

統合環境HEW (Ver 1. 2) でH-debugger 用に

新ワークスペースおよび新プロジェクト名を登録する方法 (H8S版)

ルネサスC言語用統合環境“HEW Ver1. 2 (release 9)”にH-debugger用の新ワークスペース/プロジェクト名を登録する手順方法を説明します。

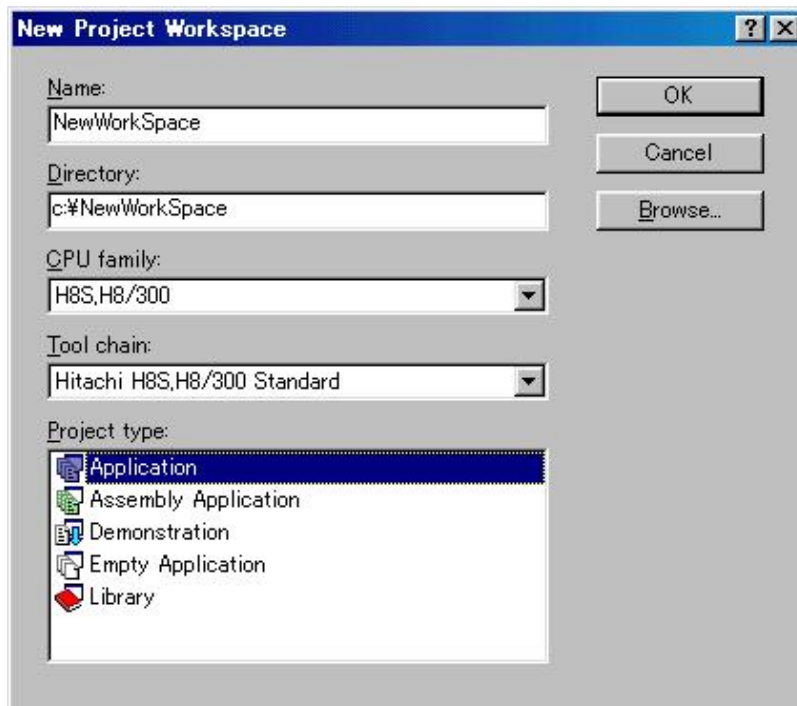
説明を明確にするために、名前等を仮に決めて例に沿って説明を進めたいと思います。

ワークスペース名： NewWorkSpace
プロジェクト名： Project1
登録モジュール名： StartupA.asm [ASM ファイル バックテール+スタートアップルチン]
Cat261p2.c [C ファイル メインモジュール]
CatPio2.c [C ファイル サブモジュール]
CPUタイプ： H8S/2612

1. 新ワークスペースの登録方法

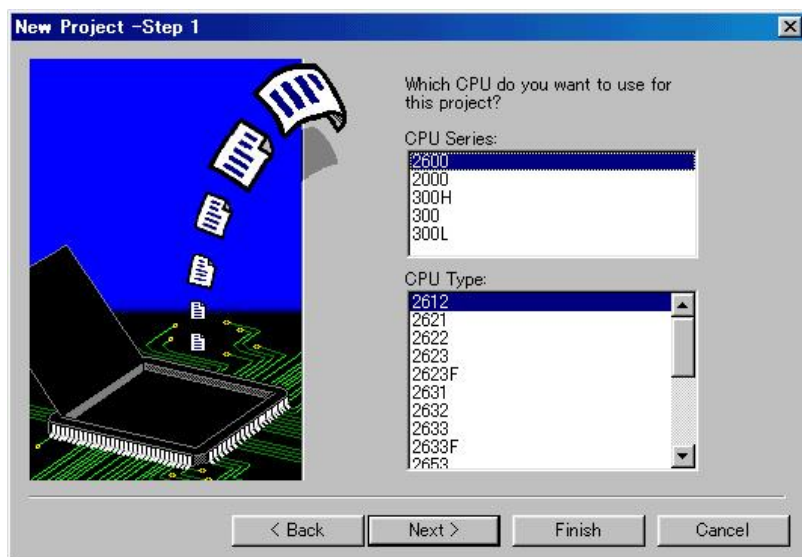
“HEW” 起動させます。

“Creat a new project workspace”をチェックしての **OK** か、もしくは **Cancel** 後に、[Fileメニュー] - [New Workspace] をクリックします。



- 1) Name : "NewWorkSpace"
- 2) Directory : "c:\¥NewWorkSpace"
- 3) Cpu family : "H8S, H8/300"
- 4) Tool chain : "HitachiH8S, H8/300Standard"
- 5) ProjectType : Application

この項目を確認後、**OK**をクリックして下さい。



- 1) CPU Series "2600" Cpu Type "2612"

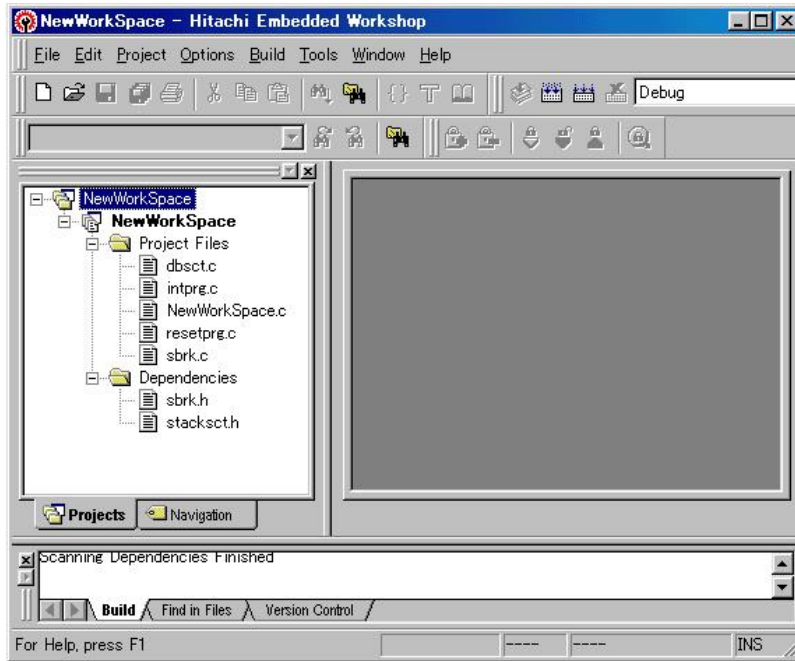
ここで、**Finish**をクリックします。

HEWは、新規ワークスペース登録時にワークスペース名とプロジェクト名を同じ名前で作成してしまいます。

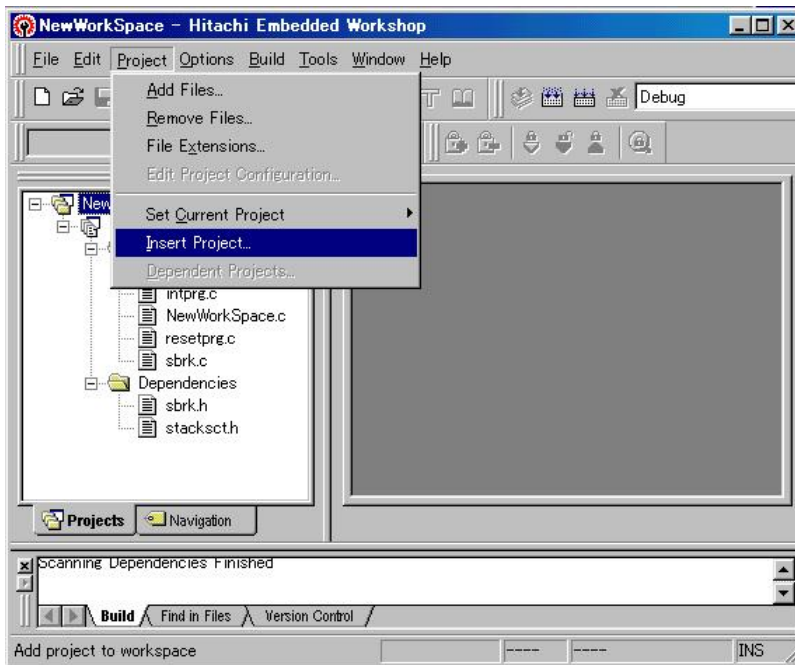
これでは、本来の目的を達成することができませんので、あえてここで **finish** を選択しました。

プロジェクト名の変更方法は、後で説明します。

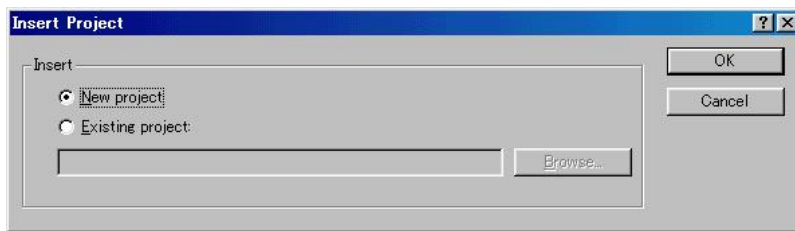
2. 希望プロジェクト名を登録する



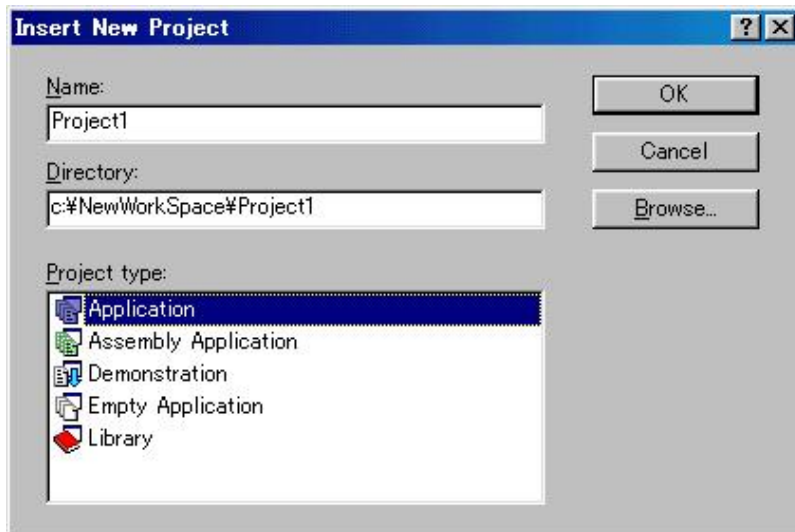
この段階では、プロジェクト名は“NewWorkSpace”です。
これを希望プロジェクト名“Project1”に変更する方法を説明します。



1) [Project] - [Insert Project] をクリックします。



2) New Project をチェック後、OK をクリック

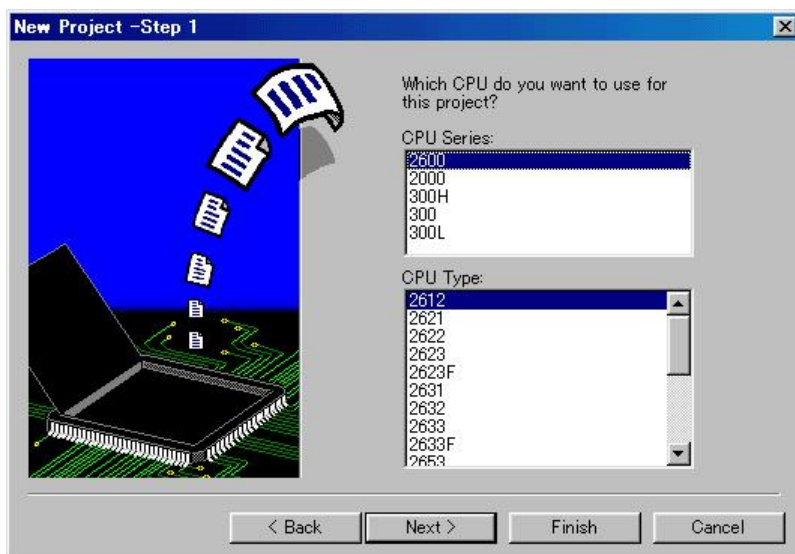


3) Name : "Project1"

4) Directory : "c:\NewWorkSpace\Project1"

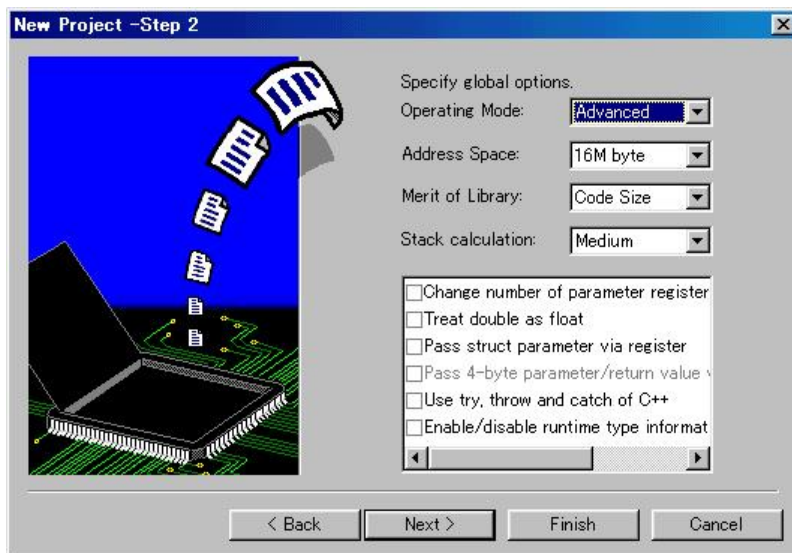
5) ProjectType: Application

6) を確認後、OK をクリックして下さい。

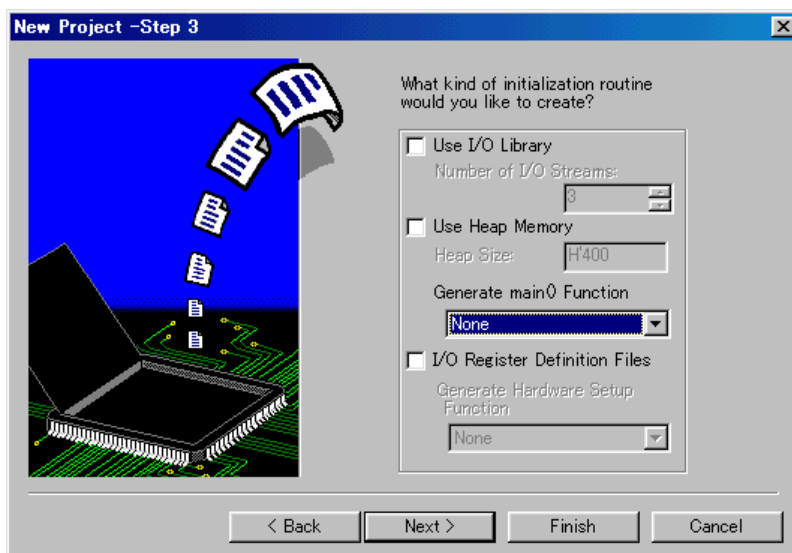


7) CPU Series "2600" Cpu Type "2612"

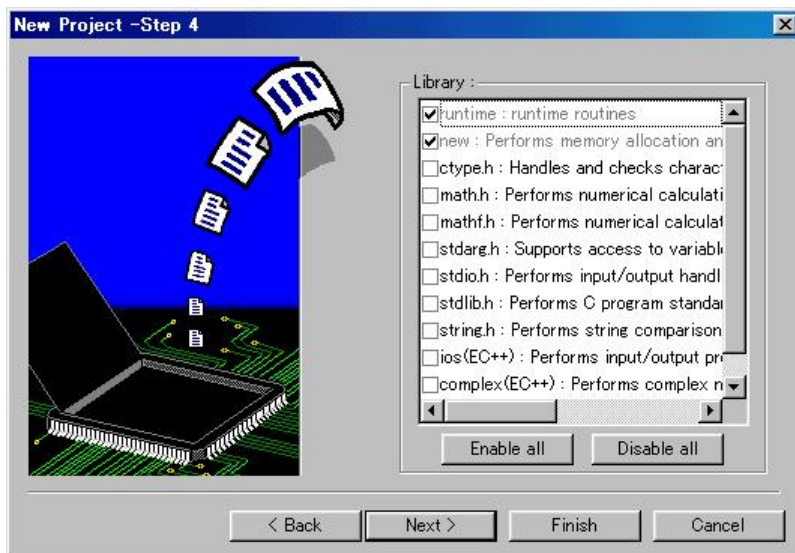
8) 確認後、Next をクリックして下さい。



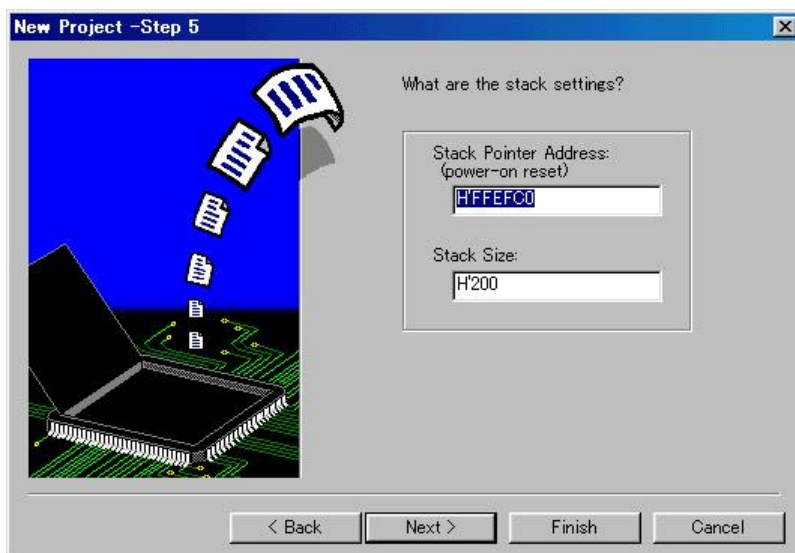
9) CPUスペックを確認後、**N e x t**をクリックして下さい。



- 1 0) この例では、HEWの用意したROM化支援ソフトは、一切使用しませんので全てのチェックを外して下さい。
- 1 1) `Generate main () Function`を“None”に選択する。
- 1 2) 確認後、**N e x t**をクリックして下さい。



1 3) C言語ライブラリの選択です。この例ではライブラリは使用しませんので、このまま、**N e x t**をクリックして下さい。



1 4) スタックボトム位置の設定ですが、この情報も使用しません。このまま、**N e x t**をクリックして下さい。



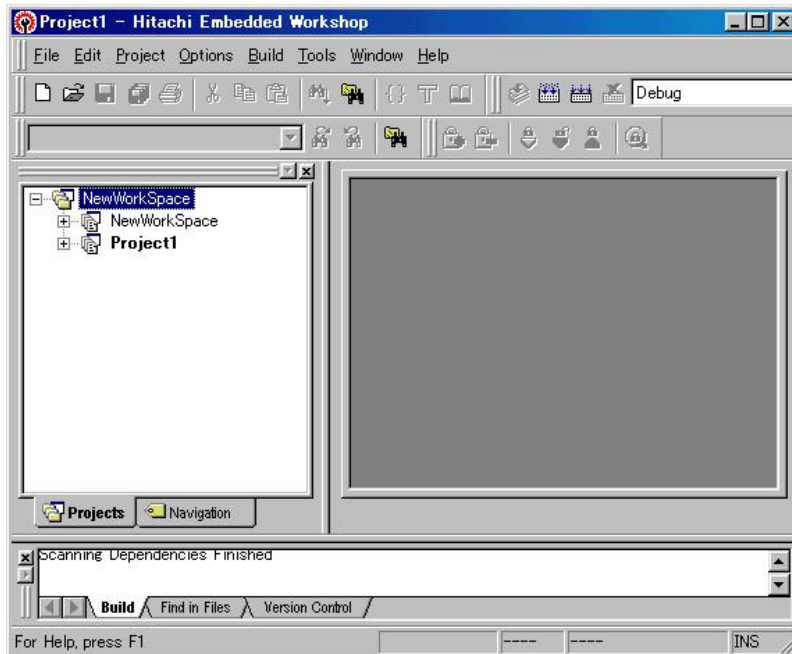
1 5) ベクターテーブル設定ファイルの設定ですが、この情報も使用しませんので、チェックを外して下さい。

1 6) この状態で、**N e x t**をクリックして下さい。



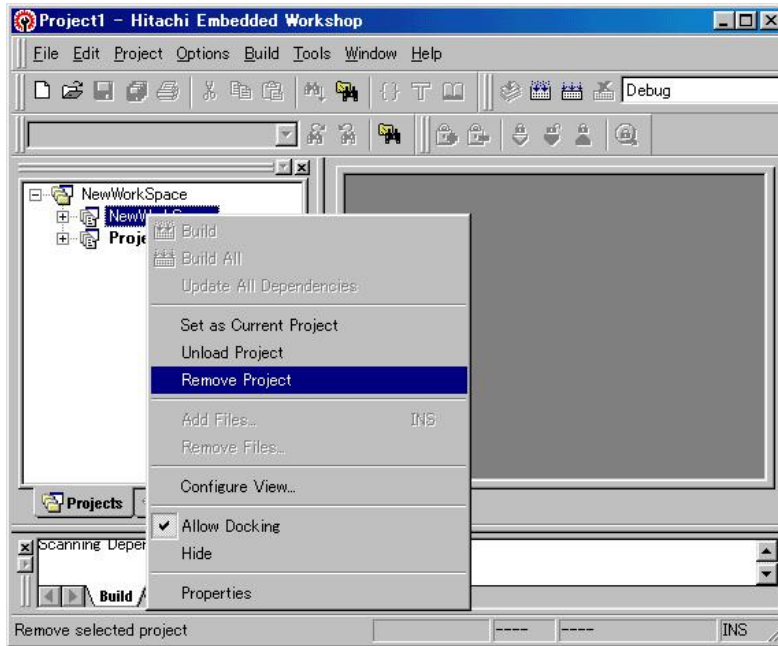
1 7) ここで最終になります。この画面での説明はセグメントを定義するためのCソースを表示していますが、このソースも使用しませんので、後でプロジェクトから **R e m o v e** をします。

1 8) この状態で、**F i n i s h**をクリックして下さい。



これまでの操作で、希望プロジェクト名“Project1”が作成できました。
今度は、“NewWorkSpace”というプロジェクト名が必要ありませんので、このプロジェクトをワークスペースからRemove（外す）する方法を説明します。

3. ワークスペースから不用プロジェクトをRemove (外す) する方法

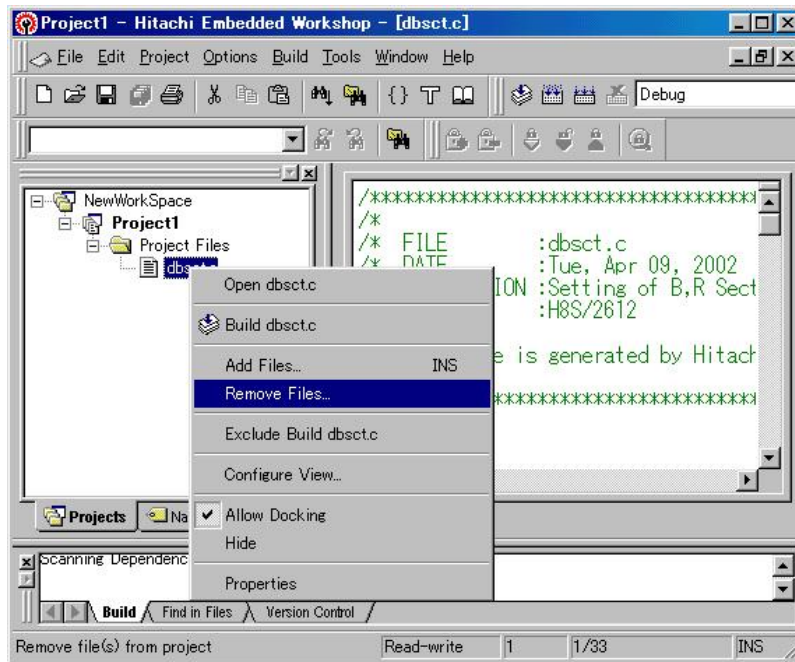


- 1) Removeしたいプロジェクト名“NewWorkSpace”にマウスポインタを置き、左ワンクリック後、右クリックをしますと、上記のようなプルダウンメニューが表示されますので、ここで“Remove Project”を選択して下さい。
この操作により、Removeすることが出来ます。

[注意 登録プロジェクトが2つ以上無いとこの操作はできません]

次にプロジェクト名“Project1”に希望モジュール(ソースファイル)を登録する前に、不用ソースファイルを削除する方法を説明します。

4. 不要なソースファイルを削除する方法

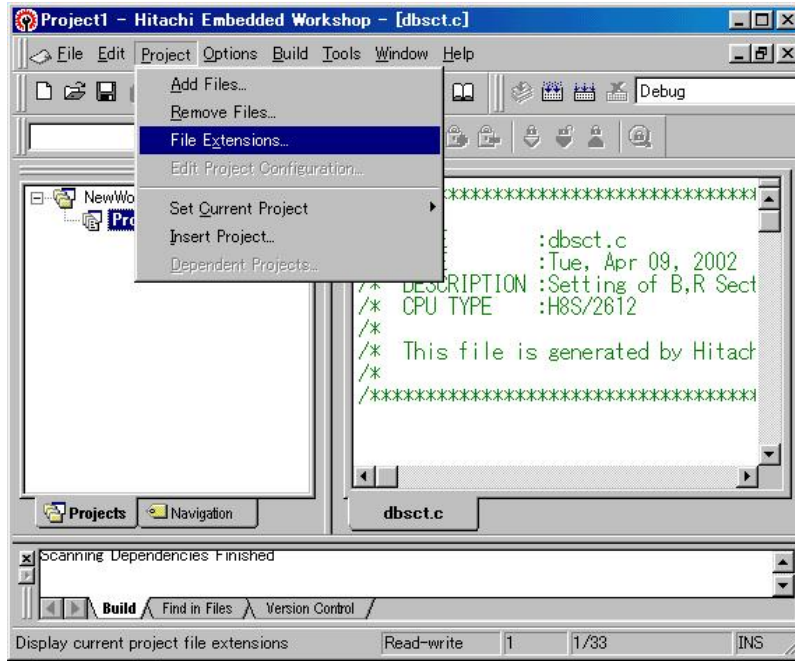


- 1) Removeしたいファイル“dbstc.c”にマウスポインタを置き、左ワンクリック後、右クリックをしますと、上記のようなプルダウンメニューが表示されますので、ここで“Remove Files”を選択して下さい。

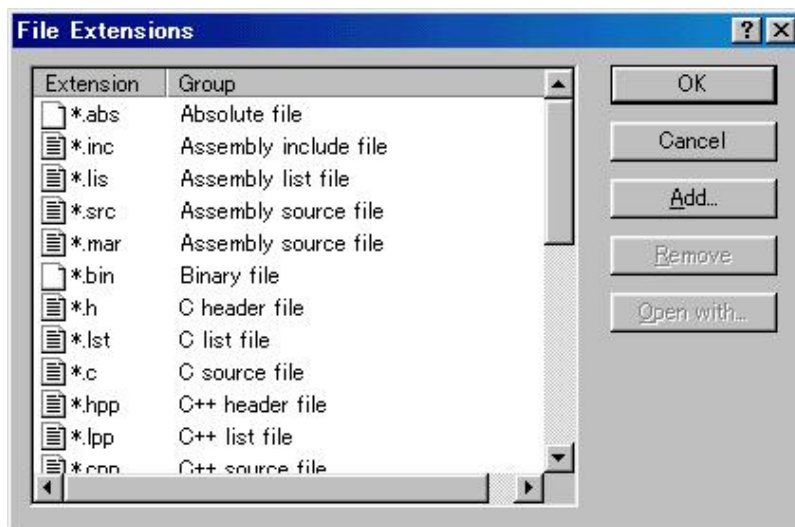
この操作により、Removeすることが出来ます。

次にプロジェクト名“Project1”に希望モジュール（ソースファイル）を登録するために、HEWに拡張子を登録する方法を説明します。

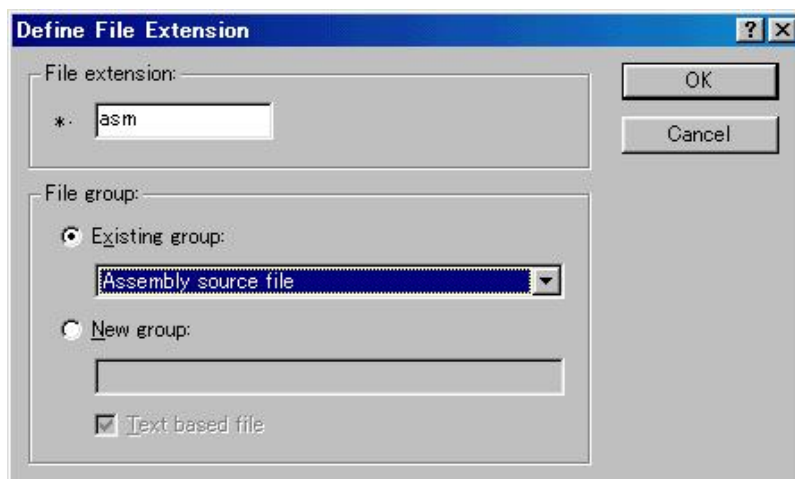
5. 拡張子を登録する方法



1) [Project] - [File Extensions] をクリックします。



2) **Add** をクリックします



- 3) File extensionを“asm”
- 4) Existing groupを“Assembly source file”
- 5) OKをクリックします

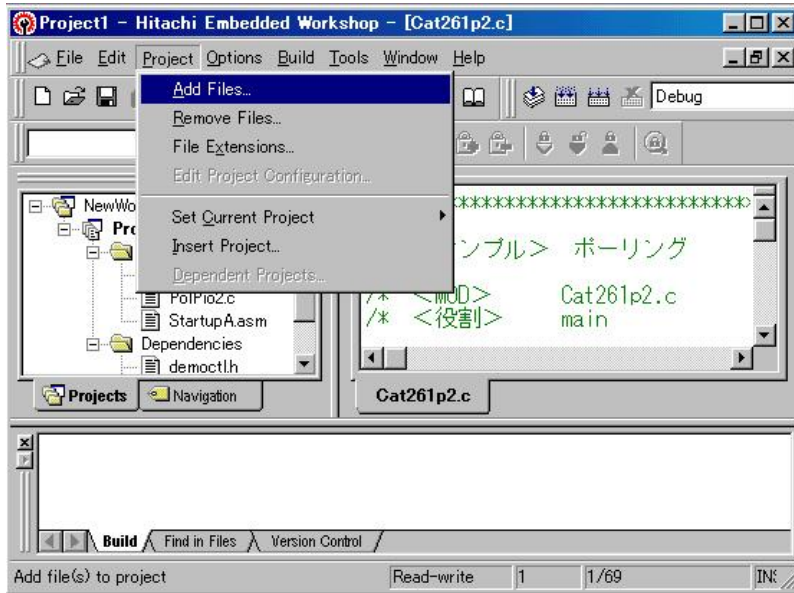
この操作により、プロジェクトに拡張子“*.asm”の登録が完了しました。

いよいよプロジェクト名“Project1”に希望モジュール（ソースファイル）を登録します。

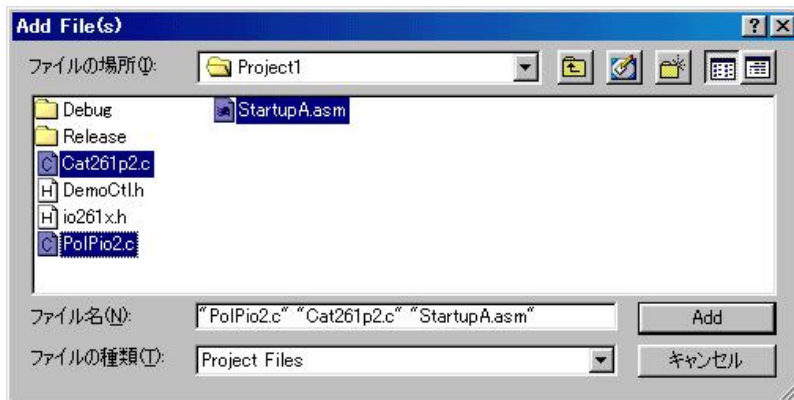
6. プロジェクトに希望モジュール（ソースファイル）を登録する方法

1) 作成済みのソースファイルを“c : ¥NewWorkSpace¥Project1”にコピーして下さい。

- (1) StartupA. asm “¥AHE261_HEW¥S1_Polling¥N02_HC_StartUp”
- (2) Cat261p2. c “
- (3) PolPio2. c “
- (4) io261x. h “
- (5) DemoCtl. h “



2) [Project] – [Add Files] をクリックします。



3) 上記3ファイルを指定して下さい

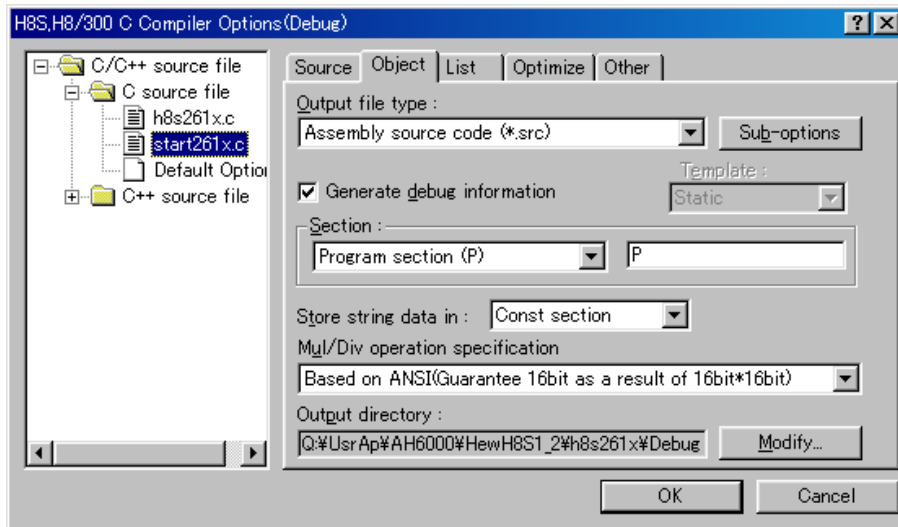
“StartupA. asm” “Cat261p2. c” “PolPio2. c”

4) 選択後、Addをクリックします。

この操作により、プロジェクトにモジュールの登録が完了しました。
次に、ツール関係の設定を説明します。

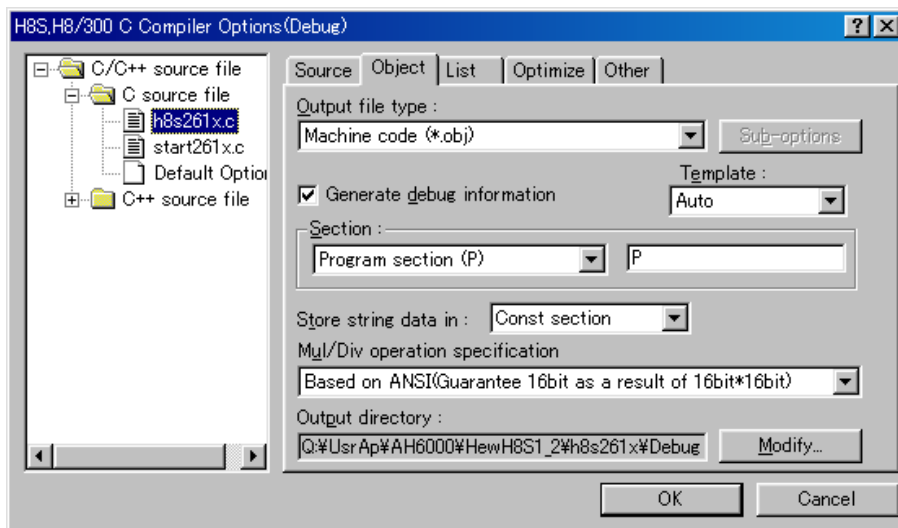
7. Cコンパイラとアセンブラの設定

1) Cソース内に「#pragma asm」のインラインアセンブラ記述がある場合



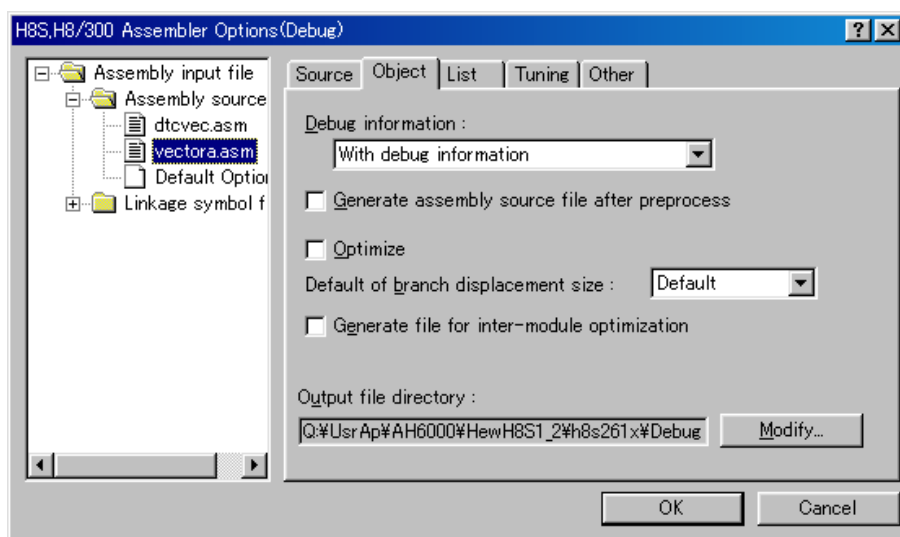
- Object タグの Output file type を「Assembly source code (*.src)」に設定する。
- この設定にしますとシンボル情報のタイプは全てラベルになります。

2) 全てC言語記述の場合



- Object タグの Output file type を「Machine code (*.obj)」に設定する。
- この設定にしますと正確なシンボル情報を得ることになります。

3) 全てアセンブラ記述の場合



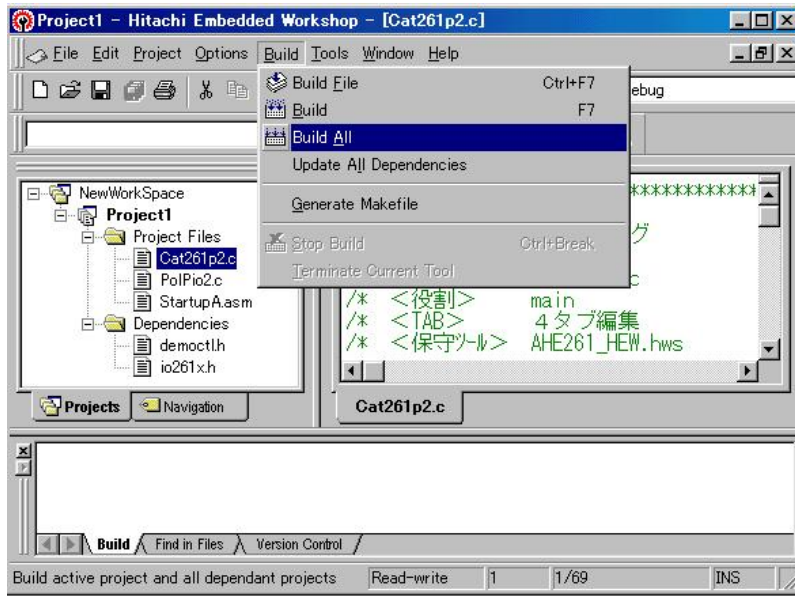
- Object タブの Debug information を「With debug information」に設定する。

8. ツール（ライブラリ）の設定

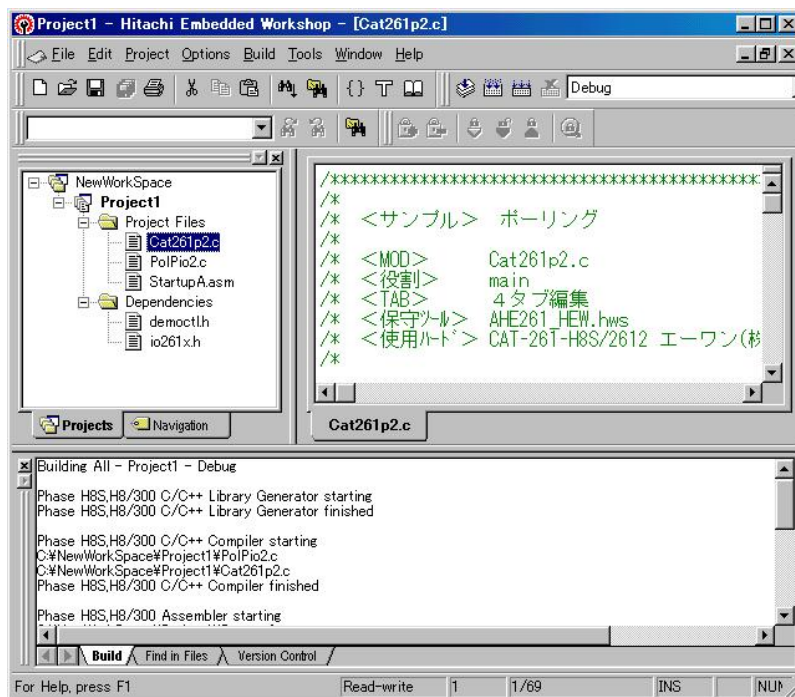
HEWは、プロジェクトごとにC言語用ライブラリを作成する仕様になっています。

ツール設定をデフォルトのまま使用しますと“Build”のたびにライブラリの作成を始めてしまいます。

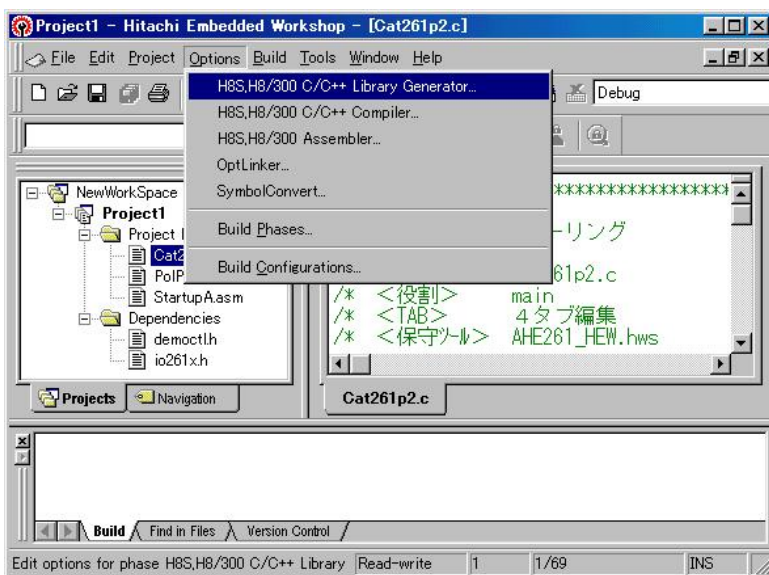
これでは、時間が掛かり過ぎますので、ライブラリ作成は1回のみにして、その後はそのオブジェクトのみを利用する方法を説明します。



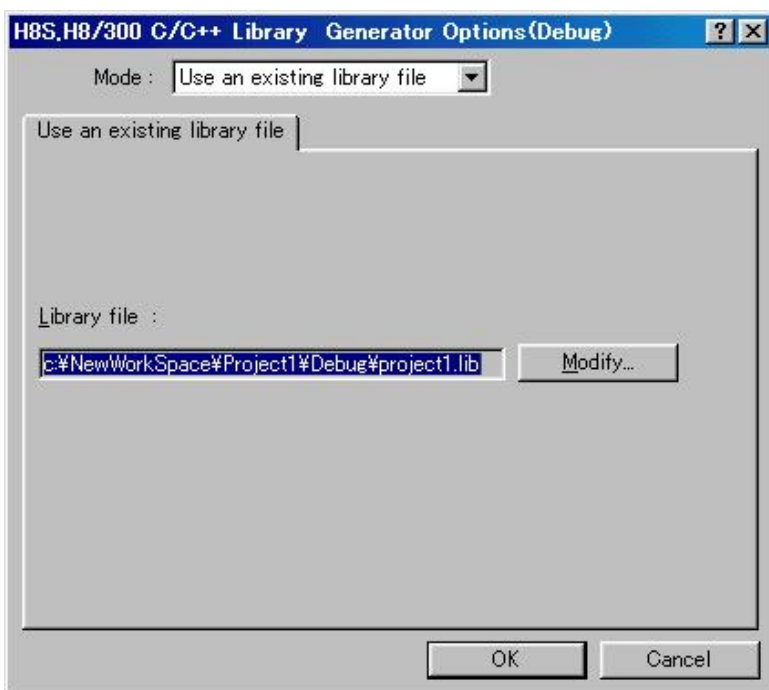
1) [Build] - [Build All] をクリックします。



2) ライブラリの作成を始めます。終了後プロジェクトのファイルをコンパイルしてリンクまで実行してまいります。この時点ではまだエラーもしくはワーニングがでますが無視してください。



3) [Option] - [...Library Generator] をクリックします。

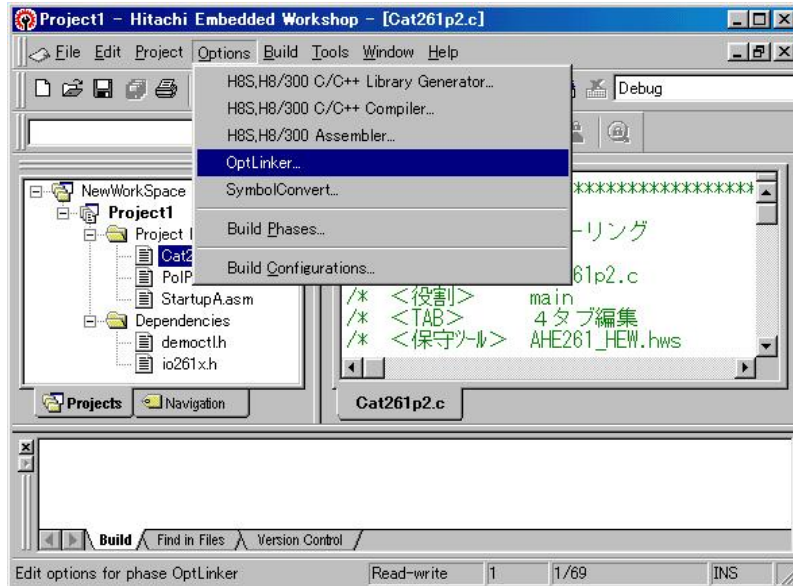


4) Modeを“Use an existing library file” にします。
この指定によりライブラリを毎回作成するのをやめて、リンク時に先ほど作成したライブラリを使用することになります。

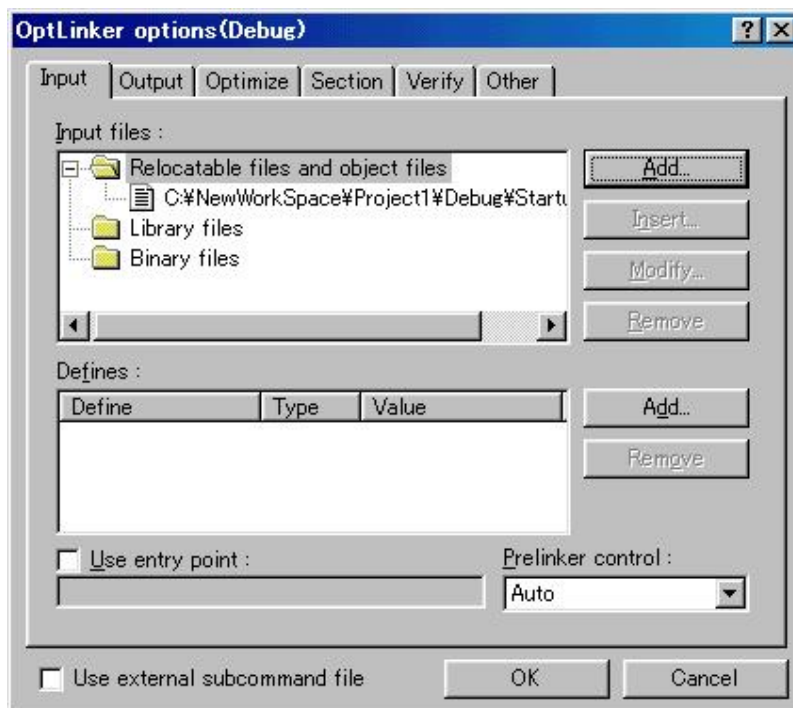
OKをクリックする

次に、リンクの設定を説明します。

9. ツール (リンク) の設定

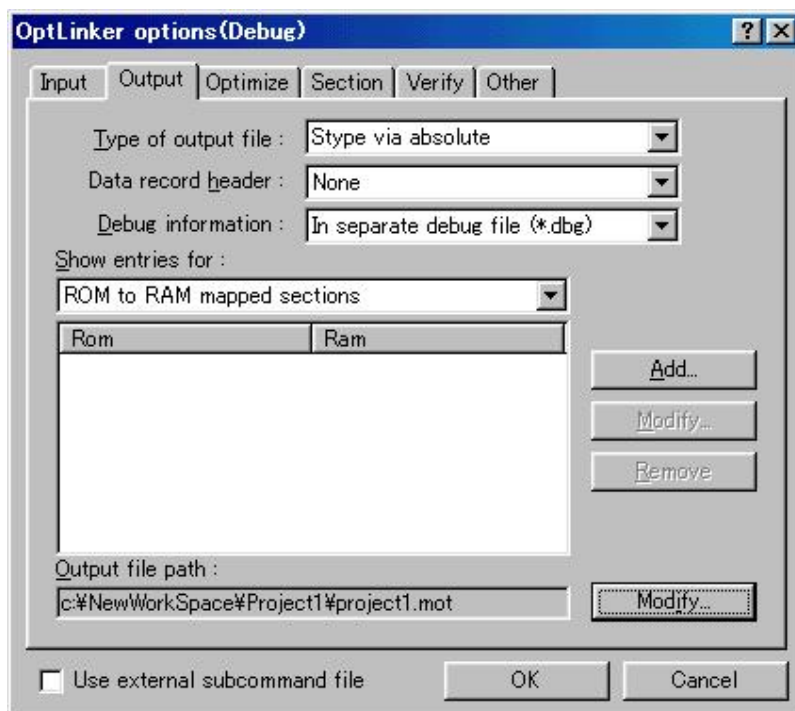


1) [Options] - [OptLinker] をクリックします。

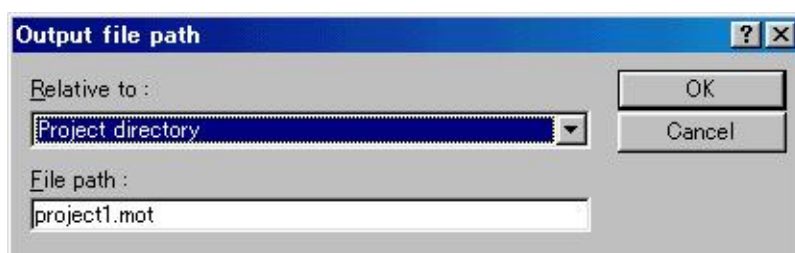


2) Inputタグ部 “Relocatable file and object file” 部に “c : ¥NewWorkspace¥Project1¥Debug¥StartupA. obi” をAddします。

この指定は、“このリスト順にモジュールをリンクしなさい”との指示になります。
(DEFのCViewにてアドレス順に他モジュールを表示させたい場合は、追加設定してください。)

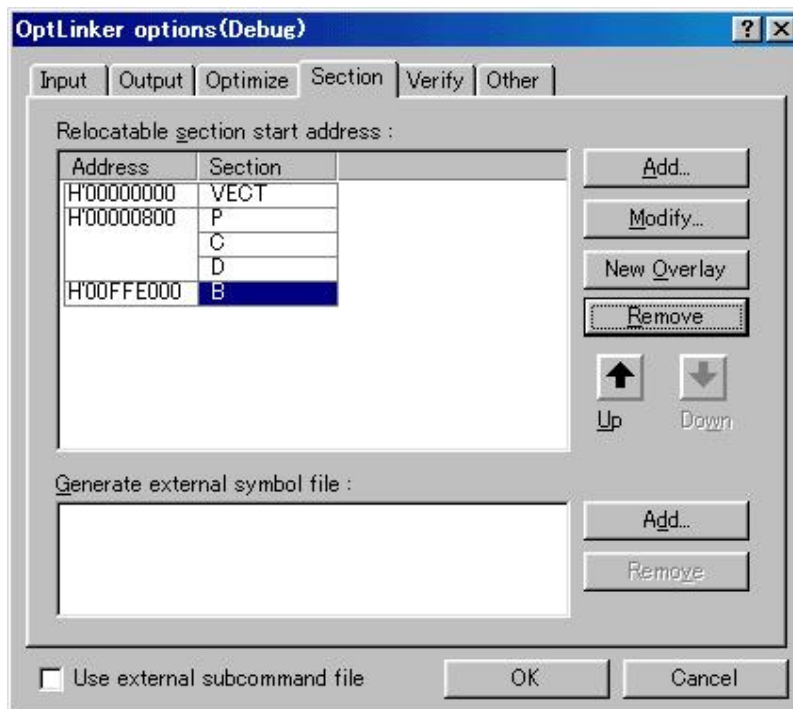


- 3) Outputタグ部をクリックする。
- 4) "Type of output file"が、"Stype via absolute"になっていることを確認します。
- 5) H-debuggerでシンボリックデバックをするためには、
Debug informationを "In separate debug file (*.dbg)" の指定が必要です。
- 6) ROM to RAM mapped section のセクション名をRemoveして下さい。
- 7) **Modify**をクリックする。



Relative to部を "Project directory" の指定にします。
これは、HEXファイルおよび、デバック情報ファイルをCソースファイルのある同じディレクトリに置くための指定です。(重要)

OKをクリックする



8) Sectionタグ部をクリックする。

9) 上記図のようにセクション指定をする。

H '00000000 VECT [StartupA. asmで作成]

H '00000800 P

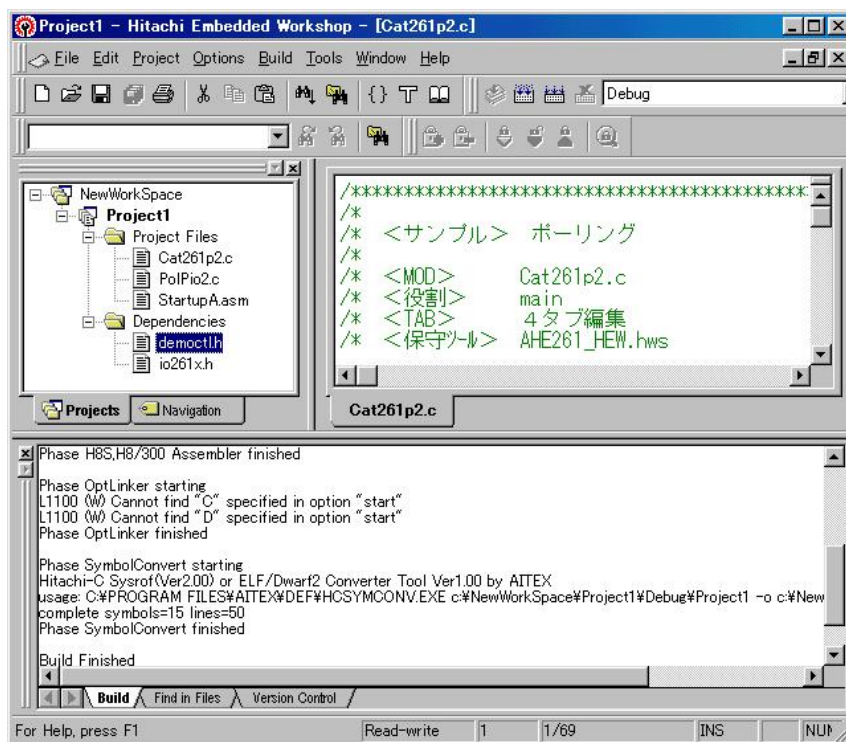
C

D

H '00FFE000 B

10) **OK**をクリックする

11) ここで、もう一度 [Build] - [Build All] をクリックする。

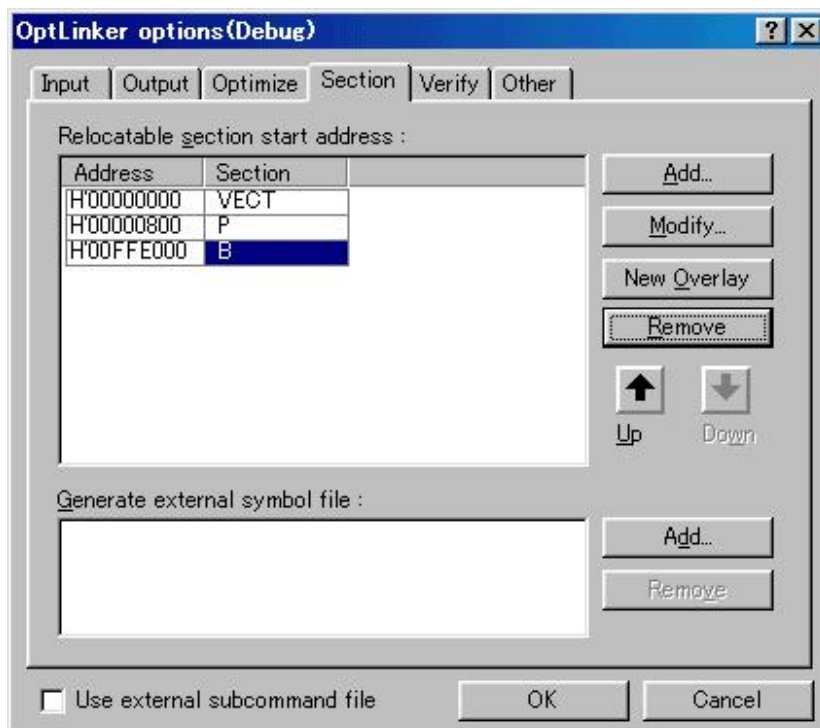


1 2) この状態での Build 結果は、上記図のようなワーニングが表示されます。

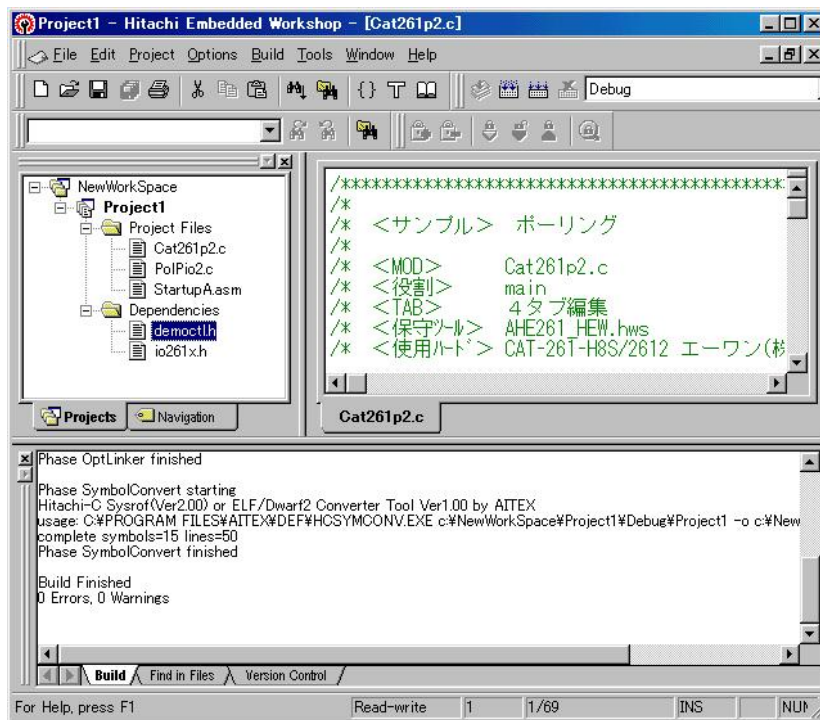
“... ‘C’ specified in option ” start “

“... ‘D’ specified in option ” start “

これは、‘C’ と ‘D’ セクションが使われていないとのメッセージですので無視しても良いのですが、気になる方は次のように削除してください。



1 3) [Options] - [OptLinker] の Section タグ部で ‘C’ と ‘D’ を Remove をして下さい。



14) ここで、最後に [Build] - [Build All] をクリックして見て下さい。

これで、Build結果がエラー無しで終了となりました。

なお、シンボルコンバータの登録方法は、別資料の“シンボルコンバータ (HEW)”をご参照下さい。

—以上—