

統合環境HEW (Ver 3.01) 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 (H8/Tiny版)

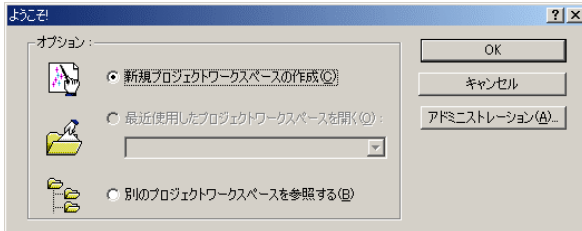
ルネサスC言語用統合環境“HEW Ver 3.01 (release1)”でH-debugger用に新ワークスペース/プロジェクト名を登録する手順方法を説明します。

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 : NewWorkSpace2
プロジェクト名 : Project2
登録モジュール名 : Vector.asm ASMファイル 割り込みベクタテーブル
 Startup.c Cファイル スタートアップ
 H83694.c Cファイル メインモジュール
CPUタイプ : H8/3694F

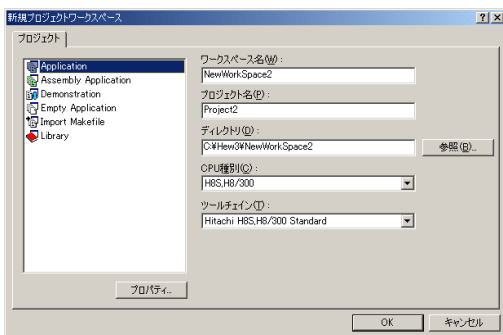
1. 新ワークスペースの登録方法

“HEW” 起動させます。



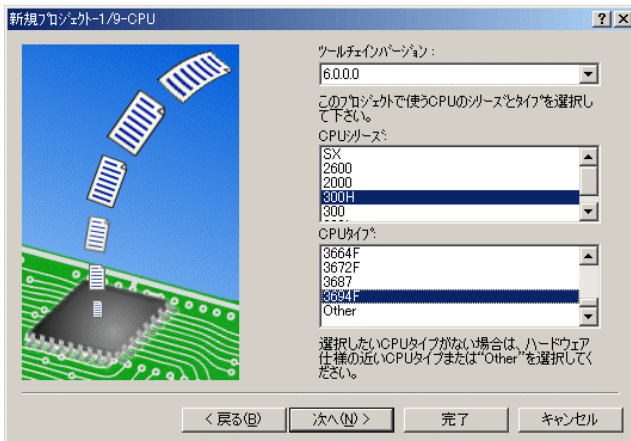
“新規プロジェクトワークスペース”をチェックしてのOKをクリックする。

もしくは、キャンセル後に、[ファイル]-[新規ワークスペース]をクリックします。



ワークスペース名	“NewWorkSpace2”
プロジェクト名	“Project2”
ディレクトリ	“C:\HEW3\NewWorkSpace2”
CPU種別	“H8S, H8/300”
ツールチェーン	“Hitachi H8S, H8/300 Standard”
プロジェクト	Application

この項目を確認後、OKをクリックして下さい。



CPU シリーズを“300H”に選択する。
CPU タイプを“3694F”に選択する。

確認後、**次へ**をクリックします。



CPUスペックを確認後、**次へ**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。

main() 関数生成を“None”に選択する。

確認後、**次へ**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、その他ライブラリを使用しません。

このまま、**次へ>**をクリックして下さい。



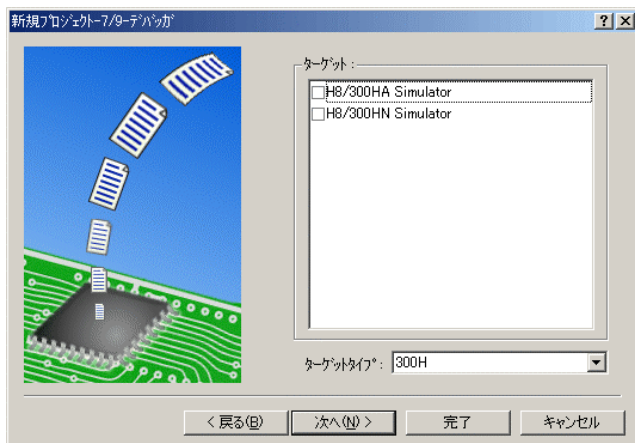
スタックボトムの設定ですが、この情報も使用しません。

このまま、**次へ>**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、**次へ>**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。

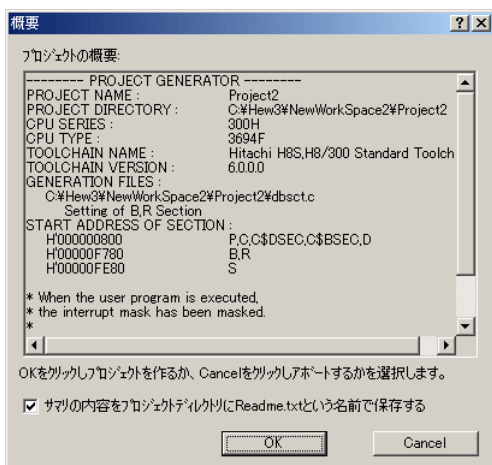
このまま、**次へ>**をクリックして下さい。



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。

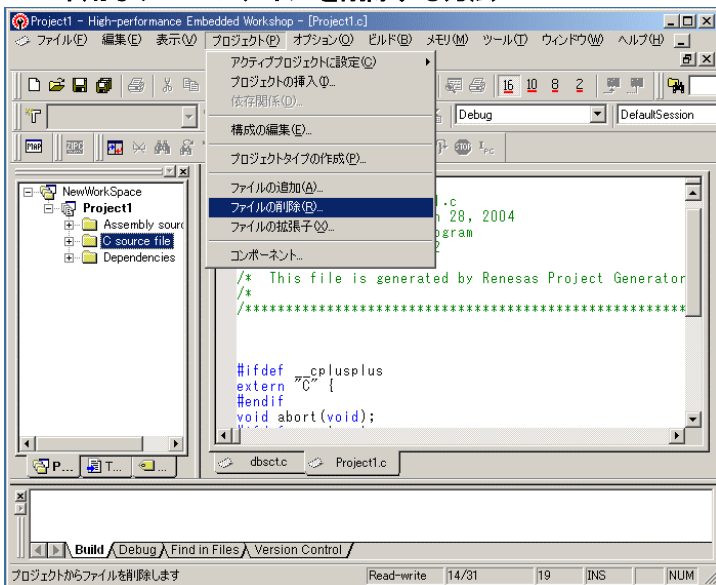
この状態で**完了**をクリックして下さい。



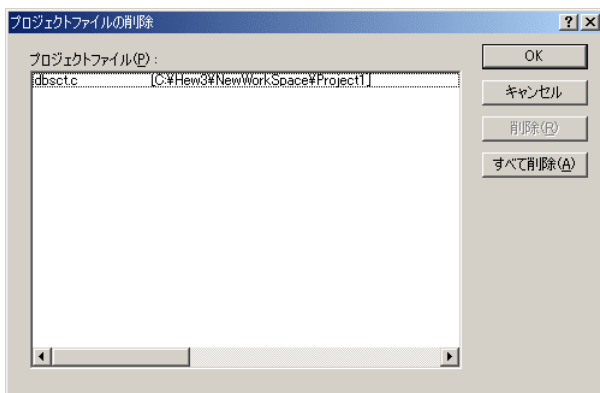
確認画面が表示されますので、**OK**をクリックして終了です。

次にプロジェクト名"Project2"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



[プロジェクト]-[ファイルの削除]をクリックします。



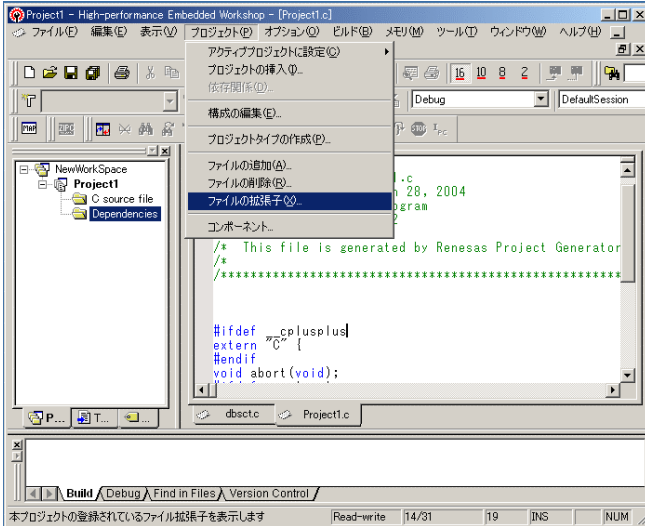
”dbsect.c”をクリックして選択すると”削除”ボタンが使用できるようになります。

削除のクリックで、”dbsect.c”を削除することが出来ます。

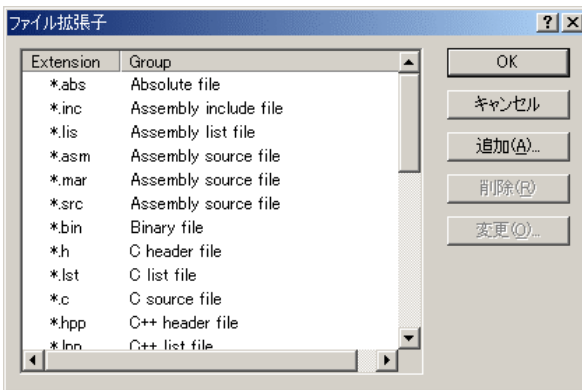
削除したらOKで終了です。

次にプロジェクト名”Project2”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。

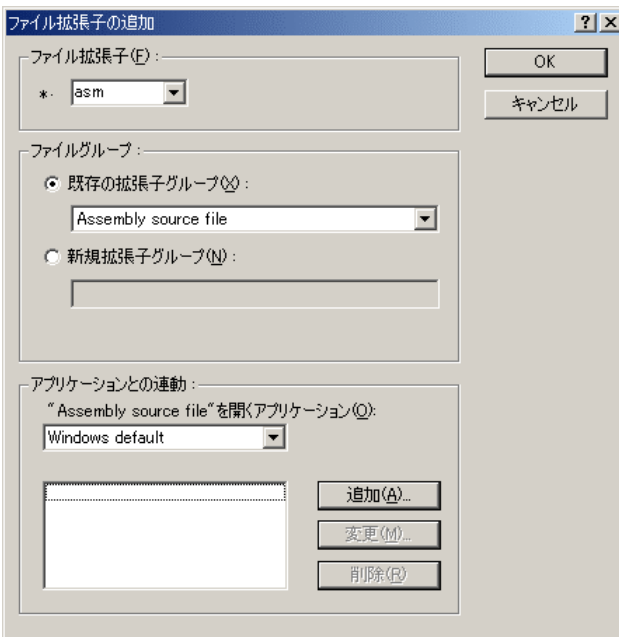
3. 拡張子を登録する方法



[プロジェクト]-[ファイルの拡張子]
をクリックします。



追加をクリックします。



ファイル拡張子に”asm”を入力する。

既存の拡張子グループ”Assembly source file”を選択する。

OKをクリックします。

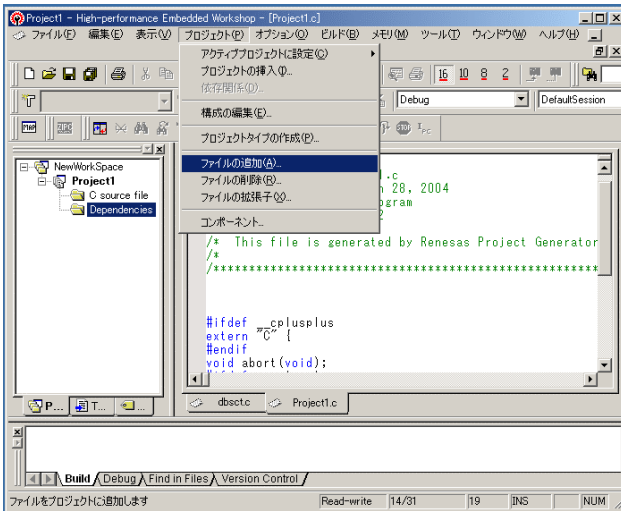
この操作により、プロジェクトに拡張子
“*.asm”の登録が完了しました。

次にプロジェクト名”Project2”に希望モジュール（ソースファイル）を登録します。

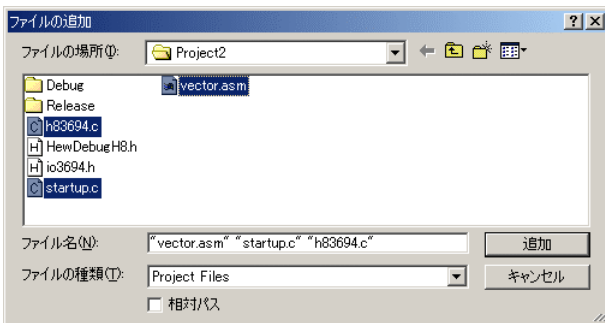
4. プロジェクトに希望モジュール（ソースファイル）を登録する方法

準備： 作成済みのソースファイルを”C:\¥Hew3¥NewWorkSpace2¥Project2”にコピーして下さい。

Vector.asm ” ¥sample¥HewH8T2_0¥H8_3694”
Startup.c (左記ソースは、製品 CD の上記フォルダに入っています。)
H83694.c
Io3694.h
HewDebugH8.h



[プロジェクト]-[ファイルの追加]をクリックします。



下記3ファイルを指定して下さい。

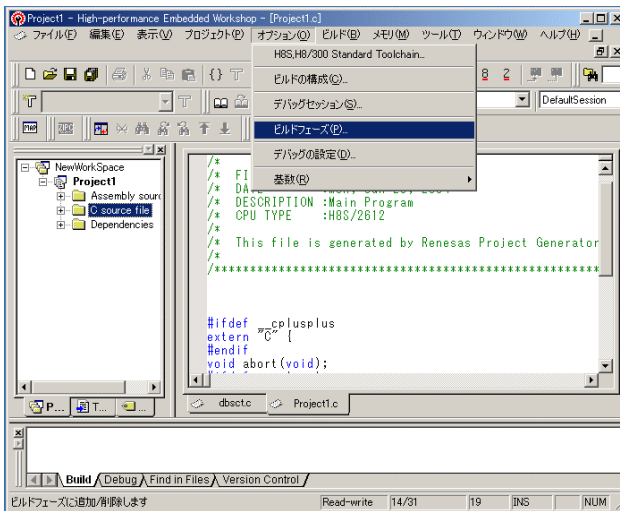
vector.asm
Startup.c
h83694.c

選択後、**追加**をクリックします。

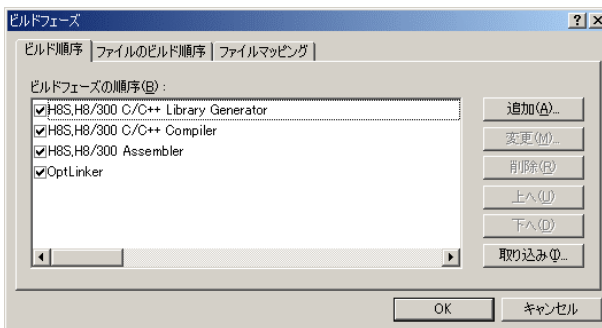
この操作により、プロジェクトにモジュールの登録が完了しました。

次に、ツール関係の設定を説明します。

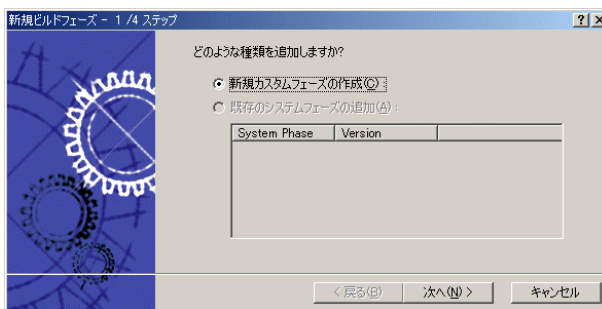
5. シンボルコンバータ「HCsymconv」を登録する。



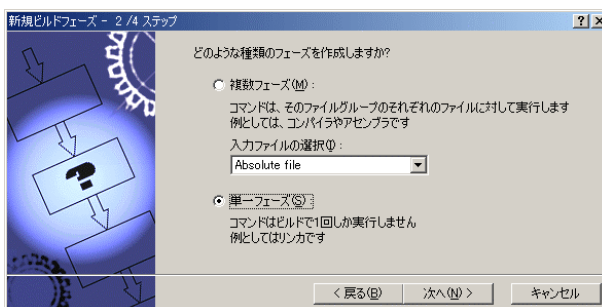
[オプション]-[ビルドフェーズ]をクリックします。



追加をクリックします。

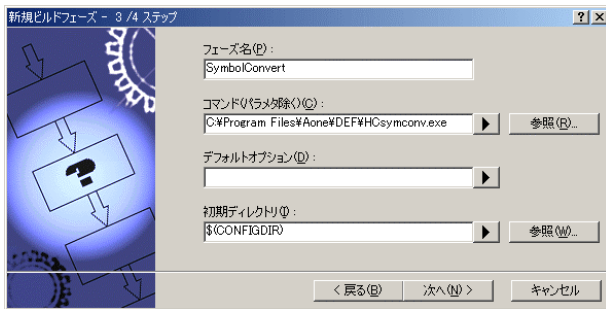


次へ>をクリックします。

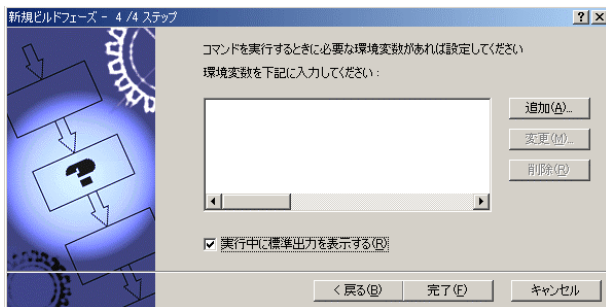


単一フェーズ側にチェックをします。

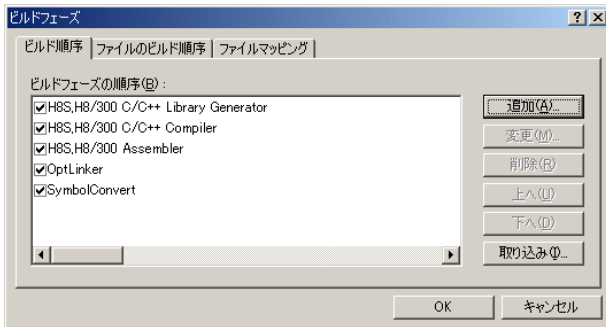
次へ>をクリックします。



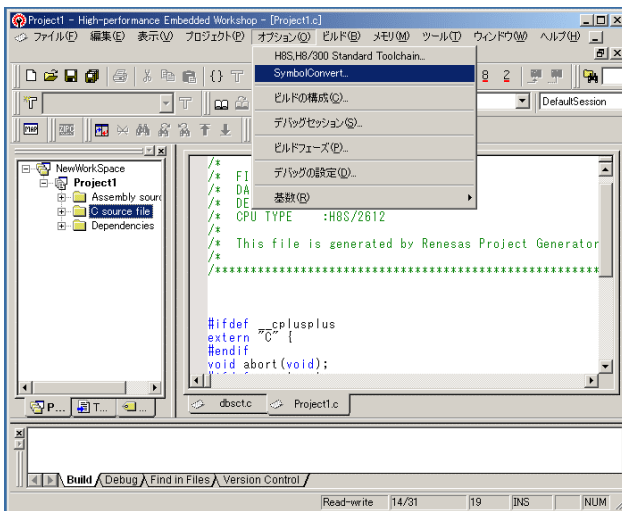
フェーズ : SymbolConvert
 コマンド : C:\ProgramFiles\Aone\DEF\HCSymconv.exe を選択する。
 (デフォルト)
 初期ディレクトリ : \$(CONFIGDIR)
 [次へ>] をクリックします。



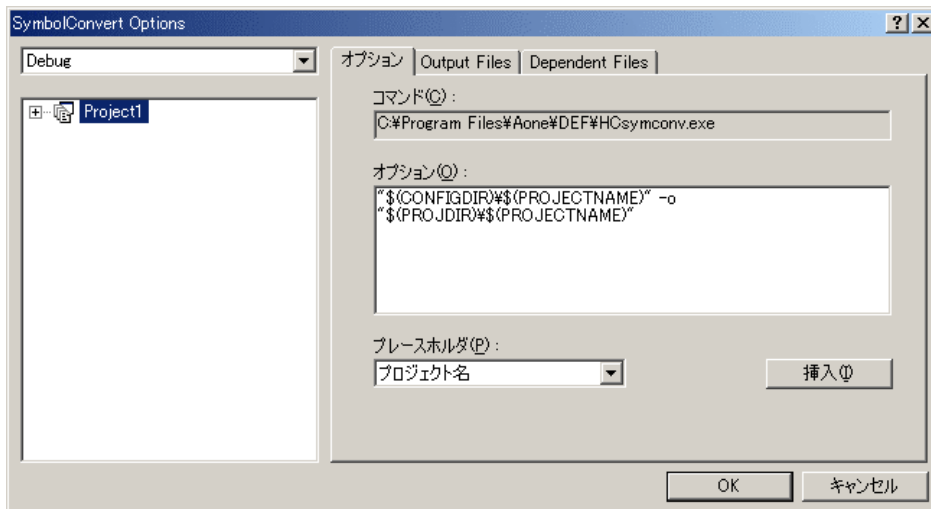
[完了] をクリックします。



[OK] をクリックします。



[オプション]-[SymbolConvert] をクリックします。



オプションに下記内容を設定する。

"\$(CONFIGDIR)\\$(PROJECTNAME)" -o "\$(PROJDIR)\\$(PROJECTNAME)"
 (入力ファイル名) (出力先名)

注意事項

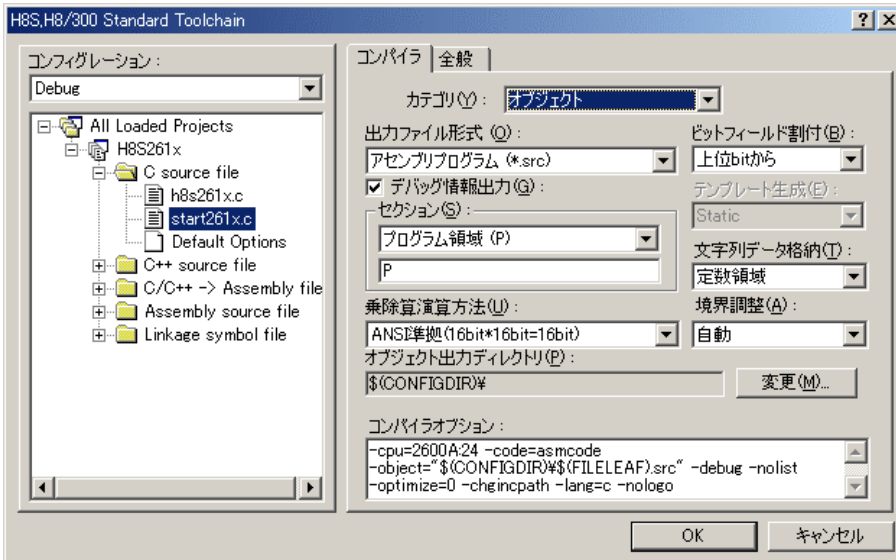
- 1) ディレクトリ名に ' ' スペースを使用している場合は、“ダブルクォートで囲んで下さい。
 “\$(CONFIGDIR)\\$(PROJECTNAME)” -o “\$(PROJDIR)\\$(PROJECTNAME)”
- 2) \$(PROJECTNAME)の先頭に「¥」記号を挿入して下さい。(手入力)
- 3) オプションSW「-o」の両端には、スペースを入れてください。(手入力)
- 4) Cソースファイルの有る場所にシンボルコンバート後のファイルを置くのが目的ですので必ず指定して下さい。

追加事項 (HCsymconv.exe スイッチ説明)

- 1) [-o] (省略可) 出力ファイル名を指定
- 2) [-r] (省略可) モジュール毎のディレクトリ情報を作成しない。ELF専用(Ver 3. 2xxから)
- 3) [-s] (省略可) ラインシンボル情報をソート (アドレス順) しない。(Ver 3. 2xxから)
- 4) [-i] (省略可) 不整合なInline情報を削除する。(Ver 3. 3xxから)
- 5) [-g] (省略可) スタティック変数をグローバル化する。(Ver 3. 6xxから)

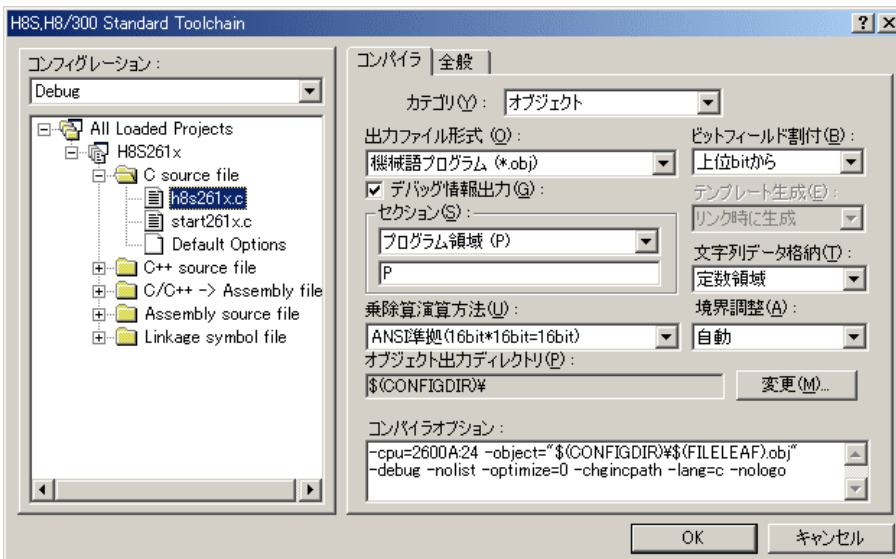
6. Cコンパイラとアセンブラの設定

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



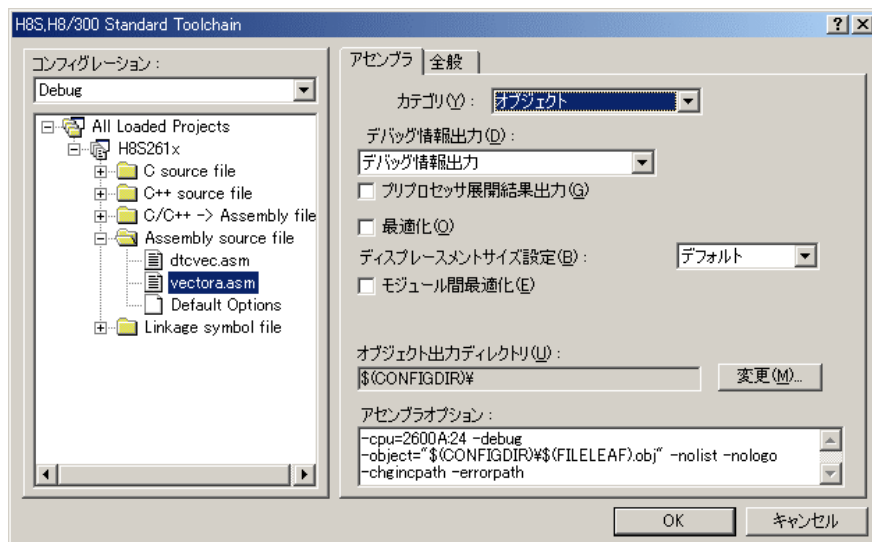
- ・カテゴリ「オブジェクト」の出力ファイル形式を「アセンブリプログラム(*.src)」に設定する。
- ・この設定にしますとシンボル情報のタイプは全てラベルになります。

全てC言語記述の場合



- ・カテゴリ「オブジェクト」の出力ファイル形式を「機械語プログラム(*.obj)」に設定する。
- ・この設定にしますと詳細なシンボル情報を得ることになります。

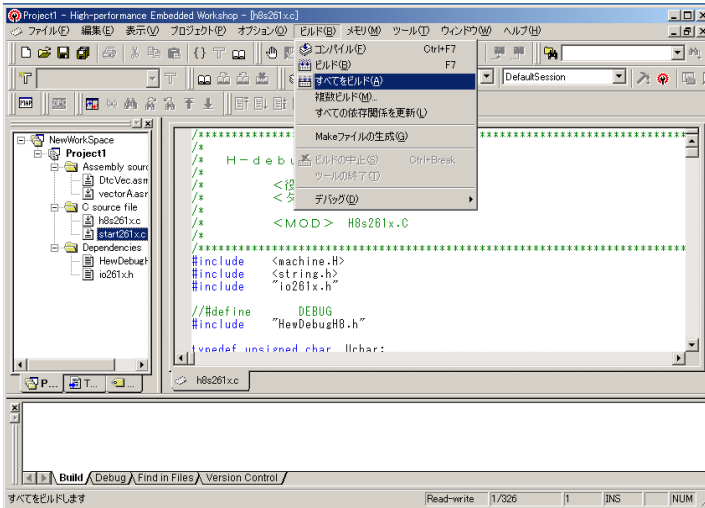
全てアセンブラ記述の場合



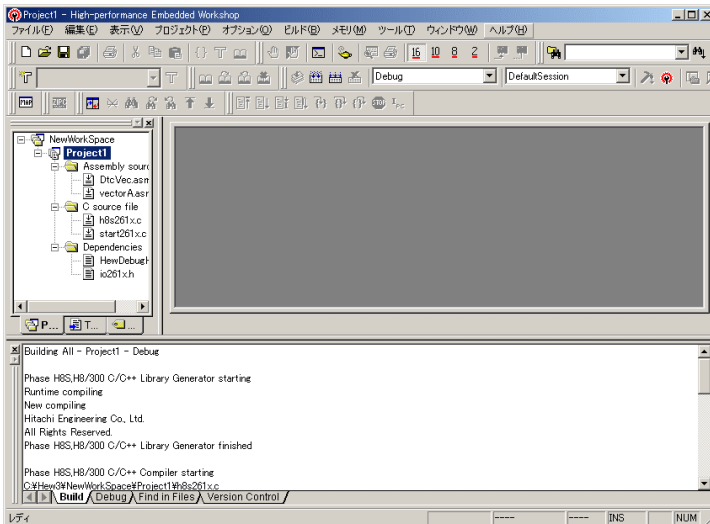
- ・カテゴリ「オブジェクト」のデバッグ情報を「デバッグ情報出力」に設定する。

7. 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。
ここで「すべてをビルド」してライブラリを作成および設定の確認をします。

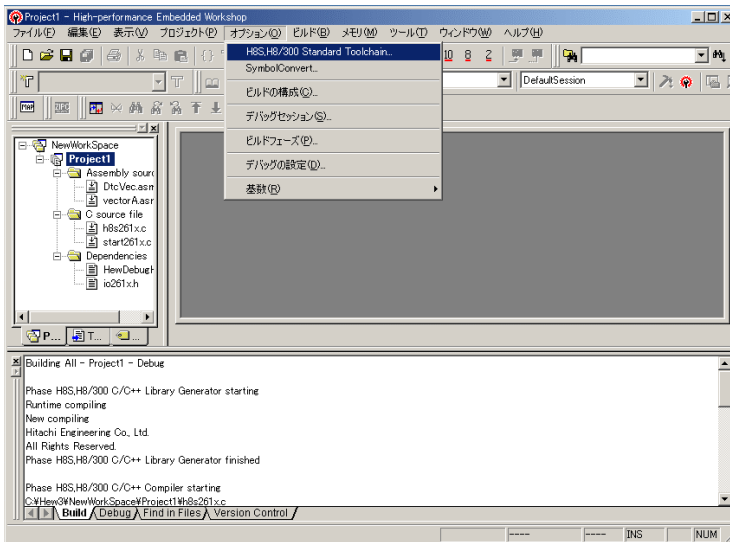


[ビルド]-[すべてをビルド]をクリックします。

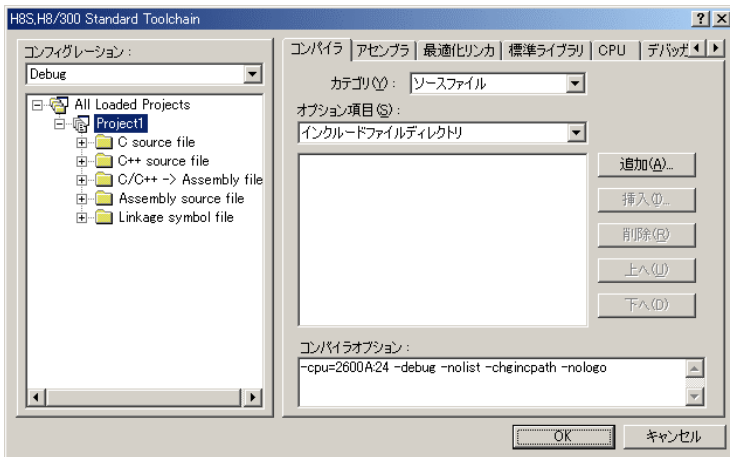


ライブラリの作成を始めます。

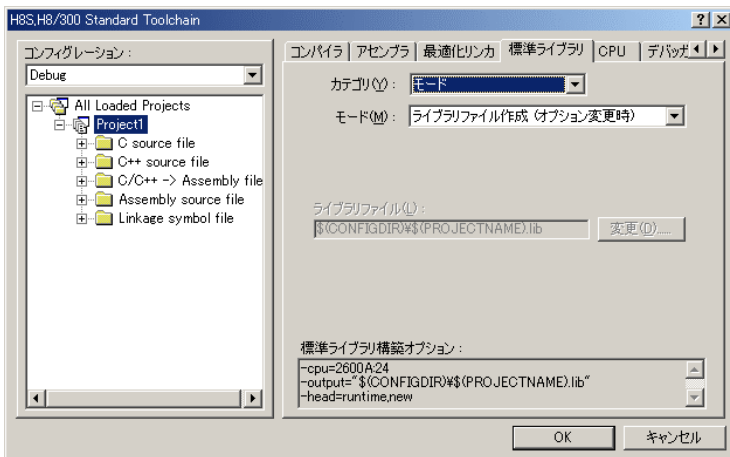
終了後プロジェクトのファイルをコンパイルしてリンクまで実行します。この時点ではまだエラーもしくはワーニングが出ますが無視してください。



[オプション]-
[H8S,H8/300Standard Toolchain]
をクリックします。



[標準ライブラリ]タグをクリッ
クする。



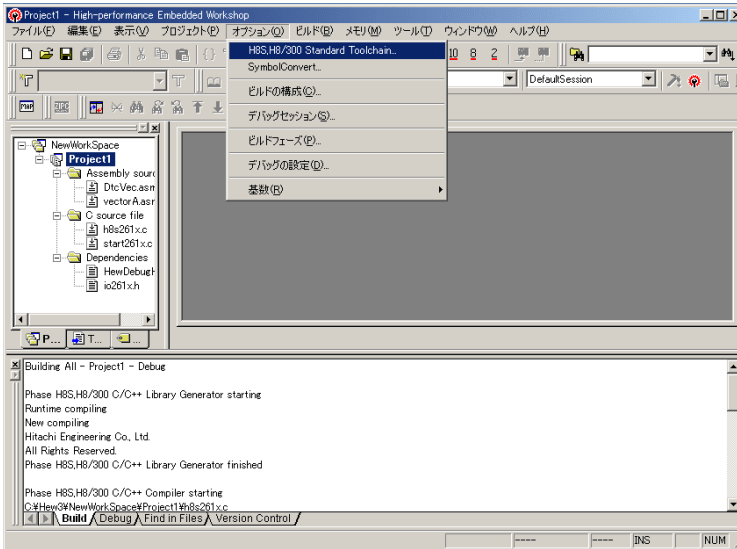
カテゴリのモードが
「ライブラリファイル作成 (オ
プション変更時)」指定になっ
ている事を確認します。

この指定によりオプション変更
時のみライブラリを作成する事
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。

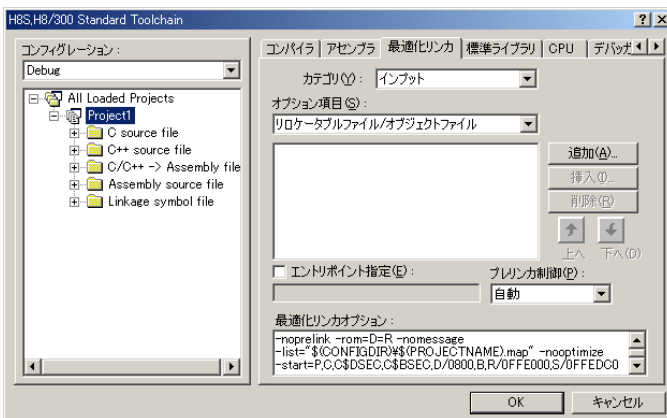
8. ツール（リンク）の設定



[オプション]-
[H8S, H8/300Standard Toolchain]
をクリックします。



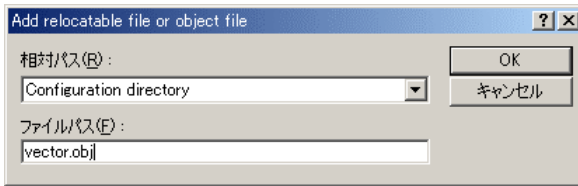
「最適化リンク」タグをクリックする。



カテゴリの「インプット」を選択する。

オプション項目の
「リロケータブルファイル/オブジェクトファイル」を選択する。

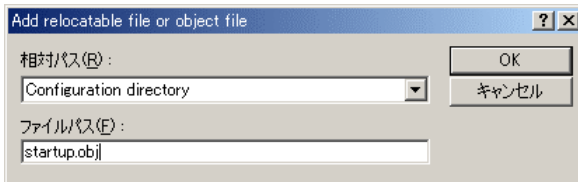
追加をクリックする。



相対パス「Configuration Directory」に選択します。

ファイルパス「vector.obj」と入力します。

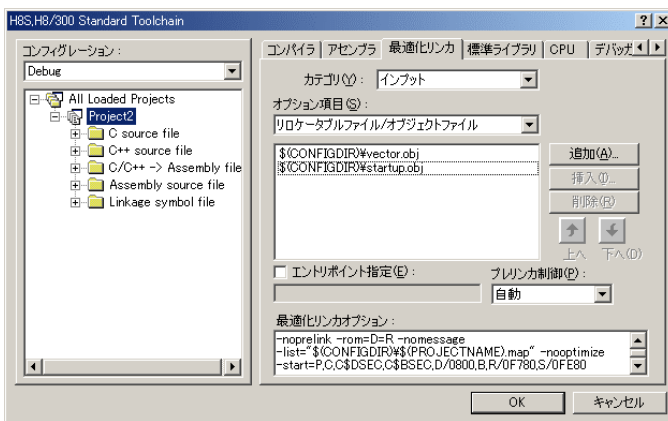
入力を確認してOKをクリックします。



相対パス「Configuration Directory」に選択します。

ファイルパス「startup.obj」と入力します。

入力を確認してOKをクリックします。

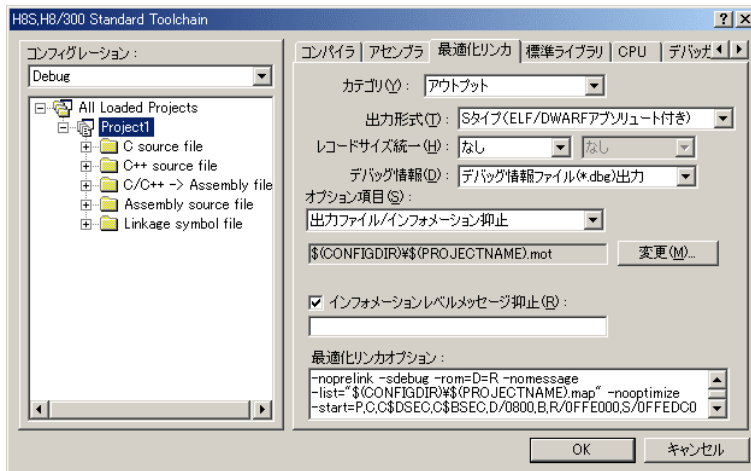


ファイルがセットされたのを確認します。

この指定は、“最初に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。)

(重要) 特に「startup.obj」の先頭に SP を設定するコードが記述してありますので P セクションの先頭を 0 x 8 0 0 番地にする為に必要な設定です。



カテゴリの「アウトプット」を選択する。

出力形式の

「Sタイプ(ELF/DWARFアプリアルト付き)」を選択する。

デバッグ情報の

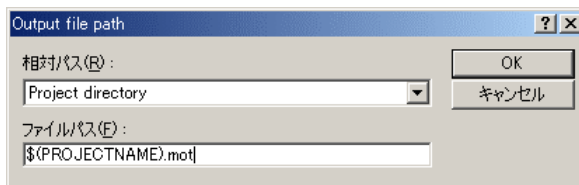
「デバッグ情報ファイル(*.dbg)出力」を選択する。

オプション項目の

「出力ファイル/インフォメーション抑止」の

変更をクリックします。

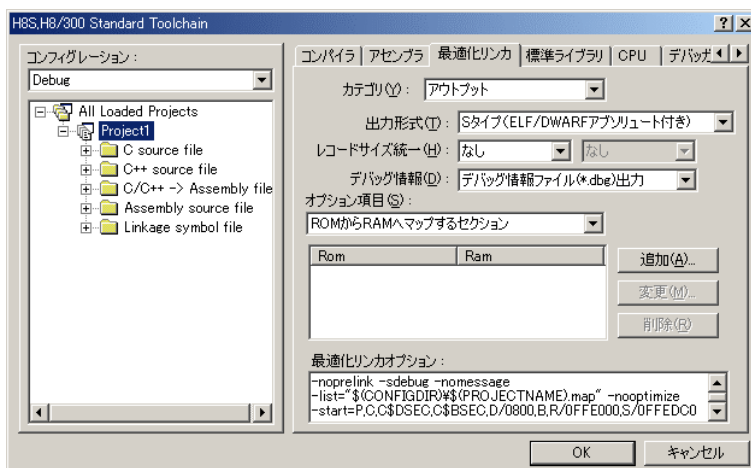
(重要) シンボリックデバッグを可能にするために必要な設定です。



相対パスを「Project directory」に設定します。

OKをクリックする。

(重要) この指定は、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に必要です。



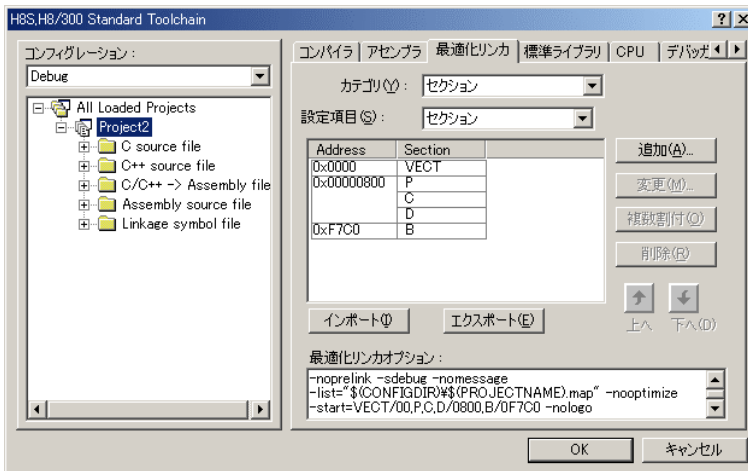
オプション項目の

「ROMからRAMへのマップするセクション」を選択する。

Rom/RamのD/Rを選択クリックする。

削除をクリックする。

これで、未使用セクション名がRemoveされます。



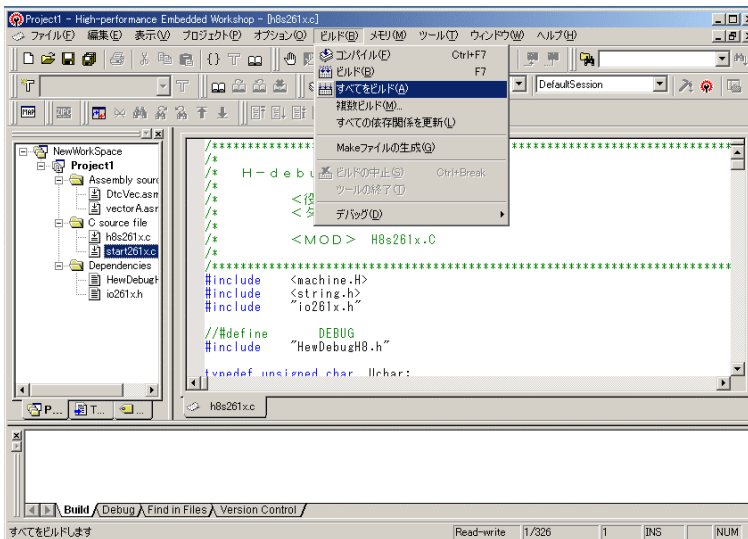
「セクション」を選択する。

下図のようにセクション指定をする。

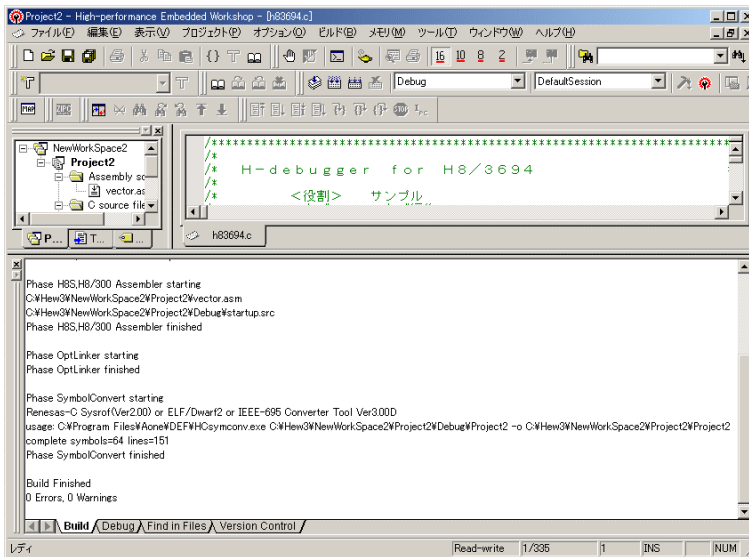
0x0000	VECT
0x0000800	P
	C
	D
0xF7C0	B

OKをクリックする。

(重要) セクション名：VECTは、「vector.asm」で宣言しています。
 0xF7C0 (0xF780) は、ソースブレイクを使用する場合の例です。
 DEFバージョン6.50xからソースブレイクを使用する場合は、モニタワーク方式をスタック方式に選択する必要があります。



[ビルド] - [すべてをビルド]をクリックします。



「0 Error 0 Warnings」になり作業終了です。

以上