

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>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パソコン機器、産業用ロボット

高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等

8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエーペンギング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

お客様各位

資料中の「日立製作所」、「日立XX」等名称の株式会社ルネサス テクノロジへの変更について

2003年4月1日を以って三菱電機株式会社及び株式会社日立製作所のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ(フラッシュメモリ・SRAM等)を含む半導体事業は株式会社ルネサス テクノロジに承継されました。従いまして、本資料中には「日立製作所」、「株式会社日立製作所」、「日立半導体」、「日立XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

ルネサステクノロジ ホームページ (<http://www.renesas.com>)

2003年4月1日
株式会社ルネサス テクノロジ
カスタマサポート部

ご注意

安全設計に関するお願い

- 弊社は品質、信頼性の向上に努めていますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

- 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
- 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
- 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
- 本資料に記載した情報は、正確を期すため、慎重に制作したものですが万一本資料の記述誤りに起因する損害をお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
- 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任は負いません。
- 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
- 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
- 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

HEW Builder High-performance Embedded Workshop 2

ユーザーズマニュアル

(Windows®98/Me、Windows NT®4.0、
Windows®2000およびWindows®XP用)

ご注意

1. 本書に記載の製品及び技術のうち「外国為替及び外国貿易法」に基づき安全保障貿易管理関連貨物・技術に該当するものを輸出する場合、または国外に持ち出す場合は日本国政府の許可が必要です。
2. 本書に記載された情報の使用に際して、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等の権利に対する保証または実施権の許諾を行うものではありません。また本書に記載された情報を使用した事により第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
3. 製品及び製品仕様は予告無く変更する場合がありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書をお求めになりご確認ください。
4. 弊社は品質・信頼性の向上に努めておりますが、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途にご使用をお考えのお客様は、事前に弊社営業担当迄ご相談をお願い致します。
5. 設計に際しては、特に最大定格、動作電源電圧範囲、放熱特性、実装条件及びその他諸条件につきましては、弊社保証範囲内でご使用いただきますようお願い致します。
保証値を越えてご使用された場合の故障及び事故につきましては、弊社はその責を負いません。
また保証値内のご使用であっても半導体製品について通常予測される故障発生率、故障モードをご考慮の上、弊社製品の動作が原因でご使用機器が人身事故、火災事故、その他の拡大損害を生じないようにフェールセーフ等のシステム上の対策を講じて頂きますようお願い致します。
6. 本製品は耐放射線設計をしておりません。
7. 本書の一部または全部を弊社の文書による承認なしに転載または複製することを堅くお断り致します。
8. 本書をはじめ弊社半導体についてのお問い合わせ、ご相談は弊社営業担当迄お願い致します。

このマニュアルについて

このマニュアルでは基本的な HEW システムについて述べています。ここには、HEW の基本的な使い方に関する情報、および HEW 環境のカスタマイズに関する章が含まれます。HEW のビルド機能およびデバッグ機能の詳細は、このマニュアルに付随した他の 2 冊のマニュアルで説明します。

このマニュアルでは C/C++ 言語、アセンブリ言語の書き方や、オペレーティングシステムの使い方、個々のデバイスに適したプログラムの書き方などについては説明していません。それらについては、各々のマニュアルを参照してください。

注意 本マニュアルに記載されている HEW の画面は英語版 Windows® 上で取得したものです。日本語版 Windows® では一部日本語表示されます。また、本マニュアルではディレクトリ区切り子としてバックスラッシュ ‘\’ を使用していますが、日本語版 Windows® 上ではバックスラッシュの代わりに円記号を使用してください。

Microsoft, MS-DOS, Windows, Windows NT は米国 Microsoft 社の米国およびその他の国における登録商標です。

Visual SourceSafe は Microsoft 社の米国およびその他の国における商標です。

IBM は International Business Machines Corporation の登録商標です。

その他、記載されている製品名は各社の商標または登録商標です。

このマニュアルの記号

このマニュアルで使われている記号の意味を説明します。

表 1: 記号一覧

記号	意味
[Menu->Menu Option]	太字と ‘->’ はメニューオプションを示します（例 [File->Save As...]）
FILENAME.C	大文字の名前はファイル名を示します
“文字列の入力”	下線は入力する文字列を示します（“”を省く）
Key + Key	キー入力を示します。例えば、CTRL+N キーでは CTRL キーと N キーを同時に押します
（「操作方法」マーク）	このマークが左端にあるとき、その右の文章は何かの操作方法を示します

目次

1. 概要	1
1.1 ワークスペース、プロジェクト、ファイル	1
1.1.1 ツールバー	2
1.1.2 ワークスペースウィンドウ	2
1.1.3 アутプットウィンドウ	3
1.2 HEW を起動する	3
1.3 新規ワークスペースを作成する	4
1.4 ワークスペースを開く	5
1.5 ワークスペースを保存する	6
1.6 ワークスペースを閉じる	6
1.7 古いワークスペースの使用	6
1.8 HEW を終了する	6
2. ビルドの基本	7
2.1 ビルド処理	7
2.2 プロジェクトファイル	8
2.2.1 プロジェクトにファイルを追加する	8
2.2.2 プロジェクトからファイルを削除する	10
2.2.3 ビルドからプロジェクトファイルを除外する	11
2.2.4 ビルドへプロジェクトファイルを入れる	11
2.3 ファイル拡張子とファイルグループ	12
2.4 ファイルのビルド方法を設定する	17
2.5 ビルドのコンフィグレーション	17
2.5.1 ビルドコンフィグレーションを選択する	18
2.5.2 ビルドコンフィグレーションを追加、削除する	19
2.6 プロジェクトをビルド実行する	20
2.6.1 プロジェクトをビルド実行する	20
2.6.2 1つのファイルをビルド実行する	20
2.6.3 ビルド実行を中止する	20
2.6.4 複数のプロジェクトをビルド実行する	21
2.6.5 アутプットウィンドウ	21
2.6.6 アутプットウィンドウの内容の制御	22
2.7 ファイル依存関係	22
2.8 ワークスペースウィンドウの構成	23
2.8.1 各ファイルの下に依存を表示する	23
2.8.2 標準ライブラリファイルのインクルードを表示する	23
2.8.3 ファイルのパスを表示する	24
2.9 アクティブプロジェクトを設定する	24
2.10 ワークスペースにプロジェクトを追加する	25
2.11 プロジェクト間の依存関係を指定する	26
2.12 ワークスペースからプロジェクトを削除する	26
2.13 ワークスペースの相対プロジェクトパス	27
3. ビルドの応用	29
3.1 ビルド実行の復習	29
3.1.1 ビルドとは?	29

3.2	カスタムビルドフェーズを作成する.....	31
3.3	ビルドのフェーズ順序	35
3.3.1	ビルドのフェーズ順序	36
3.3.2	ビルドファイルのフェーズ順序	38
3.4	カスタムビルドフェーズのオプション設定.....	39
3.4.1	Options タブ	40
3.4.2	Output Files タブ	41
3.4.3	Dependent Files タブ	42
3.5	ファイルのマッピング	43
3.6	ビルドを管理する	45
3.7	ビルドの出力のログを取る.....	46
3.8	ツールチェインのバージョンを変更する.....	47
3.9	外部デバッガを使う	48
3.10	メイクファイルの生成	49

1. 概要

この章では HEW の基本概念を説明します。Windows® アプリケーションに慣れていないユーザのために、次章以降で必要となる情報を提供します。

1.1 ワークスペース、プロジェクト、ファイル

ワードプロセッサでドキュメントを作成、修正できるのと同じように、HEW ではワークスペースを作成、修正できます。

ワークスペースはプロジェクトを入れる箱と考えることができます。同じように、プロジェクトはプロジェクトファイルを入れる箱と考えることができます。したがって各ワークスペースにはプロジェクトが一つ以上あり、各プロジェクトにはファイルが一つ以上あります。この構成を図 1.1 に示します。

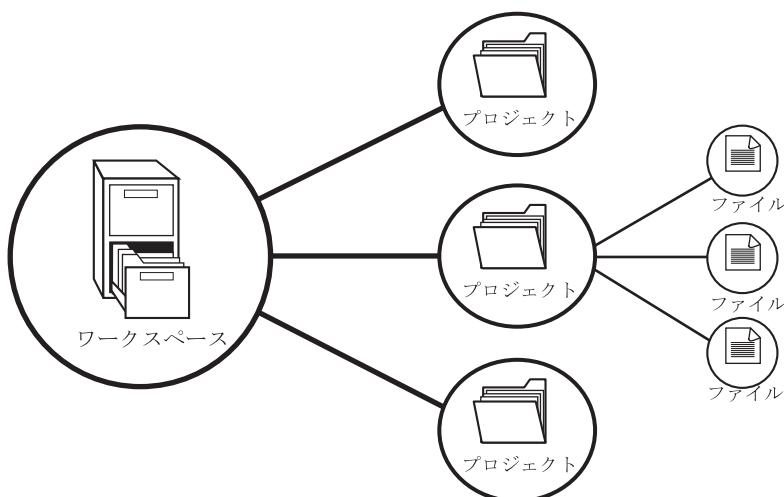


図 1.1: ワークスペース、プロジェクト、ファイル

ワークスペースでは関連したプロジェクトを一つにまとめることができます。例えば、異なるプロセッサに対して 1 つのアプリケーションを構築しなければならない場合、または、アプリケーションとライブラリを同時に開発している場合などに便利です。さらに、ワークスペース内でプロジェクトを階層的に関連づけることができます。つまり、1 つのプロジェクトを構築すると、その子プロジェクトが最初に構築されます。

ワークスペースを活用するには、ユーザは、まずワークスペースにプロジェクトを追加して、そのプロジェクトにファイルを追加しなければなりません。

1.1.1 ツールバー

ツールバーにより、使う頻度の高いオプションを簡単に利用できます。デフォルトでは“Bookmarks”、“Debug”、“Debug Run”、“Editor”、“Search”、“Standard”、“Templates”、“Version Control”的 8 つのツールバーがあります(図 1.3～図 1.10 参照)。ツールバーの作成や変更は [Tools->Customize...] メニュー オプションで指定できます。詳細については Hitachi Embedded Workshop 2.0 ユーザーズマニュアル 5 章「環境のカスタマイズ」を参照してください。



図 1.2: Standard ツールバー

1.1.2 ワークスペースウィンドウ

HEW を起動した場合、ワークスペースウィンドウにはタブが 1 つだけあります。これが“Projects”タブです。ワークスペースを開くと、ワークスペースウィンドウに 2 つのデフォルトタブを表示します。“Projects”タブには現在のワークスペース、プロジェクト、ファイルを示します(図 1.3)。アイコンをダブルクリックしてプロジェクトファイルや個々のファイルを開くことができます。

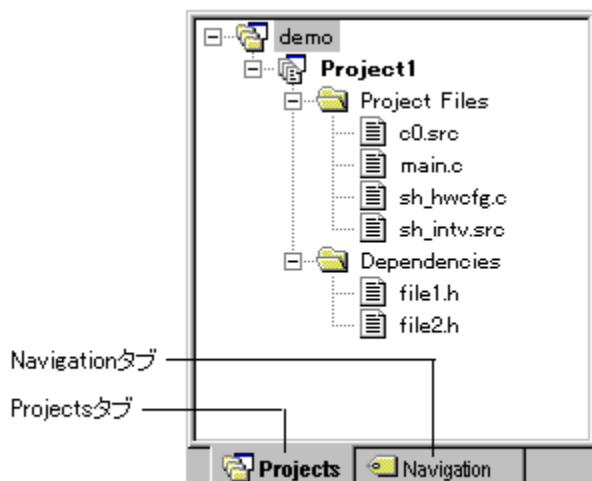


図 1.3: Workspace ウィンドウ Projects タブ

“Navigation”タブによりプロジェクトファイルの中のテキスト部へジャンプできます。“Navigation”タブに実際に表示される内容は、現在、何がインストールされているかによって異なります。図 1.4 には例えば ANSI 規格 の C 関数一覧を示します。ワークスペースウィンドウの詳細については 2 章「ビルドの基本」を参照してください。

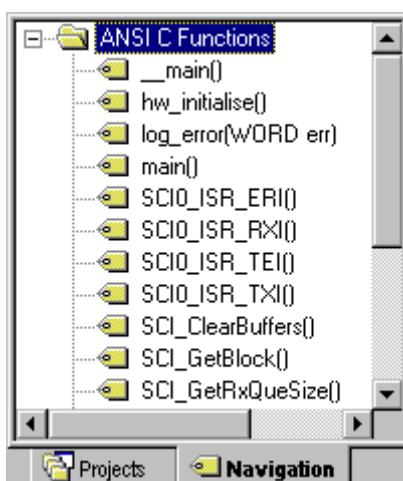


図 1.4: Workspace ウィンドウ Navigation タブ

1.1.3 アウトプットウィンドウ

デフォルトではアウトプットウィンドウに4つのタブが表示されます。“Build”タブには任意のビルド実行（コンパイラ、アセンブラーなど）の出力を示します。ソースファイルにエラーがある場合、“Build”タブにはエラーとソースファイル名と行番号が表示されます。エラーをダブルクリックすると、ソースファイルの行にジャンプするので、エラー箇所をすばやく発見できます。

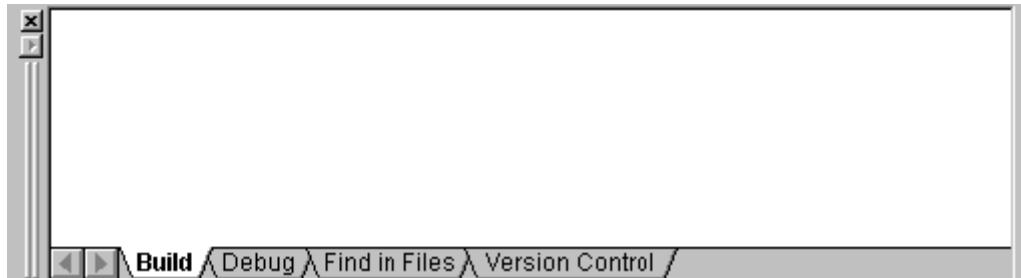


図 1.5: アウトプットウィンドウ

1.2 HEW を起動する

HEW を起動するには Windows® の“スタート”メニューを開き、“プログラム”を選択し、“Hitachi Embedded Workshop 2”を選択し、HEW のショートカットを選びます。デフォルトで図 1.6 にしめす“Welcome!”ダイアログボックが開きます。

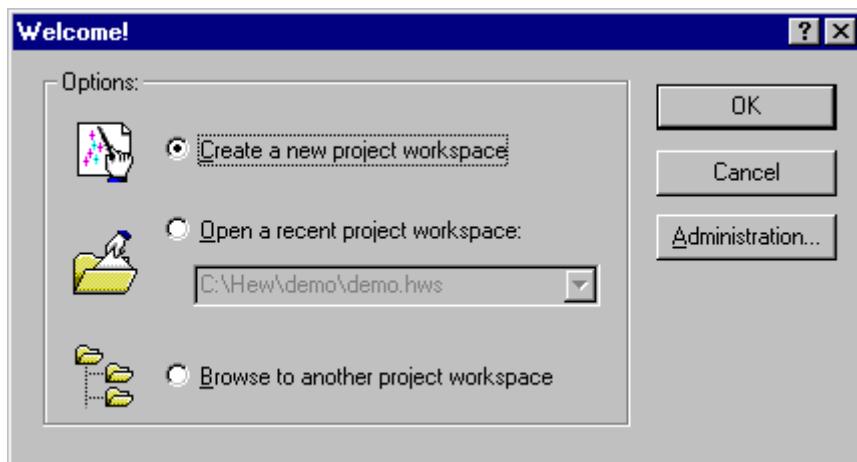


図 1.6: Welcome! ダイアログボックス

新規ワークスペースを作成するには“Create a new project workspace”を選択し、“OK”ボタンをクリックしてください。最近開いたワークスペースを開くには“Open a recent project workspace”を選択し、ドロップダウンリストから開きたいワークスペースを選択し、“OK”ボタンをクリックしてください。ワークスペースファイルを指定してワークスペースを開くには“Browse to another project workspace”を選択し、“OK”ボタンをクリックしてください。HEW にツールを登録したり、HEW からツールの登録を外したりするには“Administration”ボタンをクリックしてください。ワークスペースを開かないで HEW を使うには“Cancel”ボタンをクリックしてください。

1.3 新規ワークスペースを作成する

⌚ 新規にワークスペースを作成するには

- “Welcome!”ダイアログボックス（図1.6）から“Create a new project workspace”オプションを選び、“OK”ボタンをクリックするか、[File->New Workspace...]を選んでください。“New Project Workspace”ダイアログボックスが表示されます（図1.7）。

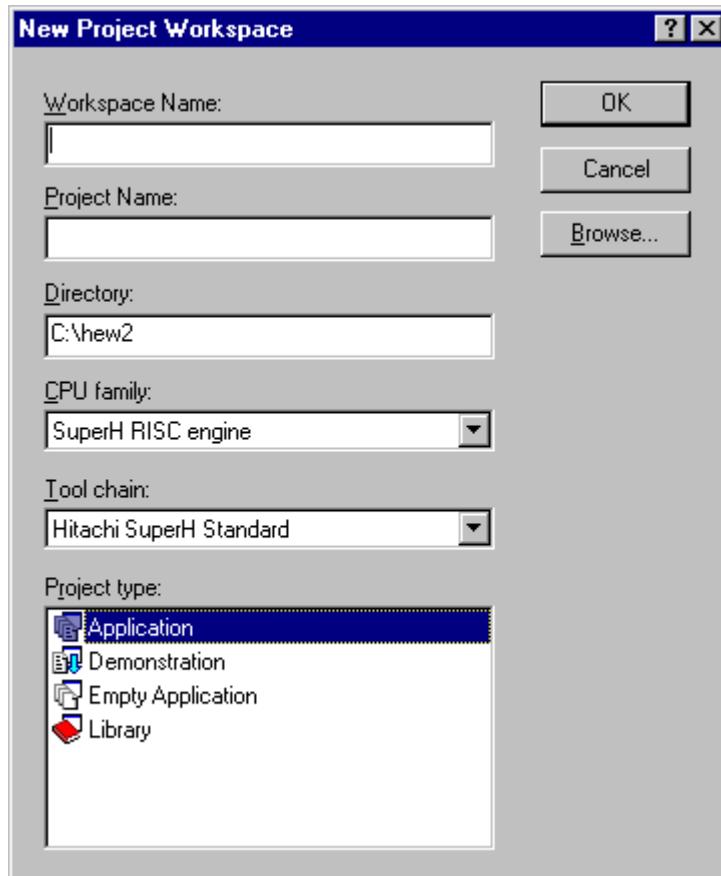


図 1.7: New Project Workspace ダイアログボックス

- “Workspace Name”フィールドに新規ワークスペース名を入力してください。新規ワークスペース名は32文字以内で、半角英数字、半角下線を使用できます。ワークスペース名を入力すると、自動的にワークスペースのサブディレクトリおよびプロジェクト名が“Directory”フィールドのディレクトリパスに追加されます。このディレクトリパスを使用する以外の方法もあり、この場合ワークスペース名とプロジェクト名が異なります。“Browse...”ボタンをクリックしてワークスペースを作成するディレクトリを選んだり、“Directory”フィールドに、ワークスペースを作成するディレクトリを手入力することができます。
- ワークスペースの基盤となるCPUファミリおよびツールチェインを選んでください。これらはワークスペースを作成した後で変更することができないので注意してください。
- 新規ワークスペースを作成するとき、HEWは自動的にプロジェクト名フィールドで指定されたプロジェクトを作成して、新規ワークスペースに追加します。“Project type”リストには、使用可能なプロジェクトの種類（アプリケーション、ライブラリなど）を表示します。作成するプロジェクトの種類をリストから選んでください。表示されるプロジェクトの種類は、現在のCPUファミリおよびツールチェインの組み合わせに有効な全種類となります。
- “OK”ボタンをクリックすると、新規のワークスペースとプロジェクトが作成されます。

注意 同一ディレクトリにすでにワークスペースが存在する場合は、ワークスペースを作成できません。

1.4 ワークスペースを開く

⌚ ワークスペースを開くには

1. “Welcome!”ダイアログボックス(図1.6)から“Browse to another project workspace”オプションを選んで“OK”ボタンをクリックするか、[File->Open Workspace...].を選んでください。“Open Project Workspace”ダイアログボックスが表示されます。
 2. 開きたいワークスペースファイルを選びます(.HWSファイルのみ)。
 3. “Open”ボタンをクリックしてワークスペースを開いてください。ワークスペースを開くときに情報を表示するように設定されている場合、ワークスペースプロパティダイアログボックスが表示されます(図1.8)。設定されていない場合、ワークスペースを開きます。
- ワークスペースプロパティダイアログボックスが表示されるかどうかはワークスペースプロパティダイアログボックスの“Show workspace information on workspace open”チェックボックス、または、“Tools Options”ダイアログボックス“Workspace”タブの“Display workspace information dialog on opening workspace”チェックボックスの有無によります。詳細は、Hitachi Embedded Workshop 2.0ユーザーズマニュアル5章、「環境のカスタマイズ」のワークスペースオプションの設定に関する説明を参照してください。ワークスペースプロパティダイアログボックスで“OK”をクリックするとワークスペースを開きます。“Cancel”ボタンをクリックするとワークスペースを開きません。

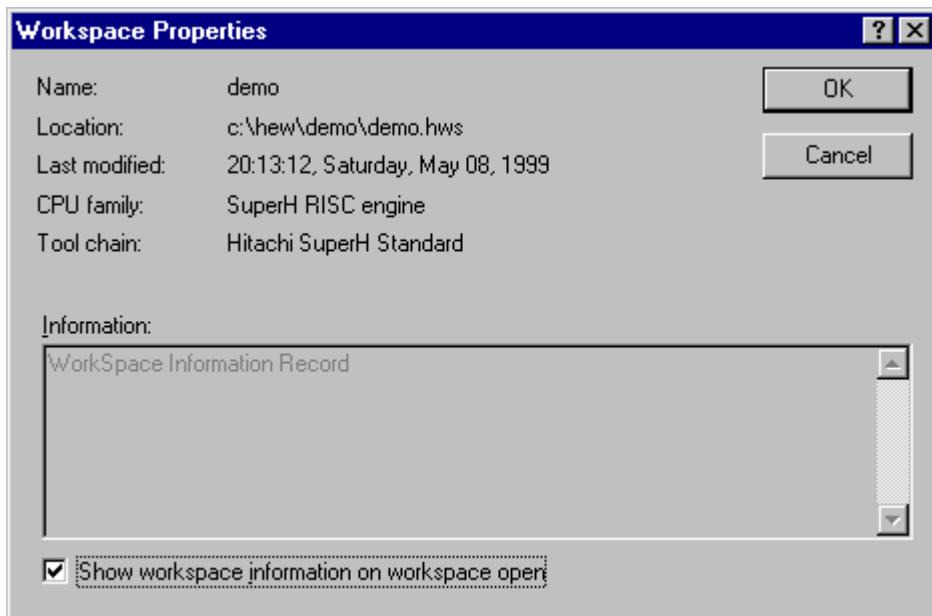


図 1.8: ワークスペースプロパティダイアログボックス

HEWでは、最近開いたファイル5つまでを“File”メニューの“Recent Workspaces”サブメニューに追加します。最近利用したファイルを再び開きたいときに便利です。

⌚ 最近使ったワークスペースを開くには

1. “Welcome!”ダイアログボックスから“Open a recent project workspace”を選び、ドロップダウンリストからワークスペース名を選び、“OK”ボタンをクリックします。
または、
2. [File->Recent Workspace]メニュー>オプションを選び、そのサブメニューからワークスペース名を選びます。

注意 HEWでは、一度に1つのワークスペースしか開けません。したがって、すでに開いているワークスペースがあるときに別のワークスペースを開こうとすると、すでに開いているワークスペースを閉じてから新しいワークスペースを開きます。

1.5 ワークスペースを保存する

[File->Save Workspace]メニュー オプションを選ぶと、HEW のワークスペースが保存できます。

1.6 ワークスペースを閉じる

HEW のワークスペースを閉じるには、[File->Close Workspace] メニュー オプションを選んでください。ワークスペースまたはそのプロジェクトに変更があった場合は、保存するかどうかを選んでください。

[File->Save Workspace]メニュー オプションを選ぶと、HEW のワークスペースが保存できます。

1.7 古いワークスペースの使用

HEW では、以前のバージョンの HEW で作成したワークスペースも開くことができます。これによって問題が発生することではなく、ワークスペースファイルの細部における違いもワークスペースを開いたときにアップグレードします。

注意 アップグレードしたプロジェクトは元の古い環境では使えません。以前の環境で使う場合は、初期のワークスペースまたはプロジェクトファイルのバックアップを保存しておく必要があります。

1.8 HEW を終了する

HEW を終了するには [File->Exit]を選ぶか、Alt+F4 キーを押下するか、システムメニューから“閉じる”オプションを選んでください。(システムメニューは HEW タイトルバーの最も左上側にあるアイコンをクリックすると開きます。) ワークスペースが開いているときは、前節で説明したワークスペースを閉じる操作を行います。

2. ビルドの基本

この章では HEW の一般的な機能を説明します。より高度な機能については 3 章 「ビルドの応用」を参照してください。

2.1 ビルド処理

ビルド処理の一般例を図 2.1 に示します。HEW のインストール時に提供されるツールによってビルド処理は変わるので、図 2.1 の例とは少し異なるかもしれません（例えば、コンパイラが無いなど）。いずれにせよ、原則は同じです。ビルドの各ステップまたはフェーズにおいて、1 セットのプロジェクトファイルについてビルド処理を行います。それが完了すると、次のステップまたはフェーズに移ります。

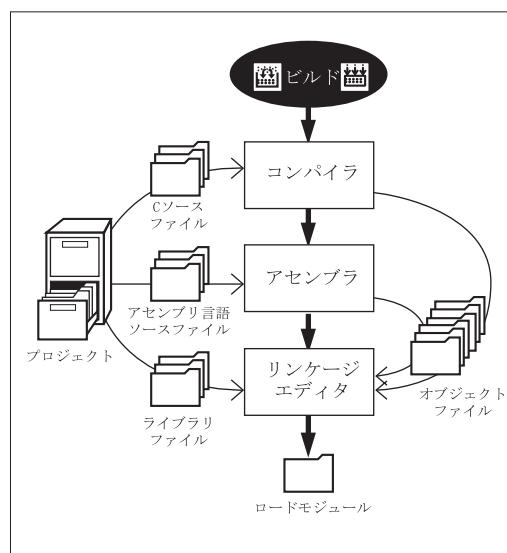


図 2.1: ビルド処理の一般例

図 2.1 に示す例では、第一のフェーズがコンパイラ、第二のフェーズがアセンブラ、そして最後のフェーズがリンクエディタです。コンパイラのフェーズでは、プロジェクトの C ソースファイルを順次コンパイルします。アセンブラのフェーズでは、アセンブリ言語のソースファイルを順次アセンブルします。リンクエディタのフェーズでは、すべてのライブラリファイルと、コンパイラフェーズとアセンブラフェーズからの出力ファイルをリンクして、ロードモジュールを作成します。このモジュールは、HEW のデバッガ機能でダウンロードし、使用します。

ビルド処理をカスタマイズする方法はいくつかあります。例えば、独自のフェーズを追加したり、あるフェーズを無効にしたり、フェーズを削除できます。これらのビルド実行の応用については、3 章「ビルドの応用」を参照してください。

この章では、ビルドの一般的な原則や基本機能を説明します。

2.2 プロジェクトファイル

HEW を使ってアプリケーションをビルド処理するには、まず、どのファイルをプロジェクトに追加して、各ファイルをどのようにビルド処理すべきかを指定しなければなりません（図 2.2）。

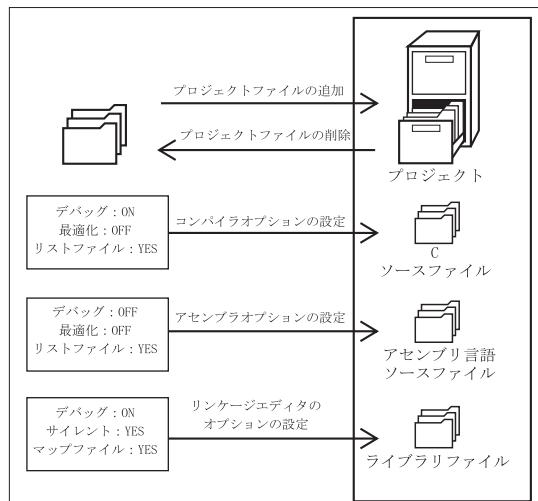


図 2.2: プロジェクトの編集

2.2.1 プロジェクトにファイルを追加する

アプリケーションをビルド実行する前に、まず、アプリケーションを構成するファイルを指定しなければなりません。

● プロジェクトにファイルを追加するには

- [Project->Add Files...] を選ぶか、ワークスペースウィンドウのポップアップメニューから [Add Files...] を選ぶ（図2.3参照）か、ワークスペースウィンドウを選んでINSキーを押下してください。

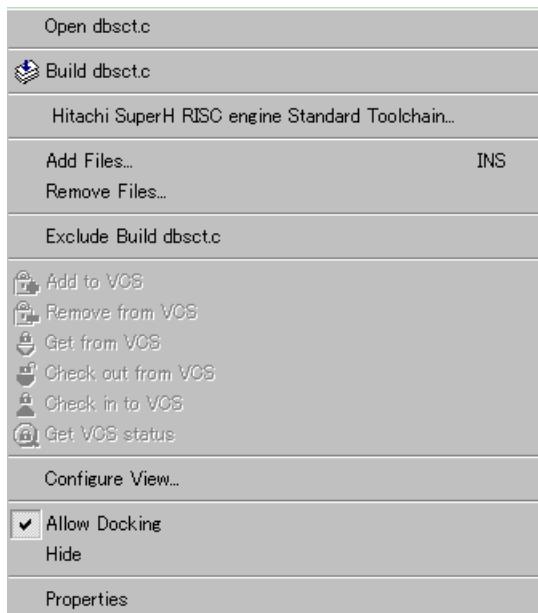


図 2.3: Project ポップアップメニュー

- “Add”ダイアログボックスが表示されます。
- 追加するファイルを選び“Add”ボタンをクリックしてください。

プロジェクトに新しいファイルを追加するには、数多くの方法があります。これを以下で説明します。

エディタウィンドウ内で開いたファイルを右クリックすると、エディタウィンドウにポップアップメニューが表示されます（図 2.4）。ファイルがすでにプロジェクト内にある場合、“Add File to Project”メニュー オプションは使用できません。“Add File to Project”を選択すると、現在のプロジェクトにファイルを追加できます。



図 2.4: エディタウィンドウポップアップメニュー

HEW では、Windows® Explorer からワークスペースウィンドウにファイルを“ドラッグしてドロップ”することができます。こうしたファイルは自動的にプロジェクトに追加され、ドラッグされた先のフォルダに表示されます。

注意 プロジェクトに追加するファイルが、HEW の認識できない形式のファイルであっても、プロジェクトに追加されます。このファイルに関して、いくつかの機能が使用できなくなります。エディタ内でファイルを開く代わりにワークスペースウィンドウ内でこのファイルをダブルクリックすると、ファイルを開く動作が Windows オペレーティングシステムに受け渡されます。ファイルを Windows® Explorer 内で開いたかのように、ファイルを開くデフォルト動作が実行されます。現在、定義されている拡張子については、“File Extensions”ダイアログボックスをご覧ください。（この章の後半にある「ファイルの拡張子とファイルグループ」を参照してください。）

2.2.2 プロジェクトからファイルを削除する

プロジェクトからファイルを削除できます。ファイルの削除は、一つでも、複数でも、すべてのファイルをまとめてでもできます。

⌚ プロジェクトからファイルを削除するには

- [Project->Remove Files...] を選択するか、ワークスペースウィンドウの“Projects”タブのポップアップメニューから [Remove Files...] を選択（図 2.5）してください。“Remove Project Files”ダイアログボックスが表示されます（図2.6）。

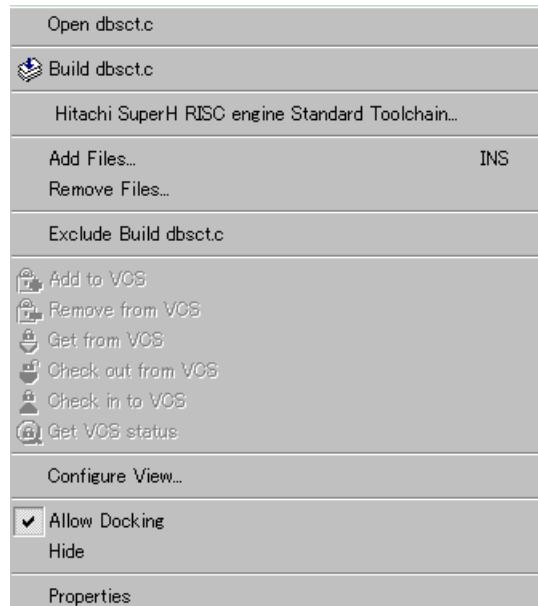


図 2.5: Project タブポップアップメニュー

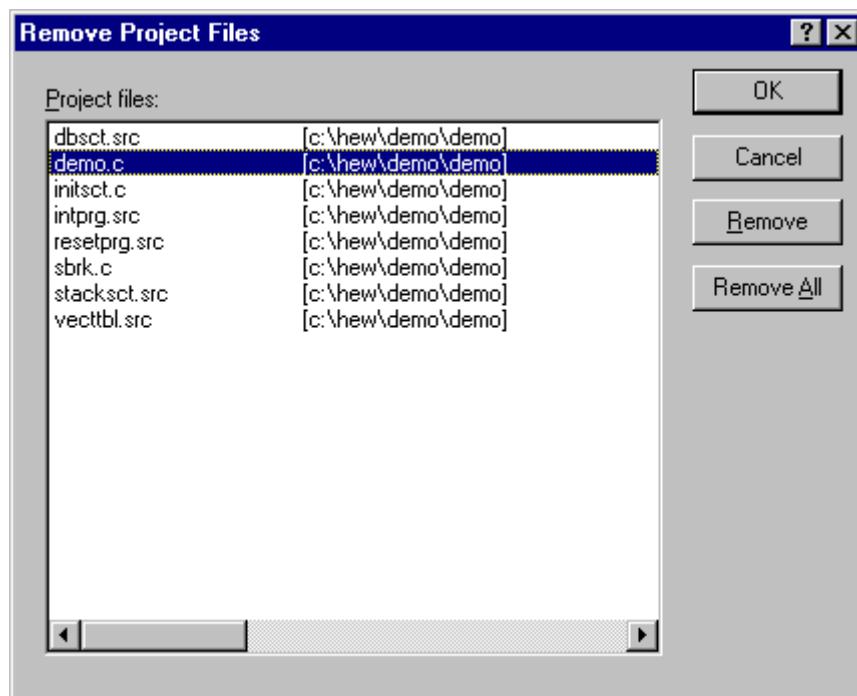


図 2.6: Remove Project Files ダイアログボックス

2. “Project files”リストから削除したいファイルを選んでください。複数のファイルを選ぶこともできます。
3. 選んだファイルを削除するには“Remove”ボタンをクリックしてください。すべてのプロジェクトファイルを削除するには“Remove All”ボタンをクリックしてください。
4. “OK”ボタンをクリックするとプロジェクトからファイルを削除します。

⌚ 選んだファイルをプロジェクトから削除するには

1. ワークスペースウィンドウの“Projects”タブで削除したいファイルを選んでください。複数のファイルを選ぶときはSHIFTキーまたはCTRLキーを押下してください。
2. DELキーを押してください。選んだファイルが削除されます。

2.2.3 ビルドからプロジェクトファイルを除外する

プロジェクトのファイルは、個々にビルドから除外することができます。

⌚ ビルドからプロジェクトのファイルを除外するには

1. ワークスペースウィンドウの“Projects”タブで、ビルドから除外したいファイルを右マウスボタンでクリックしてください。
2. ポップアップメニュー（図2.5）から[Exclude Build ファイル名]を選んでください。すると、ファイルのアイコンに赤いバツ印がつけられ、ビルドから除外されます。

2.2.4 ビルドへプロジェクトファイルを入れる

除外したプロジェクトのファイルは再びビルドに入れることができます。

⌚ 除外したファイルをビルドに再び入れるには

1. ワークスペースウィンドウの“Projects”タブでファイルを右マウスボタンで選んでください。
2. ポップアップメニューから[Include Build ファイル名]を選んでください。赤いバツ印が外され、ファイルがビルド可能になります。

2.3 ファイル拡張子とファイルグループ

HEW は拡張子でファイルを識別します。拡張子は使用するツールによって定義されます。例えば、コンパイラを使用すると拡張子.c が“C source file”グループに入り、コンパイラのフェーズに入力されます（図 2.1、ビルド処理の一般例）。さらに、独自の拡張子を定義することもできます。例えば、プロジェクトでアセンブリ言語ソースファイルを使っている場合、デフォルトの拡張子が.src だとします。.src の代わりに違う拡張子（例: .asm）を使うとき、新しい拡張子を定義してそれを.src ファイルと同様に扱うように HEW に要求できます。

ファイルの拡張子は、“File Extensions”ダイアログボックスで表示、変更できます（図 2.7）。このダイアログボックスを表示するには [Project->File Extensions...]を選んでください。このダイアログボックスには現在のワークスペースで定義されたすべての拡張子とファイルグループを表示します。

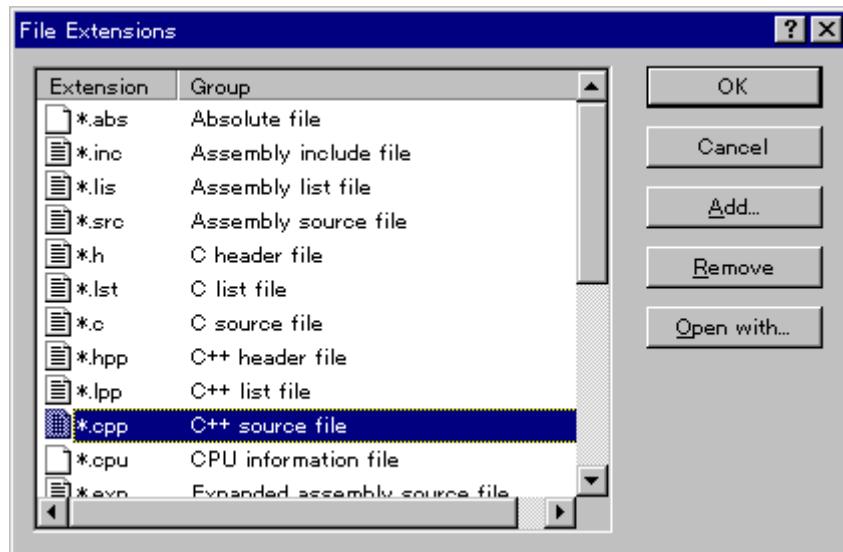


図 2.7: File Extensions ダイアログボックス

図 2.7 に示す“File Extensions”リストは 2 列に分かれています。左の列にはファイル拡張子、右の列にはファイルグループを表示します。図 2.8 に示すように、同じグループに多くのファイル拡張子が存在する場合があります。例えば、1 つのプロジェクト内でアセンブリ言語のソースファイルにいくつかの拡張子がある場合があります（例: .src, .asm, .mar など）。

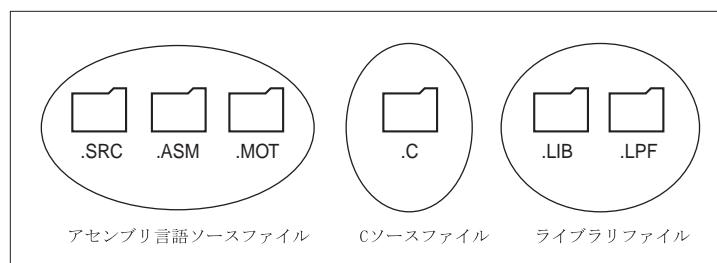


図 2.8: ファイル拡張子とファイルグループ

新しい拡張子を作成するときは、その拡張子がすでに定義されたファイルグループに属するのか、新しいファイルグループを作成する必要があるのか検討してください。新しい種類のファイルを追加するときには新しいファイルグループを作成してください。作成方法を以下に説明します。

⌚ 新規ファイルグループに新規ファイル拡張子を作成するには

1. メニューバーから[Project->File Extensions...] を選んでください。“File Extensions”ダイアログボックスが表示されます(図 2.7)。
2. “Add...”ボタンをクリックしてください。“Add File Extension”ダイアログボックスが表示されます(図2.9)。
3. “File extension”フィールドに定義する拡張子を入力してください。ピリオド(.) の入力は不要です。ドロップリストには、現在のプロジェクトで未定義のすべての拡張子が含まれます。この拡張子のうち1つを選ぶと、ファイル拡張子フィールドにテキストが自動的に追加されます。
4. “Extension belongs to a new group”オプションを選んで新しいファイルグループを定義する名前を入力してください。
5. この段階では、関連するアプリケーションの変更が可能です。“Open”的ドロップダウンリストでは次の4つから選択が可能です。
 - “Editor”
 - “None”
 - “Other”
 - “Windows default”
 “Editor”を選択すると、ワークスペースウィンドウ内のファイルを開く機能によって、ファイルをHEWエディタ内で開くことができます。“None”を選択した場合、ファイルを開く機能を使用してもファイルを開くことはできません。“Other”を選択すると、ファイルを開くための他のツールを構築できます。詳細は、「アプリケーションとファイルグループを関連付けるには」を参照してください。“Windows default”を選択すると、ワークスペースウィンドウ内のファイルを開く機能によって、開いたファイルがWindowsオペレーティングシステムに受け渡されます。また、Windows® Explorerで定義されたとして、このファイル拡張子のデフォルト動作を選択します。
6. “OK”ボタンをクリックすると“File Extension”リストに拡張子が追加されます。

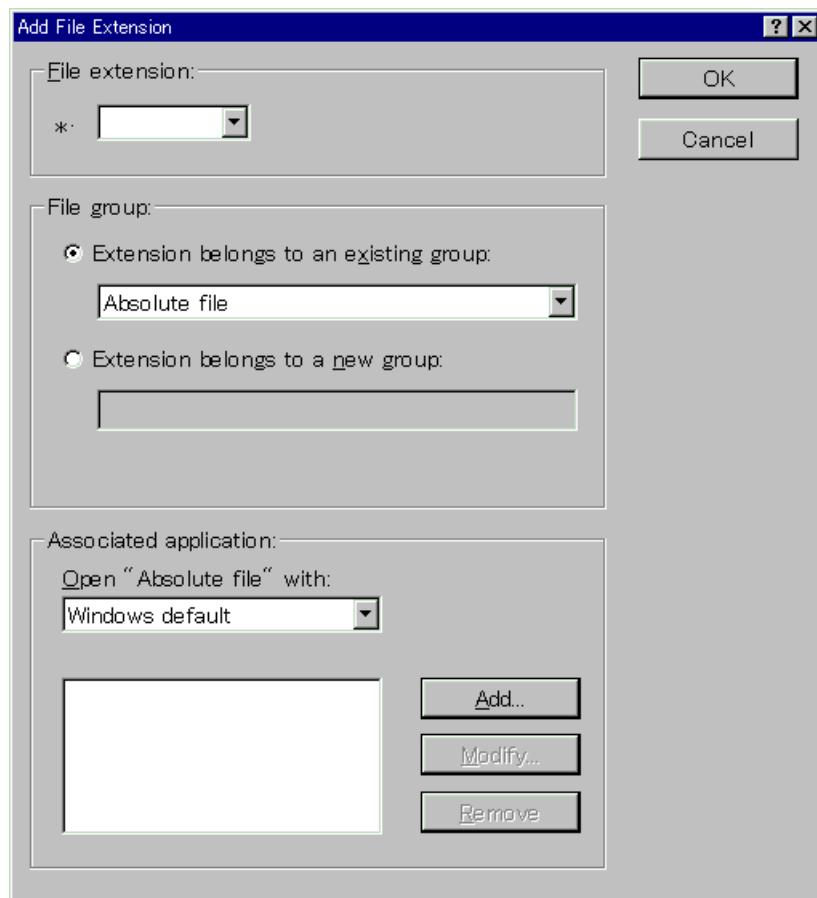


図 2.9: Add File Extension ダイアログボックス
(新規グループ)

ビルダに認識されない拡張子をもつプロジェクトを使っているときは、新しい拡張子を作成し、(例えば、フェーズ内のデフォルトの拡張子が.asm でビルダの認識する拡張子が.src のとき) 新しい拡張子を既存のファイルグループに追加することが必要です。追加方法を下記に説明します。

⌚ 新規ファイル拡張子を既存のファイルグループに追加するには

1. メニューバーから [Project->File Extensions...] を選んでください。“File Extensions”ダイアログボックスが表示されます(図 2.7)。
2. “Add...”ボタンをクリックすると “Add File Extension”ダイアログボックスが表示されます(図2.10)。
3. “File extension”フィールドに定義する拡張子を入力してください。ピリオド(.) の入力は不要です。ドロップリストには、現在のプロジェクトで未定義のすべての拡張子が含まれます。この拡張子のうち1つを選ぶと、ファイル拡張子フィールドにテキストが自動的に追加されます。
4. “Extension belongs to an existing group”オプションを選んでこの新しい拡張子をどのファイルグループに追加するか指定してください。
5. “OK”ボタンをクリックすると“File Extensions”リストに拡張子が追加されます。

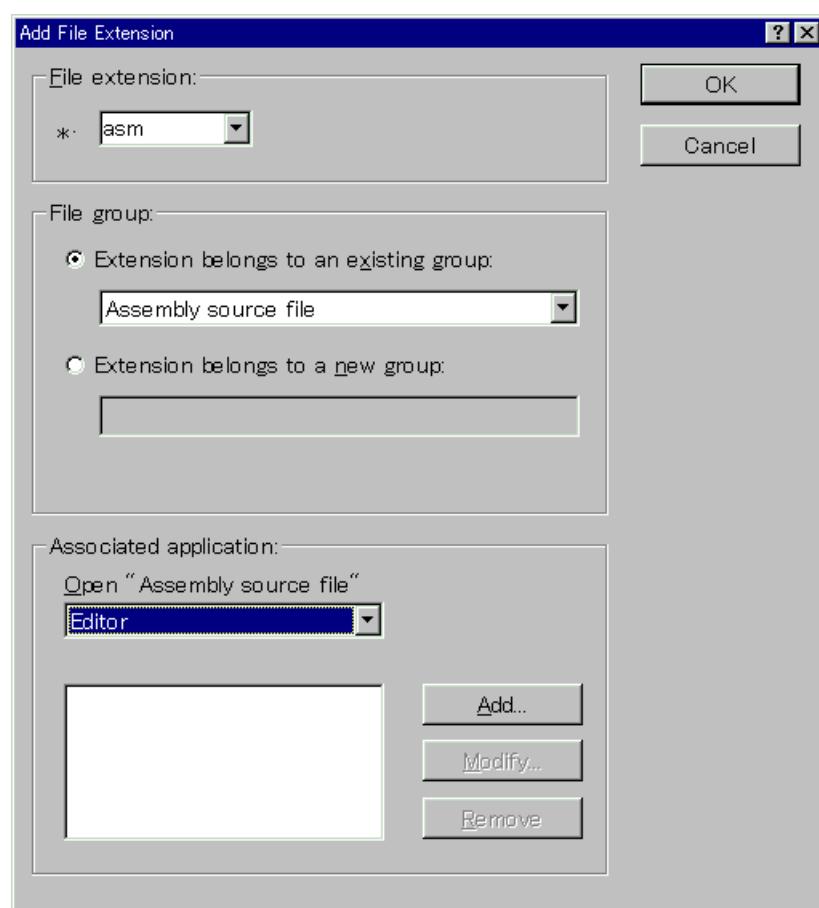


図 2.10: Add File Extension ダイアログボックス
(既存グループ)

“File Extensions”ダイアログボックスでは、エディタでファイルを開く指定だけでなく、ファイルグループとアプリケーションとの関連付けができます。これを行うと、ワークスペースウィンドウの“Projects”タブでファイルをダブルクリックすると、適切なアプリケーションでファイルを開きます。図 2.11 にワードプロセッサと拡張子.DOC の関連付けを示します。

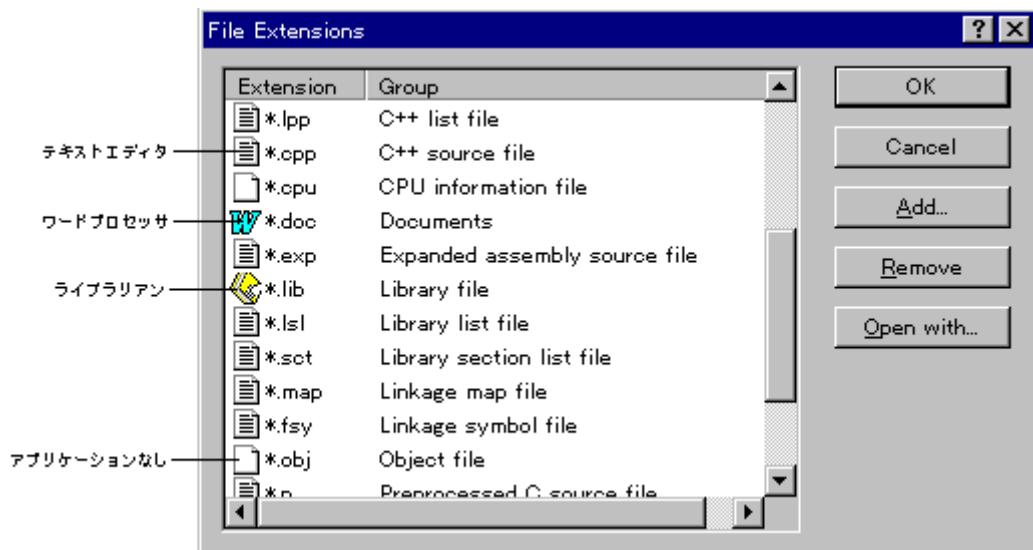


図 2.11: ファイルグループとアプリケーション

⌚ アプリケーションとファイルグループを関連付けるには

1. “File Extensions”ダイアログボックスで関連付けるファイルグループを選んでください(図2.11)。
2. “Open with...”ボタンをクリックしてください。“Modify File Extension”ダイアログボックスが表示されます(図 2.12)。

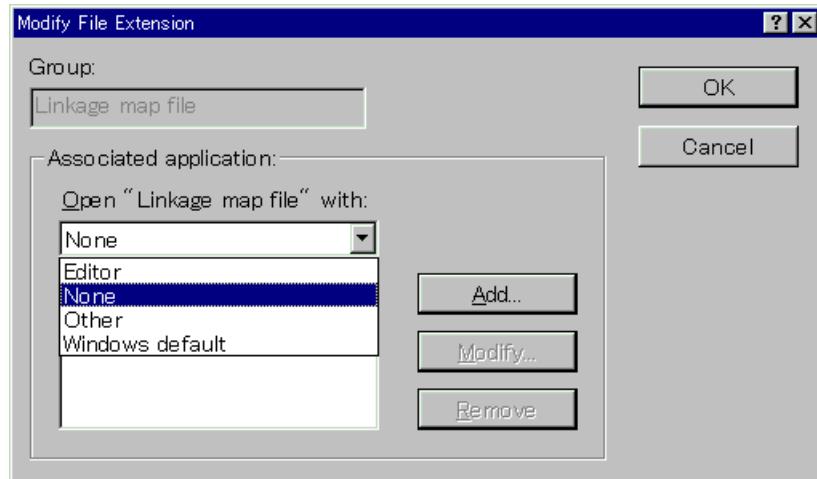


図 2.12: Modify File Extension ダイアログボックス

3. 関連付けをしない場合は“None” выбираете。HEWのエディタまたは他のエディタでこの種類のファイルを開くには“Editor”を選択します。特定のアプリケーションでこの種類のファイルを開くには“Other”を選択します。“Other”を選択すると、すでに定義されたアプリケーションをドロップダウンリストから選ぶか、または、新しいアプリケーションを定義することができます。
4. 新しいアプリケーションを定義する場合は“Add...”をクリックしてください。“Add Application”ダイアログボックスが表示されます(図2.13)。

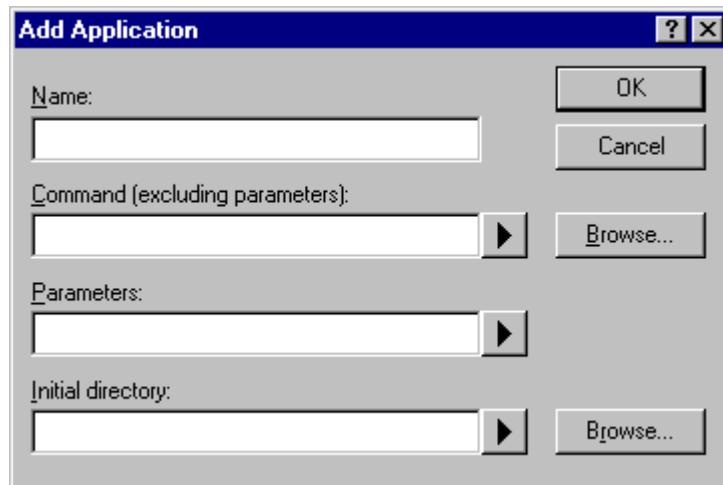


図 2.13: Add Application ダイアログボックス

5. “Name”フィールドにアプリケーション名を入力します。“Command”フィールドにアプリケーションのフルパスを入力します(パラメータは含めません)。“Parameters”フィールドにファイルを開くのに必要なパラメータを入力します。必ず\$(FULLFILE) プレイスホルダを使って入力ファイルを指定してください。(プレースホルダの詳細と使用方法については、Hitachi Embedded Workshop 2.0ユーザーズマニュアル付録 C、「プレースホルダ」を参照してください。)“Initial directory”フィールドにアプリケーションを実行させる初期ディレクトリを入力します。“OK”ボタンをクリックするとアプリケーションが定義されます。
6. アプリケーションを変更するには“Modify...”ボタンを押してください。“Modify Application”ダイアログボックスが表示されます。このダイアログボックスは、“Name”フィールドが変更できないことを除いて“Add Application”ダイアログボックスと同じです。設定を変更してから“OK”ボタンをクリックします。
7. “OK”ボタンをクリックすると選んだファイルグループに対するアプリケーションが関連付けられます。

2.4 ファイルのビルド方法を設定する

プロジェクトに必要なファイルを追加したら、次のステップは HEW に各ファイルのビルドを指示することです。このためには、“Options”メニューからメニュー オプションを選ばなければなりません。このメニューの内容は使用するツールによって異なります。例えば、コンパイラ、アセンブラー、リンクエディタを使う場合、それに対応する 3 つのメニュー オプションがあります。

❸ ビルドフェーズにオプションを設定するには

1. オプションを変更するフェーズを“Options”メニューから選んでください。
2. オプションを指定するダイアログボックスが表示されます。
3. オプションを設定して、“OK”ボタンをクリックしてください。

詳細は、コンテキスト依存のヘルプボタンをクリックするか、ヘルプを参照したい場所を選び F1 キーを押してください。

2.5 ビルドのコンフィグレーション

HEW では、ビルドのコンフィグレーションの中に、すべてのビルドのオプションを格納できます（図 2.14）。つまり、すべてのオプションを格納してそれらに名前を付けることができます。後にそのコンフィグレーションを選ぶと、各ビルドフェーズの各オプションを復帰させることができます。また、こうしたビルドのコンフィグレーションにより、ユーザはビルドのコンフィグレーション用にデバッガの設定を行うことができます。つまり、コンフィグレーションごとに異なるエンドプラットフォームをターゲットとすることができます。（詳細は、Hitachi Embedded Workshop 2.0 HEW デバッガユーザーズマニュアル 2 章「HEW プロジェクトをデバッグする」を参照してください。）

図 2.14 に“Default”、“MyDebug”、“MyOptimized”の 3 つのビルドコンフィグレーションを示します。“Default”ビルド構成では、各フェーズ（コンパイルとアセンブル）が標準設定されています。“MyDebug”ビルド構成では、各ファイルがデバッグ情報付きでビルドされています。“MyOptimized”ビルド構成では、各ファイルが最大限に最適化されデバッグ情報はありません。このプロジェクトの開発者は、オプションを設定するダイアログボックスに戻ってこれらを設定することなく、これらのビルドコンフィグレーションのうちどれでも選ぶことができます。

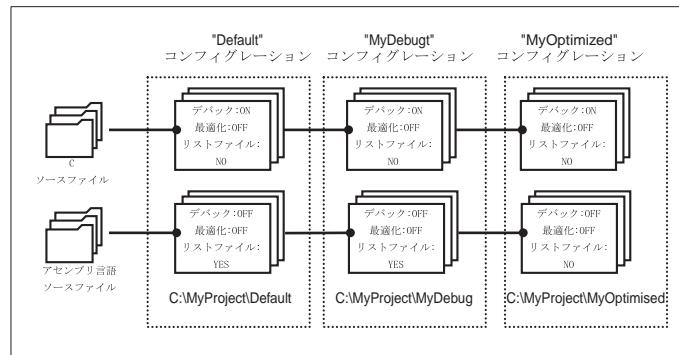


図 2.14: ビルドコンフィグレーションとファイルオプション

2.5.1 ビルドコンフィグレーションを選択する

使用するビルドコンフィグレーションを設定する方法には二つあります。

- ビルドコンフィグレーションを追加するには
1. ツールバーのドロップダウンリストボックス（図2.13）から選んでください。

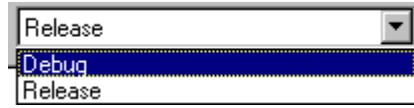


図 2.15: ツールバーの選択

または

1. [Options->Build Configurations...] を選ぶと、“Build Configurations”ダイアログボックスが表示されます（図2.14）。

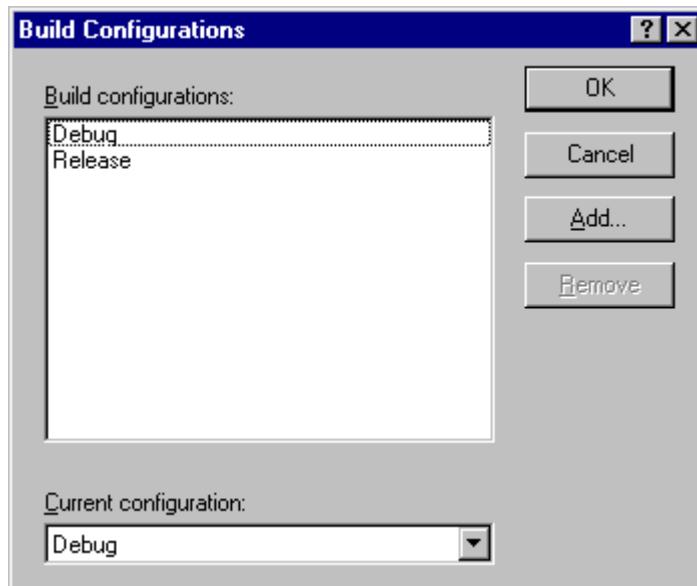


図 2.16: Build Configurations ダイアログボックス

2. “Current configuration”から使用するビルド構成を選んでください。
3. “OK”ボタンをクリックするとビルドコンフィグレーションが選ばれます。

2.5.2 ビルドコンフィグレーションを追加、削除する

ビルドコンフィグレーションの設定を他のビルドコンフィグレーションからコピーして追加したり、ビルドコンフィグレーションを削除したりできます。これらの操作を以下に説明します。

② ビルドコンフィグレーションを追加するには

1. [Options->Build Configurations...] を選ぶと“Build Configurations”ダイアログボックスが表示されます（図 2.16）。
2. “Add...”ボタンをクリックすると“Add Configuration”ダイアログボックスが表示されます（図 2.17）。

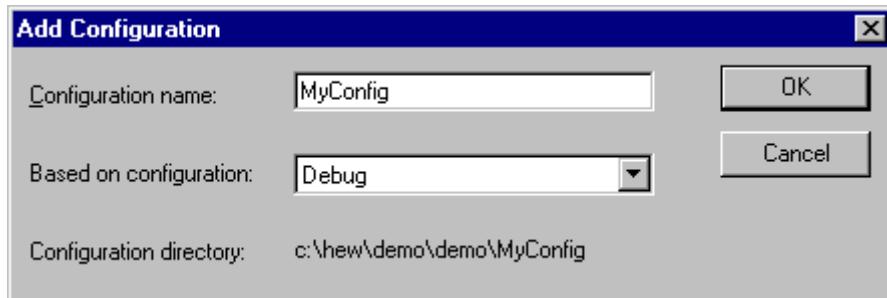


図 2.17: Add Configuration ダイアログボックス

3. “Configuration name”フィールドに新しいビルドコンフィグレーション名を入力してください。入力すると、下に表示されるディレクトリがビルドコンフィグレーションに使われるディレクトリに変わります。“Based on configuration”フィールドのドロップダウンリストにある既存コンフィグレーションの中から、コンフィグレーションの設定をコピーする元となるコンフィグレーションを選択します。両方のダイアログボックスの“OK”ボタンをクリックすると新しいビルドコンフィグレーションが作成されます。

③ ビルドコンフィグレーションを削除するには

1. [Options->Build Configurations...] を選ぶと“Build Configurations”ダイアログボックスが表示されます（図 2.16）。
2. 削除するビルドコンフィグレーションを選び“Remove”ボタンをクリックしてください。
3. “OK”ボタンをクリックすると“Build Configurations”ダイアログボックスを閉じます。

2.6 プロジェクトをビルド実行する

ビルド実行の概要は図 2.1 を参照してください。

2.6.1 プロジェクトをビルド実行する

“Build”オプションでは前回のビルド実行後に変更のあったファイルだけをコンパイルまたはアセンブルします。さらに、前回のビルド実行以後に変更のあったファイルに依存するソースファイルを再ビルド実行します。例えば、“test.c”にファイル “header.h”が含まれており “header.h”が前回のビルド実行以後に変更された場合、ファイル “test.c”が再コンパイルされます。

⌚ ビルド実行するには

- [Build->Build] を選ぶか、“ビルド”ツールバー[ボタン] をクリックするか、F7キーを押下してください。または、ワークスペースウィンドウの“Projects”タブのプロジェクトアイコン上で右マウスボタンをクリックしてポップアップメニューから[Build]を選んでください。

“Build All”オプションでは変更の有無に関わらず、すべてのソースファイルをコンパイルまたはアセンブルして、新しく作成されたオブジェクトファイルをすべてリンクします。

⌚ “Build All”処理を実行するには

- [Build->Build All] を選ぶか、“全ビルド”ツールバー[ボタン] をクリックしてください。または、ワークスペースウィンドウの“Projects”タブのプロジェクトアイコン上で右マウスボタンをクリックしてポップアップメニューから[Build All]を選んでください。

プロジェクトファイルのどれかにエラーが起こると、“Build”処理も“Build All”処理も終了します。

2.6.2 1つのファイルをビルド実行する

プロジェクトにある 1 つのファイルをビルド実行できます。

⌚ 1つのファイルをビルド実行するには

- プロジェクトウィンドウからビルド実行するファイルを選んでください。
- [Build->Build File]を選ぶか、“ビルドファイル”ツールバー[ボタン] をクリックするか、CTRL+F7キーを押してください。または、ワークスペースウィンドウの“Projects”タブのプロジェクトアイコン上で右マウスボタンをクリックして、ポップアップメニューから [Build <ファイル名>]を選んでください。

2.6.3 ビルド実行を中止する

ビルド実行を途中で中止できます。

⌚ ビルド実行を中止するには

- [Build->Stop Build] を選ぶか、“ビルド中止”ツールバー[ボタン] をクリックしてください。その時点のファイルのビルド実行を完了後、ビルド実行は中止されます。
- アウトプットウィンドウに“Build Finished”というメッセージが表示されるのを確認してから操作を続けてください。

⌚ ビルド実行中のツールを強制的に中止するには

- [Build->Terminate Current Tool]を選んでください。ビルダはツールの実行をすぐに中止します。

注意 中止したツールによって出力されたファイルは有効ではない場合があります。作成した出力ファイルをすべて削除して、そのフェーズを再実行してください。

2.6.4 複数のプロジェクトをビルド実行する

HEW を使用することで、複数のプロジェクトやコンフィグレーションのビルド処理が行えます。

⌚ 複数のプロジェクトをビルド実行するには

1. [Build->Build Multiple]を選択してください。図2.18に示します。
2. 複数のビルド実行では、どのプロジェクトまたはコンフィグレーションを処理するかを選択できます。どのプロジェクトまたはコンフィグレーションを実行するか選択するには、実行したいプロジェクトとコンフィグレーションの組み合わせの横にあるチェックボックスを選んでください。例えば、プロジェクト “hewtest2”全体にビルド実行したい場合、“hewtest2-Debug”、“hewtest2-Release”を選んでチェックし、その他のボックスのチェックを外してください。
3. 選んだ項目に満足したら、“Build”ボタンをクリックしてください。選んだプロジェクトやコンフィグレーションのビルド実行をHEWが行います。
4. また、選んだ項目に対して、全ビルド実行したい場合、“Build all”ボタンをクリックしてください。
5. 通常のビルド実行と同じ方法で、ビルドの結果がビルドウィンドウに表示されます。

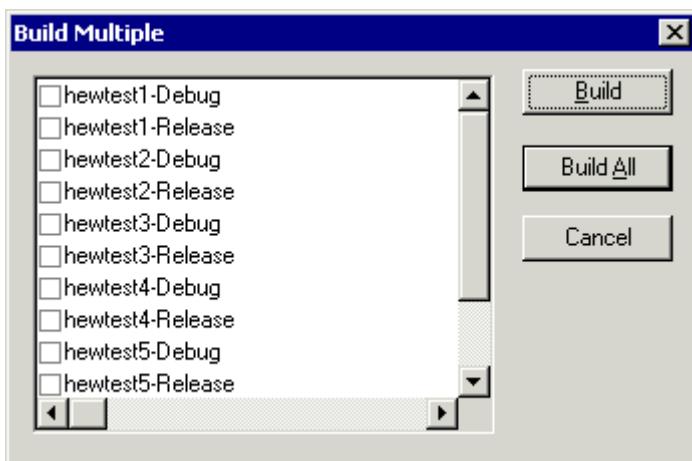


図 2.18 Build Multiple ダイアログボックス

2.6.5 アウトプットウィンドウ

ツール（コンパイラ、アセンブラー、リンクエディタなど）が実行されると、その出力がアウトプットウィンドウに表示されます。エラーまたはウォーニングが起きると、エラーメッセージまたはウォーニングメッセージと、ソースファイル名と行番号が表示されます。すぐにエラーまたはウォーニングが発生したところをエディタで見るには、表示されたエラーメッセージまたはウォーニングメッセージをダブルクリックします。

2.6.6 アウトプットウィンドウの内容の制御

ビルド実行の途中に低レベル情報（ファイルに適用したコマンドラインオプションなど）を表示すると便利です。HEW では、“Build”、“Build All”、“Build File”処理中、アウトプットウィンドウにそのオプションを表示するかどうかを“Tools Options”ダイアログボックスで指定できます。

⌚ ビルド実行中に低レベル情報の表示の有無を指定するには

1. [Tools->Options...]を選ぶと“Options”ダイアログボックスが表示されます。
2. “Build”タブ（図2.19）を選んでください。
3. “Show”グループのチェックボックスを以下のように設定します。“Command line”にはツール実行時のコマンドライン表示の有無を指定します。“Environment”にはツール実行時の環境変数の表示の有無を指定します。“Initial directory”にはツールが起動されるディレクトリパスの表示の有無を指定します。

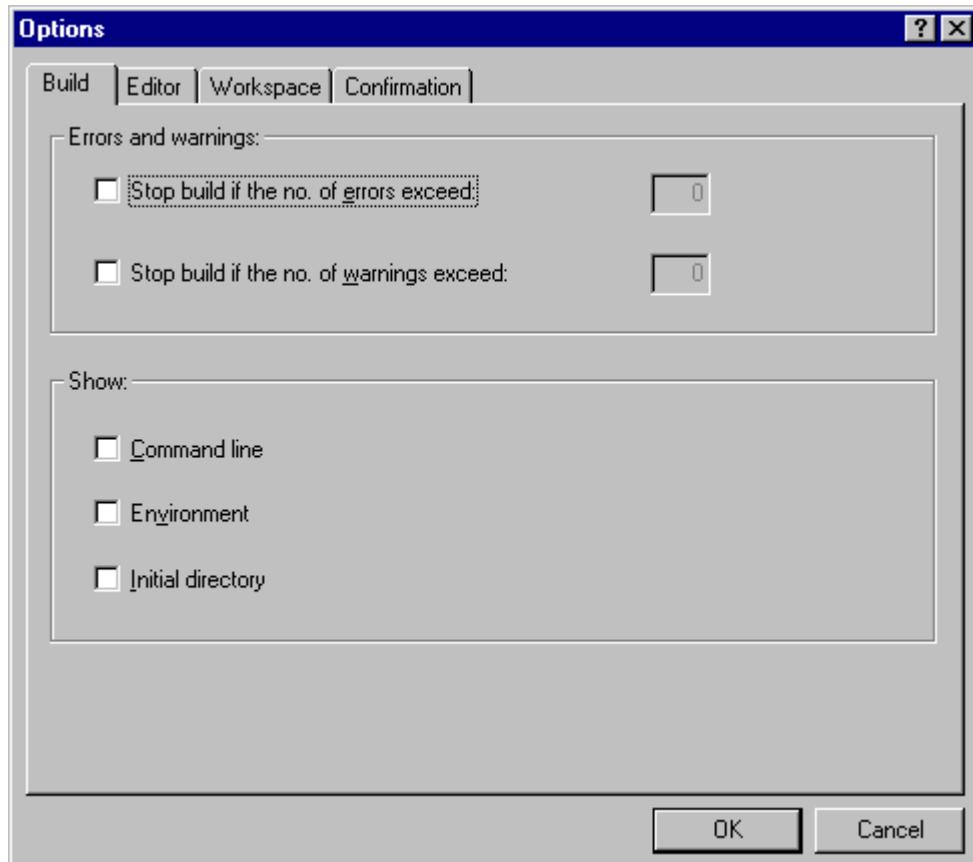


図 2.19: Options ダイアログボックス Build タブ

2.7 ファイル依存関係

多くの場合プロジェクトにはファイル間の依存関係があります。例えば、1つの C ファイルはいくつかのヘッダファイルをインクルードします。複雑なプロジェクトでは、ソースファイルが他のインクルードファイルに依存するため、管理が難しくなります。しかし、HEW にはファイル依存関係をスキャンする機能があり、そのプロジェクトにあるすべてのファイルの依存関係をチェックできます。スキャンが完了すると、プロジェクトのファイル依存関係を示す最新のリストをプロジェクトウィンドウに表示します。

⌚ プロジェクトのファイル依存関係を更新するには

1. [Build->Update All Dependencies]を選んでください。または、ワークスペースウィンドウの“Projects”タブのプロジェクトアイコンを右マウスボタンでクリックしてポップアップメニューから[Update All Dependencies]を選んでください。

最初に、すべてのファイル依存は“Dependencies”フォルダに含まれています（図 2.20.(i)）。

2.8 ワークスペースウィンドウの構成

ワークスペースウィンドウの“Projects”タブの中を右マウスでクリックすると、ポップアップメニューが表示されます。その中から、“Configure View...”メニューオプションを選び、以下の情報の表示方法を設定してください。以下に、“Configure View”ダイアログボックスの各オプションについて説明します。

2.8.1 各ファイルの下に依存を表示する

“Show dependencies under each file”を選ぶと、依存ファイルがそれをインクルードするソースファイルの下に平坦な構造で表示されます（ファイル自体がフォルダになります）。これを図 2.20.(ii)に示します。このオプションを選ばないと、別のフォルダにすべてのファイル依存を示します（図 2.20.(i)）。

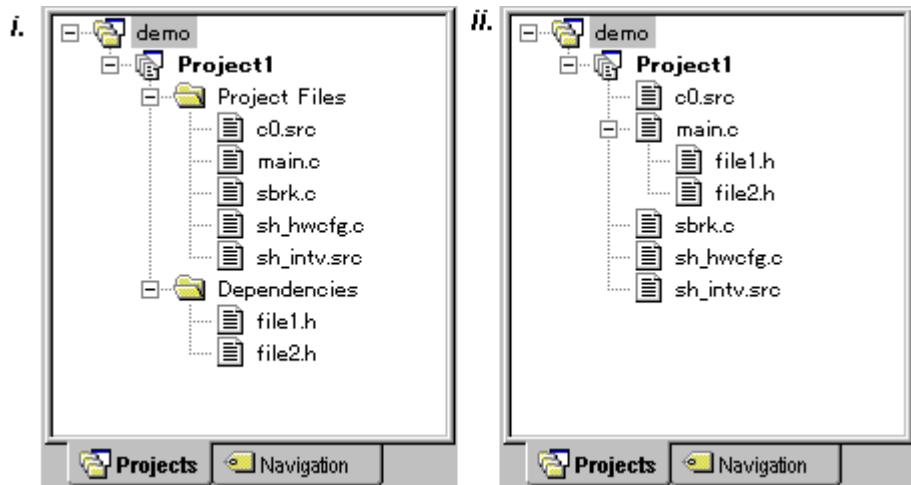


図 2.20: 各ファイルの依存

2.8.2 標準ライブラリファイルのインクルードを表示する

デフォルトでは、標準インクルードパスの依存ファイルは表示されません（図 2.21.(i)）。例えば、C コードで #include <stdio.h>などのインクルード文を書くと、“stdio.h”は依存ファイルとして表示されません。そのようなシステムインクルードファイルを表示するには、“Show standard library includes”オプションを選んでください（図 2.21.(ii)）。

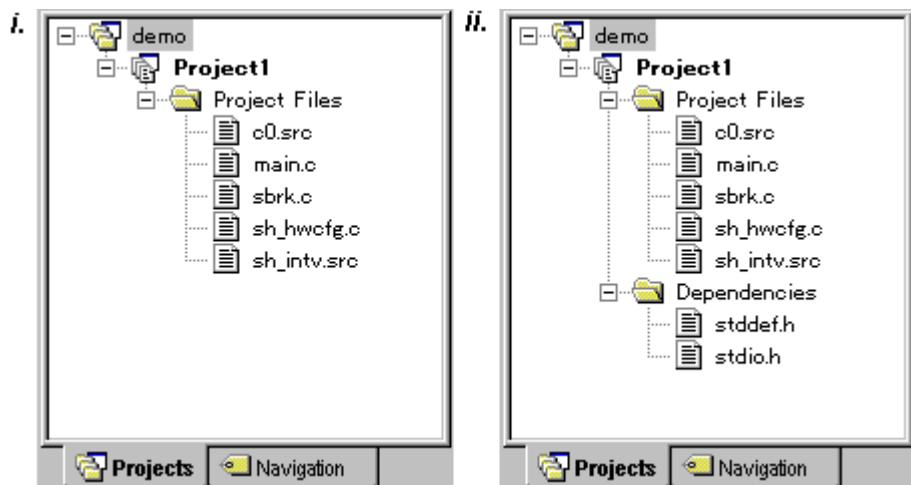


図 2.21: 標準ライブラリファイルのインクルード

2.8.3 ファイルのパスを表示する

“Show file paths”を選ぶと、ワークスペースウィンドウのすべてのファイルがフルパス（ドライブ名からのパス）で表示されます（図 2.22）。

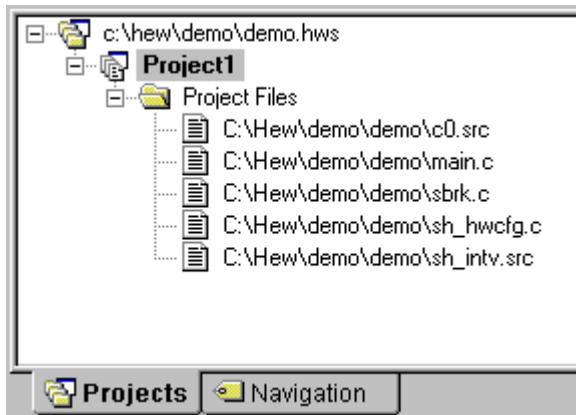


図 2.22: ファイルのパスの表示

2.9 アクティブプロジェクトを設定する

ワークスペースには複数のプロジェクトを含めることができます、ひとつだけがアクティブです。このアクティブプロジェクトでビルド動作とデバッグ動作が実行されます。そして、そのプロジェクト用のビルダやデバッガのオプションを変更できます。また、アクティブプロジェクトは太字で表示されます。

- ⌚ プロジェクトをアクティブにするには
 - 1. ワークスペースウィンドウの “Projects”タブからプロジェクトを選んでください。
 - 2. 右マウスボタンをクリックしてポップアップメニューから[Set as Current Project] オプションを選んでください。
または、
 - 1. 非アクティブのプロジェクトを [Project->Set Current Project] サブメニューから選んでください。

2.10 ワークスペースにプロジェクトを追加する

ワークスペースを作成したとき、最初はプロジェクトが一つしかありません。しかし、後で、新しいプロジェクトや既存のプロジェクトを追加することができます。

⌚ ワークスペースに新しいプロジェクトを追加するには

1. [Project->Insert Project...]を選んでください。“Insert Project” ダイアログボックスが表示されます(図 2.23)。
2. “New Project” オプションを設定してください。
3. “OK” ボタンをクリックしてください。“Insert New Project” ダイアログボックスが表示されます。
4. “Name” フィールドにプロジェクト名を入力してください。32文字以内で、半角英数字、半角下線が入力できます。プロジェクト名を入力すると、HEWは自動的にサブディレクトリを追加します。これは不要であれば削除できます。
5. “Browse...” ボタンをクリックしてプロジェクトを作成するディレクトリを選んでください。または、“Directory” フィールドにディレクトリを入力できます。
6. “Project type” リストには使用可能なプロジェクトの種類を示します(アプリケーション、ライブラリなど)。このリストから作成するプロジェクトの種類を選んでください。
7. “OK” ボタンをクリックすると、プロジェクトが作成されワークスペースに追加されます。

注意 新しいプロジェクトを作成するとき、CPU やツールチェインは指定できません。これらのプロパティはワークスペースにより定義されています。つまり、同じワークスペース内のすべてのプロジェクトは同じ CPU とツールチェインであることになります。

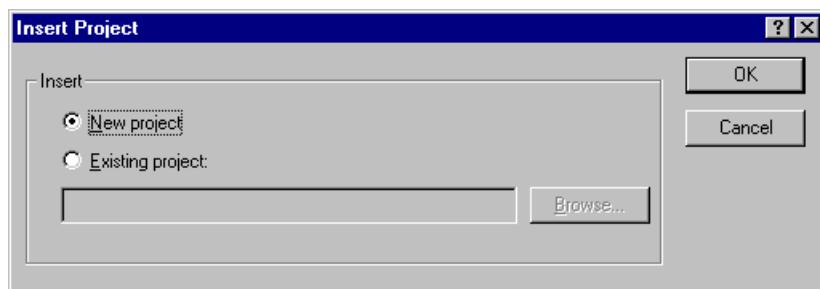


図 2.23: Insert Project ダイアログボックス

⌚ 既存のプロジェクトをワークスペースに追加するには

1. [Project->Insert Project...]を選んでください。“Insert Project” ダイアログボックスが表示されます。
2. “Existing Project” オプションを設定してください。
3. プロジェクトデータベースファイル(.HWPファイル)へのフルパスを入力するか、“Browse...” ボタンをクリックしてプロジェクトデータベースファイルを指定してください。
4. “OK” ボタンをクリックするとそのプロジェクトがワークスペースに追加されます。

注意 既存のプロジェクトをワークスペースに追加するとき、プロジェクトの CPU とツールチェインがワークスペースのものと一致しなければなりません。一致しない場合、そのプロジェクトはワークスペースに追加できません。

2.11 プロジェクト間の依存関係を指定する

ワークスペースのプロジェクトは、他のプロジェクトに依存することができます。ビルド処理をすると、依存プロジェクトが最初にビルドされます。これは、ワークスペースのプロジェクトを他のプロジェクトが使用しているときなどに便利です。例えば、ワークスペースに2つのプロジェクトがあるとします。1つはアプリケーションプロジェクトに含まれたライブラリだとします。この場合2番目のアプリケーションのビルド前にライブラリは正確にビルドされ、また最新でなくてはなりません。そのため、ライブラリをアプリケーションプロジェクトの依存プロジェクトに指定します。こうすると、最新でないライブラリが先にビルドされます。

依存プロジェクトをビルドするとき、HEWは依存プロジェクトがアクティブプロジェクトのビルドコンフィグレーションになるようにします。上記の例では、アクティブプロジェクトのビルドコンフィグレーションが“Debug”であるとき、HEWは、依存プロジェクトで“Debug”ビルドコンフィグレーションが選択されるようにします。このような一致したコンフィグレーションが存在しない場合、HEWは依存プロジェクトで最近使われたコンフィグレーションを使用します。

⌚ 依存プロジェクトを作るには

1. [Project->Dependent Projects]を選んでください。“Dependent Projects”ダイアログボックスが表示されます(図 2.24)。
2. 依存させたいプロジェクトを選んでください。“Dependent projects”リストに(選択したプロジェクト以外の)ワークスペース内のすべてのプロジェクトが表示されます。
3. “Dependent projects”リストには各プロジェクトにチェックボックスがあります。選んでプロジェクトが依存するプロジェクトのチェックボックスをチェックしてください。
4. “OK”ボタンをクリックしてください。

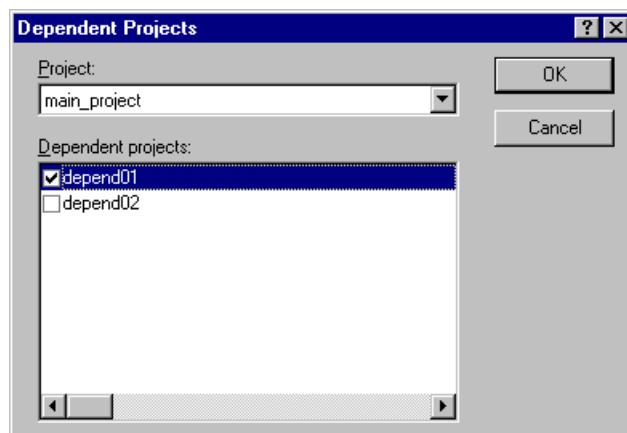


図 2.24: Dependent Projects ダイアログボックス

2.12 ワークスペースからプロジェクトを削除する

⌚ ワークスペースからプロジェクトを削除するには

1. ワークスペースウィンドウの“Projects”タブを選び、右マウスボタンでポップアップメニューを表示してください。
2. [Remove Project]オプションを指定してください。
または
1. ワークスペースウィンドウの“Projects”タブからプロジェクトを選んでください。
2. DELキーを押してください。

注意 ワークスペースから現在使用中のプロジェクトを削除することはできません。

2.13 ワークスペースの相対プロジェクトパス

HEW では、プロジェクトをワークスペースに追加する場合、相対パスを使用してワークスペースに追加することができます。これにより、プロジェクトをワークスペースディレクトリの上に置くことができ、HEW ワークスペースの再配置も正確にできます。プロジェクトは常にワークスペースと相対的であるため、プロジェクトがワークスペースの上のディレクトリにある場合、再配置のあと、HEW は、同じ相対場所でプロジェクトを見つけようとします。このことは、複数のワークスペース間で共有したプロジェクトを使用するとき、特に便利です。

HEW の古いバージョンでは、このプロジェクトは再配置されておらず、オリジナルのプロジェクトパスをアクセスしようとした。また、ワークスペースディレクトリのサブディレクトリにあったプロジェクトを再配置することのみ可能でした。これは現在のバージョンでも、HEW の標準的な動作です。

❸ プロジェクト相対パスフラグを変更するには

1. ワークスペースウィンドウでプロジェクトを選択してください。
2. マウスの右ボタンをクリックしてプロパティを選択してください。
3. “Project relative file path”チェックボックスをクリックし、相対ファイルパスの特徴を切り替えてください（図2.25）。
4. “OK”ボタンをクリックしてください。

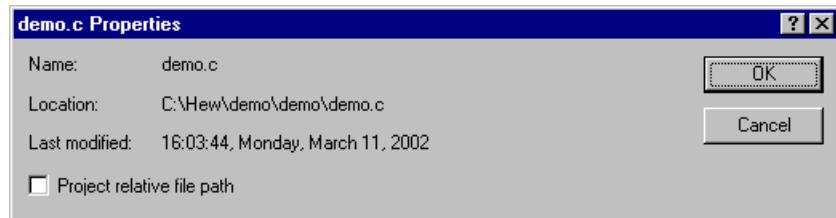


図 2.25: Properties ダイアログボックス

3. ビルドの応用

この章ではより高度なビルドの概念を説明します。

3.1 ビルド実行の復習

2章「ビルドの基本」では、ビルド実行をコンパイラ、アセンブラー、リンクエディタを用いて説明しました（図2.1）。HEWではこれが一般的な実行環境です。しかし、ビルド実行を変更（フェーズの追加や削除など）するためには、ビルドの機能についてさらに知ることが必要です。

3.1.1 ビルドとは？

プロジェクトのビルド実行とは、複数の特定の入力ファイルに複数のツールを適用して期待する出力を得ることです。つまり、オブジェクトファイルを得るために、Cソースファイルにコンパイラを適用したりアセンブリ言語のソースファイルにアセンブラーを適用したりします。ビルドの各ステップ、または、各「フェーズ」において、様々な入力ファイルの集まりに各種ツールを適用します。図3.1にビルド処理を示します。

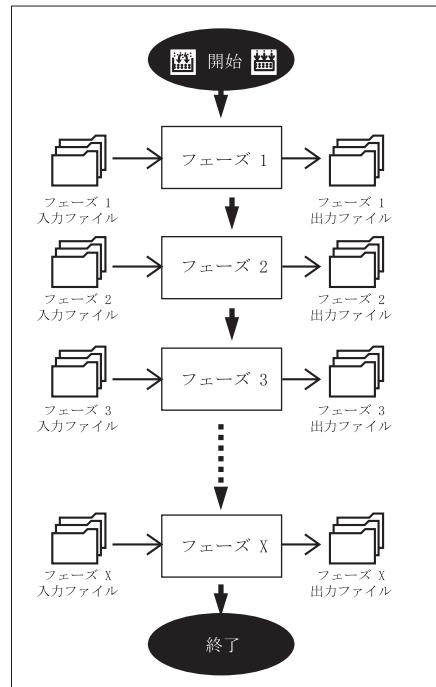


図 3.1: ビルド実行

HEWではビルド処理を変更できます。[Options->Build Phases...]を選んで“Build Phases”ダイアログボックス（図3.2）を表示します。左には現在のプロジェクトで定義されたフェーズを示します（図3.2）では標準のビルドフェーズを示します）。この章では“Build Phases”ダイアログボックスが提供する様々な機能について説明します。

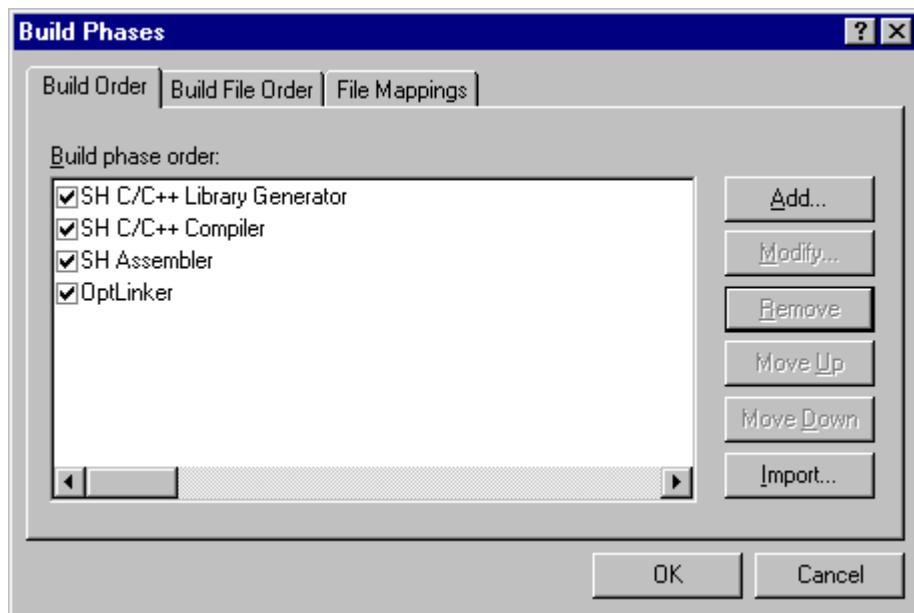


図 3.2: Build Phases ダイアログボックス

3.2 カスタムビルドフェーズを作成する

標準のビルド実行の前後または途中に他のツールを実行する場合、独自のビルドフェーズ（カスタムビルドフェーズ）を作成します。カスタムビルドフェーズを略してカスタムフェーズということがあります。

[Options->Build Phases...] を選ぶと “Build Phases” ダイアログボックス（図 3.2）が表示されます。“Add...” ボタンをクリックしてください。新しいビルドフェーズを作成するための“New Build Phase” ダイアログボックス（図 3.3a）が表示されます。

ステップ1（図 3.3a）では、カスタムビルドフェーズを新規に作成するか、システムビルドフェーズを追加するかを選択します。システムビルドフェーズは、使用しているツールチェイン（コンパイラ、アセンブラー、リンクエディタ、ライブラリアンなど）内で定義済みのすぐに使用できるフェーズ、または、ユーティリティフェーズ（例えば、ファイルコピー、ソースコード複雑度解析ツールなど）です。システムフェーズがこれ以上ない場合、“Add an existing system phase” ボタンが非アクティブになります。（システムビルドフェーズを略してシステムフェーズといいます。）

“Create a new custom phase”を選んでカスタムビルドフェーズを作成してください。

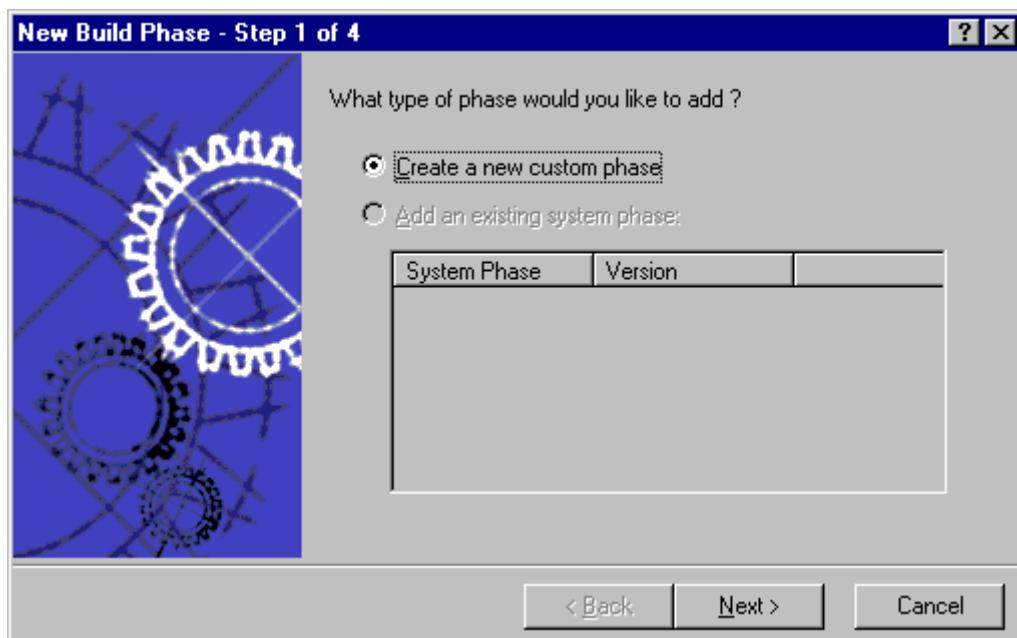


図 3.3a: New Build Phase ダイアログボックス (ステップ 1)

3 ビルドの応用

ステップ 2(図 3.3b)では作成するフェーズの種類を選びます。2つの選択肢(“Multiple phase”または“Single phase”)があります。マルチプルフェーズを実行すると特定のファイルグループに属するプロジェクト内の各ファイルにコマンドが適用されます。例えば、“Select input file group”フィールドに“C source file”を選ぶと、プロジェクト内の各ファイルに1回ずつコマンドが実行されます。シングルフェーズを選ぶとビルド実行中に一度だけ実行されます。

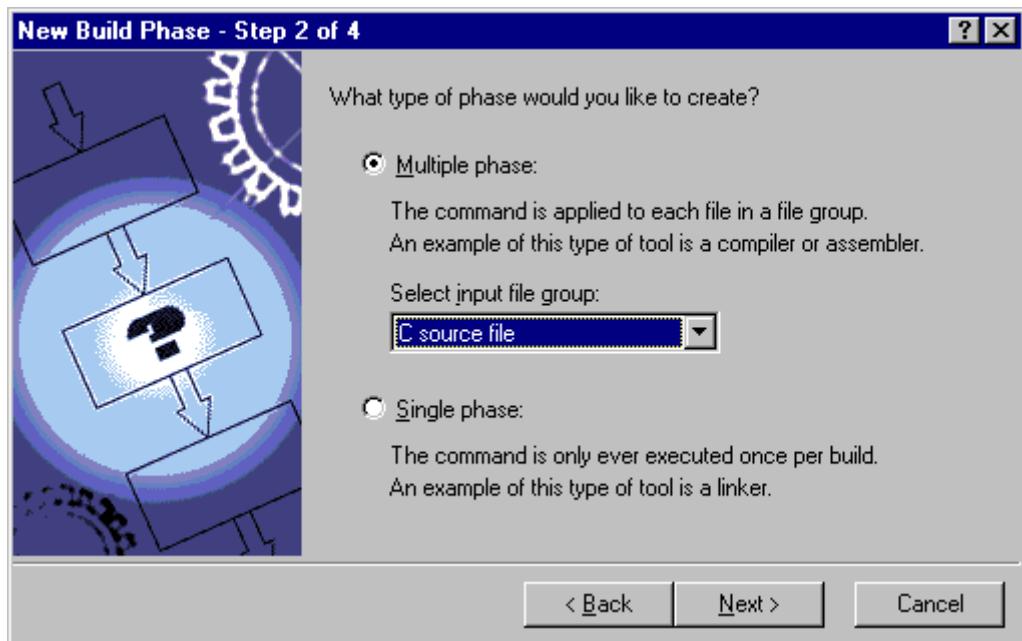


図 3.3b: New Build Phase ダイアログボックス (ステップ 2)

入力ファイルグループリストは、そのプロジェクト用に定義された現在のファイルグループを含みます。入力ファイルグループリストの中の“Multiple Groups...”エントリを選択すると、複数の入力ファイルグループを定義することができます。このリストのエントリは、図 3.3c に示すダイアログボックスを表示します。

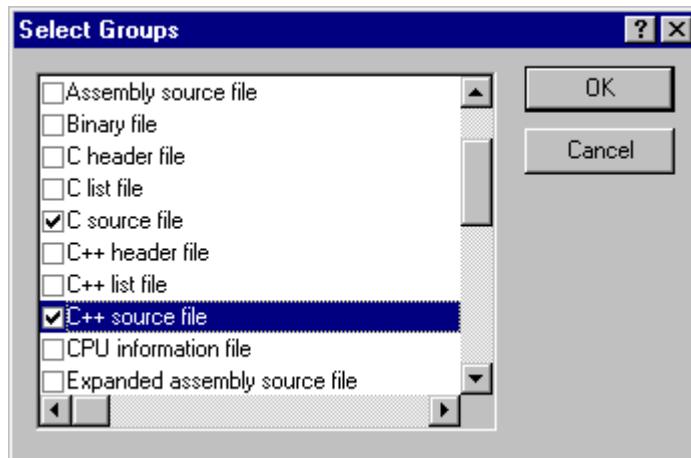


図 3.3c: 複数の入力ファイルグループの変更

選択すると、入力ファイルグループは“Multiple Groups...”と表示されます。プロジェクトに追加されているカスタムフェーズ用に、複数の入力ファイルグループをこのダイアログボックスで選ぶことができます。ファイルグループを選択するには、ファイルグループ名の隣にあるボックスをチェックしてください。このダイアログボックスでは、1つ以上のファイルグループが選択できます。

ステップ 3(図 3.3d)では、新しいビルドフェーズについての基礎的な情報を入力します。“Phase name” フィールドにフェーズ名を入力します。“Command” フィールドにプログラムファイルのパスを入力します（コマンドラインオプションは含めません。オプションは HEW のメニューバーの“Options”メニューで指定します）。“Default options” フィールドにフェーズのデフォルトのオプションを指定します。デフォルトオプションはプロジェクトに新しいファイルを追加するときに付加されるオプションです。“Initial directory” フィールドにはそのプログラムをどのディレクトリから実行するか（つまり、ツール実行前にどこにカレントディレクトリを設定するか）を入力します。

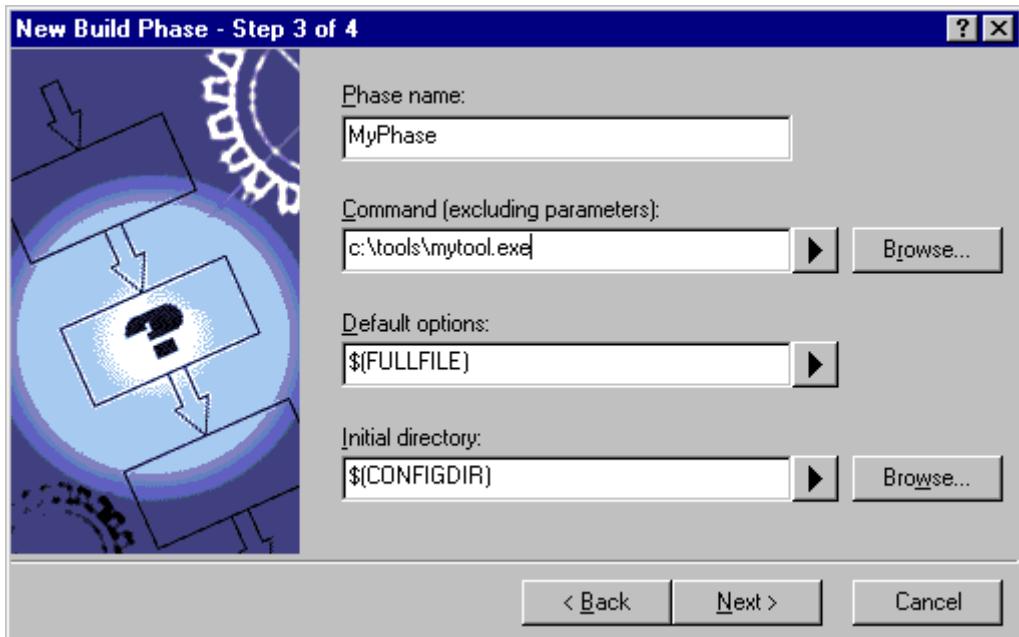


図 3.3d: New Build Phase ダイアログボックス (ステップ 3)

最後のステップ 4 (図 3.3e) では、そのフェーズに必要な環境変数を指定します。

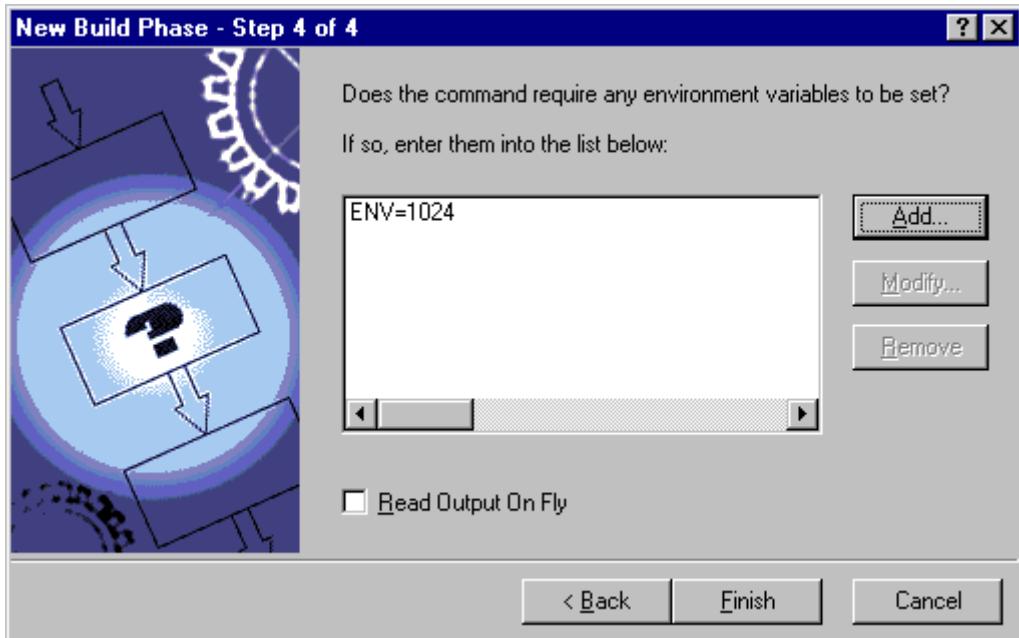


図 3.3e: New Build Phase ダイアログボックス (ステップ 4)

3 ビルドの応用

新しい環境変数を追加するには、“Add...”ボタンをクリックしてください。図 3.4 のダイアログボックスが表示されます。新しい環境変数を追加するには“Variable”フィールドに環境変数名を入力して “Value” フィールドに環境変数の値を入力して “OK”ボタンをクリックします。

環境変数を変更するには、ステップ 4 のダイアログボックスのリストから環境変数を選んで、“Modify...” ボタンをクリックします。“Variable” フィールドと “Value” フィールドを変更して “OK”ボタンをクリックすると、リストに変更した変数が追加されます。

環境変数を削除するには、ステップ 4 のダイアログボックスのリストから削除する環境変数を選んで、“Remove” ボタンをクリックします。

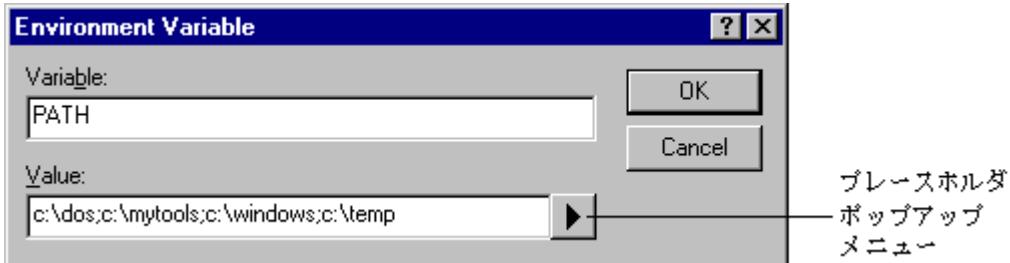


図 3.4: Environment Variable ダイアログボックス

追加するツールが動作中に出力を表示できる場合、ステップ 4 のダイアログボックスの”Read Output On Fly”オプションを使用してください。出力が発生することにツールの出力を表示します。このオプションが off に設定されると、HEW はツールに表示されている全出力を保存し、ツールが動作を終了したとき出力ウィンドウに表示します。ただ、ツールが長時間かかる作業を実行中である場合、実行の進行状況を見るのが難しいため、このオプションは問題となることがあります。

注意 “Read Output On Fly”を用いると、特定のオペレーティングシステムで特定のツールを使用するとき問題を引き起こすことがあります。もしツールが HEW の中でロックアップ、またはフリーズするといった問題がありましたら、”Read Output On Fly”オプションのチェックを外して下さい。

指定した内容で新しいフェーズを作るには、“Finish” ボタンをクリックしてください。デフォルトでは“Build Phases” ダイアログボックス（図 3.2）の“Build Order”タブの“Build Phase Order”リストの最後に新しいフェーズが追加されます。

3.3 ビルドのフェーズ順序

図 3.5 の標準的ビルドでは、コンパイラの前、アセンブラーの前、リンクエディタの前、リンクエディタの後、の四ヶ所にフェーズを追加できます。ビルドの順序の中で好きな場所にカスタムフェーズを追加したりシステムフェーズを移動したりできます。ビルド処理を正しく実行させるためには、カスタムフェーズの出力が他のフェーズに入力される場合、フェーズの順序を正しく設定することが必要です。

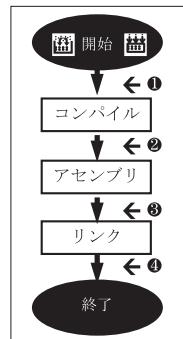


図 3.5: 標準的ビルド処理

“Build Phases” ダイアログボックスでビルドフェーズの順序を変更できます。このダイアログボックスにはフェーズの順序に関するタブが二つあります。“Build Order” タブと“Build File Order” タブです。

3.3.1 ビルドのフェーズ順序

“Build Order” タブ(図 3.6)は、“Build”(■)または“Build All”(■)操作で実行されるフェーズの現在の順序を示します。各フェーズの左にあるチェックボックスのチェックの有無はそのフェーズの有効/無効を示します。このチェックボックスをチェックするとそのフェーズが実行されます。

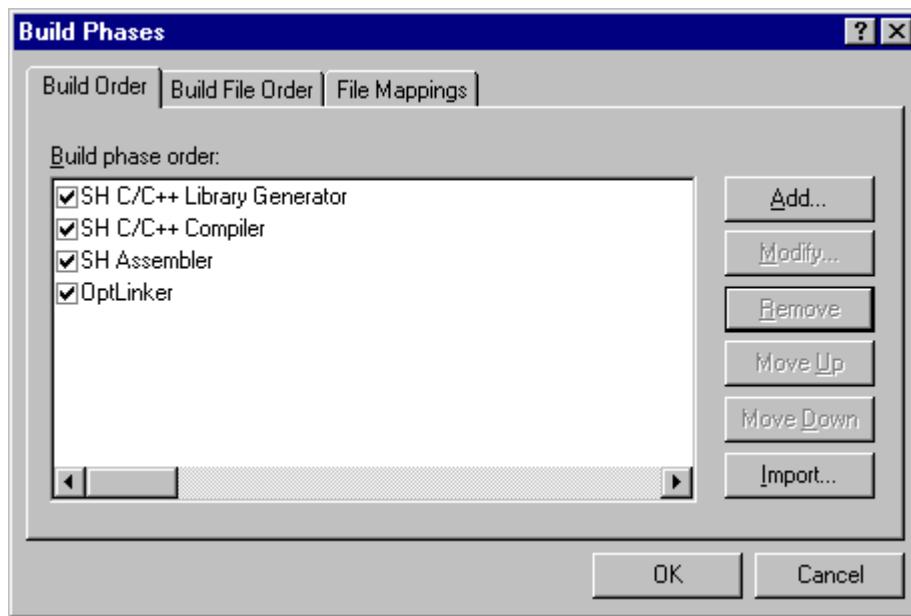


図 3.6: Build Phases ダイアログボックス Build Order タブ

また、以下の操作ができます。

⌚ フェーズを削除するには

1. 削除するフェーズを選んでください。
2. “Remove” ボタンをクリックしてください。

⌚ システムフェーズのプロパティを表示するには

1. プロパティを表示するシステムフェーズを選んでください。
2. “Modify...”ボタンをクリックしてください。

⌚ フェーズを移動するには

1. 移動するフェーズを選んでください。
2. “Move Up” または “Move Down” ボタンをクリックすると上下に移動します。

⌚ フェーズを取り込むには

1. “Import” ボタンをクリックしてください。ダイアログボックスが表示され、カスタムフェーズを取り込むための既存のプロジェクトを見ることができます。
2. カスタムフェーズを取り込みたいプロジェクトの位置を選んでください。選択すると、取り込み可能なプロジェクトのカスタムフェーズを並べたダイアログボックスが表示されます。
3. フェーズ名を選択し、プロパティをクリックすると、カスタムフェーズの詳細が表示されます。そのフェーズが必要な機能を果たすかどうか、これを見て決定することができます。
4. どのフェーズを取り込むか決定したら、そのフェーズをリスト上でハイライト表示し、“OK”ボタンをクリックしてください。ビルド順序で一番下にあるビルドフェーズダイアログボックスに、そのフェーズが追加されます。

⌚ カスタムフェーズを変更するには

1. 変更するカスタムフェーズを選んでください。
2. “Modify...”ボタンをクリックしてください。“Modify MyPhase”ダイアログボックスの“Command”タブが表示されます（図3.7）。
3. 必要に応じてフィールドの内容を変更してください。
4. 入力ファイルがなくてもフェーズの実行を中断したくない場合、“Don't check for input file(s) existence before executing” チェックボックスをチェックしてください。

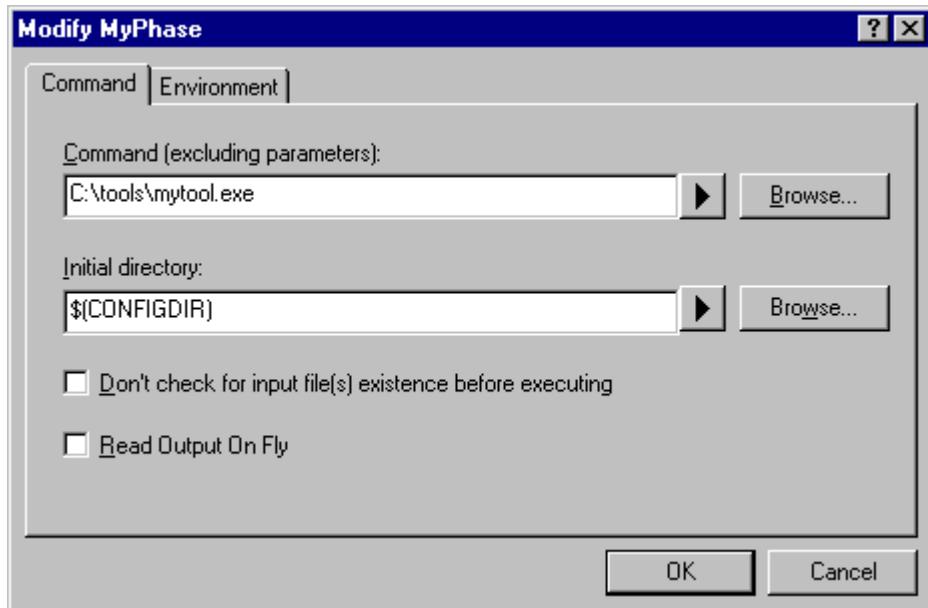


図 3.7: Modify MyPhase ダイアログボックス Command タブ

5. “Environment”タブ（図 3.8）を選んでフェーズの環境設定を行ってください。
6. 環境変数の追加は“Add...”ボタン、変更は“Modify...”削除は“Remove”ボタンを使用してください。操作は前節と同じです。
7. 変更後“OK”ボタンをクリックしてください。

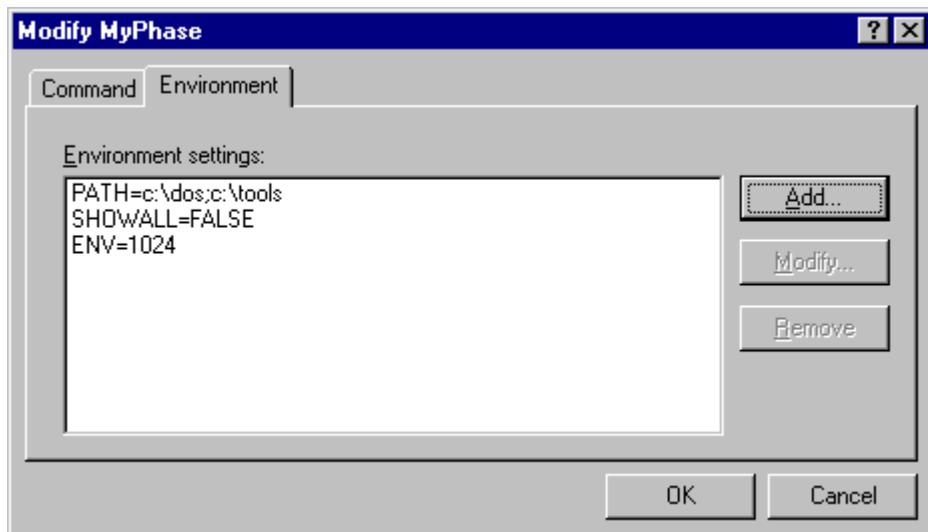


図 3.8: Modify MyPhase ダイアログボックス Environment タブ

3.3.2 ビルドファイルのフェーズ順序

ワークスペースウィンドウから C ソースファイルを選んで [Build->Build File] を選ぶか、 を押すと、ファイルがコンパイルされます。同じように、ワークスペースウィンドウからアセンブリ言語ソースファイルを選んで [Build->Build File] を実行すると、ファイルがアセンブルされます。ファイルグループと実行するフェーズの関係は“Build Phases”ダイアログボックスの“Build File Order”タブ（図 3.9）で管理されています。

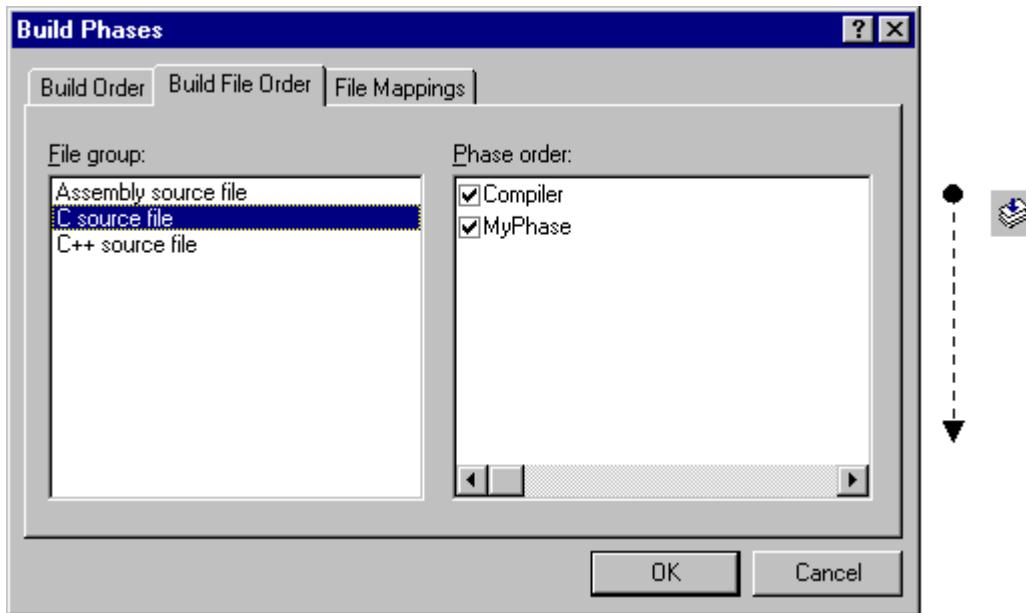


図 3.9: Build Phases ダイアログボックス Build File Order タブ

リストには、“File group” リストボックスのファイルグループに対してビルドファイル操作を選択したとき実行されるすべての現在のフェーズを表示します。図 3.9 では“C source file” ファイルグループが選ばれており、“Compiler” フェーズと “MyPhase” フェーズが関連付けられています。

“Build Order”タブのリストに新しいエントリを追加すると、自動的に“Build File Order”タブの“Phase order”リストに新しいエントリが追加されます。例えば、“C source file”を入力とするフェーズを追加します。このフェーズは“Build File”操作を“C source file”に適用する時に実行されるフェーズのリストに自動的に追加されます。
[Build->Build File...]を選んだときに実行したくないフェーズがある場合、“Phase order”リストのフェーズ名の左にあるチェックボックスのチェックを外してください。

3.4 カスタムビルドフェーズのオプション設定

カスタムフェーズを定義後、フェーズ実行時に使用するコマンドラインオプションを指定します。定義されたフェーズにはそれぞれ [Options]メニューにオプション設定用のメニューがあります。指定するフェーズのオプションを選んでください。起動するダイアログボックスは、選んだカスタムフェーズがシングルフェーズかマルチブルフェーズかによって異なります（図 3.3b “Single phase”/“Multiple phase”ラジオボタン指定）。

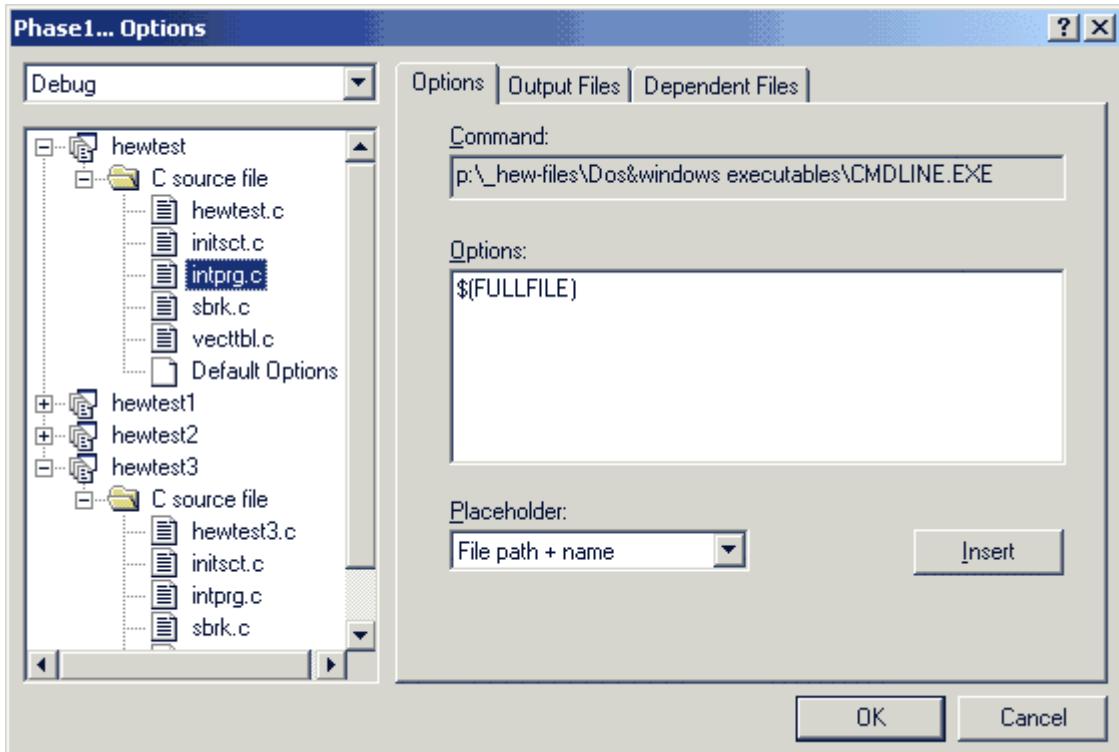


図 3.10: カスタムフェーズのオプションダイアログボックス (Multiple Phase)

図 3.10 にカスタムフェーズオプションダイアログボックスを示します。左側にはプロジェクトとファイルのリストがあります。Windows® Explorer と同様の方法で、オプションを変更するプロジェクトとファイルを複数、一回以上選ぶことができます。右側には 3 つのオプションタブがあります。選んだファイルに適用するオプションをここで設定してください。

また、ダイアログボックス左上のコンフィグレーションリストで、どのコンフィグレーションの情報を表示するか選択できます。各コンフィグレーションは“Multiple configurations...”という名のエントリと一緒に並べられています。“Multiple configurations...”を選択すると、ダイアログボックスが表示され、複数のコンフィグレーションを選択できます。

3.4.1 Options タブ

“Options” タブ（図 3.11）ではフェーズに渡されるコマンドラインオプションを定義できます。“Command” フィールドではフェーズを定義したときに入力したコマンドを表示します（図 3.3d）。 “Options” フィールドにはコマンドに渡すコマンドライン引数を入力してください。プレースホルダを挿入する場合は、対応するプレースホルダを “Placeholder” ドロップダウンリストボックスから選び、“Insert” ボタンをクリックしてください。プレースホルダの詳細については、Hitachi Embedded Workshop 2.0 ユーザーズマニュアル 付録 C、「プレースホルダ」を参照してください。

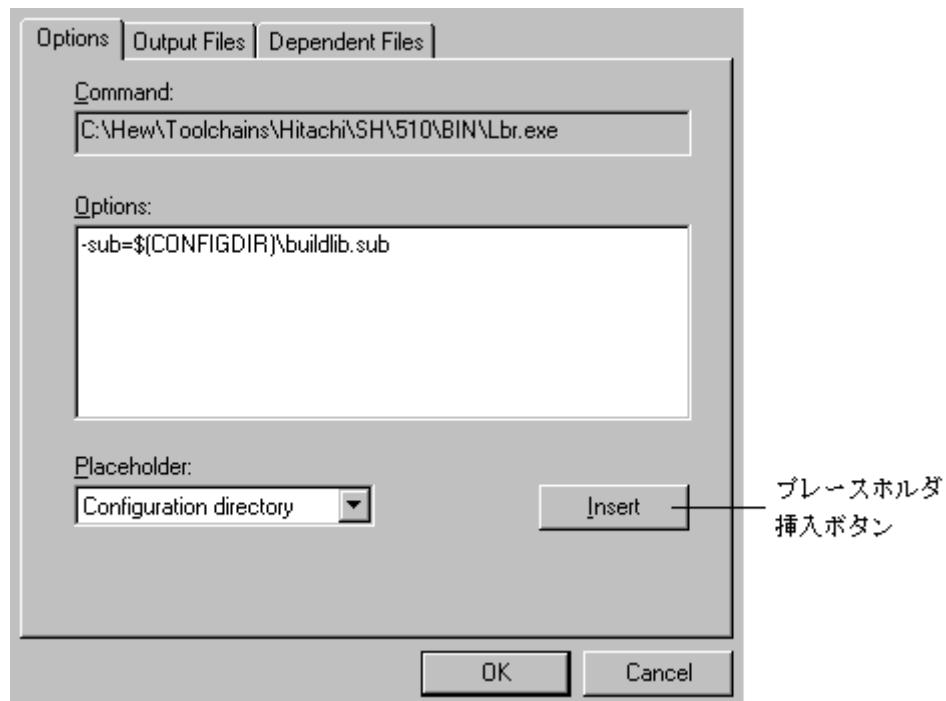


図 3.11: カスタムフェーズのオプション Options タブ

3.4.2 Output Files タブ

“Output Files”タブ(図 3.12)ではフェーズで作成される出力ファイルを指定します。HEW では、ファイルがこのフェーズを通過する前に、出力ファイルの日付が入力ファイルの日付より古いことをチェックしています。出力ファイル作成後入力ファイルが変更された場合、入力ファイルに対してこのフェーズが実行されます。出力ファイルが最新の場合、入力ファイルに対してこのフェーズは実行されません。

注意 出力ファイルを指定しない場合フェーズは常に実行されます。

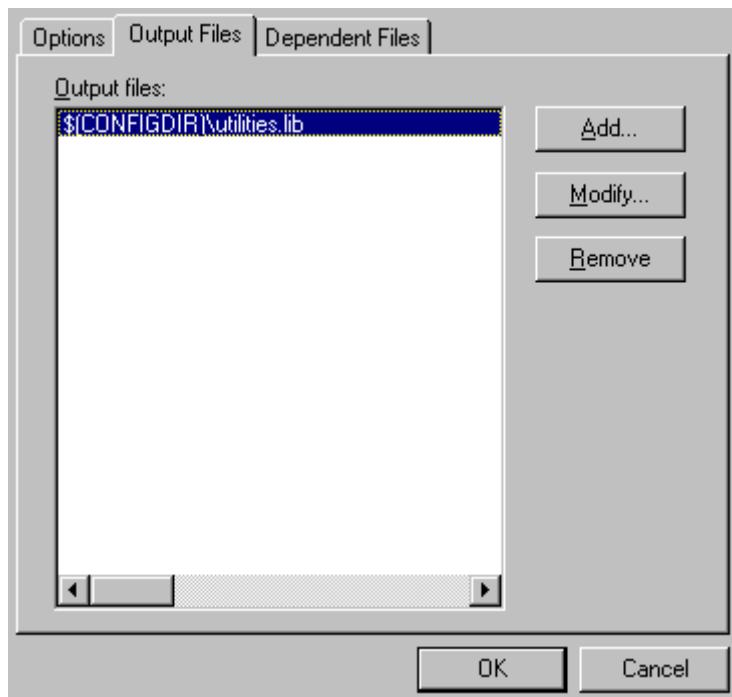


図 3.12: カスタムフェーズのオプション Output Files タブ

⌚ 出力ファイルを追加するには

1. “Add...”をクリックしてください。“Add Output File”ダイアログボックスが表示されます(図 3.13)。
2. ファイル名を入力するか、“Browse...”ボタンで選んでください。
3. “OK”ボタンをクリックすると、リストに出力ファイルを追加します。



図 3.13: Add Output File ダイアログボックス

⌚ 出力ファイルを変更するには

1. 変更する出力ファイルを選んでください。
2. “Modify...”をクリックすると“Modify Output File”ダイアログボックスが表示されます(タイトル以外は図 3.13と同様)。
3. フィールドを変更して“OK”ボタンをクリックしてください。変更した項目がリストに追加されます。

⌚ 出力ファイルを削除するには

1. 削除する出力ファイルを選んでください。
2. “Remove”ボタンをクリックしてください。

3.4.3 Dependent Files タブ

“Dependent Files”タブ（図 3.14）ではフェーズに必要な依存ファイルを指定します。HEW では、各ファイルがこのフェーズを通過する前に、依存ファイルの日付が入力ファイルの日付より新しいか否かをチェックしています。チェック後、依存ファイルの日付が新しい場合（つまり、入力ファイル作成後に依存ファイルが変更された場合）このフェーズでファイルが実行されます。依存ファイルの日付が入力ファイルの日付より古い場合、このフェーズは実行されません。

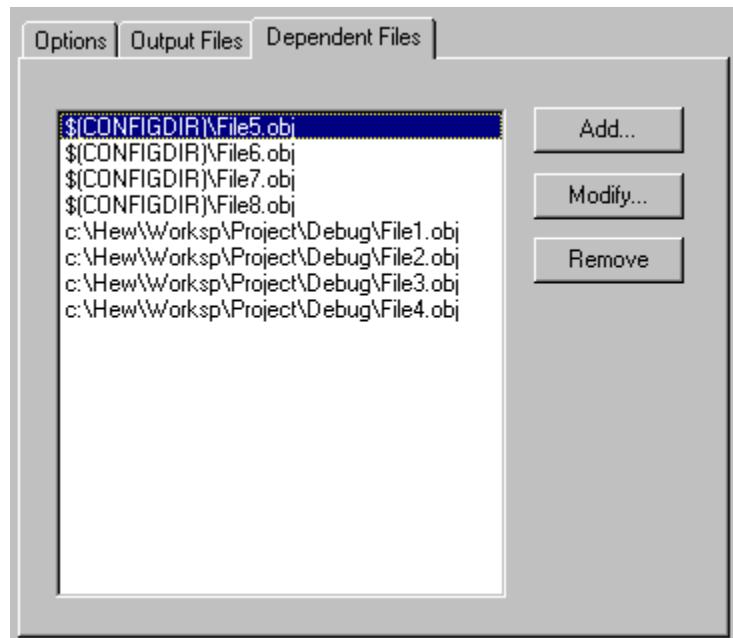


図 3.14: カスタムフェーズのオプション Dependent Files タブ

⌚ 依存ファイルを追加するには

1. “Add...”ボタンをクリックしてください。“Add Dependent File”ダイアログボックスが表示されます（図 3.15）。
2. ファイル名を入力するか、“Browse...”ボタンで選んでください。
3. “OK”ボタンをクリックすると、リストに出力ファイルを追加します。



図 3.15: Add Dependent File ダイアログボックス

⌚ 依存ファイルを変更するには

1. 変更する依存ファイルを選んでください。
2. “Modify...”をクリックすると“Modify Dependent File”ダイアログボックスが表示されます（タイトル以外は図 3.15と同じ）。
3. フィールドを変更して“OK”ボタンをクリックすると変更した項目がリストに追加されます。

⌚ 依存ファイルを削除するには

1. 削除するファイルを選んでください。
2. “Remove”ボタンをクリックしてください。

3.5 ファイルのマッピング

デフォルトでは、フェーズに入力されるファイルはプロジェクトから取得したのものだけです。つまり、“New Build Phase”ダイアログボックス(図 3.3b)の“Select input file group” ドロップダウンリストに指定した種類のプロジェクトファイルだけです。もし前のフェーズから出力されたファイル（中間ファイル）をフェーズで使いたい場合は、“Build Phases” ダイアログボックス（図 3.16）の“File Mappings” タブで定義してください。

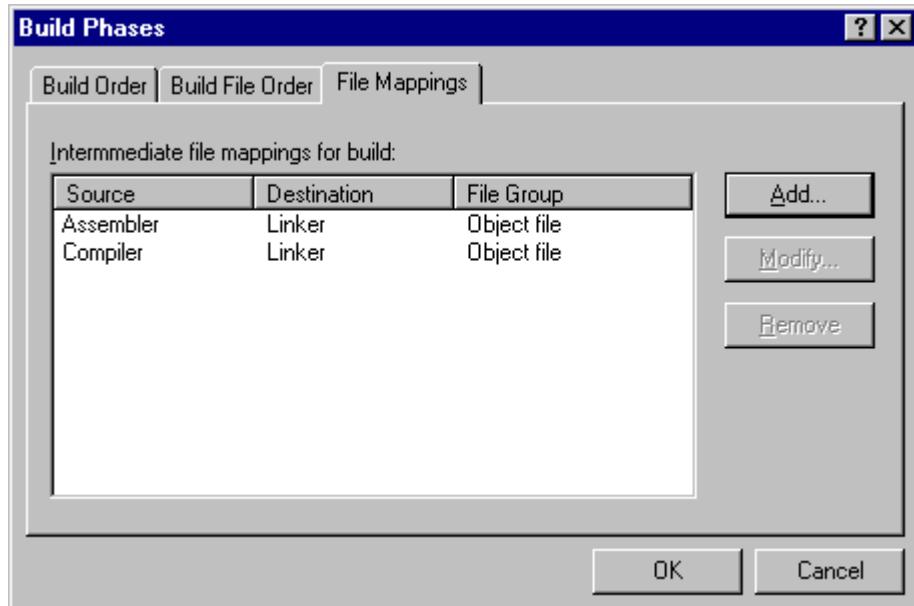


図 3.16: Build Phases ダイアログボックス File Mappings タブ

ファイルマッピングでは、あるフェーズ（出力元フェーズという）で作成したある種類の出力ファイル（中間ファイル）を、他のフェーズ（出力先フェーズという）に渡すように指定します。プロジェクトファイルに加えて中間ファイルも渡されます。

3 ビルドの応用

⌚ ファイルのマッピングを追加するには

1. “Add...”ボタンをクリックしてください。“Define File Mapping” ダイアログボックスが表示されます(図 3.17)。
2. “File group” ドロップダウンリストボックスから、フェーズ間で渡したいファイルグループを選んでください。
3. “Source phase” ドロップダウンリストボックスから出力元フェーズ(ファイルを作成するフェーズ)を選んでください。
4. “Destination phase” ドロップダウンリストボックスから出力先フェーズ(ファイルを渡す先のフェーズ)を選んでください。
5. “OK”ボタンをクリックすると新しいマッピングが追加されます。

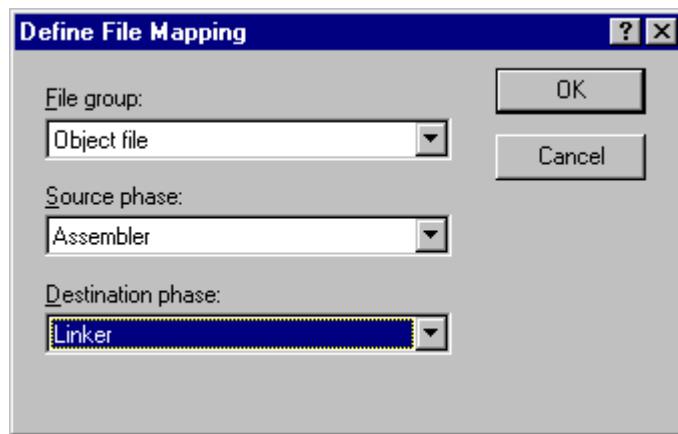


図 3.17: Define File Mapping ダイアログボックス

⌚ ファイルマッピングを変更するには

1. 変更するマッピングを選んでください。
2. “Modify...” ボタンをクリックしてください。“Define File Mapping” ダイアログボックスが表示されます(図 3.17)。
3. 必要に応じてオプションを変更してください。
4. “OK”ボタンをクリックすると変更が有効になります。

3.6 ビルドを管理する

デフォルトでは、HEW はビルドのすべてのフェーズを実行し、途中で致命的なエラーが起きたときだけ中止します。これは “Options” ダイアログボックスの “Build” タブ（図 3.18）で変更することができます。



図 3.18: Options ダイアログボックス Build タブ

[Tools->Options...] を選んでダイアログボックスを表示してください。ある一定の回数を超えるエラーが起きた後にビルドを中止したいとき、“Stop build if the no. of errors exceed” チェックボックスをチェックして右のフィールドにエラー数を入力してください。ある一定の数を超えるウォーニングが表示されたときにビルドを中止したいとき、“Stop build if the no. of warnings exceed” チェックボックスをチェックして右のフィールドにウォーニング数を入力してください。

注意 上記設定にかかわらず、致命的エラーが発生した場合、ビルドは停止します。

“Build” タブでは、エラー数やウォーニング数の制限のほかに、コマンドライン、環境、初期ディレクトリの表示の有無を設定することができます。表示するには、それぞれのチェックボックスをチェックしてください。

3.7 ビルドの出力のログを取る

ファイルに各ビルドの結果を保存したいときには、[Tools->Customize...] を選んで“Customize”ダイアログボックスを表示して“Log”タブを選んでください（図 3.19）。 “Generate log file” チェックボックスをチェックして、“Path” フィールドにログファイルのフルパスを入力するか、“Browse...”ボタンをクリックしてパスを選択してください。

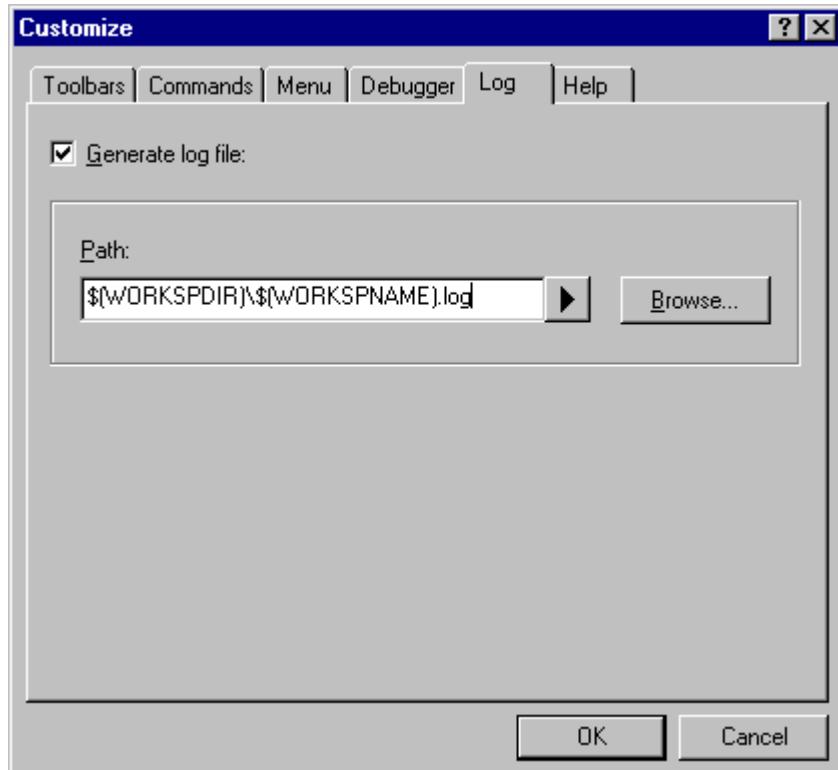


図 3.19: Customize ダイアログボックス Log タブ

3.8 ツールチェインのバージョンを変更する

同じツールチェインの2つ以上のバージョンがHEWに登録されているとき図3.20に示す“Change Toolchain Version”ダイアログボックスでバージョンを選択できます。このダイアログボックスを表示されるには[Tools->Change Toolchain Version...]を選択してください。このダイアログボックスの“Available versions”ドロップダウンリストからバージョンを選択し、“OK”ボタンをクリックしてください。

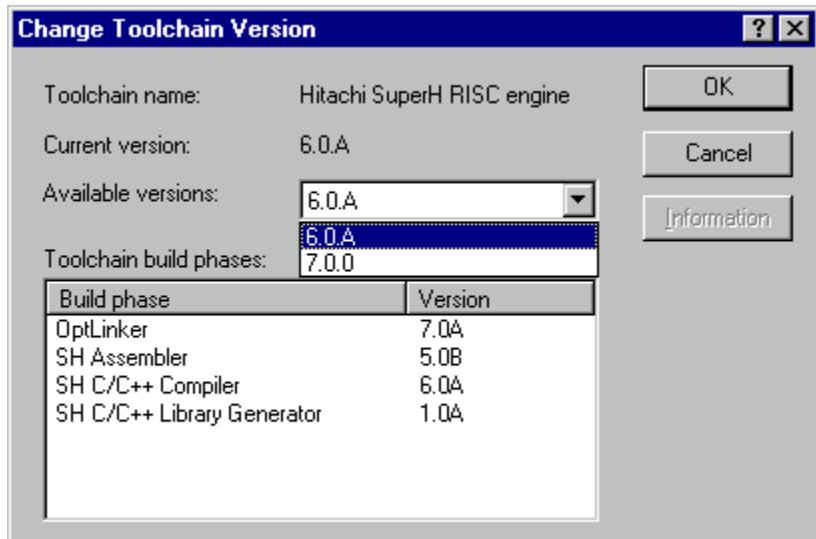


図 3.20: Change Toolchain Version ダイアログボックス

ツールチェインを構成するツールの情報を表示するには“Change Toolchain Version”ダイアログボックスの“Toolchain build phases”リストからツールを一つ選択し、“Information”ボタンをクリックしてください。ツール情報ダイアログボックス(図3.21)にそのツールの情報が表示されます。このダイアログボックスを閉じるには“Close”ボタンをクリックしてください。

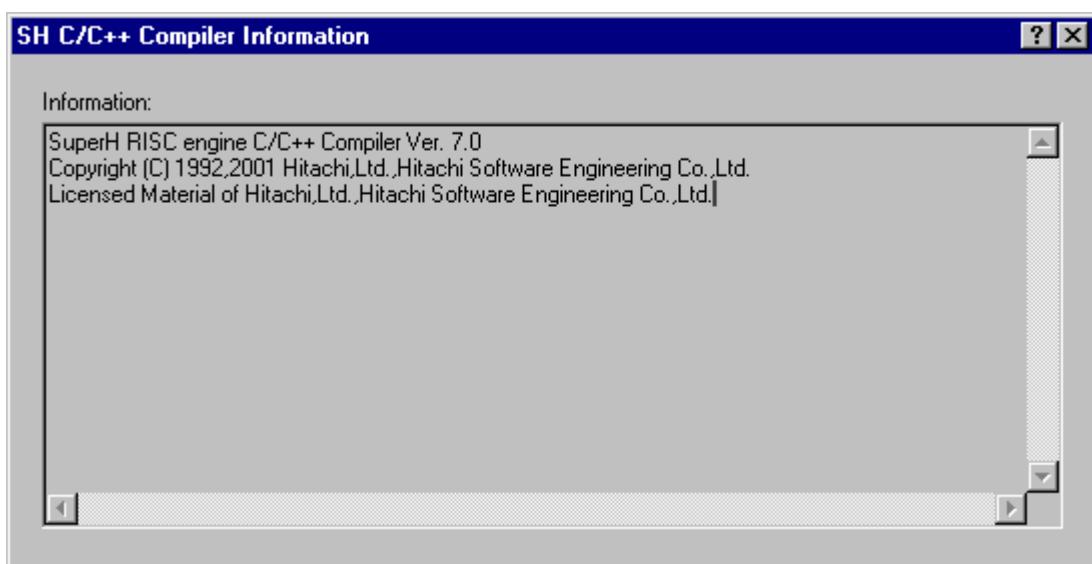


図 3.21: ツール情報 ダイアログボックス

3.9 外部デバッガを使う

HEW は外部デバッガツールを起動することができます。もし他のデバッガを使いたい場合は、“Tools”メニューに加えなければいけません。詳細は、Hitachi Embedded Workshop 2.0 ユーザーズマニュアル 5 章「環境のカスタマイズ」を参照してください。

“Customize” ダイアログボックス（図 3.22）の “Debugger” タブに日立デバッギングインターフェース(HDI)に関する情報を設定します。一部のターゲットが現在新しい環境でサポートされていない場合は、古いバージョンのデバッガを使用することができます。[Tools-> Customize...]を選んでダイアログボックスを表示させ、“Debugger”タブを選んでください。

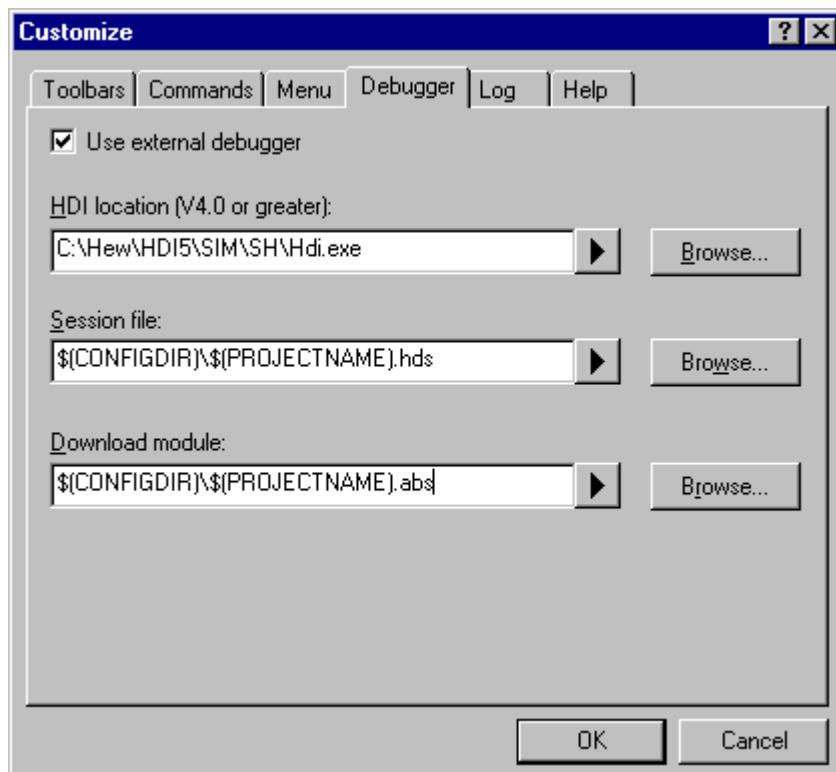


図 3.22: Customize ダイアログボックス Debugger タブ

外部デバッガを使用する場合は、“Use external debugger”をチェックしてください。そして以下の項目を設定してください。1つは HDI の実行ファイルの場所です。HDI はバージョン 4.0 以降のものを指定してください。そうでない場合、動作を保証できません。2つめはセッションファイルの情報です。HDI に切り替わったとき、どのセッションをロードするかを HDI に伝えます。もう1つはダウンロードモジュールの場所です。これはビルド後にダウンロードモジュールが変更された場合 HEW が自動的に HDI に切り替わるために必要です。

指定したセッションファイルで HDI を起動するには、次に示す“外部デバッガの起動”ツールバー ボタンをクリックしてください。



ビルド後、ダウンロードモジュールが更新された場合、HEW から HDI に戻り、すぐデバッグできるようになります。HDI を使用しているときにソースウィンドウのどれかをダブルクリックすると、HEW に戻りダブルクリックした行のソースファイルを開いた状態になります。

3.10 メイクファイルの生成

HEW では、メイクファイルを生成することができます。メイクファイルを使用すると、完全に HEW をインストールしていくなくても、現在のプロジェクトをビルドすることができます。HEW をインストールしていない相手にプロジェクトを送ったり、メイクファイルを含むビルド全体をバージョン管理したりする場合に便利です。

⌚ メイクファイルを生成するには

1. メイクファイルを生成するプロジェクトが現在のプロジェクトであることを確認してください。
2. プロジェクトをビルドするビルドコンフィグレーションが現在のコンフィグレーションであることを確認してください。
3. [Build>Generate Makefile...] を選んでください。
4. このメニューを選ぶと、ダイアログボックスが表示され、ワークスペースのどの部分をメイクファイルに追加するか尋ねます（図3.23参照）。
5. 手持ちのメイクファイルに対応するラジオボタンを選択し、“OK”ボタンをクリックしてください。

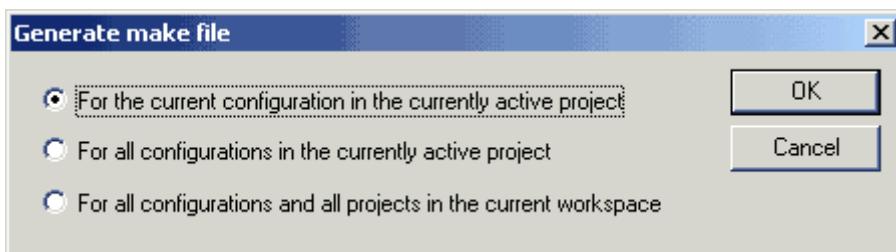


図 3.23: Generate makefile ダイアログボックス

HEW は現在のワークスペースディレクトリ内に“make”サブディレクトリを作り、その中にメイクファイルを生成します。メイクファイルの名前は、現在のプロジェクトやコンフィグレーションに拡張子.mak を付けたものです(例：project_debug.mak)。HEW により生成されたメイクファイルは、HEW をインストールしたディレクトリにある実行ファイル HMAKE.EXE で実行できます。ただし、ユーザが変更したメイクファイルは実行できません。

⌚ メイクファイルを実行するには

1. コマンドウィンドウを開き、メイクファイルが生成された“make”ディレクトリに移行してください。
2. HMAKEを実行してください。コマンドラインはHMAKE.EXE <メイクファイル名>です。

注意 生成したメイクファイルが移動可能か否かは、プロジェクト自体が移動可能か否かに依存します。たとえば、出力ディレクトリやインクルードファイルディレクトリへのフルパスを含むコンパイラオプションがあると、異なるインストール環境下の別のユーザがビルドした場合、失敗する可能性が高いでしょう。一般的に、できるだけプレースホルダを使用して、フルパス、または特定のパスの使用はなるべく避けてください。

HEW Builder
High-performance Embedded Workshop 2
ユーザーズマニュアル



ルネサス エレクトロニクス株式会社
神奈川県川崎市中原区下沼部1753 ☎211-8668

ADJ-702-351A