

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## 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のですが、誤りが無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、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と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。

=== 必ずお読み下さい ===

M32Rファミリー用 Cコンパイラパッケージ

V.4.30 Release 00

リリースノート 第4版

株式会社ルネサス ソリューションズ

2008年 2月 27日

この度はM32Rファミリー用Cコンパイラパッケージ(以下、CC32Rと略します) V.4.30 release 00 を採用いただきまして誠に有難うございます。

本資料はCC32Rのインストール方法および電子マニュアルの補足等について説明します。電子マニュアルの該当項目をご覧になる場合は、併せてこのリリースノートをご覧いただきますようお願い申し上げます。

本資料記載内容は、特性改良などにより予告なしに変更することがあります。

Active X、Microsoft、MS-DOS、Visual Basic、Visual C++、WindowsおよびWindows NTは、米国Microsoft Corporationの米国およびその他の国における商標または登録商標です。  
HP-UXは、米国Hewlett-Packard Companyのオペレーティングシステムの名称です。  
Sun、Solaris、JavaおよびすべてのJava関連の商標およびロゴは、米国およびその他の国における米国Sun Microsystems, Inc.の商標または登録商標です。  
UNIXは、The Open Groupの米国ならびにその他の国における登録商標です。  
IBMおよびATは、米国International Business Machines Corporationの登録商標です。  
HP 9000は、米国Hewlett-Packard Companyの商品名称です。  
SPARCおよびSPARCstationは、米国SPARC International, Inc.の登録商標です。  
Intel、Pentiumは、米国Intel Corporationの登録商標です。  
i386、i486、MMXは、米国Intel Corporationの商標です。  
AdobeおよびAcrobatは、Adobe Systems Incorporated (アドビシステムズ社)の登録商標です。  
NetscapeおよびNetscape Navigatorは、米国およびその他の諸国のNetscape Communications Corporation社の登録商標です。  
その他すべてのブランド名および製品名は個々の所有者の登録商標もしくは商標です。

=====

COPYRIGHT(C) 2008 **Renesas Technology Corp.** and **Renesas Solutions Corp.**

# 1. V.4.20 Release 1からの変更点

---

前バージョンからの変更点です。

## 64bit整数演算関数ライブラリの追加 (標準ライブラリ)

64bit 整数演算を行う関数群を追加しました。C 言語の整数型と同様に、加減乗除、ビット演算、シフト、比較などの演算を 64bit の範囲で行うことが可能です。

詳しくは、ユーザズマニュアル《コンパイラ編》「第 13 章 64bit 整数演算ライブラリ」を参照ください。

## ベースレジスタ機能のセクション指定対応 (C コンパイラ)

アクセス制御ファイルの「ワイルドカード指定」に、セクション付きの指定ができるようになりました。これにより、`#pragma SECTION` でベースレジスタを適用するシンボルの出力先を変更しておき、変更セクションを「セクション指定つきワイルドカード」に指定して、複数のシンボルに一括してベースレジスタを適用、といった使い方ができます。

詳しくは、ユーザズマニュアル《コンパイラ編》「A.1 ベースレジスタ機能」の「A.1.7.2 アクセス制御ファイルの文法」(3)オブジェクト登録行 を参照ください。

## アクセス制御ファイル出力機能の拡張 (マップジェネレータ)

ロードモジュールからアクセス制御ファイルを作成する際、`static` 宣言されたオブジェクトや、ローカル変数も出力するようにしました。

(アクセス制御ファイルの作成にはデバッグ情報が必要です。)

## リンクエラー時のマップ出力に対応 (リンカ)

マップ生成(-M)オプション指定時にリンクエラーが発生していても、仮のリンクマップファイルを生成するようにしました。正常リンク時とアドレスが異なる場合があるためあくまで仮の情報ですが、セクションの配置などが原因で発生したリンクエラーの原因を解析するには有効です。

## csvシンボルマップファイル出力に対応 (マップジェネレータ)

csv 形式のシンボルマップファイルを生成する機能を追加しました。オプションは `-c` または `-c16` です。csv 形式ですので、Microsoft Excel 等の表計算ソフトに入力してアドレスとシンボルの対比表を作成することができます。

(csv シンボルマップの出力にはデバッグ情報が必要です。)

詳しくは、ユーザズマニュアル《アセンブラ編》「第 3 部 マップジェネレータ」「第 5 章 csv シンボルマップファイル作成機能」を参照ください。

## Sフォーマットのソート出力 (ロードモジュールコンバータ)

S1,S2,S3 レコード(データ)をアドレス順に出力するようにしました。

(他のレコードは従来と同じ行に出力します。)

## コメントのネストの警告抑止オプションを追加 (C コンパイラ)

コメント中に記述されたコメントの開始指定(`/*` や `//`)に対する警告を抑止するオプション(`-warn_suppress_nested_comment`)を追加しました。

## シンボル数の上限(-XXオプション)のデフォルト値を変更 (C コンパイラ)

-XX=シンボル数 オプションを指定しなかった場合の、シンボル数の上限のデフォルトを

40,000 に変更しました。(V.4.20 では 10,000)

### 統合開発環境HEW 3.01.04 を収録しました

添付の HEW を 3.01.04 にリビジョンアップしました。

なお2004年9月1日時点で、HEWの最新版は3.01.05となっており、このパッケージに添付されるHEWは最新版ではございませんのでご注意ください。お手数ですが、HEW 3.01.05の詳細を下記ツールニュースにてご確認ください、必要に応じダウンロードのうえ更新くださるようお願いいたします。

RSO-HEW-040801D「HEW 3.01.05へのリビジョンアップのお知らせ」(2004年8月1日発行)

[ <http://tool-support.renesas.com/jpn/toolnews/n040801/tn5.htm> ]

### 次の8つの問題点を解消しました

- ポインタを用いてメモリ内容を読み出す場合に読み出すアドレスが不正になる問題 ( C コンパイラ )  
[RENASAS TOOL NEWS 2004年08月01日 : RSO-M3T-CC32R-040801D 掲載]
- ビットフィールドを使った変数の変更が以降の参照に反映されない問題 ( C コンパイラ ) [RENASAS TOOL NEWS 2004年05月01日 : RSO-M3T-CC32R\_2-040501D 掲載]
- 累乗関数(powf5, powf, およびpow)の演算が不正になる問題 ( 標準ライブラリ )  
[RENASAS TOOL NEWS 2004年05月01日 : RSO-M3T-CC32R\_1-040501D 掲載]
- Windows版で標準ライブラリが正常に再構築できない問題 ( 標準ライブラリ。P C 版のみ ) [RENASAS TOOL NEWS 2004年02月01日 : RSO-M3T-CC32R\_2-040201D 掲載]
- 指数部を持つ浮動小数点定数の値が不正になる問題 ( C コンパイラ )  
[RENASAS TOOL NEWS 2004年02月01日 : RSO-M3T-CC32R\_1-040201D 掲載]
- printfなどの書式付き出力関数を使用して 0.5未満の値を表示が正しくない問題 ( 標準ライブラリ )  
[RENASAS TOOL NEWS 2004年01月16日 : RSO-M3T-CC32R-040116D 掲載]
- ポインタ型の仮引数を使った同一の間接参照を、関数をはさんで 2つ以上実行する場合に、関数呼び出し後の間接参照を関数呼び出し前の値に置き換えてしまう問題 ( C コンパイラ )  
[RENASAS TOOL NEWS 2003年12月16日 : RSO-M3T-CC32R-031216D 掲載]
- 並列の関係にある文または式で float型の演算のみが異なる場合に、これらの式を誤って削除する問題 ( C コンパイラ )  
[RENASAS TOOL NEWS 2003年11月16日 : RSO-M3T-CC32R-031116D 掲載]

### マニュアル正誤表

添付マニュアルの正誤表を示しますので、下記内容に訂正のうえご参照ください。

#### 【 C/C++コンパイラ編 】

ページ	箇所	誤	正
CC32R-iii (5/395)	第7章 組み込み用アプリケーションの作成	7.4 HEW におけるスタートアップファイル start.ms について	7.4 High-performance Embedded Workshop におけるスタートアップファイル start.ms について
cc32R-9 (22/395)	2.2 前バージョンとの互換性 -MAP オプションのエラー処理の緩和	cc32Rのリンクマップファイル出力オプション、-Mオプション (または、lnk32Rの-Mオプション) を指定していた場合、...	cc32Rのリンクマップファイル出力オプション、-MAPオプション (または、lnk32Rの-Mオプション) を指定していた場合、...

《次頁へ続く》

《次頁からの続き》

ページ	箇所	誤	正
cc32R-11 (24/395)	3.1.3 コマンド行の記述規則 図3.1 cc32Rコマンドの入力書式	<code>-warn_suppressed_nested_comment</code>	<code>-warn_suppress_nested_comment</code>
cc32R-54 (67/395)	4.6 C言語記述上の制限事項、表4.20 C言語記述上の制限事項「宣言」/「外部識別子、内部識別子、マクロ名における有効文字数」	最大31文字まで。	最大240文字まで。
cc32R-104 (117/395)	7.4 (タイトル)	7.4 HEW におけるスタートアップファイル start.ms について	7.4 High-performance Embedded Workshop におけるスタートアップファイル start.ms について
cc32R-104 (117/395)	7.4 HEW におけるスタートアップファイル start.ms について 本文	HEW でプロジェクトを新規作成する際に作成する、start.ms は、TM およびユーザーズマニュアルで使用しているstart.ms を、HEW 用に変更したものです。	High-performance Embedded Workshop でプロジェクトを新規作成する際に作成する、start.ms は、TM およびユーザーズマニュアルで使用しているstart.ms を、High-performance Embedded Workshop 用に変更したものです。
cc32R-104 (117/395)	7.4 HEW におけるスタートアップファイル start.ms について 本文	内容は基本的に同じですが、HEW からアセンブラ as32R の-D オプションに次のシンボルを指定することで、start.ms のパラメータを操作するようになっています。	内容は基本的に同じですが、High-performance Embedded Workshop からアセンブラ as32R の-D オプションに次のシンボルを指定することで、start.ms のパラメータを操作するようになっています。
cc32R-104 (117/395)	7.4 HEW におけるスタートアップファイル start.ms について 表7.3 (表のタイトル)	表7.3 HEW におけるスタートアップファイル start.ms のシンボルの意味	表 7.3 High-performance Embedded Workshop におけるスタートアップファイル start.ms のシンボルの意味
cc32R-104 (117/395)	7.4 HEW におけるスタートアップファイル start.ms について 表7.3 HEW におけるスタートアップファイル start.ms のシンボルの意味	HEWのプロジェクト作成ダイアログでの項目名	High-performance Embedded Workshop のプロジェクト作成ダイアログでの項目名
cc32R-267 (280/395)	10.2.8 レジスタ (Registers) ANSI規格 6.5.1 記憶クラス指定子 <register宣言できるオブジェクトの数>	記憶クラス指定子registerは無視されます。	記憶クラス指定子registerは認識されませんが、全てのregister指定されたオブジェクトがレジスタに割り付けられるとは限りません。実際にどのオブジェクトをレジスタに割り付けるかは、コンパイラが判断します。 register宣言できるオブジェクト数に制限はありません。

## 2. インストールを始める前に

---

インストールを開始するにあたり、以下の点にご注意ください。

### ライセンスIDについて

インストールの途中でライセンスIDを入力する必要がありますので、インストールを始める前にライセンスID証書によりライセンスIDをご確認ください。

なお、オンラインバージョンアップあるいはバージョンアップ購入により本バージョンを入手された場合は、バージョンアップ前のライセンスIDを用いてインストールしてください(本バージョンの CC32R は、V.1.00 Release 4 以降のライセンスIDでインストールできます)。

### 必要なメモリおよびHDDの容量

CC32Rを快適に使用するためには、64Mバイト以上のメモリと500Mバイト以上の空きハードディスク領域が必要です。

### 動作環境

Windows 3.1およびWindows NT 3.5x以前のバージョンでは動作しません。

### ファイル名、ディレクトリ名

インストール先のディレクトリ名や、コンパイラに指定するソースプログラムのファイル名やディレクトリ名は、次の注意事項に従ってください。

- ・ 漢字や全角文字を含むディレクトリ名、ファイル名は使用できません。
- ・ ファイル名に使用するピリオド( . )は1つのみ使用可能です。
- ・ ネットワークパス名は使用できません。ドライブ名に割り当ててご使用下さい。
- ・ ショートカットは使用できません。
- ・ 空白を含むディレクトリ名、ファイル名は使用できません。
- ・ ". . ."表記を用いて2つよりも上のディレクトリを指定することはできません。
- ・ パス指定を含めたファイル名の長さが128文字以上になるものは使用できません。

### SQLint ( MISRA C ルールチェッカ ) を導入される場合

SQLintのインストールは、CC32Rのインストールの後に行ってください。

SQLintの後にインストールされたCC32Rからは、SQLintの機能を利用することはできません。

詳しくは、SQLintに添付のマニュアルまたはリリースノートを参照ください。

### 古いバージョンのCC32R がインストールされている場合

古いCC32Rがインストールされている場合は、この古いCC32Rをアンインストールしてから新しいバージョンをインストールしてください。

CC32Rをアンインストールするには、Windowsの [ コントロールパネル ] - [ アプリケーションの追加と削除 ] から、 [ CC32R V.x.xx Release x ] ( xにはバージョン番号等が入ります ) を選択し、 [ 追加と削除(R) ] をクリックしてください。

### ユーザ登録のお願い

製品のインストールが完了しましたら、ユーザ登録を行ってください。

ユーザ登録されないと、バージョンアップ情報の提供や技術サポートなどのサービスを受けることができません。

インストールすると以下のファイルが生成されます。

C:\¥mtool¥support¥cc32r¥regist.txt

「C:\¥mtool」はデフォルトでインストールした場合のディレクトリ名です。

この regist.txt のファイル内容をすべてコピー & ペーストして以下の電子メールアドレス宛てに送付してください。

regist\_tool@renesas.com

電子メールをご利用になれない場合は、regist.txt のファイル内容をプリントアウトしてファクシミリで送付ください。送付先はライセンス ID 証書に記載されています。  
(ファクシミリの場合、ユーザ登録の完了まで多少日数がかかることがあります。できるだけ電子メールをご利用ください。)

(株)ルネサス テクノロジーの個人情報保護方針につきましては、ルネサステクノロジーのホームページ「個人情報保護について」をご覧ください。

ホームページ：

<http://japan.renesas.com/fmwk.jsp?cnt=privacy.htm&fp=/privacy/&site=i>

ユーザ登録でご提供頂きました個人情報、お客様のサポート活動に活用させて頂き、そのために必要な範囲で(株)ルネサス テクノロジー、およびその関係会社、ならびに特約店に、電子データ、書面により提供させて頂きますので、ご了承の程お願い申し上げます。なお、提供を希望されない場合は、提供を停止させて頂きますので、お問い合わせ時にその旨ご連絡ください。その場合、サポート範囲が制約される場合がございます。

## 環境変数の設定

次の環境変数は必ず設定して下さい。設定しない場合や不正なパスを指定した場合は、動作中にエラーが発生することがあります。(ただし、PC版でTMからCC32Rを動作させる場合は設定の必要はありません)

M32RBIN  
M32RLIB  
M32RINC  
M32RTMP

## ウイルスチェックプログラム

ウイルスチェックプログラムが常駐した状態でインストーラを起動すると正常に動作しない場合があります。その場合はウイルスチェックプログラムの常駐を解除してからインストーラを起動し直してください。

## CC32R の最新情報を入手するには

以下のURLにホームページを公開しています。こちらでは、ルネサス開発環境に関する最新情報が掲載されていますのでご活用ください。

<http://www.renesas.com/jp/tools>



## 3. CC32Rのインストール

### 3.1 インストール手順

(1) 次のA.またはB.のいずれかの方法に従ってインストーラを起動してください。

**[A. 添付CDから製品をインストールする場合]**

ご使用の言語に合わせ、表1に示すそれぞれのインストーラを直接起動してください。  
(ドライブ名はご使用のPCにより異なりますので別途ご確認ください。)

表1 CC32Rのインストーラ (CD-ROM のドライブ名を Q: とした場合)

対応OS	言語	インストーラ (ファイル名)
Windows 98 / SE Windows Me Windows NT 4.0	日本語	Q:\¥CC32R¥WINJ¥SETUP.EXE
Windows 2000 Windows XP	英語	Q:\¥CC32R¥WINE¥SETUP.EXE

**[B. オンラインバージョンアップの場合]**

ホームページに記載している方法でインストーラをダウンロードし、インストールしてください。  
オンラインバージョンアップの場合、最初にインストーラを展開するディレクトリ(「ファイルを保存するフォルダ」)を尋ねられます。このディレクトリは、中身がないか存在しないディレクトリを指定してください。もし中身のあるディレクトリを指定して、「ファイルを上書きしてもよろしいですか?」と聞かれた場合は、確認のうえ「すべてはい(E)」を押して続行するか、「キャンセル」を押してインストーラの起動からやり直して中身のないディレクトリを指定し直してください。

(2) 表示されるメッセージに従ってインストールを実行してください。

**【HEWのインストールについて】**

HEWでCC32Rをご利用する場合は、次の方法でCC32RインストーラからHEWのインストールを行ってください。

手順 でHEWのインストーラが起動されない場合は、一旦CC32Rのインストールを中止し、先にHEWのみをインストールしてください。その後、CC32Rのインストーラを起動してください。  
HEWは2004年8月1日に3.01.05にリビジョンアップしましたが、本バージョンにはこれよりも古いHEW 3.01.04を収録しております。お手数ですが HEW 3.01.05の詳細を下記ツールニュースにてご確認ください、必要に応じダウンロードのうえ更新くださるようお願いいたします。  
RSO-HEW-040801D「HEW 3.01.05へのリビジョンアップのお知らせ」(2004年8月1日発行)  
[<http://tool-support.renesas.com/jpn/toolnews/n040801/tn5.htm>]

CC32Rインストーラを起動すると、HEWのインストーラが起動します。その指示に従ってHEWのインストールを行ってください。

CC32Rのインストーラは一時停止しています。HEWのインストーラの終了後に自動的に再開します。

**【TMのインストールについて】**

CC32RインストーラからTMのインストーラも起動できるようになりました。  
(すでにTMがインストールされている場合は、アンインストールしておいてください。)

インストール中に「TMのインストーラを起動しますか?」と尋ねられます。そこで「はい(Y)」を押せばTMのインストーラを起動します。

CC32Rのインストーラは、「TMのインストールが完了したらOKを押してください」という画面で一時停止しています。TMのインストーラの終了を確認したら、「OK」を押してCC32Rのインストールを再開してください。

**【ユーザ情報の入力について】**

インストールの途中、会社名や連絡先などのお客様の情報の入力がありますが、これはユーザ登録のために必要なファイルを作成するのに使用しています。

## 3.2 環境設定

DOSプロンプト上でCC32Rを利用する場合は、表2の環境変数を設定してください。  
表の中の「自動」は、インストーラがAUTOEXEC.BATを書きかえます。デフォルトの設定でインストールを実行した場合は、これらの設定は必要ありません。

表2 CC32Rの環境変数設定例 (C:¥mtoolをインストール先ディレクトリに選択した場合)

環境変数	設定例	備考
M32RBIN	SET M32RBIN=C:¥mtool¥bin32R	自動
M32RLIB	SET M32RLIB=C:¥mtool¥lib32R	自動
M32RINC	SET M32RINC=C:¥mtool¥inc32R	自動
M32RTMP	SET M32RTMP=C:¥mtool¥TMP	自動
M32RKIN	SET M32RKIN=sjis	
M32RKOUT	SET M32RKOUT=sjis	
コマンドパス	PATH %M32RBIN%;%PATH%	自動

- ・ 設定例は、インストーラのデフォルトのインストール先ディレクトリ (C:¥MTOOL) に製品をインストールした場合です。インストーラのデフォルトと異なるディレクトリに製品をインストールした場合は、実際のディレクトリにあった内容に変更してください。
- ・ M32RKIN, M32RKOUTは日本語処理機能の文字コードを指定します。  
PC版では自動的にsjisが設定されますので、入力も出力もShift-JISを使用する場合はこれらの環境変数は設定の必要はありません。
- ・ HEW,またはTMから利用する場合は、環境変数の設定は必要はありません。

## 3.3 電子マニュアルの参照方法

電子マニュアルを参照するには、Acrobat Reader (アドビシステムズ社) などのPDFファイル表示プログラムが必要です。必要に応じてインストールしてください。

- ・ Acrobat Readerについて

次のアドビシステムズ社のホームページからダウンロードしてください。

Acrobat Readerの動作環境や最新情報なども掲載されています。

<http://www.adobe.co.jp/> (日本)

<http://www.adobe.com/> (米国)

電子マニュアルを参照するには次の方法があります。

1. スタートメニューから電子マニュアルを開く  
インストール後は、電子マニュアルはWindowsのスタートメニューに登録されます。  
[ スタート ] [ プログラム(P) ] [ RENESAS-TOOLS ]  
[ CC32R V.x.xx Release x ]  
(xにはバージョン番号等が入ります) から、目的の電子マニュアルを選択してください。
2. 電子マニュアルファイルをダブルクリックして開く  
デフォルトのインストール先 (C:¥mtool) でインストールした場合、電子マニュアルをC:¥mtool¥manualの下にインストールします。これらのファイル (拡張子.pdf) をダブルクリックすると、Acrobat Readerでファイルを見ることができます。)  
電子マニュアルのファイル名を表3に示します。

表3 電子マニュアルファイル一覧

言語	マニュアルタイトル	PDFファイル名
日本語	M3T-CC32Rユーザーズマニュアル (Cコンパイラ編)	CC32Ruj.pdf
	M3T-CC32Rユーザーズマニュアル (アセンブラ編)	AS32Ruj.pdf
	マップビューワ ユーザーズマニュアル	mapuj.pdf
英語	M3T-CC32R User's Manual <C Compiler>	CC32Rue.pdf
	M3T-CC32R User's Manual <Assembler>	AS32Rue.pdf
	Map Viewer User's Manual	mapue.pdf

【補足】 (マップビューワのマニュアルについて)

マップビューワのマニュアルは、NC30WAとの組み合わせをベースに記述されていますが、CC32Rとの組み合わせでも同様に利用することができます。

## 4. V.4.30 Release 00 使用上のご注意

本バージョンにおいて確認されている使用上の注意事項です。

### 右辺が符号なしであるシフト演算のコンパイルが正常に行われない問題 (C コンパイラ)

RENESAS TOOL NEWS [2006年03月01日 : RSO-M3T-CC32R-060301D] 掲載 (予定)

整数の定数式において、右辺が符号なしであるシフト演算を含む場合、次のような警告メッセージを表示します。

```
"xxxx", line XX: warning: shift count greater than number of bits  
(xxxx はファイル名を、XX には行番号をそれぞれ表示します。)
```

この定数式をビットフィールドメンバのサイズを表す場所に記述すると、本来とは異なるサイズとして認識します。それ以外では、コンパイルエラーが発生する場合があります。

#### 発生条件

次の(1) ~ (3)全ての条件を満たす場合に発生します。

- (1) シフト演算を含んだ整数の定数式がある。
- (2) (1)の定数式において、シフト演算の右オペランド(ビット数を表す)が符号なし整数型である。
- (3) (1)の定数式を次の(a) ~ (e)のいずれかの値を表すのに用いている。
  - (a) 構造体のビットフィールドメンバの大きさ
  - (b) 配列の大きさ
  - (c) 列挙定数の値
  - (d) caseラベルの定数
  - (e) (a) ~ (d)以外の数値を表す箇所

#### 【注意】

(3)(a)に該当する場合、コンパイラは、ビットフィールドの大きさを本来とは異なり、宣言型をサイズと解釈してコード生成を行います。このため、もし、前述の警告が表示された場合、該当する定数式をビットフィールドメンバのサイズを意味する場所に記述していないかどうかご確認ください。

(3)(b) ~ (d)のいずれかに該当する場合は、コンパイルエラーとなります。

(3)(e)に該当する場合は、2.「内容」の警告メッセージは表示しますが、生成コードに影響はありません。

#### 発生例

[ソースファイル例1] --- 構造体のビットフィールドメンバの大きさ  
警告メッセージを表示します。

メンバb01は4ビットではなく、16ビット(short型のサイズ)となります。

[sample1.c]

```
-----  
struct SBtag {  
    short b01:1<<(unsigned)2;    /* 発生条件(1),(2)および(3a) */  
    short b02:1;  
};  
-----
```

[ソースファイル例2] --- 配列の大きさ

警告メッセージに加え、次のエラーを表示します:

```
error: array: subscript must be positive, non-zero, integral value
```

[sample2.c]

```
-----  
int array[8>>1u];                /* 発生条件(1),(2)および(3b) */  
-----
```

-----  
[ソースファイル例3] --- 列挙定数の値

警告メッセージに加え、次のエラーを表示します:

error: enumeration-constant out of range

[sample3.c]

-----  
enum Etag {  
 A,  
 B = 2<<(unsigned)2 /\* 発生条件(1),(2)および(3c) \*/  
};  
-----

[ソースファイル例4] --- caseラベルの定数

警告メッセージに加え、次のエラーを表示します:

error: unable to evaluate case label (out of range?)

[sample4.c]

-----  
int func4(int key)  
{  
 switch (key) {  
 case ((3\*8)>>(7UL-5)): /\* 発生条件(1),(2)および(3d) \*/  
 return 1;  
 }  
 return 0;  
}  
-----

[ソースファイル例5] --- その他の値が記述できる箇所

警告メッセージを表示します。生成コードには影響しません。

[sample5.c]

-----  
int data = 0x8000>>12u; /\* 発生条件(1),(2)および(3e) \*/  
-----

## 回避策

以下の回避例のように、シフト演算子の右パラメータにキャスト (signed) を付加して、シフト演算子の右パラメータを符号付き整数にしてください。

[sample1.c の変更例]

-----  
struct SBtag {  
 short b01:1<<(signed)(unsigned)2; /\* 右パラメータに(signed)を付加\*/  
 short b02:1;  
};  
-----

[sample2.c の変更例]

-----  
int array[8>>(signed)1u]; /\* 右パラメータに(signed)を付加 \*/  
-----

[sample3.c の変更例]

-----  
enum Etag {  
 A,  
 B = 2<<(signed)(unsigned)2 /\* 右パラメータに(signed)を付加 \*/  
};  
-----

[sample4.c の変更例]

-----  
int func3(int key)  
{  
 switch (key) {  
 case ((3\*8)>>(signed)(7UL-5)): /\* 右パラメータに(signed)を付加 \*/  
 return 1;  
 }  
}  
-----

```
    return 0;  
}
```

-----  
[sample5.c の変更例]  
-----

```
int data = 0x8000>>(signed)12u; /* 右パラメータに(signed)を付加 */  
-----
```

## #pragma ADDRESSで指定された構造体、配列または共用体を含むオブジェクトのアクセスが不正になる問題 (C コンパイラ)

RENESAS TOOL NEWS [2005年02月16日 : RSO-M3T-CC32R-050216D] 掲載

集成体型(配列型および構造体型)または、集成体型をメンバに持つ共用体に#pragma ADDRESS を指定した場合、これらの先頭以外の要素に対し、アクセスまたは&でアドレスを取り出す処理を行うと、不正な(先頭と同じアドレスを用いた)コードを生成します。

### 発生条件

次の(1) ~ (3)全ての条件を満たす場合に発生します。

- (1) 次のいずれかのオブジェクトを、関数の外で定義している。
  - (a) 構造体
  - (b) 配列
  - (c) 構造体または配列をメンバに持つ共用体
- (2) (1)のオブジェクトに#pragma ADDRESS を指定している。
- (3) (1)のオブジェクトの、先頭アドレス以外に配置されたメンバまたは要素に対し、次のいずれかの処理を行っている
  - (a) アクセス(読み出しあるいは書き込み)。
  - (b) &演算子による配置アドレスの取得

### 発生例

[ソースファイル例1]

[sample1.c]

```
-----  
#pragma ADDRESS data1 0x10000 /* 条件(2) */  
struct stg1 {  
    int a;  
    int b;  
} data1; /* 条件(1)(a) */  
  
void func1(void)  
{  
    data1.b = 3; /* 条件(3)(a) */  
}  
-----
```

[ソースファイル例2]

[sample2.c]

```
-----  
#pragma ADDRESS data2 0x20000 /* 条件(2) */  
short data2[10]; /* 条件(1)(b) */  
short *d2;  
  
void func2(void)  
{  
    d2 = &data2[3]; /* 条件(3)(b) */  
}  
-----
```

[ソースファイル例3]

[sample3.c]

```
-----  
union utg3 {  
    double a[10];  
    struct {  
-----
```

```
        int b;
        double c[5];
    } d;
} data3;                                /* 条件(1)(c) */
#pragma ADDRESS data3 0x30000          /* 条件(2) */
void func3(int i)
{
    data3.d.c[i] =                      /* 条件(3)(a) */
    data3.a[4];                          /* 条件(3)(a) */
}
```

-----

## 回避策

下記の例のように、マクロを使ってアドレスをポインタにキャストしたアクセスに変更してください。

[sample1.c の変更例]

```
-----
#define data1 ( *(struct stg1*) 0x10000 ) /* マクロに変更 */
struct stg1 {
    int a;
    int b;
} /* data1 */ ; /* data1の定義は削除し、
                構造体タグstg1の宣言のみにする */
void func1(void)
{
    data1.b = 3;                          /* マクロにより置換される */
}
```

-----

[sample2.c の変更例]

```
-----
#define data2 ( (short *) 0x20000 ) /* マクロに変更 */
/* short data2[10]; */ /* data2の定義は削除する */
short *d2;
void func2(void)
{
    d2 = &data2[3];                       /* マクロにより置換される */
}
```

-----

[sample3.c の変更例]

```
-----
union utg3 {
    double a[10];
    struct {
        int b;
        double c[5];
    } d;
} /* data3 */ ; /* data3の定義は削除し、
                構造体タグutg3の宣言のみにする */
#define data3 ( *(union utg3*) 0x30000 ) /* マクロに変更 */
void func3(int i)
{
    data3.d.c[i] =                      /* マクロにより置換される */
    data3.a[4];                          /* マクロにより置換される */
}
```

-----

## 配列の引数を含む関数へのポインタにキャストを関数呼び出しに用いるとエラーになる問題 (Cコンパイラ)

RENESAS TOOL NEWS [2004年12月01日 : RSO-M3T-CC32R-041201D] 掲載

キャストを使って引数に配列を含む関数へのポインタに型変換し、関数呼び出しを行う場合、

キャスト中の仮引数と関数呼び出しの実引数の型が適合する場合でも、次のような型不一致のエラーを表示します。

```
error: type of argument does not match with prototype
```

#### 発生条件

次の(1),(2)の条件を全て満たすCソースファイル上の記述に対して発生します。

- (1) キャストにより、任意のポインタまたは整数を、関数へのポインタに型変換し、ポインタを介さず関数呼び出しに使っている。
- (2) キャストの関数の仮引数のうち少なくともひとつが配列型である。

#### 発生例

[例1]

[sample1.c]

```
-----  
void func1(int, short arr[]);  
short array[3];  
  
void foo1(void *ptr1)  
{  
    (*(void*)(int, short arr[]))ptr1(2,array);    /* 条件(1)(2) */  
}  
-----
```

[例2]

[sample2.c]

```
-----  
typedef struct AAA TYPE_A;  
int func2(TYPE_A arr[][3]);  
char *ptr2;  
  
int foo2(TYPE_A array[][3])  
{  
    int ans;  
    ans = (*(int*)(TYPE_A arr[][3]))ptr2(array);    /* 条件(1)(2) */  
    return ans;  
}  
-----
```

[例3]

[sample3.c]

```
-----  
typedef float TYPE_F[3];  
int func3(TYPE_F);  
TYPE_F array;  
  
void foo3(void)  
{  
    (*(void*)(TYPE_F))0x123400(array);    /* 条件(1)(2) */  
}  
-----
```

#### 回避策

下記の例のように、当該箇所をポインタ変数を介した関数呼び出しに変更してください。

[sample1.c の変更例]

```
-----  
void func1(int, short arr[]);  
short array[3];  
  
void foo1(void *ptr1)  
{  
    void (*callptr1)(int, short arr[]) /* 当該関数のポインタを定義 */  
        = (void (*)(int,short arr[]))ptr1; /* 同、初期化 */  
  
    (*callptr1)(2,array);    /* ポインタを使った呼び出しに変更 */  
}  
-----
```

[sample2.c の変更例]

```
-----  
typedef struct AAA TYPE_A;  
int func2(TYPE_A arr[][3]);  
char *ptr2;  
  
int foo2(TYPE_A array[][3])  
{  
    int ans;  
    int (*callptr2)(TYPE_A arr[][3]); /* 当該関数のポインタを定義 */  
    callptr2 = (int (*)(TYPE_A arr[][3]))ptr2; /* 同、初期化 */  
  
    ans = (*callptr2)(array); /* ポインタを使った呼び出しに変更 */  
    return ans;  
}  
-----
```

[sample3.c の変更例]

```
-----  
typedef float TYPE_F[3];  
int func3(TYPE_F);  
void *ptr3;  
TYPE_F array;  
  
void foo3(void)  
{  
    void (*callptr3)(TYPE_F); /* 当該関数のポインタを定義 */  
    callptr3 = (void (*)(TYPE_F))0x123400; /* 同、初期化 */  
  
    (*callptr3)(array); /* ポインタを使った呼び出しに変更 */  
}  
-----
```

## char型の2次元配列を初期化子で初期化する場合にエラーになる問題 (Cコンパイラ)

RENESAS TOOL NEWS [2004年09月16日 : RSO-M3T-CC32R-040916D] 掲載

char型の2次元配列を、文字列リテラルを含んだ特定の形式の初期化子を使って、宣言と同時に初期化する記述に対し、コンパイラが次のエラーを発生します。

```
cg32r: "xxxx", line XX: internal error: illegal IL, size of initializer is  
larger than name size.
```

(xxxx はファイル名が、XX には行番号がそれぞれ表示されます。)

### 発生条件

次の(1) ~ (4)の条件全て満たすCソースファイル上の記述に対して発生します。

- (1) char型(ポインタ型除く)の2次元配列を初期化子付きで宣言している。
- (2) (1)の配列宣言に大きさを指定していない。
- (3) (1)の初期化子には、文字列リテラル "... " を2つ以上記述している。
- (4) (3)のうち、1つめの文字列リテラルを、{"..."} のように中括弧で囲んでいる。

### 発生例

[ソースファイル例1] --- 静的な初期化の例

[sample1.c]

```
-----  
char array1[][2] = { /* 発生条件(1),(2) */  
    {"ab"}, /* 発生条件(3),(4) */  
    "cd", /* 発生条件(3) */  
    "ef" /* 発生条件(3) */  
};  
-----
```

[ソースファイル例2] --- 動的な初期化の例

[sample2.c]



```
-----  
extern void array_func(char [][][6]);  
void func2(void)  
{  
    char array2[][6] = { /* 発生条件(1),(2) */  
        {"5678"}, /* 発生条件(3),(4) */  
        {"1234"}, /* 発生条件(3) */  
    };  
    array_func(array2);  
}  
-----
```

## 回避策

次のいずれかの方法で回避できます。

### (1) 配列のサイズを省略しない記述に変更する。

[sample1.c の回避例]

```
-----  
char array1[3][2] = { /* 個数3を省略せずに記述 */  
    {"ab"},  
    {"cd"},  
    {"ef"}  
};  
-----
```

[sample2.c の回避例]

```
-----  
extern void array_func(char [][][6]);  
void func2(void)  
{  
    char array2[2][6] = { /* 個数2を省略せずに記述 */  
        {"5678"},  
        {"1234"},  
    };  
    array_func(array2);  
}  
-----
```

### (2) 配列のサイズを省略しない記述に変更する。

[sample1.c の回避例]

```
-----  
char array1[][2] = {  
    "ab", /* {"ab"}, から 中括弧 { } をはずす */  
    "cd",  
    "ef"  
};  
-----
```

[sample2.c の回避例]

```
-----  
extern void array_func(char [][][6]);  
void func2(void)  
{  
    char array2[][6] = {  
        "5678", /* {"5678"}, から 中括弧 { } をはずす */  
        {"1234"},  
    };  
    array_func(array2);  
}  
-----
```

**右辺より左辺の型サイズが小さい代入式の後に、その代入式の右辺と等価な式を用いる場合に、等価な式の演算が不正になる問題 (C コンパイラ)**

RENESAS TOOL NEWS [2008年01月16日 : 080116/tn1] 掲載

右辺より左辺の型サイズが小さい代入式の後に、その代入式の右辺と等価な式を含む別式がある場合、二つ目の式を演算する際に代入式の右辺の演算結果を小さいサイズに型変換し

た結果の値が使用されます。

#### 発生条件

次の(1)～(7)の全ての条件を満たす場合に発生することがあります。

- (1) コンパイル時に使用している最適化オプションが次の(a)か(b)のいずれかに該当する。
  - (a) -O, -O2, -O3, -O6, -O7のいずれかを使用している。
  - (b) -Ospace, -Otimeのいずれかを使用し、かつ、-O0, -O1, -O4, -O5 のいずれも使用していない。
- (2) 整数型の変数をひとつ以上使い、かつ演算結果が整数型になる式がある。  
整数型の変数がひとつだけある場合も該当する。
- (3) (2)の演算結果を、結果の整数型よりもサイズが小さい整数型のauto変数に代入している。
- (4) (2)と等価な式を、(3)の代入式よりも後に実行される別の式の一部に使用している。
- (5) (3)を実行してから(4)を実行するまでに、(2)で参照されるいずれの変数も値が変更される可能性がない。
- (6) (3)のauto変数が、レジスタではなく、スタックに割り付けられている。
- (7) 変数がひとつだけの場合も含み、(2)の演算結果がレジスタに割り付けられている。

#### 注意：

発生条件(6)および(7)は、コンパイラの処理によってスタックやレジスタへの割付が決定されるため、条件に該当するかどうかは、ソースコードから判断することはできません。実際にコンパイルして得られる生成コードを確認してください。

#### 発生例

val\_ulong >> 8 の演算において、val\_ucharに代入された後の値を val\_ulongの値として使用するため、val\_ulong >> 8 の結果が正しい値となりません。

ただし、以下の例で省略している部分のプログラム内容によっては問題が発生しない場合があります。

[ソースファイル例]

[sample.c]

```
-----  
unsigned long func_ulong(unsigned short);  
void func(void)  
{  
    unsigned long val_ulong;  
    unsigned char val_uchar;  
    unsigned int val_another;  
    unsigned short val_ushort;  
    /*  
    . . . . .  
    (途中省略)  
    . . . . .  
    */  
    val_ulong = func_ulong(val_ushort);  
    val_uchar = val_ulong; /* 発生条件(2)(3) */  
    val_another = val_ulong >> 8; /* 発生条件(4)(5) */  
    /*  
    . . . . .  
    (途中省略)  
    . . . . .  
    */  
}
```

#### 回避策

次のいずれかの方法で回避できます。

- (1) レベル2の最適化を抑止する。  
オプション-O, -O2, -O3, -O6 または -O7 のいずれかを使用している場合は-O0,-O1,-O4,または-O5のいずれかに変更する。  
オプション-Ospaceまたは-Otimeを使用する場合は、同時に-O0,-O1,-O4,または-O5のいずれかを使用する。
- (2) 条件(3)と(4)の式の順番を入れ換える。

[sample.c の回避例]

```
-----  
unsigned long func_ulong(unsigned short);  
void func(void)  
{  
    . . . . .  
    val_ulong = func_ulong(val_ushort);  
    val_another = val_ulong >> 8; /* 発生条件(4)(5) */  
    val_uchar = val_ulong; /* 発生条件(2)(3) */  
    . . . . .  
}
```

### 参考

本問題は、V.5.01 Release 00で改修済みです。最新版をご使用ください。

## 5. ソフトウェア一覧

インストール後に作成されるディレクトリとファイルを表4に示します。

HEW, TMのインストーラが展開するファイルは含みません。

表4 インストール後のディレクトリ・ファイル一覧

ディレクトリ名	ファイル名	説明
bin32R	cc32R.exe as32R.exe lnk32R.exe lib32R.exe lmc32R.exe map32R.exe	コンパイルドライバ (V.2.11.00.000) アセンブルドライバ (V.2.03.00.000) リンカ (V.1.11.00.000) ライブラリアン (V.1.02.01.000) ロードモジュールコンバータ (V.1.12.00.000) マップジェネレータ (V.1.21.00.000)
lib32R	cpre.exe cfrt.exe postpar.exe opt.exe cg32R.exe a032R.exe a132R.exe alis32R.exe parafilt.exe cmerge.exe m32RcR.lib m32RcRM.lib m32RcRL.lib m32RcR.stk m32RcRM.stk m32RcRL.stk	プリプロセッサ (V.2.06.00.000) パーザ (V.2.23.00.000) ポストパーザ (V.1.00.01.000) グローバルオブティマイザ (V.1.25.00.000) コードジェネレータ (V.4.03.00.000) マクロプロセッサ (V.1.00.01.000) アセンブラ (V.4.04.01.000) リストプロセッサ (V.1.01.00.000) パラレルプロセッサ (V.1.00.02.000) Cソースマージプロセッサ (V.1.01.01.000) Cライブラリ (スモールモデル) " (ミディアムモデル) " (ラージモデル) " スタック使用量計算ファイル (m32RcR.lib) " " (m32RcRM.lib) " " (m32RcRL.lib)
inc32R	assert.h, ctype.h, errno.h, float.h, limits.h, locale.h, long64.h, math.h, mathf.h, setjmp.h, signal.h, stdarg.h, stddef.h, stdio.h, stdlib.h, string.h, time.h	Cライブラリヘッダ
	cstddef, cstdio, cstdlib, exception, new, stdexcept, typeinfo, new_ecpp.h, new_std.h	Cライブラリ予約ヘッダ 【注1】
inc32R¥sys	assert.h, ctype.h, errno.h, float.h, limits.h, locale.h, math.h, mathf.h, setjmp.h, signal.h,stdarg.h, stddef.h, stdio.h, stdlib.h, string.h, time.h	システム定義ヘッダ 【注2】
inc32R¥com	ANSI_errno.h, def.h, SBPP	
UnSpt32R 【注3】	strip32R.exe abslist.exe stk32R.exe license.sj license.txt cmerge.sj abslist.sj stk.sj cmerge.txt abslist.txt stk.txt	デバッグ情報除去ユーティリティ (V.1.00.00.000) ABSリストユーティリティ (V.1.00.05.000) スタック計算ユーティリティ (V.1.00.00.000) ユーティリティ取扱い条件 (日本語) ユーティリティ取扱い条件 (英語) ユーティリティマニュアル (日本語) " ( " ) " ( " ) ユーティリティマニュアル (英語) " ( " ) " ( " )

lib32R¥src	~.c ~.h ~.ms floatlow¥~.mo stack¥~.stk objects¥~¥~.mo	Cライブラリ 再生成用構成ファイル
	BUILD.bat CLEAN.bat	Cライブラリ ビルド用ツール "
support¥cc32r	userinfo.txt regist.txt	インストール情報
smp32R	start.ms	スタートアップ、低水準サンプル
bin	mapviewer.exe map_inspect.dll mapviewer.hlp mapviewer.cnt	マップビューワ (V.3.00.00) マップビューワ用DLL マップビューワ ヘルプファイル マップビューワ ヘルプ設定ファイル
lib32R	tmcpp.exe	TM用 依存関係解析プロセッサ (V.4.02.00)
	cc32r.mkt	TM用 make基本設定ファイル
	~.opt	TM用 オプション設定ファイル
manual	CC32Ruj.pdf AS32Ruj.pdf CC32Rue.pdf AS32Rue.pdf mapuj.pdf	ユーザーズマニュアルCコンパイラ編〔日本語〕 ユーザーズマニュアルアセンブラ編〔日本語〕 ユーザーズマニュアルCコンパイラ編〔英語〕 ユーザーズマニュアルアセンブラ編〔英語〕 マップビューワ マニュアル〔日本語/英語 (自動選択)〕
hew	~.det ~.dll ~.tbp	HEW設定ファイル

||||||| ご注意 |||||||

- 【注1】 inc32Rに含まれるCライブラリ予約ヘッダは、将来の拡張のために配置しているものです。  
現在は使用できませんので、インクルードはしないでください。
- 【注2】 inc32R¥sysに含まれるシステム定義ヘッダはCC32Rが参照しますので、変更および削除しないでください。変更および削除した場合の動作は保証されません。
- 【注3】 UnSpt32Rの各ディレクトリ内に含まれるプログラムは、ライセンス形態やサポート等の取り扱いが他のCC32Rの構成物とは異なっております。  
ディレクトリ内のlicense.sjの内容をご確認ください。