

統合環境HEW (Ver 2. 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 Ver 2. 2 (release15)”でH-debugger 用に
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。

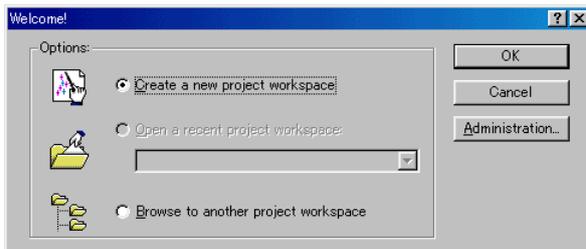
ワークスペース名 : NewWorkSpace
プロジェクト名 : Project1
登録モジュール名 : StartupA.asm ASMファイル ベクタブル・スタートアップルチン
 Cat261p2.c Cファイル メインモジュール
 PolPio2.c Cファイル サブモジュール
CPUタイプ : H8S/2612

※HEW Ver2.1(release10)の場合も、画面表示に多少違いがありますが同様の操作で実行出来ます。

※この説明書では、一部HEW Ver2.1(release10)画面が使用されています。

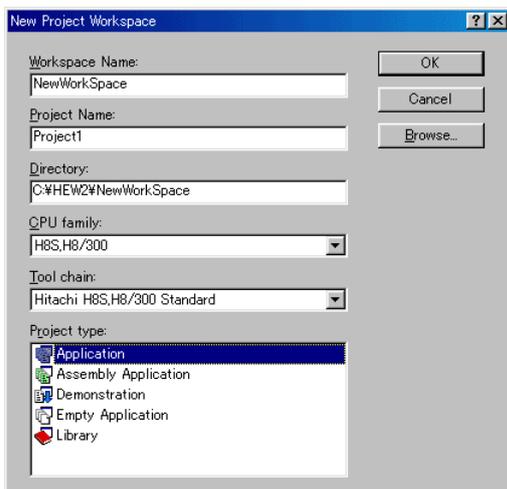
1. 新ワークスペースの登録方法

“HEW” 起動させます。



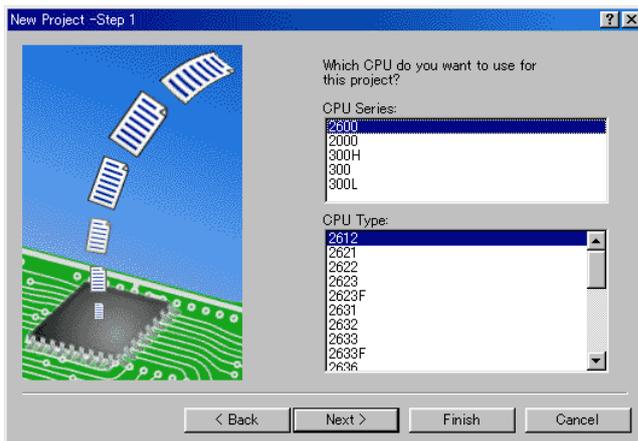
“Creat a new project workspace” をチェックしてのOKをクリックする。

もしくは、Cancel後に、[File]-[New Work Space]をクリックします。



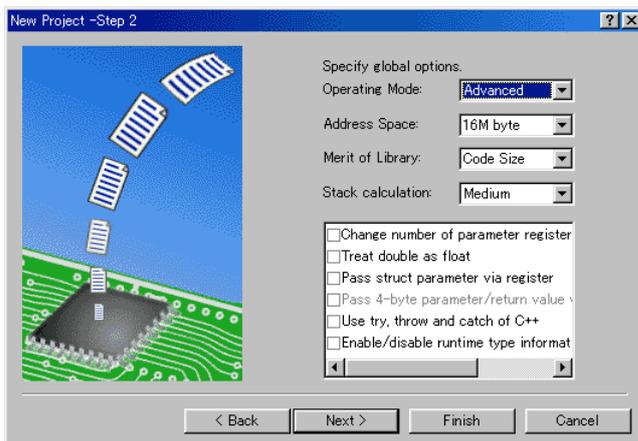
Workspace Name “NewWorkSpace”
Project Name “Project1”
Directory “C:\HEW2\NewWorkSpace”
CPU family “H8S, H8/3000”
Tool chain “Hitachi H8S, H8/300
 Standard”
Project Type Application

この項目を確認後、OKをクリックして下さい。

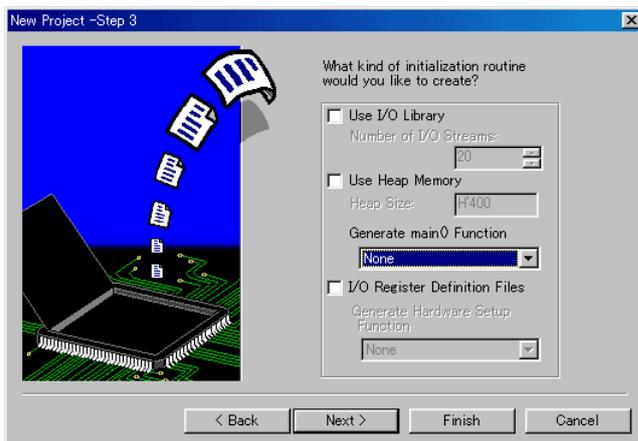


CPU Series を “2600” に選択する。
CPU Type を “2612” に選択する。

確認後、**N e x t**をクリックします。



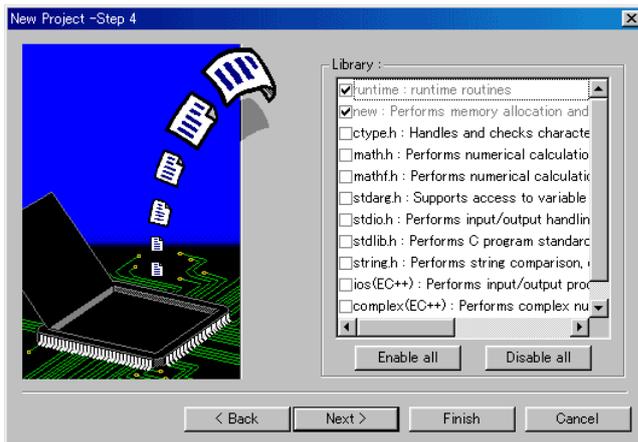
CPUスペックを確認後、**N e x t**をクリックして下さい。



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。

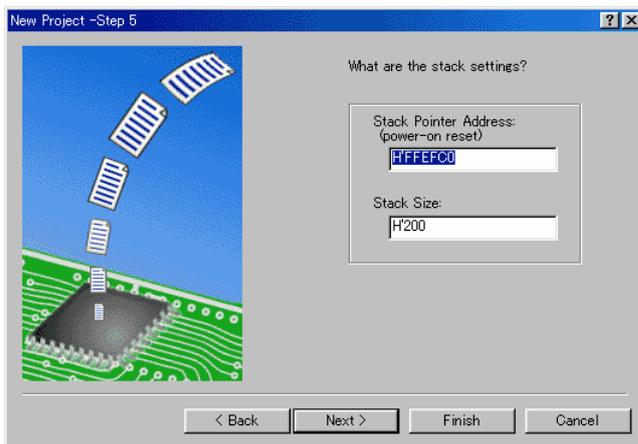
Generate main() Function を”None” に
選択する。

確認後、**N e x t**をクリックして下さい。



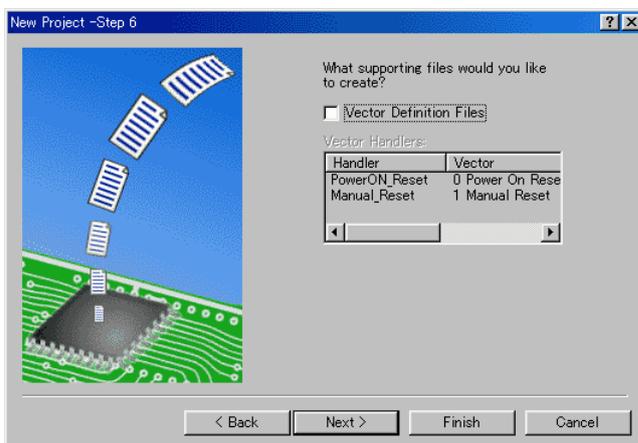
C言語ライブラリの選択です。この例では、ライブラリを使用しません。

このまま、**N e x t**をクリックして下さい。



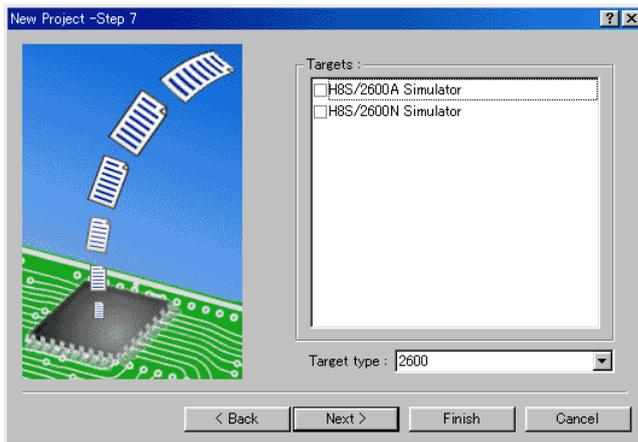
スタックボトムの設定ですが、この情報も使用しません。

このまま、**N e x t**をクリックして下さい。



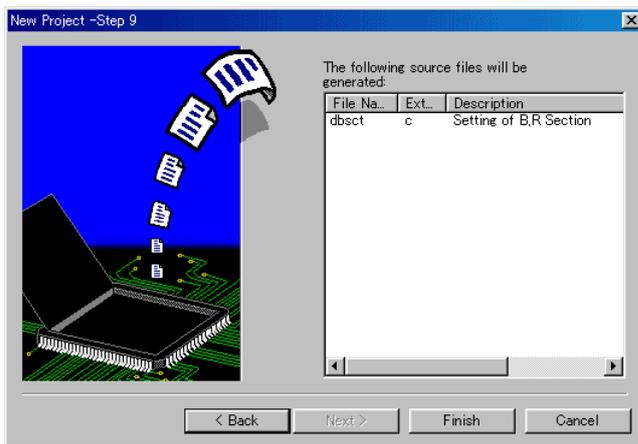
ベクターテーブル設定ファイルの設定ですが、この情報も使用しませんのでチェックを外してください。

確認後、**N e x t**をクリックして下さい。



シミュレータの設定ですが、この情報も使用しません。

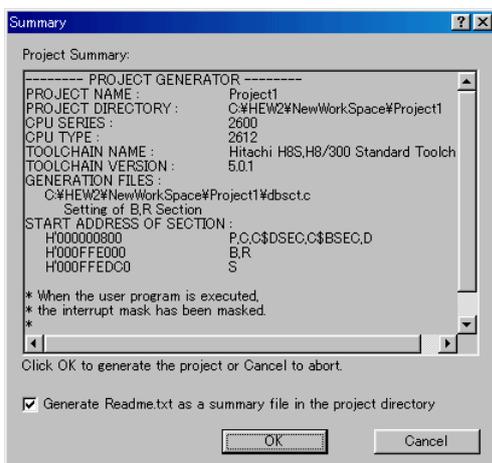
このまま、**Next**をクリックして下さい。



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Removeします。

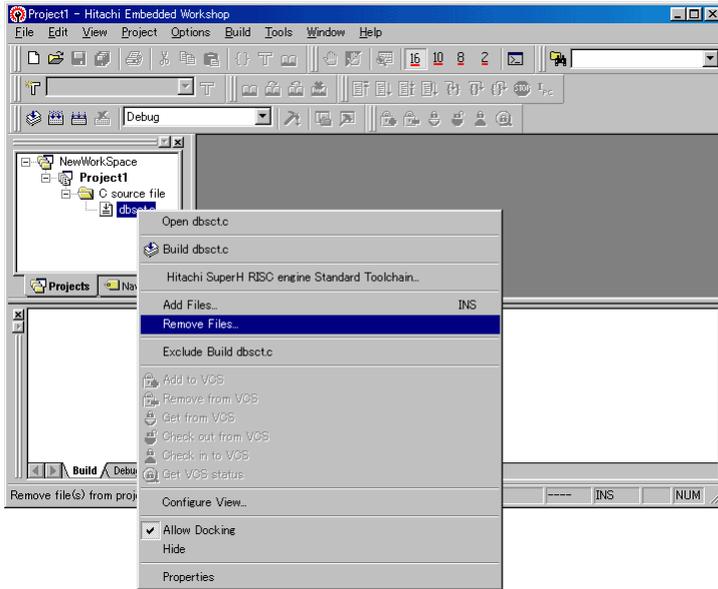
この状態で、**Finish**をクリックして下さい。



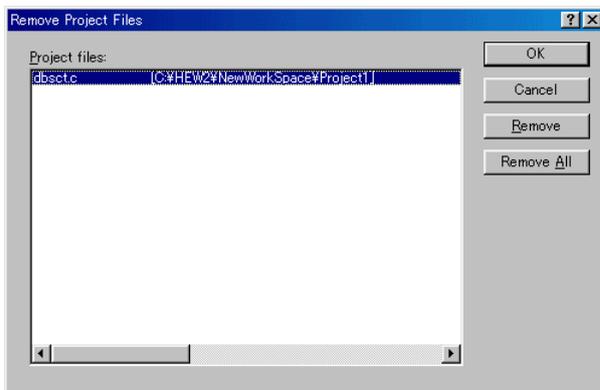
確認画面が表示されますので、OKをクリックして終了です。

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）を登録する前に、不用ソースファイルを削除する方法を説明します。

2. 不要なソースファイルを削除する方法



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。



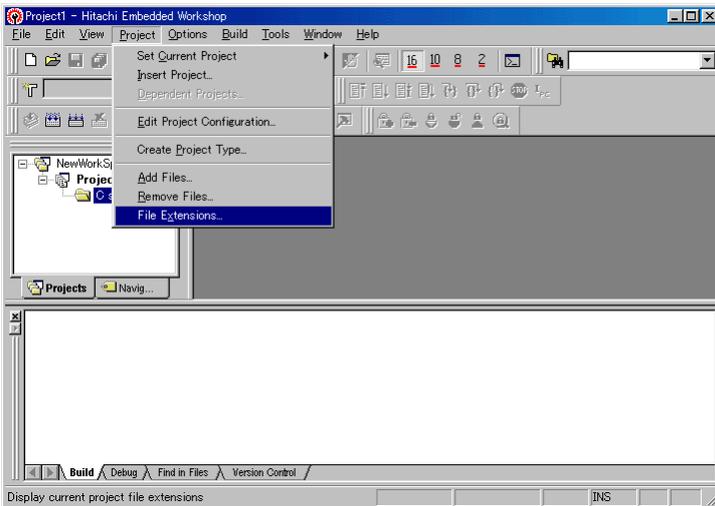
” dbstc.c” をクリックして選択すると” Remove” ボタンが使用できるようになります。

R e m o v e のクリックで、” dbstc.c” を Remove することが出来ます。

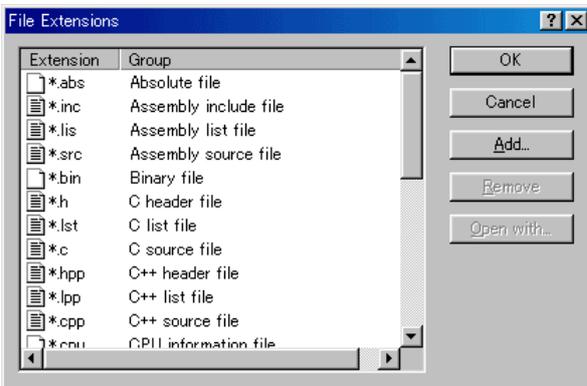
Remove したら **O K** で終了です。

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。

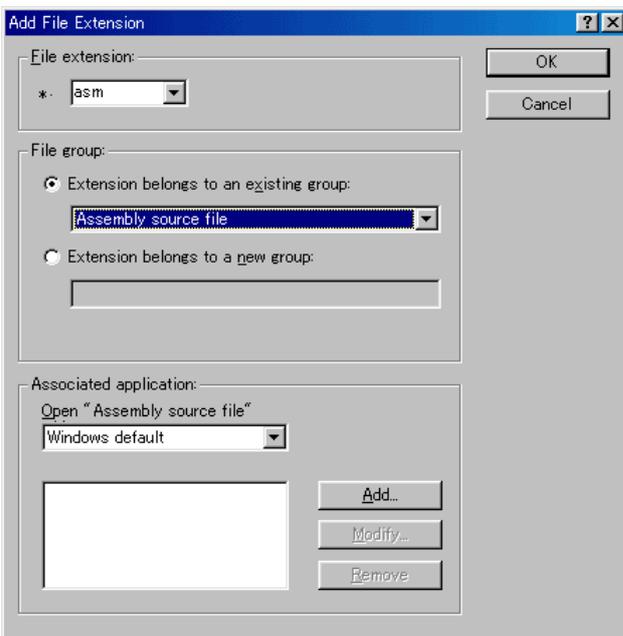
3. 拡張子を登録する方法



[Project]-[File Extension]をクリックします。



Addをクリックします。



File extensionの”asm”を入力する。

Existing groupの”assembly source file”を選択する。

OKをクリックします。

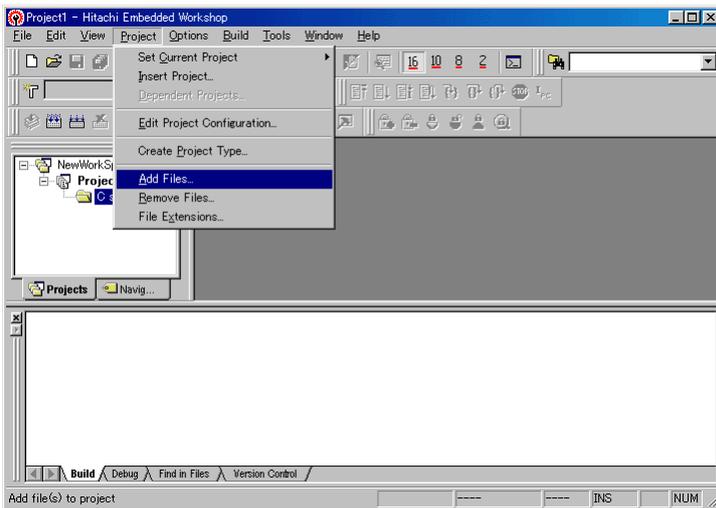
この操作により、プロジェクトに拡張子”*.asm”の登録が完了しました。

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。

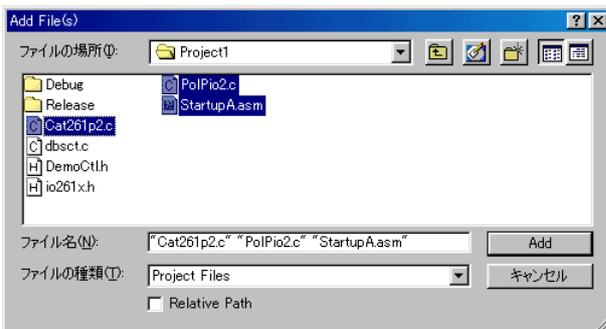
4. プロジェクトに希望モジュール（ソースファイル）を登録する方法

準備： 作成済みのソースファイルを” C:\Hew2\NewWorkSpace\Project1” にコピーして下さい。

StartupA.asm ” ¥ AHE261_HEW¥S1_Polling¥N02_HC_StartUp¥Cat261p2”
Cat261p2.c (左記ソースは、製品CDの上記フォルダに入っています。)
PolPio2.c
io261x.h
DemoCt1.h



[Project]-[Add Files]をクリック
します。



下記3ファイルを指定して下さい。

StartupA.asm
Cat261p2.c
PolPio2.c

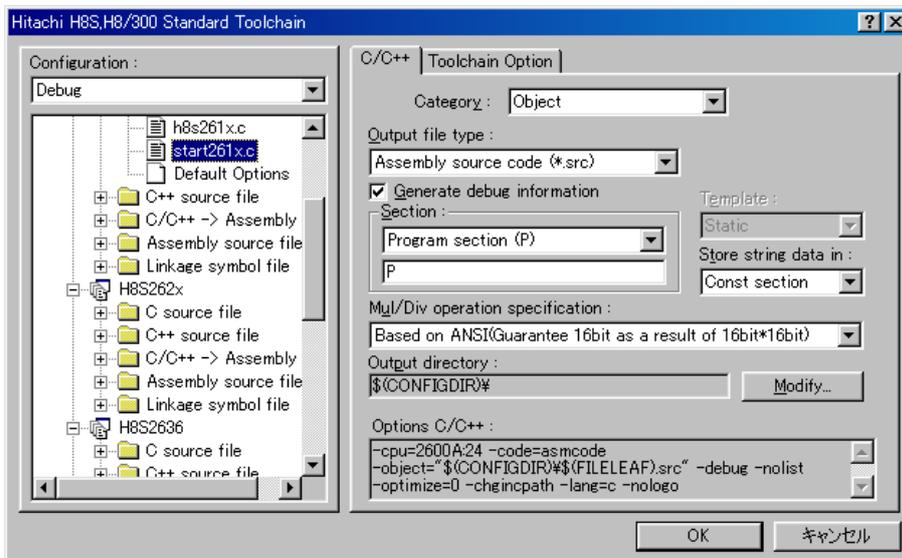
選択後、**A d d**をクリックします。

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。

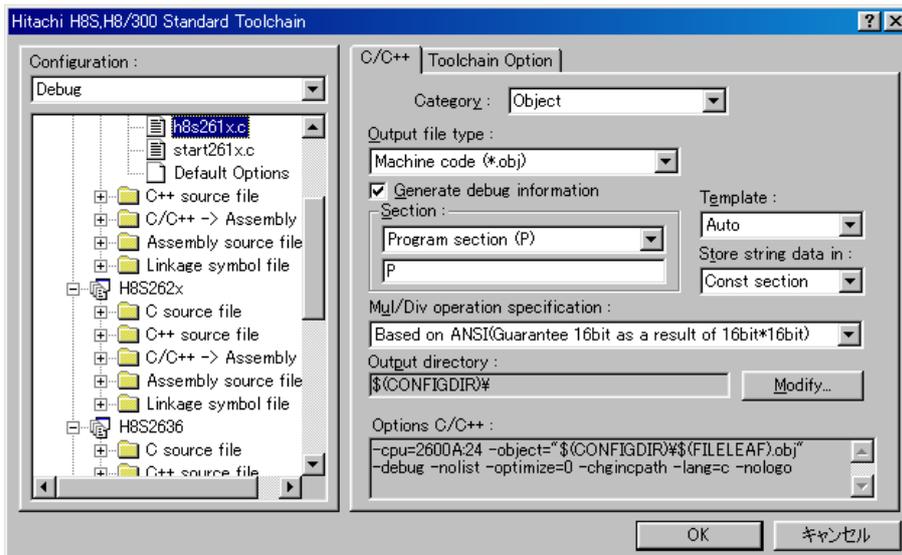
5. Cコンパイラとアセンブラの設定

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



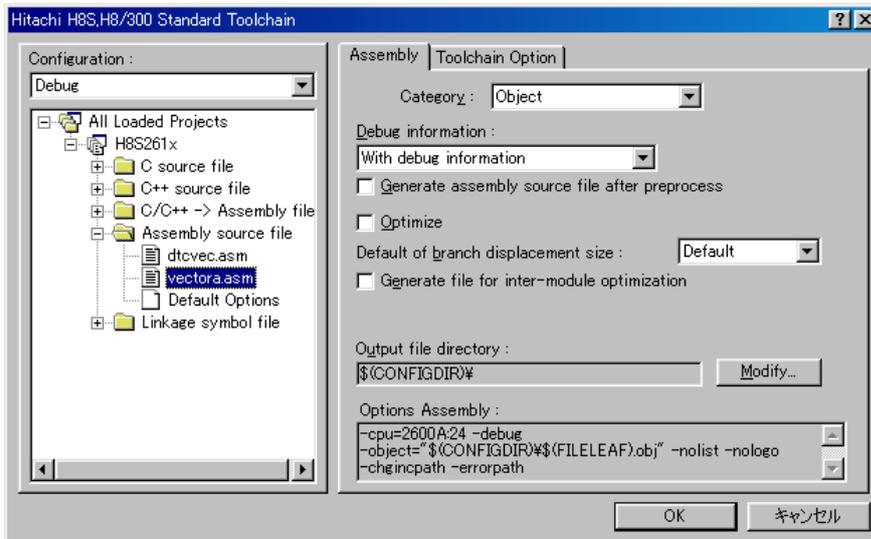
- ・カテゴリ「Object」のOutput file typeを「Assembly source code(*.src)」に設定する。
- ・この設定にしますとシンボル情報のタイプは全てラベルになります。

2) 全てC言語記述の場合



- ・カテゴリ「Object」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となります。

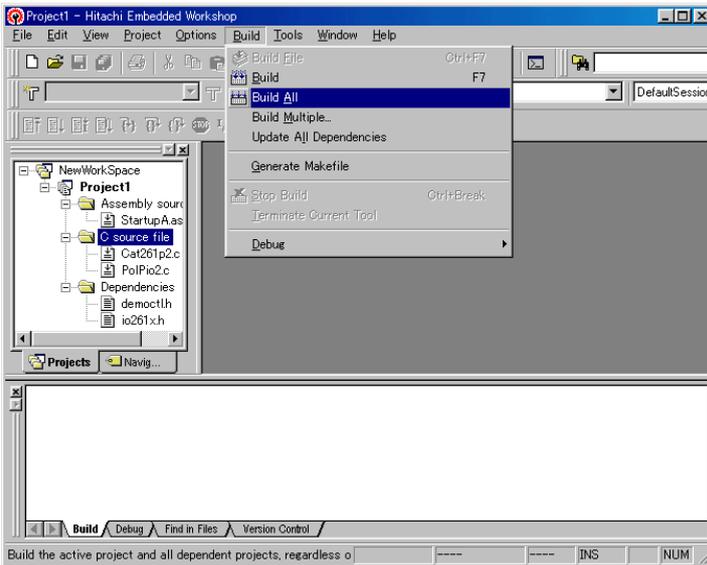
3) 全てアセンブラ記述の場合



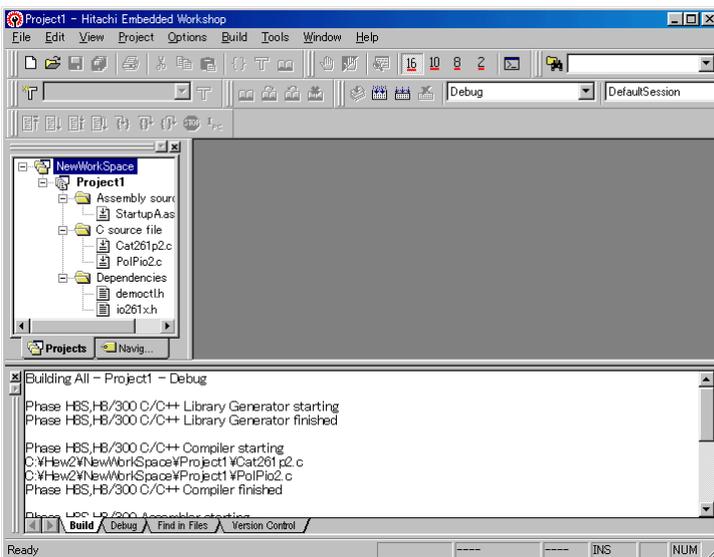
- ・カテゴリ「Object」の Debug information を「With debug information」に設定する。

6. 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で使用しますと“Build”のたびにライブラリの作成を始めてしまいます。
ここでは、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。

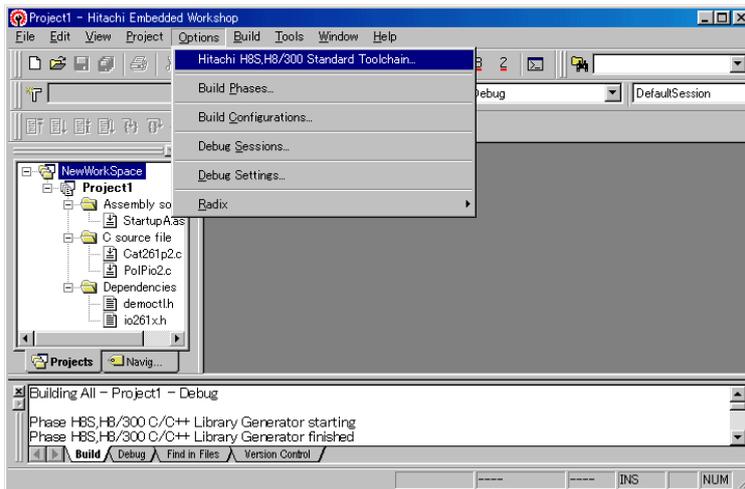


[Build]-[Build All]をクリックします。

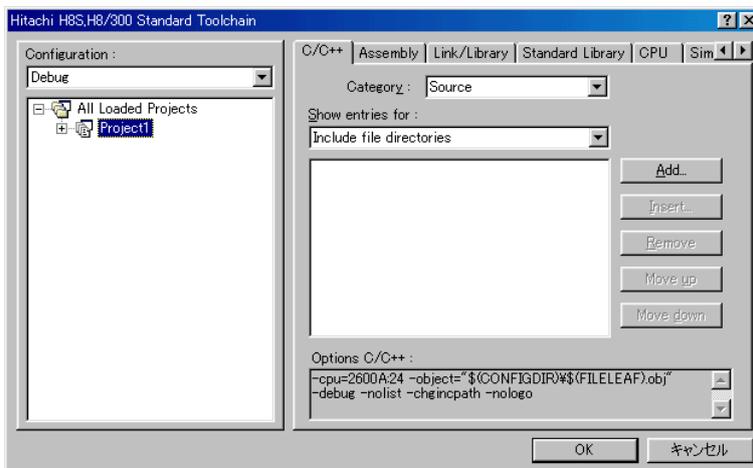


ライブラリの作成を始めます。

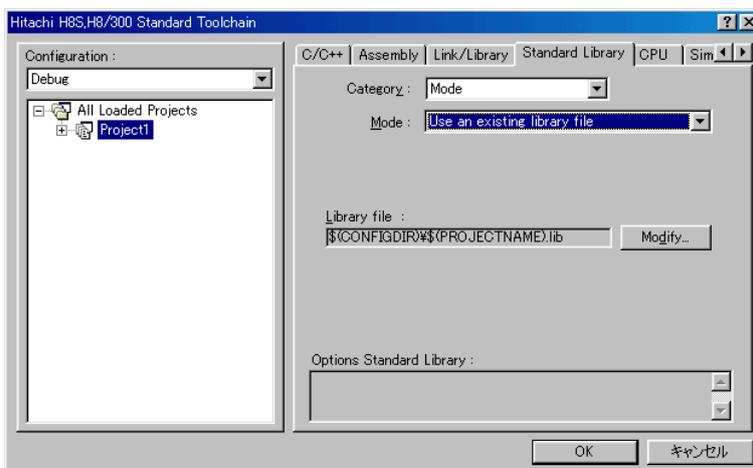
終了後プロジェクトのファイルをコンパイルしてリンクまで実行してまいります。この時点ではまだエラーもしくはワーニングが出ますが無視してください。



[Option]-[Hitachi H8S,H8/300 Standard Toolchain]をクリックします。



[Standard Library]タグをクリックする。



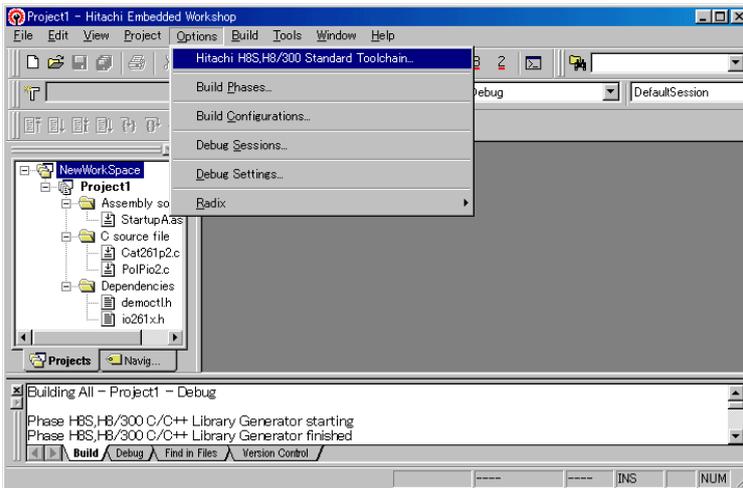
Modeを” Use an existing library file” にします。

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。

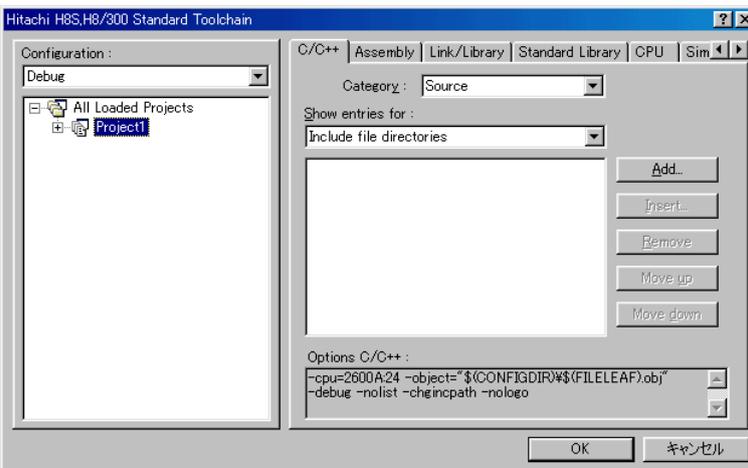
OKをクリックする。

次に、リンカの設定を説明します。

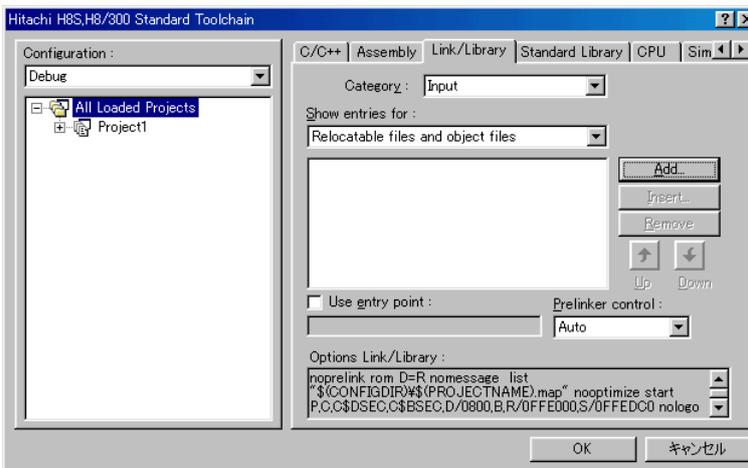
7. ツール (リンカ) の設定



[Option]-[Hitachi H8S,H8/300 Standard Toolchain]をクリックします。



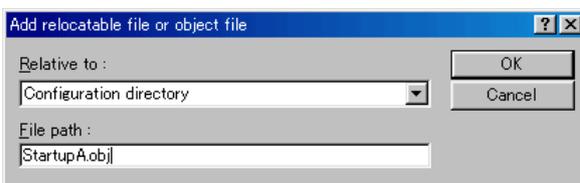
[Link/Library] タグをクリックする。



Category の " Input " を選択する。

Show entries for の " Relocatable file and object file " を選択する。

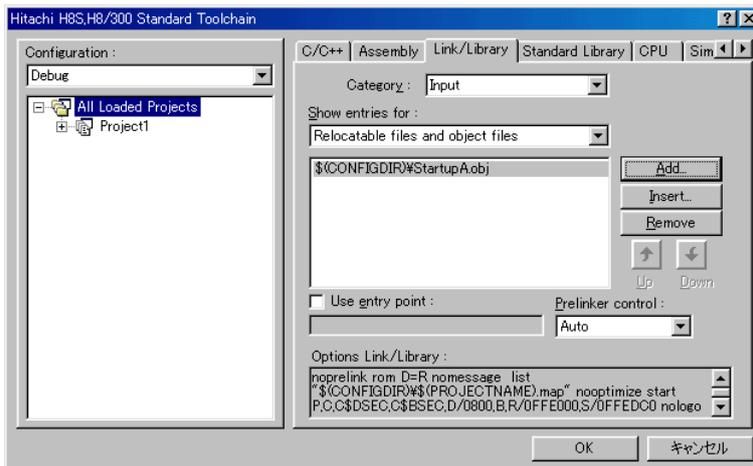
A d d をクリックする。



Relative to を " Configuration Directory " に選択します。

File Path を " StartupA.obj " と入力します。

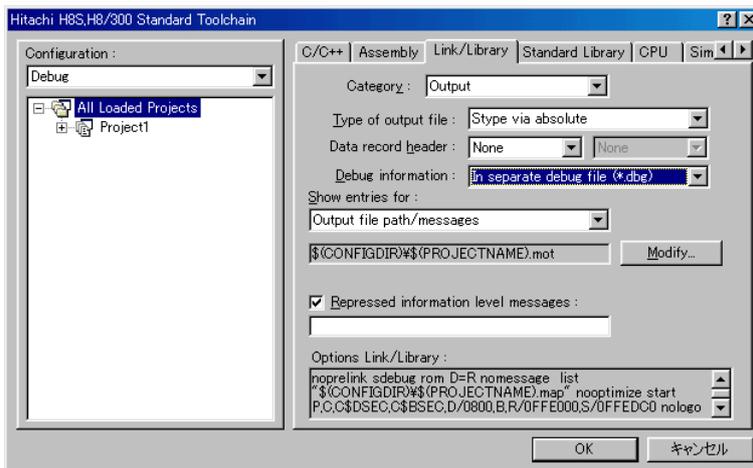
入力を確認して **OK** をクリックします。



ファイルがセットされたのを確認します。

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 C View 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。)

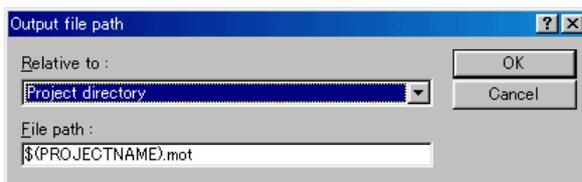


Category の”Output”を選択する。

”Type of output file ”が、”Stype via absolute”になっていることを確認します。

H-debugger でシンボリックデバックをするために、Debug information を ”In separate debug file (*.dbg)”を選択します。

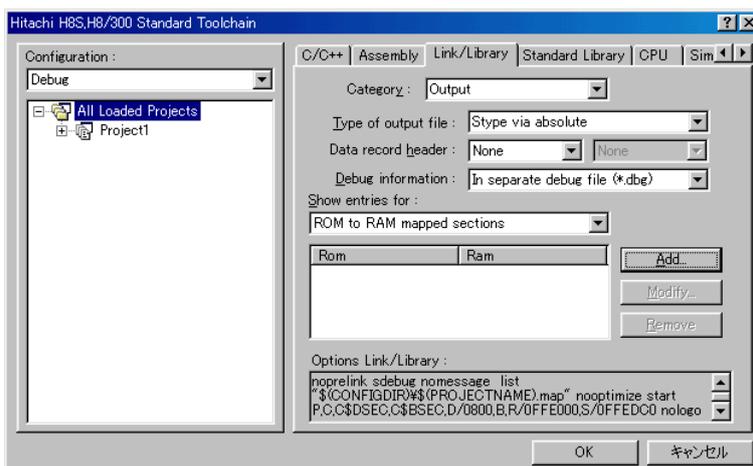
Output file Path/messages を設定するため **Modify** をクリックします。



Relative To を”Project directry”に設定します。

OKをクリックする。

(重要) これは、HEX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。

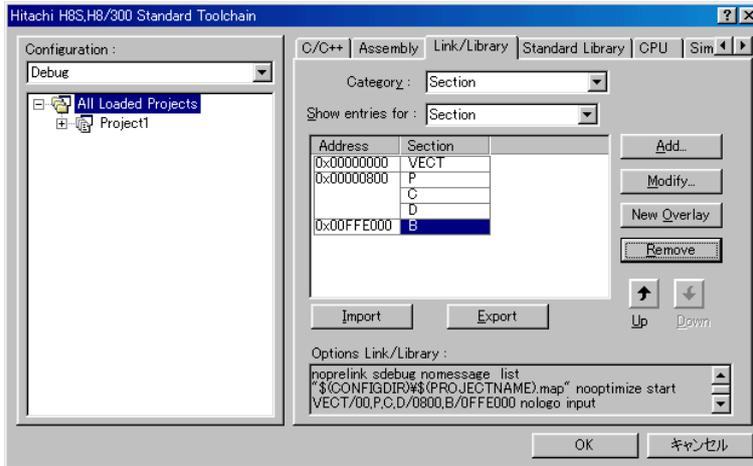


Show entries for の”ROM to RAM mapped sections”を選択する。

Rom/Ram の D/R を選択クリックする。

Remove をクリックする。

これで、セクション名が Remove されます。



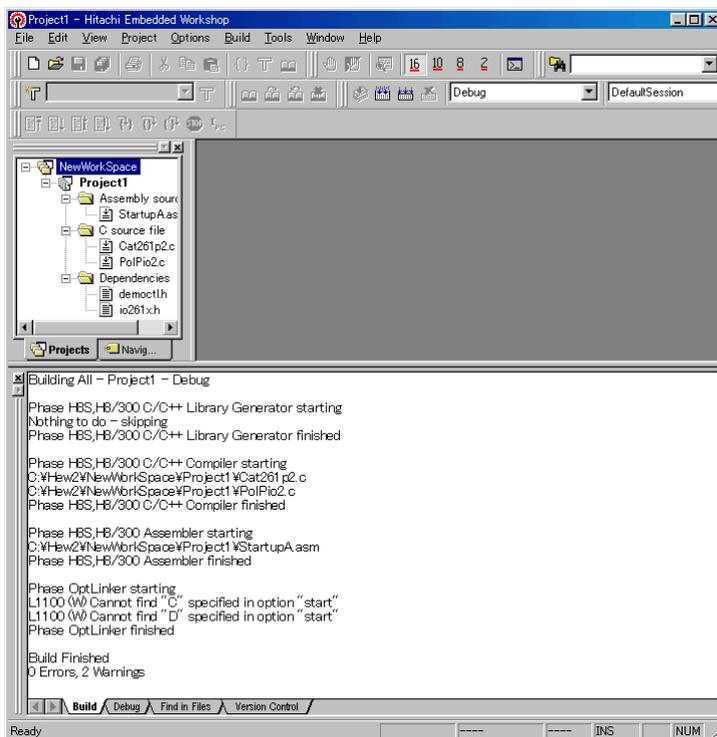
Category の” Section” を選択する。

下図のようにセクション指定をする。

0x00000000	VECT [StartupA.asm]
0x00000800	P C D
0x00FFE000	B

OK をクリックする。

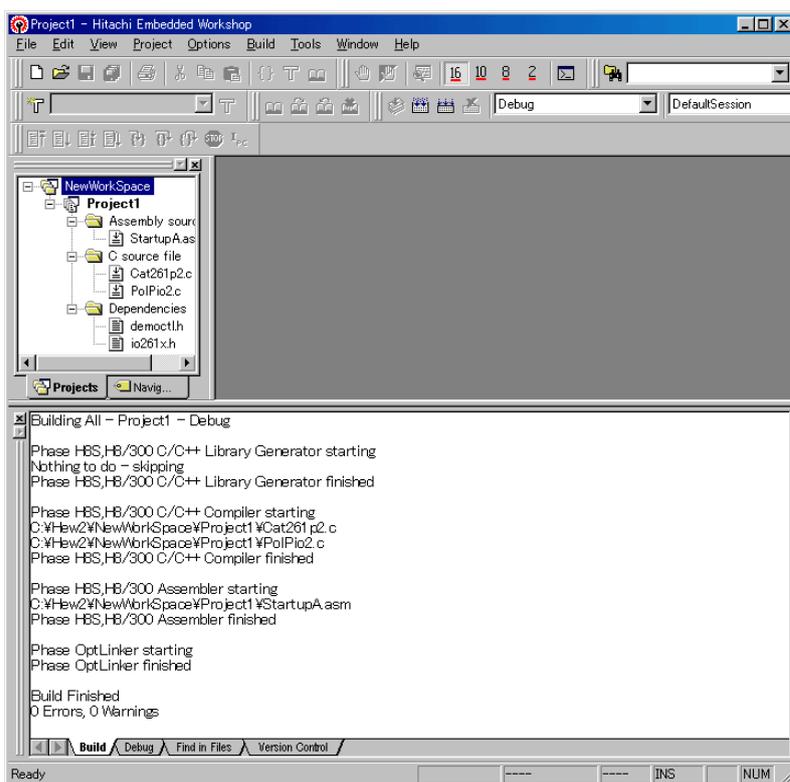
ここで、再度 [Build] - [Build All] をクリックします。



このじょうたいでの Build 結果は、左記のように2つのワーニングが表示されます。

これは、'C' と 'D' のセッションが使用されていないと言うメッセージですので無視してもかまいませんが、気になる方は、前項のセクション指定にて 'C' と 'D' を Remove してください。

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ーの登録方法は、別資料の“シンボルコンバータ (HEW)”をご参照下さい。

—以上—