall not be used.

なお、この機能を使用する場合は共用体を出力しないため、同一アドレスに複数の IOR が配置されている場合、アクセス・サイズが最大の IOR のみを出力します。

本機能の ON/OFF に関わらず、I/O ヘッダ・ファイルの前半部にビットフィールドの定義を出力します。この定義は MISRA-C:2004 ルールの指摘の対象になります。

4.8 デバッグ・ツールの機能追加

4.8.1 ウォッチパネルの保存機能の追加【RX】【RH850】【RL78】

ウォッチパネルにウォッチ・データを展開して保存する機能を追加しました。

この機能により、配列、ポインタ型変数、構造体／共用体、レジスタ（部分名がついているもののみ）などのウォッチ式の展開表示が可能な場合、展開表示していなくても展開状態で保存をすることが可能です。

第5章 注意事項

本章では、注意事項について説明します。

5.1 CS+全体の注意事項

5.1.1 ファイル名に関する注意事項

フォルダ名、ファイル名に関しては次の注意事項があります。

- ・フォルダ名、ファイル名

Windows のエクスプローラーで作成することのできないフォルダ名とファイル名は、使用しないでください。

- ・ソース・ファイル名とロード・モジュール・ファイル名とプロジェクト・ファイル名

ファイル名は、a-z, A-Z, 0-9, .(ピリオド), _(アンダスコア), +, - のいずれかの文字で構成されます。

ファイル名の先頭と最後に,.(ピリオド)の文字は使えません。

ファイル名の先頭に「+」(プラス)／「-」(マイナス)は使えません。

英大文字(A-Z), 英小文字(a-z)は区別されません。

ファイル名は、パスを含めて最大 259 文字です。

ファイル名が同じソース・ファイルは使用しないで下さい。異なるパスに存在していても区別できません。

- ・上記以外のファイル名

Windows のファイル名規約に準拠します。

なお、ファイル名には次の文字は使えません。

¥ / : * ? " < > | ;

ファイル名の先頭と最後に,.(ピリオド) とスペースは使えません。

英大文字(A-Z), 英小文字(a-z)は区別されません。

ファイル名は、パスを含めて最大 259 文字です。

- ・フォルダ名

Windows のファイル名規約に準拠します。

なお、ファイル名には次の文字は使えません。

() , =

5.1.2 パネル表示に関する注意事項

使用するハードウェア環境が CS+ の推奨サポート環境を下回るスペックである場合、[プロパティ] パネルのサイズを小さくすると表示内容が乱れることがあります。

その場合には、分割パネル領域から [プロパティ] パネルを外に出してください。

- ・ドッキング可能を ON にして、ドッキング・パネル化する

- ・フローティングを ON にして、フローティング・パネル化する

5.1.3 ユーザーアカウント制御(UAC)機能に関する注意事項

Windows Vista / Windows 7においてUAC 機能を無効にした場合、管理者権限をもたないユーザでプロジェクトを作成や開いた場合で、かつ、デバイス依存情報をインストールしていない場合、デバイス依存情報のインストールが開始されますがインストールに失敗します。UAC 機能を無効にする場合は、管理者権限でログインしてプロジェクトを作成してください。

5.1.4 Windows の更新プログラムに関する注意事項

マイクロソフト株式会社より公開された、Windows 用の更新プログラム（KB2393802）を適用している場合、パソコンがブルースクリーンになる障害に該当することがあります。この障害に対しては、パソコン等の各メーカーより提供される修正プログラムを適用してください。

5.1.5 弊社製リアルタイム OS に関する注意事項

弊社製の RX ファミリー用のリアルタイム OS を使用する場合には、CS+のインストール・フォルダを括弧がないフォルダに変更してインストールしてください。64bit 版の Windows にインストールする場合には、¥Program Files (x86) がデフォルトのインストール・フォルダになり、フォルダ名に括弧がある場合エラーになります。

5.1.6 マイクロコントローラ変更に関する注意事項

マイクロコントローラを変更する場合には、次の注意事項があります。

- ・ 同じファミリー（RH850, RX, RL78）内の、同じビルド・ツールに対応しているマイクロコントローラへのみ変更が可能です。
- ・ マイクロコントローラを変更する際は、デバッグ・ツールを接続していない状態にしてください。
- ・ マイクロコントローラを変更する前に、プロジェクトを保存する必要があります。
- ・ 端子配置（設計ツール）、コード生成（設計ツール）、デバッグ・ツール（ウォッチ登録情報除く）の情報は、マイクロコントローラの変更後、引き継がれません。

5.1.7 プラグイン管理機能に関する注意事項

プラグインの管理ダイアログの基本機能タブにおいて、開発対象となるマイクロコントローラ用プラグインのチェックは、外さないことを推奨します。

開発対象ではないマイクロコントローラ用のビルド・ツール・プラグイン、デバッグ・ツール・プラグインのチェックを外してください。たとえば、ビルド・ツール・プラグインのみチェックを外すとデバッグ・ツールでダウンロードするファイルが見つからずエラーとなります。

5.1.8 エディタ・パネルに関する注意事項

- ・ ページ設定ダイアログが使用できません。
- ・ 印刷プレビューのツールバーにコピーボタンがありますが、使用できません。
- ・ 変数、ラベルを選択して、コンテキスト・メニューの「関数へジャンプ」機能を使用した場合、変数、ラベルにジャンプしません。
- ・ 関数へジャンプ機能で、別ファイルに定義されている static 関数には移動できません。
- ・ メイン・プロジェクトとサブプロジェクトに、パスの違う同名のソース・ファイルが登録されていて、

メイン・プロジェクトとサブプロジェクトのロード・モジュールを両方ダウンロードしたとき、次のようになります。

- 当該ファイルでは、メイン・プロジェクトのアドレスが表示される
 - 当該ファイルの逆アセンブルから「ソースヘジャンプ」を行うと、メイン・プロジェクトに登録されているファイルが開く
 - どちらのプロジェクトから当該ファイルを開いても1つのファイルしか開けない
 - ・ 無名の構造体ではスマート・エディットが正しく動作しません。
 - ・ 関数の引数に関数呼び出しが含まれる場合、ツールチップに誤った情報が表示されます。
 - ・ クラスの配列やクラスのパインタの配列に対して、メンバ変数、メンバ関数の補完が正しく動作しません。
 - ・ メンバ名を途中まで入力して、ctrl+"を入力しても補完機能が正しく動作しません。
 - ・ アウトライン（折り畳み/展開）は、プロジェクトに登録したファイルのみが対象です。そのため、プロジェクトに登録していないファイルをCS+のエディタで表示しても、アウトライン表示にはなりません。
 - ・ Windows 8, windows8.1 では、アンチエイリアスが有効になり表示が不鮮明になる場合があります。
 - ・ 混合表示モードにて、行番号を指定してジャンプすると、逆アセンブルのコードが挿入されて表示されるため、指定行が画面上に表示されない場合があります。
 - ・ 構造体がネストしている場合に、3段目以降は、スマート・エディットの機能は使用できません。また、ツールチップの情報も表示されません。
 - ・ 「#ifdef - #endif」の直後のコードでは、スマート・エディットでメンバの候補が表示されません。また、ツールチップが表示されません。
 - ・ 「#ifdef - #endif」中の最初の変数については、「#endif」以降のコードで、スマート・エディットでメンバの候補が表示されません。また、ツールチップが表示されません。
 - ・ プロジェクトに含まれないファイルを開き、ブックマークを設定し、プロジェクトを閉じます。そして、そのファイルのブックマークの設定を変更した後、プロジェクトを開きなおします。ブックマークダイアログを開いたとき、ブックマークダイアログには、プロジェクトを閉じたときのブックマークが表示され、ソース・ウインドウでは、プロジェクトを閉じた後のブックマークが表示され、差異が生じます。このような場合には、一旦、ファイルを閉じ、再度開いてください。ブックマークダイアログに表示されていたブックマークがソース・ウインドウに表示されます。
 - ・ 矩形選択（Alt キーを押下してマウスによる範囲選択）を行い、最終行以降に行が追加されて貼り付けされる場合、貼り付け位置によらず行の先頭から貼り付けられます。貼り付け後に必要なスペースを挿入してください。
 - ・ 名前を付けて保存ダイアログにおいてファイルを保存する場合、拡張子を入力しない場合には、ファイルの種類ドロップダウン・リストで選択されている最初の拡張子が自動的に付加されます。ただし、ファイルの種類ドロップダウン・リストで選択されている拡張子および Windows で登録されている拡張子を付加してファイル名を入力した場合、拡張子は付加されません。
- 自動的に拡張子が付加されてしまった場合は、エクスプローラー等でファイルをリネームしてください。

5.1.9 PM+から CS+プロジェクトへの移行に関する注意事項

PM+ V6.00/V6.10/V6.11 で作成した CA850 のプロジェクトに対して、ビルド・モードを新規追加した場合、そのプロジェクトを CS+ で読み込むと以下ようになります。

1) Debug Build または Release Build が選択されている場合：

新規追加したビルド・モードの情報が変換されません。

2) 新規追加したビルド・モードが選択されている場合：

エラーとなります。

回避策として、PM+ V6.20 以上でプロジェクトを開いて保存し、保存後のプロジェクトを CS+ で読み込んで下さい。

5.1.10 プロジェクト流用時のデバッグ・ツールの設定に関する注意事項

プロジェクトを流用作成する時、作成するプロジェクトにてデフォルトで選択されているデバッグ・ツールに対してのみ、流用した設定を反映します。

ただし、RX ファミリーについては、内部処理がエミュレータ、シミュレータで共通となっている為、デバッグ・ツールの選択状態に関わらず流用した設定を反映します。

5.1.11 オンライン・ヘルプに関する注意事項

オンライン・ヘルプにおいて、検索タブ(S)を表示した状態で閉じ、再度オンライン・ヘルプを表示し、目次(C)タブを表示した場合、コーディング編、ビルド編、コンパイラ編、ビルド・ツール操作編が表示されない場合があります。

このようになった場合には、目次(C)タブを表示したままオンライン・ヘルプを閉じてから、再度オンライン・ヘルプを表示しなおしてください。

5.1.12 プロジェクト変換時の注意事項

High-performance Embedded Workshop / PM+ / 旧 CubeSuite を開いた時の〔プロジェクト変換設定〕ダイアログで、プロジェクトの変換先デバイスを切り替えた時、〔プロジェクトの種類〕で選択されていた値を初期値であるコンボボックスの先頭の値へ戻ります。

例えば、デバイスを選択し直すとプロジェクトの種類が先頭の（例えば〔アプリケーション〕）に切り替わります。

5.1.13 High-performance Embedded Workshop プロジェクト変換時の注意事項

High-performance Embedded Workshop のプロジェクトを CS+環境で読み込んだ場合、プロジェクト変換ができずエラーとなったり、ビルド実行時にエラーが発生する場合があります。

(1) CS+用のプロジェクトへ変換ができない

- ・ ルネサス エレクトロニクス社製ツールチェーンが使用されていないプロジェクト
- ・ High-performance Embedded Workshop 環境の設定ファイル (tps ファイル) が存在していないプロジェクト (tps ファイルは、High-performance Embedded Workshop 環境で一度開くと自動生成されます。)
プロジェクト変換前に一度プロジェクトを High-performance Embedded Workshop 環境で開くことで解決できます
- ・ ルネサス エレクトロニクス社製リアルタイム OS の設定ファイル (CFG ファイル) が複数存在しているプロジェクト

(2) CS+用のプロジェクトへ変換はできるが、ビルド実行でエラーが発生

- ・ プレースホルダ (\$(TCINSTALL)) を使用しているプロジェクト
\$(TCINSTALL)は、変換後のプロジェクトにそのまま残ります。

CS+は、\$(TCINSTALL)を解釈できません。オプションのパラメータに\$(TCINSTALL)を使用していた場合は、そのままオプションに渡されますので意図したビルド結果を得られない可能性があります。(ビルドでエラーが発生するなど)

\$(TCINSTALL)をプロジェクト変換後に、お客様自身で変更してください。

- ・ プレースホルダ (\$(WORKSPDIR)) を使用しているプロジェクト
プロジェクト・ファイル (拡張子 hwp) を指定して変換した場合、「%ProjectDir%¥..」 (プロジェクト・フォルダの1つ上のフォルダ) に固定で変換します。

プロジェクト・フォルダの1つ上のフォルダにワークスペースがない場合は、正しいフォルダを示さなくなりますので、ビルドでエラーが発生することがあります。

その場合、プロジェクト変換後に「%ProjectDir%¥..」を、お客様自身で変更してください。

- ・ カスタム・ビルド・フェーズを使用しているプロジェクト
カスタム・ビルド・フェーズは、削除されます。

カスタム・ビルド・フェーズは、ビルド時に実行されなくなります。

よって、カスタム・ビルド・フェーズで生成されたファイル出力を使用している場合はビルド・エラーとなる可能性があります。

プロジェクト変換後に、カスタム・ビルド・フェーズのコマンドを、各フェーズの前後実行コマンドに必要なに応じて登録してください。

- ・ カスタムプレースホルダを使用しているプロジェクト
カスタムプレースホルダは変換しません。

CS+は、カスタムプレースホルダを解釈できません。オプションのパラメータにカスタムプレースホルダを使用していた場合は、そのままオプションに渡されますので意図したビルド結果を得られない可能性があります。（ビルドでエラーが発生するなど）

プロジェクト変換後に、カスタムプレースホルダを、お客様自身で変更してください。

(3) その他

- (a)\$ (FILEDIR)は、 %FileDir%へ変換します。

変換後、パス編集ダイアログでパス編集するとき、 %FileDir%により以下のエラーが発生します。

指定したパスに存在しないフォルダが含まれています。(W0205012)

プロパティを編集する場合、 %FileDir%を別のプレースホルダまたはディレクトリに置き換えてください。

- (b)\$ (WINDIR)は、 %WinDir%へ変換します。

- (c)フォルダの表示順が異なる場合があります。

- (d)High-performance Embedded Workshop のプロジェクトでダウンロード・ファイルを指定している場合、
変換後は各デバッグ・ツールのダウンロード・ファイル一覧の 2 番目以降に登録します。

- (e)コンパイル・オプション-output=src は、 -output=obj（デフォルト）へ変換します。

- (f)ライブラリ・プロジェクトを変換時、ライブラリが標準ライブラリをリンクしていた場合は、そのリンク
設定を破棄します。（変換ログに出力されます）

- (g)ライブラリジェネレータで「既存標準ライブラリファイル指定」を指定していた場合、「標準ライブラリ・
ファイル指定なし」に変更します。結果、指定していたライブラリがリンクされません。（変換ログに出
力されます）

- (h)High-performance Embedded Workshop の「全般」タブのオプションは変換せずに破棄します。

- (i)リンカでサブコマンドファイルを指定していた場合、変換後「サブコマンドファイルを使用する」設定を
破棄してリンカのオプション設定をデフォルトにします。

- (j)リンカの-library, -input, -binary オプションで指定したファイルは、リンク順設定ダイアログのファイル
リストに表示しません。リンク順の指定対象外になります。

- (k)RTOS のコンフィグレーションファイルは、変換後に「Configuration file」カテゴリ・ノードの下に表示
しません。

- (l)RTOS オプションは変換せずに破棄します。オプション設定はデフォルトになります。

- (m)RTOS プロジェクトのビルド・モードは、プロジェクト変換後「DefaultBuild」になります。

変換後、ビルド・モードを変更してください。

- (n)RTOS プロジェクトのアセンブラ出力ファイル（ritbl.obj）のリンク順がプロジェクト変換後
High-performance Embedded Workshop と異なります。

5.1.14 新規にプロジェクトを作成する際の注意事項

【対象】RX

RX 開発環境にて、” 空のアプリケーション(CC-RX)” のプロジェクトを作成してビルドした場合、以下のエラーが出る場合があります。

** L2132 (E) Cannot find "D" specified in option "rom"

** L2132 (E) Cannot find "D_1" specified in option "rom"

** L2132 (E) Cannot find "D_2" specified in option "rom"

エラーが発生した場合は、リンク・オプションの” ROM から RAM へマップするセクション” の設定を変更してください。

5.1.15 マイクロソフト株式会社 IME に関する注意事項

マイクロソフト株式会社製の Office 2010 付属の Microsoft Office IME 2010 を使用している場合に、CS+使用時に、E2000006 エラーが出力される場合があります。

Microsoft Office IME 2010 に起因する可能性がありますので、Windows 標準の IME に戻すか、マイクロソフト株式会社より提供されている Microsoft Office IME 2010 の KB2687611 を解決するためのモジュールをインストールしてください。

5.1.16 チュートリアルの注意事項

チュートリアルでは、コード生成プラグイン、端子配置プラグイン、プログラム解析プラグインを使用します。プラグイン管理ダイアログで使用するプラグインを有効にしてください。

5.1.17 CS+の複数起動の注意事項

CS+は、同じホストマシン上で複数起動が可能ですが、次の注意事項があります。

- ・ CS+を複数起動した場合、パソコンのユーザ毎の情報ファイルは最後に書き込んだ情報が保存されます。
- ・ CS+を複数起動した場合、スタック見積もりツール（CallWalker 含む）の情報ファイルは最後に書き込んだ情報が保存されます。
- ・ 複数起動した CS+で、同一プロジェクト・ファイルを使用した場合、最後に書き込んだ情報が保存されます。
- ・ 複数起動した CS+で、同一プロジェクト・ファイルを使用した場合、同時にビルドしないでください。出力ファイルが同一のためです。

5.2 設計ツールの注意事項

5.2.1 パッケージの変更に関する注意事項

端子配置のプロパティでパッケージ名を変更した場合、端子配置図および端子配置表の入力データはクリアされます。

5.2.2 プロジェクト保存に関する注意事項

サブプロジェクトが存在するプロジェクトにて、端子配置図または端子配置表パネルが開いた状態でプロジェクトの保存を行った場合に、プロジェクト・ツリー上の最後のサブプロジェクトの端子配置図、端子配置表が必ず表示されます。

5.3 デバッグ・ツールの注意事項

文中において以下の略称を使用しています。

OCD(シリアル) : E1 エミュレータ(シリアル), E20 エミュレータ(シリアル)

OCD(JTAG) : E1 エミュレータ(JTAG), E20 エミュレータ(JTAG)

5.3.1 サブプロジェクトの追加に関する注意事項

【対 象】 全デバッグ・ツール, 全デバイス共通

メイン・プロジェクトと異なるデバイスを扱うサブプロジェクトを追加する場合、デバッグ・ツールを切断してから行ってください。

5.3.2 低消費電力モードに関する注意事項

【対 象】 全デバッグ・ツール, RX

スリープモード、ストップモードおよびスタンバイモードなどの低消費電力モード中に強制ブレークを行った場合や、ステップ実行で低消費電力モードに移行する命令を実行した場合、シミュレータとエミュレータでは以下のような動作の差があります。

- ・エミュレータ：強制ブレークにより低消費電力モードは解除されます。また、ステップ実行では低消費電力モードに移行します。
- ・シミュレータ：レジスタなどによる低消費電力モードへの移行はサポートしていません。WAIT 命令実行時にはブレークし、PC は次の命令のアドレスとなります。また、ステップ実行では低消費電力モードに移行せず、PC は次の命令のアドレスとなります。

5.3.3 任意区間のトレースに関する注意事項

【対 象】 シミュレータ, 全デバイス共通

トレース開始イベントからトレース終了イベントまでをトレースする場合、シミュレータではトレース終了イベントがトレース結果として表示されません。このため、シミュレータを使用する場合はトレース終了イベントをトレース・データとして表示させる範囲の 1 行下に設定してください。

5.3.4 サブプロジェクトの追加について

【対 象】全デバッグ・ツール, 全デバイス

デバッグ・ツール接続中にサブプロジェクトを追加すると、ダウンロード等に失敗することがあります。サブプロジェクトの追加は、デバッグ・ツール切断中にしてください。

5.3.5 ブレークポイントの設定等が不正になる注意

【対 象】全デバッグ・ツール, 全デバイス

関数名や変数名を、先頭のアンダー・バーの有無などで使い分けている場合、デバッガが誤認識してしまい、シンボル変換や、ブレークポイントの設定が不正になる場合があります。

例えば `_reset` と `__reset` という 2 つの関数が存在していた場合などが該当します。

5.3.6 同名の変数の取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

異なるソース・ファイルに無名名前空間を記述し、その中に同名の変数を定義した場合、ウォッチパネルでは、最初に見つかる変数の情報を表示します。

5.3.7 メンバ変数ポインタの取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

下記のプログラムに定義されたメンバ変数ポインタ `mp1` をウォッチパネルおよびローカル変数パネルに登録した場合、型名に `"int Foo::*"` ではなく `"int *"` と表示されます。

```
class Foo {  
    int m1;  
};  
  
int Foo::*mp1 = &Foo::m1;
```

5.3.8 レジスタ割付された共用体の取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

共用体がレジスタに割り付いている場合、共用体のメンバはレジスタの下位バイトから割り付いているとみなします。このため、ビッグエンディアンの場合はメンバの値を正しく表示できません。

5.3.9 char 型の引数を持つ同名の関数の取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

下記のように `char` 型を使用した 3 つの関数を定義した場合、`"Func(signed char)"` のアドレスを正しく表示できません。（`"Func(char)"` のアドレスを表示します。）

```
void Func(char);  
void Func(signed char);  
void Func(unsigned char);
```

5.3.10 char 型の一次元配列の取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

下記のような char 型の一次元配列がレジスタやメモリの複数個所に割り付いていた場合は、ウォッチパネルおよびローカル変数パネルに配列"array"を登録しても値のカラムに文字列を表示できません。

```
char array[5] = "ABCD";
```

5.3.11 オーバーレイ・セクションの優先セクションの変更に関する注意事項

【対 象】全デバッグ・ツール, RX

オーバーレイ・セクションの優先セクションを変更しても、デバッガの機能には直ぐには反映されません。

例えば、エディタ上のアドレス表示については、ファイルを一旦閉じ、再度開くことにより反映されます。

また、ウォッチパネル上の変数表示については、1 回ステップを実行することにより反映されます。

5.3.12 レジスタ割付された変数の取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

ローカル変数パネルの[スコープ]にて"カレント"以外を選択中は、レジスタに割り付いた変数の値は正しく表示できません。また、その変数の値を編集することも出来ません。

5.3.13 変数の割り付き位置表示の取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

以下の条件を全て満たす変数を定義した場合、ウォッチパネル、ローカル変数パネルでは、対象のメンバ変数の割り付き位置文字列が変数全体の割り付き位置文字列で表示されます。

<条件>

- (1) 定義した変数が複数のアドレスやレジスタに割り付いている。
(アドレスカラムに 2 つ以上のアドレスやレジスタ名が表示される場合)
- (2) 変数に以下の型のメンバが定義されている。
 - 構造体, クラス, 配列, 共用体のいずれか

<例>

```
struct Mem {  
    long m_base;  
};  
  
struct Sample {  
    long m_a;  
    struct Mem m_b;  <-条件(2)に該当  
};  
  
main () {  
    struct Sample obj;  
}
```

表示結果 :

"obj"	-	{ R1:REG, R2:REG }	(struct Sample)
L m_a	0x00000000	{ R1:REG }	(long)
L m_b	-	{ R1:REG, R2:REG }	(struct Base)
L m_base	0x00000000	{ R2:REG }	(long)

5.3.14 変数をキャストする際の取り扱いに関する注意事項

【対 象】全デバッグ・ツール, RX

ウォッチパネルで変数を他の型へキャストした場合, C スタイルのキャストを行いません。

このため, 例えば仮想継承クラスの基底クラスへのキャスト結果は, プログラム中で記述したキャスト結果と異なります。

```
class AAA [
    int m_aaa;
} objA;
class BBB : public AAA { //BBB は AAA を継承している
    int m_bbb;
} objB;
class CCC { //CCC は AAA を継承していない
    int m_ccc;
} objC
```

```
class AAA* pa = objA;
class BBB* pb = objB;
class CCC* pc = objC;
```

"(AAA*)pa" . . . 使用可能

"(BBB*)pb" . . . 使用可能

"(AAA*)pb" . . . 使用可能

"(CCC*)pc" . . . 使用可能

"(AAA*)pc" . . . pc の指すアドレスを型"AAA"の先頭アドレスと見做してキャストします。

プログラミング上のキャストイメージ : (AAA*)((void*)pc)

5.3.15 PC スリープ状態からの復帰に関する注意事項

【対 象】OCD(JTAG), OCD(シリアル), RX

Windows Vista または Windows 7 でデバッグ中に PC がスリープ状態または休止状態に移行した場合, 復帰後にデバッグを継続できません。

Windows Vista または Windows 7 で使用する場合は, PC がスリープ状態および休止状態に移行しない設定でご使用ください。

5.3.16 プログラム実行中のトレース停止、再開の注意事項

【対 象】全デバッグ・ツール、RX

トレース開始イベント、あるいはトレース終了イベントを設定している場合、プログラム実行中のトレース停止・再開はできません。

5.3.17 トレースのタイムスタンプについての注意事項

【対 象】OCD(JTAG)、OCD(シリアル)、RX

トレース情報に付加されるタイムスタンプは、フレーム間の経過時間がトレースクロックの 20 ビット分を超える場合、および、トレース出力でロストが発生した場合、正しい時間となりません。

5.3.18 CC-RX コンパイラのリンク・オプションの注意事項

【対 象】全デバッグ・ツール、RX

CC-RX コンパイラはリンク・オプション-sdebug には対応していません。

デバッグの際は、CC-RX コンパイラプロパティの[リンク・オプション]タブ→[出力カテゴリ]→[デバッグ情報を出力する]を-debug に設定してください。

5.3.19 リターンアウト実行時の注意事項

【対象】全デバッグ・ツール、RX

再帰呼び出しの関数でリターンアウト実行した場合、呼び出し元ではなく呼び出し先関数のリターン命令を実行したところで停止する場合があります。

5.3.20 スタートアッププログラム保護機能の注意事項

【対象】OCD(シリアル)、RX100

ユーザ・プログラム実行中に下記いずれかを実施してブレーク後、「CPUのリセット」をすると、デバッガが表示する ROM 内容と MCU の ROM 内容が一致しません。

この場合、再度ユーザ・プログラム実行して停止すると一致します。

- ・ R_FCL_ChangeSwapState 関数をコールして即座にスワップする
- ・ フラッシュ初期設定レジスタ(FISR) を操作して即座にスワップする

5.3.21 カバレッジ計測機能に関する注意事項

【対 象】E20 エミュレータ(JTAG)、RX64M

(1) ホットプラグイン接続ではカバレッジ計測できません。

ホットプラグイン接続する場合は、[デバッグ・ツール設定]タブ上の[カバレッジ]カテゴリで、[コード・カバレッジ機能を使用する]を[いいえ]に設定してください。

(2) プログラム実行中に「システムリセットを発行しました。」というエラーメッセージが表示された場合、プログラム実行開始からシステムリセット発生時点までの実行はカバレッジ測定されません。

5.3.22 for 文やインライン関数内のブレークポイント設定に関する注意事項

【対 象】全デバッグ・ツール、RX

以下のプログラムを C ソースに記述するとソースプログラム 1 行に対する命令が複数箇所に配置されますがエディタ上にはそのうちの 1 箇所のアドレスのみを表示しています。

この行に対しブレークポイントを設定した場合、ブレークするのはエディタに表示したアドレス実行時のみとなります。

1. インライン関数 (*1)
2. テンプレート関数
3. for 文、do-while 文の先頭行

*1: 最適化によりインライン展開された関数も含まれます

5.3.23 DMAC/DTC のトレース設定に関する注意事項

【対 象】 E20 エミュレータ(JTAG)、RX64M

デバッグ・ツールプロパティの[デバッグ・ツール設定]タブ内の[トレース]カテゴリの[データアクセスのバス・マスタ]を[DTAC/DTC]に設定した場合、[トレース]カテゴリの[外部トレース出力]に[トレース出力優先]を設定するとトレースが正常に動作しない可能性があります。

[CPU 実行優先]または[外部出力しない]を設定してください。

5.3.24 PLL 動作時のメイン・クロック・ソース設定に関する注意事項

【対 象】 OCD(JTAG), OCD(シリアル), RX64M

ユーザ・プログラムで PLL を使用する場合、以下の設定を行ってください。

・デバッグ・ツールプロパティの[接続用設定]タブの[クロック]カテゴリの[メイン・クロック・ソース]を[EXTAL]に設定してください。

・PLL のクロックソースに EXTAL を使用する場合は、[メイン・クロック周波数]に EXTAL の周波数を設定してください。

・PLL のクロックソースに HOCO を使用する場合は、[メイン・クロック周波数]に HOCO の周波数を設定してください。

5.3.25 トレースパネルの DMA 表示に関する注意事項

【対 象】 IECUBE, RL78, 78K0R

DMA による SFR/メモリへのアクセスがあった場合、トレースパネルに“DMA”の文字列が表示されません。(DMA アクセスによる“アドレス”、“データ”は問題なく表示されます。)

5.3.26 デバッグ・ツールプロパティの注意事項

【対 象】 全デバッグ・ツール, RH850

プロパティパネルの[接続用設定]タブから仮想マシン・スレッドを使用するを“いいえ”から変更しないでください。

5.3.27 CPU リセット、またはダウンロード後の注意事項

【対 象】 全デバッグ・ツール, RL78

ビルド・ツールで「最適化時のデバッグ情報強化を行う: はい」に設定し、CPU リセットやダウンロード後に main 関数の先頭で停止した場合、逆アセンブルパネルを開くことがあります。この様な場合、main 関数の先頭に“__nop();”を追加してください。

5.4 解析ツールの注意事項

5.4.1 解析グラフパネルに関する注意事項

- ・ エミュレータにて、内蔵トレースのタイム・タグをサポートしていない場合は、解析グラフパネルは使用できません。(E1/E20,RX)
- ・ 値の推移グラフにて、デバッグ・ツールにシミュレータを指定している場合、IOR のリアルタイム・サンプリング方式はサポートしていません。
- ・ 実行時間の割合に表示する結果は正確でない可能性があります。これはトレースのタイム・ラグ計測用カウンタが小さくオーバーフローする可能性があるためです。オーバーフローしているかはトレースパネルのタイムスタンプで確認してください。(E1/E20,RX)

5.5 Python コンソールの注意事項

5.5.1 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、外部エディタ等で作成しコピーし貼り付けてください。

5.5.2 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>>>の後に表示結果が、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。

5.5.3 フォルダやファイルへのパスに関する注意事項

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¥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、""(パス指定)の前にrを記載してください。IronPython は""の中がパスと認識します。

(例) `r"c:\¥test¥test.py"`

なお、パスの指定には¥(バックスラッシュ)ではなく/(スラッシュ)も使用可能です。

5.5.4 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、もしくはプロジェクト・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おいてある場合は、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。

5.5.5 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。

1. 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
2. 複数のプロジェクトをもつプロジェクトでアクティブ・プロジェクトを変更した場合

5.5.6 強制停止に関する注意事項

コンテキスト・メニューの[強制停止]を実行した場合、実行中のスクリプトや関数を強制停止しますが、[強制停止]した時点で実行が開始していない Hook 関数や Callback 関数がある場合は、[強制停止]後順次実行します。

5.5.7 ビルド中の Python コマンドの実行に関する注意事項

ビルド中に Python コマンドを使用しないでください。

第6章 制限事項

本章では、制限事項について説明します。

6.1 デバッグ・ツールの制限事項

文中において、以下の略称を使用しています。

OCD(シリアル) : E1 エミュレータ(シリアル), E20 エミュレータ(シリアル)

OCD(JTAG) : E1 エミュレータ(JTAG), E20 エミュレータ(JTAG)

6.1.1 デバッグ・ツールの制限事項一覧

No.	対象ツール	対象デバイス	制限事項	備考
1	OCD(シリアル) OCD(JTAG)	RX64M	認証切れエラーに関する制限事項	

6.1.2 デバッグ・ツールの制限事項詳細

No.1 認証切れエラーに関する制限事項

【対 象】OCD（シリアル）、OCD（JTAG） RX64M

【内 容】以下の条件を全て満たした場合、認証切れエラーが発生し、デバッグを継続することができなくなります。

[条件]

1. ID コードにオール FF 以外を設定しているデバイスをユーザブートモードでデバッグしている。
2. オプション設定メモリ領域へのデータが含まれるプログラムをダウンロードした後に、リセットコマンドや端子リセットまたは内部リセットが発生した場合。

【回避策】回避策はございません。

すべての商標および登録商標は、それぞれの所有者に帰属します。

ご注意書き

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

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

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



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 エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口： <http://japan.renesas.com/contact/>