

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>

日立半導体技術情報

〒 1 0 0 - 0 0 0 4
 東京都千代田区大手町 2 丁目 6 番 2 号
 (日本ビル)
 TEL (03)5201-5022 (ダイヤルイン)
 株式会社 日立製作所 半導体グループ

製品分類	開発環境		発行番号	TN-CSX-042A	Rev.	第 1 版
題名	H8S,H8/300 シリーズ C/C++コンパイラパッケージ リビジョンアップのお知らせ		情報分類	①. 仕様変更 2. ドキュメント訂正追加等 3. 使用上の注意事項 4. マスク変更 5. ライン変更		
適用製品	PS008CAS4-MWR, PS008CAS5-MWR, PS008CAS4-SLR, PS008CAS4-H7R	対象ロット等 全ロット	関連資料	H8S,H8/300 シリーズ C/C++コンパイラ、アセンブラ、最適化リンケージエディタ ユーザーズマニュアル ADJ-702-303 第 1 版		有効期限 永年

H8S,H8/300 シリーズ C/C++コンパイラパッケージを Ver. 5.0.02(Windows 版)、Ver.4.0.06(UNIX 版)にリビジョンアップしました。

次に示す製品を御使用のお客様につきましては、周知願います。

Windows 版 : H8S,H8/300 シリーズ C/C++コンパイラパッケージ(型名:PS008CAS5-MWR)の Ver.5.0, Ver.5.0.01

H8S,H8/300 シリーズ C/C++コンパイラパッケージ(型名:PS008CAS4-MWR)の Ver.4.0, Ver.4.0r1, Ver.4.0A, Ver.4.0Ar1, Ver.4.0Ar2

SPARC 版 : H8S, H8/300 Series C/C++コンパイラパッケージ(型名:PS008CAS4-SLR)の Ver.4.0, Ver.4.0A, Ver.4.0Ar1, Ver.4.0B, Ver.4.0.05

HP9000 版 : H8S, H8/300 Series C/C++コンパイラパッケージ(型名:PS008CAS4-H7R)の Ver.4.0, Ver.4.0A, Ver.4.0Ar1, Ver.4.0B, Ver.4.0Br1, Ver.4.0.05

添付:

(1) H8S,H8/300 シリーズ C/C++コンパイラパッケージ V5.0.02(Ver.4.0.06) アップデート内容
 (PS008CAS5-020725J) 3 枚

H8S,H8/300 シリーズ C/C++コンパイラ パッケージ Ver.5.0.02(Ver.4.0.06)アップデート内容

1 コンパイラ (Ver.4.0B -> Ver.4.0.03)

1.1 C++初期処理/後処理データ領域(C\$INIT),C++仮想関数表領域(C\$VTBL)の出力不正

【内容】

C++コンパイル時に、`abs16` オプションを指定してコンパイルした場合、`C$INIT,C$VTBL` セクションが生成されず、そこに割りつくオブジェクトが`$ABS16C` セクションに割り付いていた問題点を修正しました。

1.2 `abs` オプションによるセクション不正

【内容】

C++コンパイル時に、`abs8` または `abs16` オプションを使用すると、`#pragma abs8 section` または `#pragma abs16 section` で指定しているセクション名の切り替えが無効になる問題点を修正しました。

1.3 メンバ型情報不正

【内容】

構造体のメンバを関数へのポインタでそのポインタを関数のプロトタイプ形式で宣言した場合、デバッグ情報が不正となる問題点を修正しました。

【例】

```
struct sample {
    void (*fp)(int);
};
```

1.4 コンパイラの不正終了

【内容】

C++コンパイル時に、メンバ関数と同名の宣言済み関数をメンバ内で再宣言するとコンパイラが不正終了する問題点を修正しました。

【例】

```
void func();
struct S {
    void func() {
        extern void func();
    }
};
```

1.5 セクション演算子によるオブジェクト不正

【内容】

セクションアドレス演算子の結果をデータ以外の目的で絶対アドレスとして使用する場合、オブジェクト不正となる問題点を修正しました。

【例】

```
void sub(void){
    (*((void (*)(void))__sectop("P"))0);
}
```

1.6 class 内配列情報読み込み不正

【内容】

C++コンパイル時で debug オプションが指定されたとき、構造体/クラスのメンバに static 宣言した配列メンバがあり、そのメンバまたはその構造体/クラスを引数や局所変数として使用した場合、内部エラーが発生する問題点を修正しました。

【例】

```
class B {
    static char EMPTY[];
    char *m;
};
class B a;
void func(class B b) {
    a=b;
}
```

1.7 define オプションの解釈変更

【内容】

オプションで -define=マクロと指定した場合、-define=マクロ=1 と同等の動作を行っていた問題点を修正しました。

1.8 1 ビット演算時のコード不正

【内容】

1 ビットのビットフィールドのみの変数同士で、ビットフィールドの型指定子のサイズが互いに異なる時に、ビット演算(&,|,^,&=,^=)を実施した場合、不正なコードが出力されることがある問題点を修正しました。

【例】

```
struct {
    char  char1 :1;
    int   int1  :1;
} x1,x2,x3;
void sub(){
    x1.int1 |= x2.char1 | x3.char1;
}
```

<アセンブラコード>

```
_sub:
    BLD.B    #7,@_x2:32
    BOR.B    #7,@_x3:32
    BST.B    #0,R0L
    BLD.B    #7,@_x1+2:32    <正しいコード>
    BOR.B    #0,R1H          BOR.B    #0,R0L
    BST.B    #7,@_x1+2:32
    RTS
```

1.9 debug オプション指定によるインターナルエラー発生(Solaris 版のみ)

【内容】

Solaris 版コンパイラにおいて debug オプションを指定してコンパイルを行うと、コンパイラ自身の Solaris 上メモリ割付状況に応じてインターナルエラーが発生する問題点を修正しました。

2 フォーマットコンバータ (Ver.1.0C -> Ver.1.0.04)

2.1 圧縮フォルダ内のファイルの入力

【内容】

圧縮フォルダ内のファイルを入力しようとした場合、ファイルを認識できない(G3002 エラー)問題点を修正しました。

3 最適化リンケージエディタ (Ver.7.1.02 -> Ver7.1.05)

3.1 -compress オプション指定時のデバッグ情報不正

【内容】

-compress オプション指定によりデバッグ情報を圧縮した際、型情報の参照先が不正となる問題点を修正しました。

3.2 binary ファイル入力指定時のセクション属性不正

【内容】

下記条件全てを満たした際に、セクションの属性が不正となる問題点を修正しました。

- (1) object ファイルと binary ファイルを入力
- (2) object ファイル内でサイズ 0 のセクションを定義
- (3) -binary オプションでサイズ 0 のセクションを指定
- (4) ファイルの入力指定順：(2)の object ファイル (3)の binary ファイル

3.3 relocate ファイル生成時の異常終了

【内容】

下記条件全てを満たした際に異常終了する問題点を修正しました。

- (1) 入力 object ファイルの先頭：-goptimize オプション指定ありのファイル
入力 object ファイルの 2 番目以降の入力：-goptimize オプション指定なしのファイル
もしくはアセンブラ出力ファイル
- (2) 出力形式に relocate ファイルを指定
- (3) -profile オプション指定あり
- (4) 最適化指定あり

3.4 -profile オプション指定の最適化で内部エラー(8899,8876,8921)

【内容】

-goptimize オプション指定ありの object ファイルについて、-profile オプションを指定する最適化を行った際に内部エラー(8899,8876,8921)となる問題点を修正しました。

3.5 異なる CPU 種別の混在した relocatable ファイルリンク時の内部エラー(3081)

【内容】

CPU 種別の 2600a と 2600n が混在した Relocate ファイルのリンク時に、-compress,-delete,-rename オプション、もしくは最適化を指定すると内部エラーとなる問題点を修正しました。

3.6 -output オプションでアドレス範囲指定時の内部エラー(7041)

【内容】

-output オプションを指定した際、アドレス範囲で出力指定した場合に内部エラー(7041)となる場合がある問題点を修正しました。

3.7 最適化抑止指定時の不当な最適化

【内容】

下記条件を全て満たした際に、最適化抑止範囲が有効にならない問題点を修正しました。

- (1) -goptimize オプション指定ありのファイルを入力
- (2) optlnk で最適化を指定
- (3) -absolute_forbid オプションを指定
- (4) 最適化抑止範囲を二つ以上記述(二つ目以降の指定が無効となる)

3.8 relocate ファイル生成時の Object 不正

【内容】

下記条件を全て満たした際に不正な Object コードが生成される問題点を修正しました。

- (1) 入力に relocate ファイルを指定
- (2) 出力に relocate ファイルを指定
- (3) -delete,もしくは-rename オプションを指定