

C/C++コンパイラパッケージ M3T-CC32R V.5.00 Release 00 ご使用上のお願い

M32Rファミリ用C/C++コンパイラパッケージ M3T-CC32R V.5.00 Release 00の使用上の注意事項を連絡します。

- リロケータブルロードモジュールの再リンクに関する注意事項

1. 内容

該当製品のリンカlnk32Rまたはコンパイルドライバcc32Rで、-rオプションを使用してリロケータブルロードモジュールファイルを作成し、それを該当製品を使用して再度リンクすると、以下のメッセージをともなうエラーが発生してリンクできません。

```
Ink32R: error: already exist COMMON section.
```

なお、統合開発環境High-performance Embedded Workshopと使用する場合も同じエラーが発生します。

1.1 発生例

%はプロンプトを示します。

(1) lnk32Rを直接起動してリンクする場合のコマンドライン例

```
-----  
% cc32R -c sample11.c  
% cc32R -c -o sample12.mo sample12.c  
% as32R -o sample13.mo sample13.ms  
% lnk32R -r -o group1.rel -l m32RcR.lib sample11.mo : -rを使用  
sample12.mo sample13.mo  
% as32R -o start.mo start.ms  
% lnk32R -o sample.abs -SEC P=10000,C,D,B group1.rel : 再リンク  
start.mo
```

Ink32R: error: already exist COMMON section.

(2) cc32Rを起動してリンクする場合のコマンドライン例

```
% cc32R -c sample11.c
% cc32R -c -o sample12.mo sample12.c
% as32R -o sample13.mo sample13.ms
% cc32R -r -o group1.rel -l m32RcR.lib sample11.mo   : -rを使用
sample12.mo sample13.mo
% as32R -o start.mo start.ms
% cc32R -o sample.abs -SEC P=10000,C,D,B group1.rel : 再リンク
start.mo
```

Ink32R: error: already exist COMMON section.

2. 回避策

リロケータブルロードモジュールファイルを作成する代わりにライブラリ (.lib) を作成し、これを
入力ファイルとして再リンクしてください。

2.1 発生例(1)「Ink32Rを直接起動してリンクする場合」の回避例

lib32Rコマンドを使用してライブラリを作成する。

変更前 : Ink32R -r -o 出力ファイル 入力ファイル. . .

変更後 : lib32R -c 出力ファイル 入力ファイル. . .

```
% cc32R -c sample11.c
% cc32R -c -o sample12.mo sample12.c
% as32R -o sample13.mo sample13.ms
% lib32R -c group1.lib sample11.mo sample12.mo sample13.mo
      : lib32R -cに変更して.libを作成 (注)
% as32R -o start.mo start.ms
% Ink32R -o sample.abs -l m32RcR.lib -SEC P=10000,C,D,B
group1.lib start.mo      : .libを再リンク
```

注 : 必須ではありませんが、出力ファイル名の拡張子は.libに変更することをお勧めしま
す。(例では group1.rel → group1.lib に変更)

2.2 発生例(2)「cc32Rを起動してリンクする場合」の回避例

cc32Rコマンドに-mklib=cオプションを付けてライブラリを作成する。

変更前： cc32R -r -o 出力ファイル 入力ファイル. . .

変更後： cc32R -mklib=c -o 出力ファイル 入力ファイル. . .

また、cc32Rコマンドに-cオプションを使用して入力ファイルを生成している場合は、変更後も-cオプションを使用する。cc32Rまたはas32Rコマンドに -o オプションを用いて出力ファイル名を決定している場合は、拡張子を .ml にする。

```
-----  
% cc32R -mklib -c sample11.c  
% cc32R -mklib -c -o sample12.ml sample12.c  
% as32R -o sample13.ml sample13.ms      : 拡張子を.mlにする  
% cc32R -mklib=c -o group1.lib sample11.ml : -mklib=cに変更 (注)  
sample12.ml sample13.ml              : 拡張子を.mlにする  
% as32R -o start.mo start.ms  
% lnk32R -o sample.abs -l m32RcR.lib -SEC P=10000,C,D,B  
group1.lib start.mo                  : .libを再リンク  
-----
```

注: 必須ではありませんが、出力ファイル名の拡張子は.libに変更することをお勧めします。
(例では group1.rel → group1.lib に変更)

2.3 High-performance Embedded Workshopを使用している場合

リンカオプションに -rを選択してリロケータブルロードモジュールを作成している場合は、ライブラリを作成するよう以下のとおり変更ください。

- (1) リロケータブルロードモジュールを作成するプロジェクトを開く。
- (2) High-performance Embedded Workshopのメニュー「ビルド」→「Renesas M32R Standard Toolchain」を選択してRenesas M32R Standard Toolchainオプションダイアログボックスを開く。
- (3) 「Link」タブを選択する。
- (4) 「Category」リストボックスの「Input」を選択し、[-L][-I]行の下にある、標準ライブラリファイル名を控えておく(再リンクの際にこのファイル名を指定するため)。
- (5) 「Category」リストボックスの「Section」を選択する。
- (6) -r オプションのチェックをはずす。
- (7) 「Category」リストボックスの「Message」を選択する。

- (8) 中段の「User defined options」テキストボックスに、以下の文字列を追加する。
-.ml=.mo -mklib=c
- (9) OKボタンを押し、ビルドを行う。

回避策を実施する際の注意：

- 変更の対象となるlnk32Rまたはcc32Rコマンドで-e, -lまたは -L オプションを使用している場合は、再リンクの際にも使用してください。
- ライブラリはリンクする他のオブジェクトから使われていない場合、ライブラリ中のオブジェクトモジュールの一部をリンクしないことがあります。よって、今回の対策でライブラリ化したファイルについて必ずリンクする必要がある場合は再リンク時に改めて入力ファイルとして指定してください。

3. 恒久対策

次期バージョンで改修する予定です。

[免責事項]

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。ニュース本文中のURLを予告なしに変更または中止することがありますので、あらかじめご承知ください。