

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連する情報は、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連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のですが、誤りが無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、パーソナル機器、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ジング処理等、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、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と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。

# High-performance Embedded Workshop V.4.07

ユーザーズマニュアル

ルネサスマイクロコンピュータ開発環境システム

## 本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社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弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 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など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 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などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊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、弊社ホームページ (<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。
5. 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のですが、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独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。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
  - 1) 生命維持装置。
  - 2) 人体に埋め込み使用するもの。
  - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
  - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およ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ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）およびエージング処理等、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。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 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弊社営業窓口までご照会ください。

## はじめに

High-performance Embedded Workshop は、ルネサステクノロジマイクロコントローラの組み込み用アプリケーションの開発をサポートするツールです。おもな特徴をまとめると次のようになります。

- GUIを用いてコンパイラ、アセンブラ、リンカージェネレータなどのオプション設定のカスタマイズができるプロジェクトビルドシステム
- プログラムを読みやすくするシンタックス色付け機能を持つ統合テキストエディタ
- ユーザ独自のツールを実行するための環境設定
- 同一アプリケーション内でのビルドおよびデバッグを可能にする統合化デバッガ
- バージョン管理サポート

### High-performance Embedded Workshop の動作環境に関する注意事項

このユーザズマニュアル、High-performance Embedded Workshop オンラインヘルプ、および High-performance Embedded Workshop リリースノートに記載している内容は、PC および周辺機器などのすべての機器において High-performance Embedded Workshop が正常に動作することを保証するものではありません。

### High-performance Embedded Workshop が生成するサンプルのソースファイルに関する注意事項

ワークスペースを作成するとマイコンに対応したサンプルのソースファイルが自動生成されます。生成されたソースファイルのコードは一例であり、これを使用して作成したすべてのプログラムの動作を保証するものではありません。該当するマイコンのハードウェアマニュアルをご参照の上、必要に応じてコードを変更しコードの適用可否をご判断ください。

## このユーザーズマニュアルについて

High-performance Embedded Workshop システムについて述べています。High-performance Embedded Workshop の基本的な使い方に関する情報、High-performance Embedded Workshop 環境のカスタマイズ、High-performance Embedded Workshop のビルド機能、および各 High-performance Embedded Workshop 製品で共通なデバッグ機能について説明します。なお、SuperH ファミリ用のデバッガと連携して動作した際の図を使用しています。デバッガの詳細については、各製品に同梱されるエミュレータまたはシミュレータのユーザーズマニュアルおよびヘルプを参照してください。

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、個々のマイコン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、各々のユーザーズマニュアルを参照してください。

High-performance Embedded Workshop は、インストール上、各種言語にカスタマイズされています。このユーザーズマニュアルでは、High-performance Embedded Workshop アプリケーションの日本語版について説明します。

## このユーザーズマニュアルの記号

このユーザーズマニュアルで使われている記号の意味を説明します。

記号	意味
[メニュー -> メニューオプション]	'->' はメニューオプションを示します (例 [ファイル-> 名前を付けて保存])。)
FILENAME.C	大文字の名前はファイル名を示します。
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。

## このユーザーズマニュアルの図について

このユーザーズマニュアルの一部の図は、実物と違っていることがあります。

## 商標

Microsoft, MS-DOS, Visual SourceSafe, Windows, および Windows Vista は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

その他すべての会社名および製品名は、各社の登録商標または商標です。

## 製品の内容および本書についてのお問い合わせ先

株式会社 ルネサス テクノロジ

コンタクトセンタ csc@renesas.com

ホームページ <http://japan.renesas.com/tools>

# 目次

1. 概要.....	1
1.1 ワークスペースとプロジェクト.....	1
1.2 メインウィンドウ .....	1
1.2.1 タイトルバー .....	2
1.2.2 メニューバー .....	2
1.2.3 ツールバー .....	3
1.2.4 ワークスペースウィンドウ.....	6
1.2.5 エディタウィンドウ .....	8
1.2.6 アウトプットウィンドウ .....	9
1.2.7 ステータスバー .....	13
1.3 ヘルプ機能 .....	13
1.4 High-performance Embedded Workshop を起動する.....	14
1.5 新規ワークスペースを作成する.....	15
1.6 ワークスペースを開く .....	15
1.7 古いワークスペースを使用する.....	16
1.8 ワークスペースを保存する .....	16
1.9 ワークスペースを閉じる .....	18
1.10 High-performance Embedded Workshop を終了する.....	18
1.11 ツールシステム概要 .....	18
1.12 High-performance Embedded Workshop 管理情報ファイル.....	18
1.13 コンフィグレーションおよびセッションの概要.....	19
1.14 マクロ生成支援機能とテスト支援機能の概要.....	22
1.14.1 テストシナリオの例 .....	25
1.14.2 ステップ1：マクロの記録.....	27
1.14.3 ステップ2：マクロを編集する（記録内容を見る）.....	28
1.14.4 ステップ3：マクロを実行する.....	29
1.14.5 ステップ4：テストスイートを作成する.....	30
1.14.6 ステップ5：テストスイートを編集する.....	30
1.14.7 ステップ6：テストイメージファイルを作成する.....	32
1.14.8 ステップ7：プログラムを変更しテストを実行する.....	34
1.14.9 ステップ8：テスト結果を確認する（不一致の場合）.....	35
1.14.10 ステップ9：プログラムを元に戻しテストを実行する.....	36
1.14.11 ステップ10：テスト結果を確認する（一致した場合）.....	36
2. ビルドの基本.....	37
2.1 ビルド処理 .....	37
2.2 ワークスペースウィンドウの Projects タブの構成.....	37

2.3	プロジェクトファイル	40
2.3.1	プロジェクトにファイルを追加する	41
2.3.2	ファイルおよびフォルダをドラッグアンドドロップする	42
2.3.3	プロジェクトからファイルを削除する	44
2.3.4	ビルドからプロジェクトファイルを除外する	45
2.3.5	除外したプロジェクトファイルをビルドに再び入れる	45
2.4	ワークスペースウィンドウでユーザフォルダを使う	46
2.5	ファイル拡張子とファイルグループ	47
2.5.1	アプリケーションとファイルグループを関連付ける	47
2.5.2	新規ファイルグループに新規ファイル拡張子を登録する	49
2.5.3	新規ファイル拡張子を作成する	50
2.6	ビルドオプションを設定する	51
2.7	ビルドのコンフィグレーション	52
2.7.1	ビルドコンフィグレーションを選択する	52
2.7.2	新規ビルドコンフィグレーションを追加する	53
2.7.3	ビルドコンフィグレーションを削除する	53
2.8	プロジェクトをビルドする	53
2.8.1	ファイルをコンパイルする	53
2.8.2	プロジェクトをビルドする	54
2.8.3	複数のプロジェクトをビルドする	54
2.8.4	ツールの実行を中止する	55
2.8.5	ビルドの中間ファイルおよび出力ファイルを削除する	55
2.8.6	アウトプットウィンドウの Build タブの構成	57
2.8.7	アウトプットウィンドウの Build タブの内容の制御	59
2.8.8	ビルド対象ファイルをワークスペースウィンドウにマーク表示する	59
2.9	ファイル依存関係	60
2.10	表示方法を指定する	62
2.11	ワークスペースにプロジェクトを追加する	64
2.12	アクティブプロジェクトを設定する	65
2.13	プロジェクト間の依存関係を指定する	66
2.14	ワークスペースからプロジェクトを削除する	67
2.15	ワークスペースのプロジェクト相対パス	67
3.	ビルドの応用	68
3.1	ビルド実行の復習	68
3.1.1	ビルドとは?	68
3.2	カスタムビルドフェーズを作成する	69
3.3	ビルドフェーズの順序	73
3.3.1	ビルド順序タブ	74
3.3.2	ファイルのビルド順序タブ	77



3.3.3	ファイルマッピングタブ .....	78
3.4	カスタムビルドフェーズのオプション設定 .....	79
3.4.1	オプションタブ .....	80
3.4.2	出力ファイルタブ .....	80
3.4.3	依存ファイルタブ .....	81
3.5	ビルドを管理する .....	82
3.6	ビルドの出力のログを取る .....	82
3.7	ツールチェーンのバージョンを変更する .....	83
3.8	Make ファイルの生成 .....	83
3.9	Make ファイルを使ったビルド .....	86
3.10	リンク順序をカスタマイズする .....	87
4.	エディタ .....	91
4.1	エディタウィンドウ .....	91
4.2	複数のファイルを扱う .....	92
4.3	標準のファイル操作 .....	92
4.3.1	新規ファイルを作成する .....	92
4.3.2	ファイルを編集する .....	93
4.3.3	ファイルを保存する .....	93
4.3.4	ファイルを開く .....	94
4.3.5	ファイルを閉じる .....	95
4.3.6	ウィンドウを閉じるポップアップメニュー .....	96
4.4	検索とファイル内の移動 .....	97
4.4.1	文字列を検索する .....	97
4.4.2	複数のファイルで文字列を検索する .....	98
4.4.3	文字列を置換する .....	99
4.4.4	指定した行にジャンプする .....	100
4.5	ブックマーク .....	100
4.6	ファイルを印刷する .....	101
4.7	テキストのレイアウト .....	102
4.7.1	ページ設定 .....	102
4.7.2	タブを変更する .....	103
4.7.3	自動インデント .....	103
4.8	ウィンドウを分割する .....	104
4.9	エディタのフォントを変更する .....	104
4.10	シンタックスを色付けする .....	105
4.10.1	テキストの色を変更する .....	105
4.10.2	新規キーワードを作成する .....	106
4.10.3	シンタックスの色付けを有効/無効にする .....	107
4.11	テンプレート .....	107

4.11.1	テンプレートを設定する	108
4.11.2	テンプレートを削除する	109
4.11.3	テンプレートを挿入する	109
4.12	括弧の組み合わせ	110
4.13	ファイルを読み取り専用属性にする	110
4.14	デバッグ時にファイルの編集を抑止する	111
4.15	エディタカラムの管理	111
4.16	カラムのヘッダの表示/非表示を切り替える	112
4.17	エディタ内からファイルを開く	112
4.18	ツールチップウォッチ	113
4.19	式の評価	113
5.	ツール管理	115
5.1	ツールの位置	115
5.2	High-performance Embedded Workshop 登録ファイル	116
5.3	ツールを登録する	117
5.4	ツールの登録を取り消す	118
5.5	ツールのプロパティの参照と編集	118
5.6	テクニカルサポートについて	120
5.7	オンデマンドのコンポーネント	121
5.8	カスタムプロジェクトタイプ	122
6.	環境のカスタマイズ	124
6.1	ツールバーのカスタマイズ	124
6.2	ツールメニューのカスタマイズ	126
6.3	カスタムブレースホルダを使う	128
6.4	ワークスペースやプロジェクトのログ機能を使う	129
6.5	ヘルプシステムを構築する	129
6.6	キーボードショートカットをカスタマイズする	130
6.7	基本設定の影響範囲	132
6.7.1	カスタマイズダイアログボックスの影響範囲	132
6.7.2	オプションダイアログボックスの影響範囲	132
6.8	ワークスペースオプションを指定する	132
6.8.1	起動時に前回開いたワークスペースを開く	133
6.8.2	ワークスペースを開いたときにファイルを表示する	133
6.8.3	ワークスペースを開いたときにワークスペース情報を表示する	133
6.8.4	ツール実行前にワークスペースを保存する	134
6.8.5	ワークスペース保存前に確認する	134
6.8.6	セッション保存前に確認する	134
6.8.7	自動バックアップ機能を有効にする	134

6.8.8	ワークスペースを開いたときに読み込むプロジェクトを設定する	135
6.8.9	新規ワークスペースのデフォルトディレクトリを変更する	136
6.9	外部エディタを使う	136
6.10	表示するフォントのカスタマイズ	138
6.11	バーチャルデスクトップを使う	139
7.	バージョン管理	141
7.1	バージョン管理システムを選択する	142
7.2	バージョン管理設定内容のインポートとエクスポート	143
8.	カスタムバージョン管理システム	144
8.1	バージョン管理メニューオプションを定義する	144
8.1.1	システム定義メニューオプション	145
8.1.2	ユーザ定義メニューオプション	146
8.2	バージョン管理コマンドを定義する	147
8.3	引数を指定する	148
8.4	コメントを指定する	149
8.5	バージョン管理コマンドリターンコードオプション	149
8.6	ディレクトリをマッピングする	149
8.7	ディレクトリマッピング例	151
8.8	バージョン管理環境変数を設定する	152
8.9	バージョン管理システムの実行を制御する	152
8.10	ユーザ名とパスワードを変更する	153
8.11	カスタムバージョン管理システムの使用例	154
8.11.1	High-performance Embedded Workshop と RCS を接続して使用する	154
8.11.2	High-performance Embedded Workshop と CVS を接続して使用する	161
9.	Visual SourceSafe バージョン管理システム	168
9.1	ワークスペースに Visual SourceSafe を関連付ける	168
9.2	Visual SourceSafe コマンド	171
9.2.1	Visual SourceSafe にファイルを追加する	171
9.2.2	Visual SourceSafe からファイルを削除する	172
9.2.3	Visual SourceSafe からファイルの読み取り専用コピーを取得する	173
9.2.4	Visual SourceSafe からファイルの書き込み可能コピーをチェックアウトする	173
9.2.5	Visual SourceSafe にファイルの書き込み可能コピーをチェックインする	175
9.2.6	Visual SourceSafe のファイルの状態を表示する	176
9.2.7	Visual SourceSafe のチェックアウトコマンドを取り消す	176
9.2.8	Visual SourceSafe のファイル履歴を表示する	177
9.3	Visual SourceSafe コマンドのオプション	178
9.4	バージョン管理設定内容	178
9.5	ディレクトリをマッピングする	179

9.6	Visual SourceSafe コマンドを追加する .....	180
10.	ネットワークを利用したプロジェクト共有機能.....	183
10.1	Windows® XP Service Pack 2 以降を搭載した PC で本機能を使う場合の設定 .....	184
10.2	ネットワークを利用したプロジェクト共有機能を使う .....	186
10.3	アドミニストレータのパスワードを設定する.....	187
10.4	新規ユーザを追加する .....	187
10.5	パスワードを変更する .....	189
10.6	ネットワークを利用したプロジェクト共有機能を使う場合の留意事項.....	189
11.	ファイルの比較 .....	190
11.1	差分ウィンドウを開く .....	190
12.	ナビゲーション機能 .....	193
12.1	C 関数と#define ナビゲーションの構成 .....	195
12.2	C++ナビゲーションの構成 .....	195
12.3	エディタ内から定義へジャンプする.....	197
12.4	ナビゲーション項目をドラッグアンドドロップする.....	198
12.5	スマートエディタ .....	199
13.	マップ.....	201
13.1	セクション設定を管理する .....	202
13.1.1	Map Section Information ウィンドウを開く .....	203
13.1.2	セクション情報編集モードの開始と終了.....	205
13.1.3	セクショングループを追加する.....	206
13.1.4	セクションを追加する .....	207
13.1.5	オーバーレイグループを追加する.....	207
13.1.6	未登録セクションを自動登録する.....	208
13.1.7	選択項目を編集する .....	209
13.1.8	プライマリセクションに設定する.....	209
13.1.9	メモリマップを設定する .....	209
13.1.10	メモリリソースを自動的に確保する.....	210
13.1.11	セクション設定ツリーを印刷する.....	210
13.1.12	未割り当て領域を表示する.....	211
13.1.13	サイズ 0 のセクションを表示する .....	211
13.1.14	アドレスに対応するソースコードを表示する.....	211
13.1.15	セクションリストを印刷する.....	211
13.2	シンボル情報を見る .....	212
13.2.1	Map Symbol Information ウィンドウを開く .....	212
13.2.2	マップリストを印刷する .....	213
13.2.3	シンボルを検索する .....	214

13.2.4	シンボル情報をフィルタ表示する	214
13.2.5	アドレスに対応するソースコードを表示する	215
13.2.6	シンボル情報を印刷する	216
14.	コマンドライン	217
14.1	コマンドラインウィンドウを開く	217
14.2	バッチファイルを設定する	220
14.3	バッチファイルを実行する	221
14.4	コマンド実行を中断する	221
14.5	ログファイルを設定する	222
14.6	ログファイルへの出力を開始/停止する	222
14.7	ファイルの絶対パスを入力する	222
14.8	プレースホルダを挿入する	222
14.9	ウィンドウの表示内容をすべて選択する	222
14.10	選択部分をクリップボードにコピーする	222
14.11	選択部分をクリップボードに切り取る	223
14.12	クリップボードの内容を貼り付ける	223
14.13	ウィンドウの表示内容をクリアする	223
14.14	直前の操作を元に戻す	223
14.15	括弧の呼応状態を確認する	223
14.16	バッチファイルの実行状態をリセットする	223
14.17	バッチファイルを1行単位で実行する	224
14.18	バッチファイルにブレークポイントを設定する	225
15.	マクロ生成支援機能	227
15.1	マクロメニューとツールバー	227
15.2	マクロの設定ダイアログボックスを使用する	229
15.3	既存のマクロファイルをインポートする	230
15.4	マクロを記録する	231
15.5	マクロファイルにマクロを記録できる機能	232
15.5.1	マクロを記録できる機能 ( High-performance Embedded Workshop 共通 )	232
15.5.2	マクロを記録できる機能 ( デバッガ依存 )	236
15.6	マクロを実行する	244
15.7	マクロを編集する	244
15.8	マクロを割り当てる	245
15.9	アウトプットウィンドウの Macro タブの構成	246
16.	テスト支援機能	247
16.1	テストスイートを作成する	247
16.2	テストスイートを開く / 閉じる	248
16.3	テストスイートを編集する	249

16.4	テストスイートにテストを追加する.....	250
16.5	テストイメージファイルを作成する.....	252
16.6	テストイメージファイルにテストイメージデータとして保存できる機能.....	253
16.6.1	テストイメージファイルに保存できる機能（High-performance Embedded Workshop 共通）..	254
16.6.2	テストイメージファイルに保存できる機能（デバugg依存）.....	259
16.7	テストイメージファイルを比較する.....	272
16.8	テストを実行する.....	273
16.9	テストブラウザを使用する.....	274
16.10	ワークスペースウィンドウの Test タブの構成.....	275
16.11	アウトプットウィンドウの Test タブの構成.....	276
17.	デバugg機能.....	278
17.1	デバuggの準備をする.....	278
17.1.1	デバuggの前にビルドを行う.....	278
17.1.2	デバuggを選択する.....	278
17.1.3	プロジェクトの構成を編集する.....	289
17.1.4	デバuggを設定する.....	289
17.1.5	ダウンロードモジュール.....	296
17.1.6	デバuggセッション.....	308
17.2	プログラムを表示する.....	313
17.2.1	エディタウィンドウを開く.....	314
17.2.2	逆アセンブリウィンドウを開く.....	321
17.2.3	現在の PC 位置を表示する.....	328
17.2.4	PC 位置の行を強調表示する.....	328
17.3	メモリを操作する.....	328
17.3.1	メモリウィンドウを開く.....	328
17.3.2	任意のアドレスにデータを設定する.....	331
17.3.3	アドレス領域を選択する.....	332
17.3.4	任意のアドレス領域を一定データで充填する.....	332
17.3.5	任意のアドレス領域のデータを別領域にコピーする.....	333
17.3.6	任意の 2 つのアドレス領域を比較する.....	333
17.3.7	任意のアドレス領域をテストする.....	334
17.3.8	任意のアドレス領域をテキストファイルに保存する.....	334
17.3.9	任意のアドレス領域のデータを検索する.....	335
17.3.10	表示開始アドレスを変更する.....	336
17.3.11	スクロール範囲を変更する.....	336
17.3.12	指定したレジスタが示すアドレスから表示する.....	336
17.3.13	スタックポインタ位置を追従する.....	337
17.3.14	ダウンロード時に表示するラベル位置を変更する.....	337
17.3.15	ウィンドウの表示内容を手動で更新する.....	337

17.3.16	ウィンドウの表示内容の更新を抑止する	337
17.3.17	ウィンドウの表示内容を定期的に更新する	337
17.3.18	更新間隔を設定する	338
17.3.19	表示データ長を変更する	339
17.3.20	表示基数を変更する	339
17.3.21	表示コードを変更する	339
17.3.22	レイアウトを設定する	339
17.3.23	表示桁数を変更する	340
17.3.24	カバレッジ計測結果の表示/表示なしを切り替える	340
17.3.25	任意のアドレス領域を保存する	340
17.3.26	メモリにファイルをロードする	341
17.3.27	ウィンドウを分割表示する	342
17.3.28	アドレス領域を検証する	342
17.3.29	テキストの色を変更する	342
17.4	メモリ内容を画像形式で表示する	343
17.4.1	画像ウィンドウを開く	343
17.4.2	ウィンドウの表示内容を定期的に更新する	346
17.4.3	ウィンドウの表示内容を手動で更新する	346
17.4.4	更新間隔を設定する	346
17.4.5	連続フレームで画像を見る	347
17.4.6	ピクセル情報を表示する	350
17.5	メモリ内容を波形形式で表示する	350
17.5.1	波形ウィンドウを開く	350
17.5.2	ウィンドウの表示内容を定期的に更新する	352
17.5.3	ウィンドウの表示内容を手動で更新する	352
17.5.4	更新間隔を設定する	352
17.5.5	拡大表示する	353
17.5.6	縮小表示する	353
17.5.7	最初のサイズに戻す	353
17.5.8	拡大/縮小倍率を設定する	353
17.5.9	横軸のサイズを設定する	353
17.5.10	カーソルを非表示にする	353
17.5.11	サンプリング情報を表示する	353
17.6	I/O レジスタを見る	354
17.6.1	IO ウィンドウを開く	354
17.6.2	I/O レジスタ表示を展開する	356
17.6.3	I/O レジスタの内容を設定する	356
17.6.4	ウィンドウの表示内容を手動で更新する	357
17.6.5	ウィンドウの表示内容の更新を抑止する	357

17.6.6	表示する I/O レジスタを選択する .....	357
17.6.7	I/O ファイルをロードする .....	358
17.6.8	IO ウィンドウの表示内容を印刷する .....	359
17.6.9	IO ウィンドウの表示内容を保存する .....	359
17.6.10	I/O レジスタを検索する .....	359
17.6.11	次を検索する .....	359
17.7	レジスタ内容を見る .....	360
17.7.1	レジスタウィンドウを開く .....	360
17.7.2	レジスタの表示基数を変更する .....	361
17.7.3	レジスタバンクを切り替える .....	361
17.7.4	レイアウトを設定する .....	361
17.7.5	表示するレジスタを選択する .....	362
17.7.6	レジスタの内容を設定する .....	363
17.7.7	フラグの値を設定する .....	363
17.7.8	ウィンドウの表示内容を手動で更新する .....	364
17.7.9	ウィンドウの表示内容の更新を抑止する .....	364
17.7.10	ウィンドウを分割表示する .....	364
17.7.11	レジスタウィンドウの表示内容を保存する .....	364
17.7.12	レジスタの内容を使用する .....	364
17.7.13	テキストの色を変更する .....	364
17.8	基数を設定する .....	365
17.9	ターゲットマイコンをリセットする .....	365
17.10	カーソル位置に PC 値を設定する .....	365
17.11	デバッガを初期化する .....	366
17.12	デバッガを接続/接続解除する .....	366
17.13	プログラムを実行する .....	366
17.13.1	実行を継続する .....	366
17.13.2	リセットから実行を開始する .....	366
17.13.3	ブレークポイントを無視して実行する .....	367
17.13.4	カーソルまで実行する .....	367
17.13.5	条件を指定して実行する .....	367
17.13.6	リセット時に main 関数の先頭までプログラムを実行する .....	368
17.13.7	シングルステップ .....	369
17.13.8	複数のステップ .....	370
17.14	プログラムを停止する .....	370
17.14.1	停止ツールバーボタンによる停止 .....	370
17.14.2	標準のブレークポイント (PC ブレークポイント) .....	371
17.15	現在の状態を表示する .....	372
17.16	関数呼び出し履歴を見る .....	373



17.16.1	スタックトレースウィンドウを開く	373
17.16.2	ソースコードを表示する	374
17.16.3	表示形式を設定する	374
17.16.4	エンコード形式を設定する	374
17.17	外部デバッガを使う	375
17.17.1	日立デバッグインタフェースを High-performance Embedded Workshop と統合して構築する	375
17.17.2	PD デバッガを High-performance Embedded Workshop と統合して構築する	376
17.17.3	外部デバッガを High-performance Embedded Workshop と統合して構築する	376
17.18	デバッガに依存する機能	377
17.18.1	ラベルを見る	377
17.18.2	Elf/Dwarf2 のサポート	381
17.18.3	変数を表示する	384
18.	同期デバッグ機能	400
18.1	同期デバッグダイアログボックス	400
18.1.1	同期デバッグ設定を管理する	401
18.1.2	同期するセッションを定義する	402
18.1.3	同期オプションを設定する	405
18.1.4	メモリ更新オプションを設定する	406
18.1.5	同期デバッグモード	406
18.1.6	同期デバッグを開始する	407
18.1.7	同期デバッグを更新する	407
18.1.8	同期デバッグを停止する	407
18.2	同期デバッグ時に High-performance Embedded Workshop を使用する	407
18.2.1	共通の機能	408
18.2.2	Parallel モードの機能	411
18.2.3	Internal モードの機能	412
18.3	同期デバッグ時にコマンドラインウィンドウを使用する	413
18.4	用語の解説	414
19.	テクニカルサポート	415
19.1	バージョン情報の参照	415
19.2	アップデートの確認	415
19.3	不具合レポートの作成	416
	リファレンス	417
1.	メインメニュー	418
1.1	ファイルメニューオプション	418
1.2	編集メニューオプション	418
1.3	表示メニューオプション	419

1.4	プロジェクトメニューオプション	420
1.5	ビルドメニューオプション	421
1.6	デバッグメニューオプション	421
1.7	基本設定メニューオプション	422
1.8	ツールメニューオプション	423
1.9	テストメニューオプション	423
1.10	ウィンドウメニューオプション	424
1.11	ヘルプメニューオプション	424
2.	ウィンドウ	425
3.	コマンド	426
3.1	コマンド一覧 (アルファベット順)	426
3.2	コマンド一覧 (機能別)	427
4.	正規表現	430
5.	プレースホルダ	431
5.1	プレースホルダとは?	431
5.2	プレースホルダを挿入する	431
5.3	使用できるプレースホルダ	432
5.4	プレースホルダを使うにあたって	433
6.	I/O ファイルフォーマット	434
7.	シンボルファイルフォーマット	436
8.	キーボードショートカット	437
9.	デバッガでドラッグアンドドロップする	440
10.	ラベルを使用してコードを参照する	441
11.	ウィンドウのツールバー	443
12.	High-performance Embedded Workshop V.1.x 用のツールチェインでビルドする	445
13.	HMAKE ユーザガイドライン	446
13.1	コマンドライン	446
13.2	ファイルのシンタックス	446
13.3	記述部	447
13.4	コメント	449
13.5	メッセージコマンド	449

## 1. 概要

本ユーザーズマニュアルでは High-performance Embedded Workshop の機能について記載しています。この章では High-performance Embedded Workshop の基本概念を説明します。

### 1.1 ワークスペースとプロジェクト

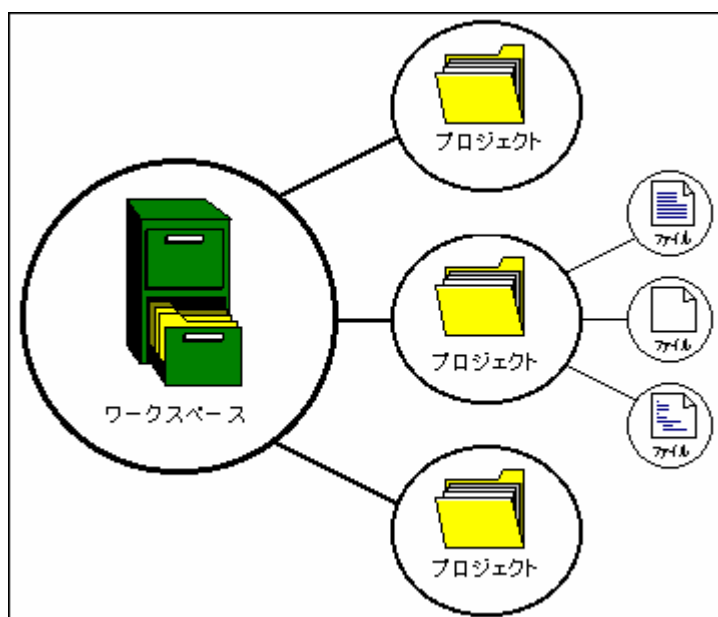
High-performance Embedded Workshop では、ワークスペースとプロジェクトという概念があります。

- ワークスペース

High-performance Embedded Workshop を用いてプログラムを作成する場合の、一番大きな管理単位です。ワークスペースには、最低 1 つのプロジェクトが必要で、この 1 つのプロジェクトは、ワークスペース作成時に自動的に作成されます。また、ワークスペースは複数のプロジェクトを持つことができます。

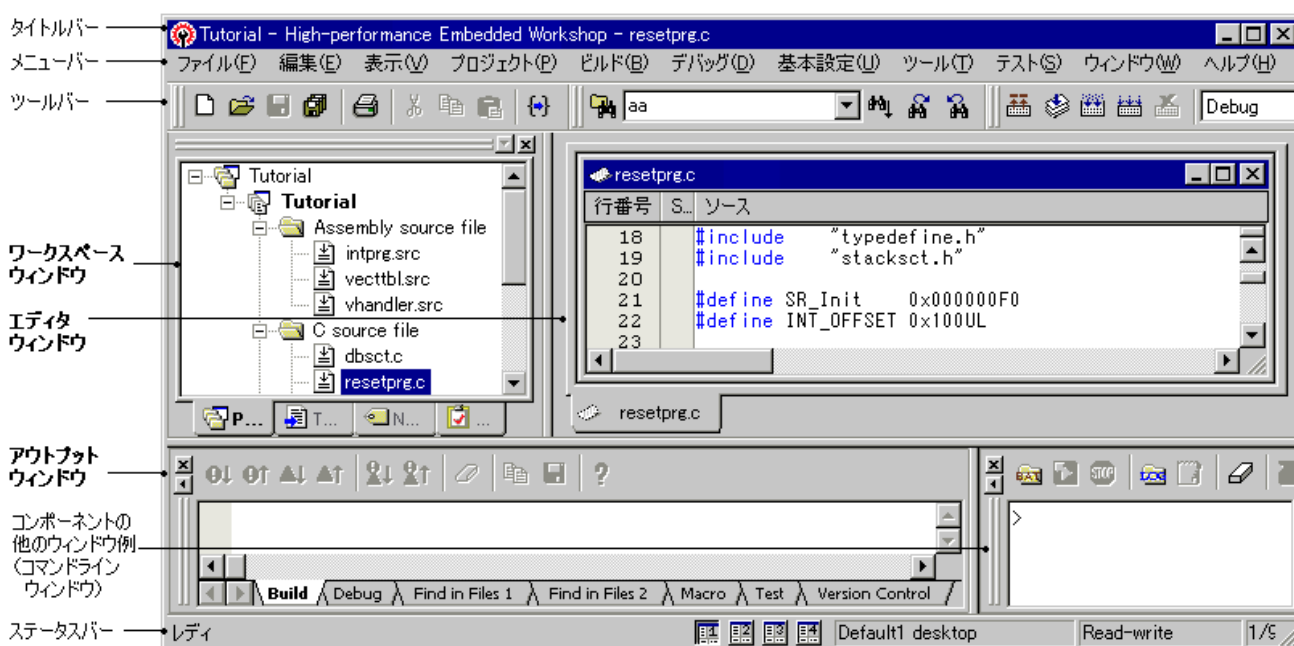
- プロジェクト

プログラムを作成する場合、特定機能をライブラリ化してモジュールを階層化する場合があります。このような場合、ライブラリ用のプロジェクトを作成、追加することができます。



### 1.2 メインウィンドウ

High-performance Embedded Workshop にはメインウィンドウが 3 つあります。ワークスペースウィンドウ、エディタウィンドウ、アウトプットウィンドウです。ワークスペースウィンドウには現在そのワークスペースにあるプロジェクトやファイルを表示します。エディタウィンドウではファイルを表示、編集できます。アウトプットウィンドウには様々な処理結果（ビルド、バージョン管理コマンドなど）を表示します。



### 1.2.1 タイトルバー

タイトルバーには、現在開いている、プロジェクト名、ファイル名を表示します。また、"最小化"ボタン、"最大化"ボタン、"閉じる"ボタンがあります。"最小化"ボタンをクリックすると Windows®のタスクバー上に High-performance Embedded Workshop を最小化します。"最大化"ボタンをクリックすると High-performance Embedded Workshop をフルスクリーンで表示します。"閉じる"ボタンをクリックすると High-performance Embedded Workshop を閉じることができます。（これは[ファイル->アプリケーションの終了]を選択するか "Alt+F4"キーを押すのと同じです）。

### 1.2.2 メニューバー

メニューバーには次の 11 のメニューがあります。[ファイル]、[編集]、[表示]、[プロジェクト]、[ビルド]\*、[デバッグ]、[基本設定]、[ツール]、[テスト]、[ウインドウ]、および[ヘルプ]です。メニューのオプションは、すべてこれらメニューの下にグループ化されています。例えば、ファイルを開きたいときには[ファイル]メニューの下のオプションを選択します。ツールをセットアップしたいときには、[ツール]メニューを選択します。

ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) 基本設定(U) ツール(T) テスト(S) ウインドウ(W) ヘルプ(H)

#### 注：

- \* High-performance Embedded Workshop V.4.01 以降のバージョンで作成したデバッグ専用プロジェクト（[Debugger only - xxxxxx]）を使用している場合は、[ビルド]メニューを表示しません。High-performance Embedded Workshop V.4.01 より前のバージョンで作成したデバッグ専用プロジェクトを使用している場合には、[ビルド]メニューを表示します。

### 1.2.3 ツールバー

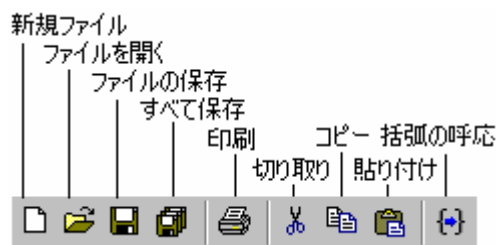
ツールバーにより使う頻度の高いオプションを簡単にアクセスできます。デフォルトでは[エディタ]、[検索]、[テンプレート]、[ブックマーク]、[デフォルト]、[標準]、[バージョン管理]、[マップ]、[ツール]、[マクロ]、[デバッグ]、[デバッグラン]、および[システムツール]の13のツールバーがあります（下の図を参照）。

デフォルトセッションの初期表示では、ツールバーには[バージョン管理]、および周辺機能のボタンは表示しません。

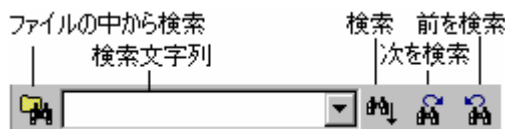
さらに、High-performance Embedded Workshop V.4.01 以降のバージョンで作成したデバッグ専用プロジェクト（[Debugger only - xxxxxx]）を使用する場合、デフォルトでは[エディタ]、[検索]、[テンプレート]、[ブックマーク]、[デフォルト]、および[標準]は表示しません。

ツールバーの作成や変更は、[基本設定 -> カスタマイズ]で指定できます（詳細は、「6.1 ツールバーのカスタマイズ」を参照してください）。

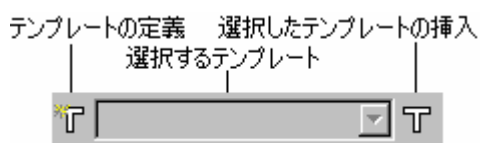
#### エディタツールバー



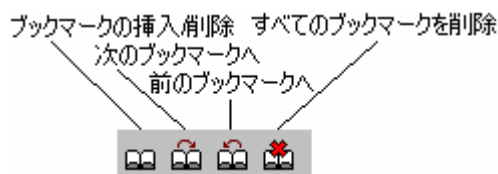
#### 検索ツールバー



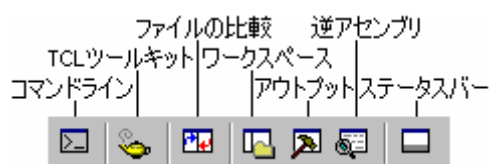
#### テンプレートツールバー



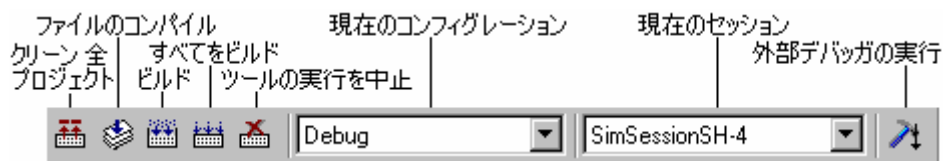
#### ブックマークツールバー



#### デフォルトツールバー

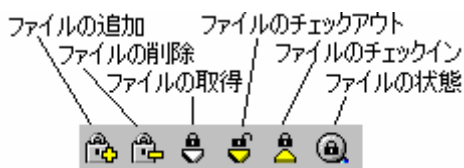


### 標準ツールバー



### バージョン管理ツールバー

現在のプロジェクトでバージョン管理ツールが選択されているときのみ使用できます。



### マップツールバー

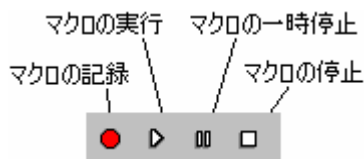


### ツールツールバー



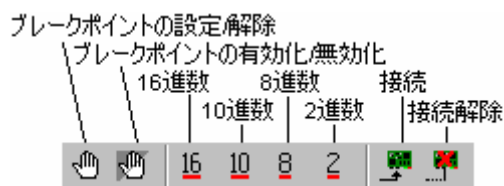
### マクロツールバー

下記の標準のマクロボタンの右側に作成したマクロを割り当てたカスタムのボタンを追加できます。



### デバッグツールバー

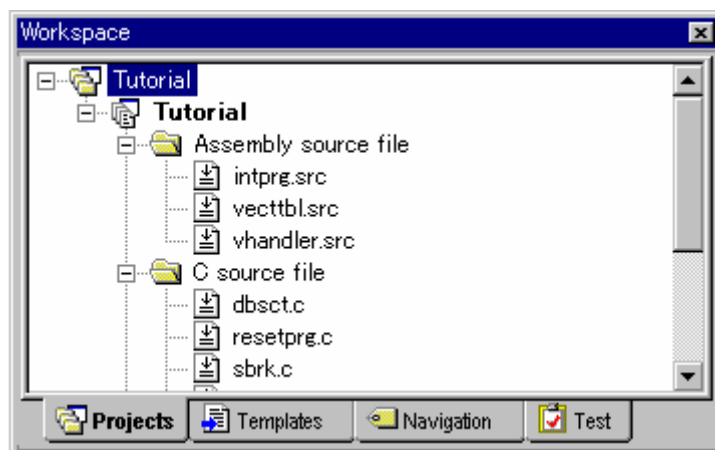
ターゲットが接続されたセッションを使用しているときのみ使用できます。





## 1.2.4 ワークスペースウィンドウ

ワークスペースウィンドウには4つのタブ（[Projects]、[Templates]、[Navigation]、[Test]）があります。



- **[Projects]タブ**

現在のワークスペース、プロジェクト、ファイルを表示します。アイコンのダブルクリックによりプロジェクトファイルや個々のファイルが開けます。[Projects]タブの詳細については「2.2 ワークスペースウィンドウの Projects タブの構成」を参照してください。

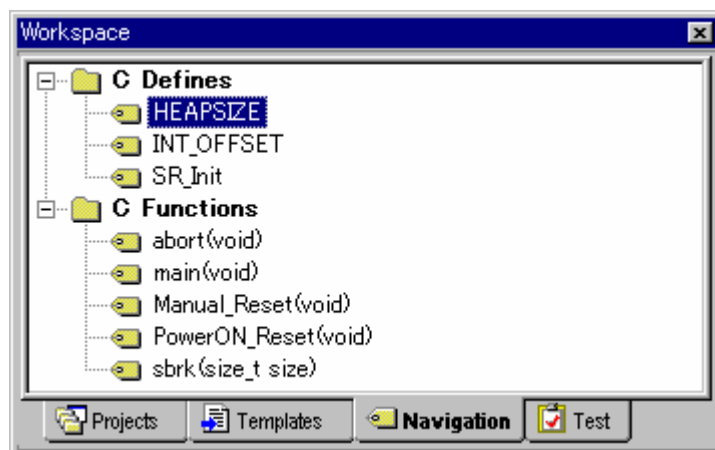
[Projects]タブのファイル名上にマウスポインタを置くと、ツールチップにファイルの絶対パスを表示します。



- **[Templates]タブ**

テンプレートの設定を表示します。テンプレートの詳細については「4.11 テンプレート」を参照してください。

- **[Navigation]タブ**





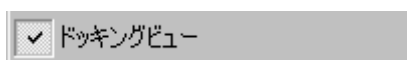
プロジェクトのファイルの中のテキスト部分へジャンプできます。[Navigation]タブに実際に表示される内容は、現在、何がインストールされているかによって異なります。上の図に例えば ANSI 規格の C マクロ名定義 (C Defines)、および C 関数一覧 (C Functions) を示します。ナビゲーションの詳細については「12. ナビゲーション機能」を参照してください。

### • [Test]タブ

テスト支援機能としてテストスイートを設定および表示します。テスト支援機能の詳細については、「16. テスト支援機能」を参照してください。

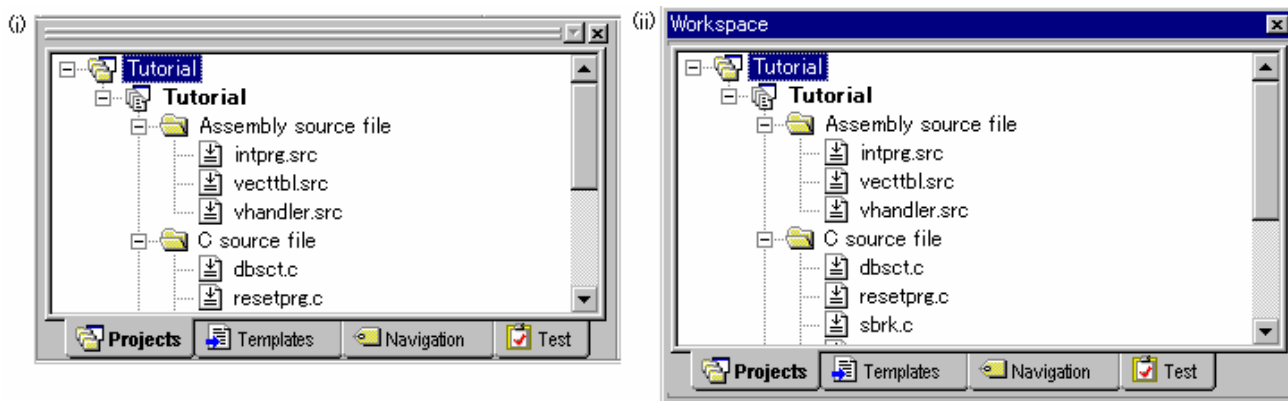
### ワークスペースウィンドウやアウトプットウィンドウのドッキングビューとは

ウィンドウ上で右クリックしてください。ポップアップメニューを表示します。

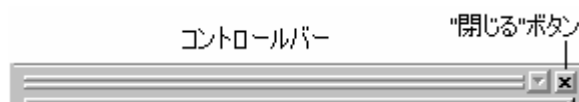


[ドッキングビュー]にチェックマークが付いている場合はドッキングできます。チェックマークが外れている場合はドッキングできません。[ドッキングビュー]を選択するとチェックマークが付いたり外れたりします。

[ドッキングビュー]にチェックマークが付いている場合、ウィンドウを High-performance Embedded Workshop メインウィンドウや他のドッキング状態のウィンドウの端にドッキングできます。同じく[ドッキングビュー]にチェックマークが付いている場合、ウィンドウを他の High-performance Embedded Workshop のウィンドウ上や High-performance Embedded Workshop メインウィンドウの外でフローティング状態にできます。下の図の(i)はドッキング状態のワークスペースウィンドウ、(ii)はフローティング状態のワークスペースウィンドウを示します。



ワークスペースウィンドウやアウトプットウィンドウがドッキング状態のとき、下の図に示すコントロールバーを表示します。ドッキング状態のウィンドウを移動したいとき、コントロールバーを移動先までドラッグしてください。



### ワークスペースウィンドウやアウトプットウィンドウをドッキング状態にするには

1. ポップアップメニューで[ドッキングビュー]にチェックマークが付いていることを確認してください。
2. 以下のいずれかの操作を選択してください。
  - フローティング状態のウィンドウのタイトルバーをダブルクリックする。
  - フローティング状態のウィンドウのタイトルバーを移動先のドッキング状態のウィンドウ、メニューバー、ツールバー、または High-performance Embedded Workshop のメインウィンドウの端までドラッグする。

### ワークスペースウィンドウやアウトプットウィンドウをフローティング状態にするには

1. ポップアップメニューで[ドッキングビュー]にチェックマークが付いていることを確認してください。
2. 以下のいずれかの操作を選択してください。
  - ドッキング状態のウィンドウのコントロールバーをダブルクリックする。
  - ドッキング状態のウィンドウのコントロールバーを High-performance Embedded Workshop のメインウィンドウや他のドッキング状態のウィンドウ、メニューバー、ツールバーの端から外れるようにドラッグする。
  - ドッキング状態のウィンドウのコントロールバーを"Ctrl"キーを押しながらドラッグする。

### ワークスペースウィンドウやアウトプットウィンドウを隠すには

以下のいずれかの操作を選択してください。

- ウィンドウの右上端にある"閉じる"ボタンをクリックする。
- フローティング状態のウィンドウの中で右クリックし、ポップアップメニューから[非表示]を選択する。

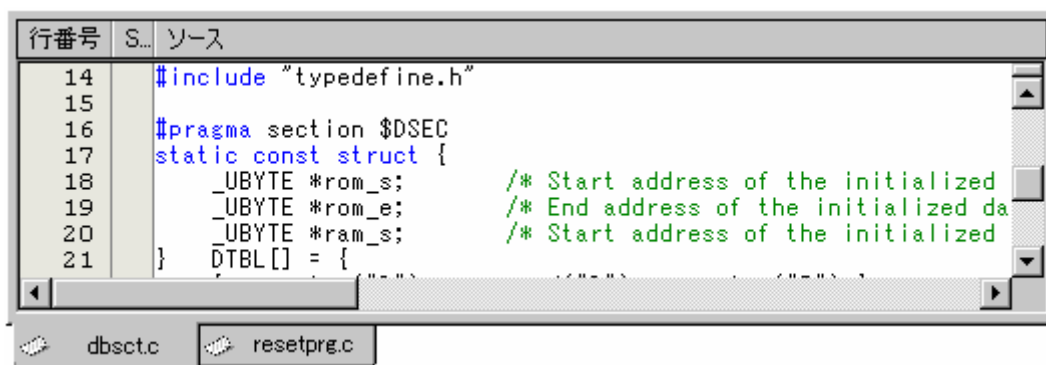


### ワークスペースウィンドウやアウトプットウィンドウを表示するには

ワークスペースウィンドウを表示するには[表示 -> ワークスペース]、アウトプットウィンドウを表示するには[表示 -> アウトプット]を選択してください。

## 1.2.5 エディタウィンドウ

エディタウィンドウではプロジェクトのファイル进行操作します。同時に複数のファイルを開いたり、任意の順序にファイルを切り替えたり、並べ替えたり、編集できます。デフォルトでは、エディタウィンドウはブック形式で表示します。つまり、各ファイルにタブが付いていてファイル間の行き来を容易にできます（下の図参照）。



エディタには、ウィンドウ内の左側に余白（カラム）があります。これにより、ブックマークとブレークポイント\* の位置を簡単に設定できます。カラムの用途やその表示する情報について知りたい場合は、そのカラム上にマウスポインタを置いてください。これを説明するツールチップとしてポップアップウィンドウを開きます。

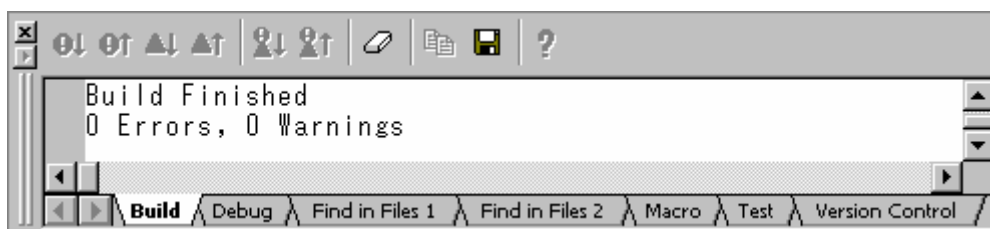
エディタウィンドウは[表示形式]ダイアログボックスでカスタマイズできます。[表示形式]ダイアログボックスは[基本設定 -> 表示の形式]メニューで開きます。[表示形式]ダイアログボックスでは、エディタウィンドウのフォントやテキストの色、タブ文字などが変更できます。また、High-performance Embedded Workshop でインストールした他のウィンドウの外観も変更できます。High-performance Embedded Workshop エディタ以外のエディタを使う場合は、使用するエディタを[オプション]ダイアログボックスで指定してください。[オプション]ダイアログボックスは、[基本設定 -> オプション]から開くことができます。エディタの使用方法については、「4. エディタ」を参照してください。

注：

\*. デバッガに接続している場合のみブレークポイントを設定できます。

## 1.2.6 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7つのタブ（[Build]、[Debug]、[Find in Files1]、[Find in Files2]、[Macro]、[Test]、[Version Control]）があります。



### [Build]タブ

ビルド実行（コンパイラ、アセンブラなど）の出力を表示します。ソースファイルにエラーがある場合、アウトプットウィンドウの[Build]タブにアイコン、およびエラーメッセージをソースファイル名、行番号とともに出力します。

エラーメッセージの表示に関するツールバーボタンまたはポップアップメニューオプションを選択すると、エラーメッセージ出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。

エラーメッセージ出力行をダブルクリックすると、該当ソースがある場合はエディタにエラーメッセージの該当ソース行を表示します。




エラーメッセージの表示に関する操作を行った場合は、ステータスバーにエラーメッセージを表示します。

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。  
基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
-		次のエラーの出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。
-		前のエラーの出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。
-		次のウォーニングの出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。
-		前のウォーニングの出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。
次エラー/ウォーニング/情報の表示		次のエラー、ウォーニング、またはインフォメーションの出力行を強調表示し、該当ソース行がある場合はエディタに該当ソース行を表示します。
前エラー/ウォーニング/情報の表示		前のエラー、ウォーニング、またはインフォメーションの出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。
ヘルプ		この出力行に関するヘルプを表示します。
ジャンプ	-	この出力行に関するソース行をエディタに表示します。
クリア		ウィンドウ内容をクリアします。
保存		ウィンドウ内容をテキストファイルに保存します。
コピー		ウィンドウ内容の選択部分を Windows® クリップボードにコピーします。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。

[Build]タブではビルド実行により出力されたエラーメッセージ行の左側の余白（カラム）にエラーメッセージに対応したアイコンを表示します。




アイコン名	アイコン	エラーメッセージレベル
Build Error		エラー
Build Warning		ウォーニング
Information		インフォメーション

## [Debug]タブ

あらゆるデバッグ処理の出力を表示します。情報を表示する必要のあるデバッグツールから、アウトプットウィンドウの[Debug]タブに出力します。

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。  
基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
クリア		ウィンドウ内容をクリアします。
保存		ウィンドウ内容をテキストファイルに保存します。
コピー		ウィンドウ内容の選択部分を Windows® クリップボードにコピーします。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。




**[Find in Files 1]および[Find in Files 2]タブ**

ファイル内から検索操作の結果を表示します。ファイル内から検索を使用するには、[編集 -> ファイルから検索]を選択するか、ツールバーの[ファイルの中から検索]ボタンをクリックしてください。ファイル内から検索の使い方の詳細については、「4.4.2 複数のファイルで文字列を検索する」を参照してください。

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
ジャンプ	-	関連するソース行を表示します。
クリア		ウィンドウ内容をクリアします。
保存		ウィンドウ内容をテキストファイルに保存します。
コピー		ウィンドウ内容の選択部分を Windows® クリップボードにコピーします。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。




**[Macro]タブ**

マクロ生成支援機能として、現在のマクロの記録状況を表示します。[ツール -> マクロの記録]による記録の開始から[ツール -> マクロの停止]による記録の終了までに High-performance Embedded Workshop マクロファイルに記録した High-performance Embedded Workshop コマンドラインのコマンドなどの情報を表示します。マクロを記録しながら記録内容を確認できます。マクロ生成支援機能の詳細については、「15. マクロ生成支援機能」を参照してください。

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
クリア		ウィンドウ内容をクリアします。
保存		ウィンドウ内容をテキストファイルに保存します。
コピー		ウィンドウ内容の選択部分を Windows® クリップボードにコピーします。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。




**[Test]タブ**

テスト支援機能として、現在のテスト実行の結果および進行状況を表示します。テスト実行の進行状況は、現在実行中のテストと、残りのテストの数で構成されています。エラーが発生した場合、このウィンドウに表示されます。テスト支援機能の詳細については、「16. テスト支援機能」を参照してください。

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
クリア		ウィンドウ内容をクリアします。
保存		ウィンドウ内容をテキストファイルに保存します。
コピー		ウィンドウ内容の選択部分を Windows® クリップボードにコピーします。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。




### [Version Control]タブ

バージョン管理操作の結果を表示します。このタブは、バージョン管理システムを使っているときだけ表示します。バージョン管理の詳細については、「7. バージョン管理」を参照してください。

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
クリア		ウィンドウ内容をクリアします。
保存		ウィンドウ内容をテキストファイルに保存します。
コピー		ウィンドウ内容の選択部分を Windows® クリップボードにコピーします。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。

"Shift+ESC"キーを押すとアウトプットウィンドウを閉じます。

アウトプットウィンドウは文字列の色、背景色、およびフォントを他のウィンドウと同じようにカスタマイズできます。

さらにアウトプットウィンドウの[Build]タブでは、エラーメッセージ行の文字列を他の行の文字列と異なる色で強調表示されるようにカスタマイズが可能です。

### 既存の色をカスタマイズするには

1. [基本設定 -> 表示の形式]を選択してください。[表示形式]ダイアログボックスを開きます。
2. [Output]アイテムをダイアログボックスの左側のツリーから選択し展開します。
3. 色を変更したいカテゴリを選択します。次の表に示すカテゴリが選択できます。

カテゴリ	カラータブの前面の色 (デフォルト)	カラータブの背景の色 (デフォルト)	カスタマイズされるタブ	カスタマイズされる出力内容
Text	SYSTEM	SYSTEM	すべてのタブ	すべて
Build Error Text	黒	白	Build	エラー行
Build Warning Text	黒	白	Build	ウォーニング行
Information Text	黒	白	Build	インフォメーション行

4. [カラー]タブの[前面]リストと[背景]リストの選択を変更します。
5. [OK]ボタンをクリックします。



### 1.2.7 ステータスバー

現在の High-performance Embedded Workshop の状態に関する様々な情報を表示します。下の図にステータスバーを示します。



ステータスバーのカスタマイズ機能によって、ステータスバー領域およびその内容をカスタマイズできます。ステータスバー領域には、特定のターゲットによって必要なターゲット関連の情報も表示します。

ステータスバー領域の最下部には一般的なアプリケーション関連の情報（以前のバージョンと同様の内容）を表示します。

ターゲット関連の情報をその上に表示します。表示情報はすべてターゲットに依存するため、表示されない場合もあります。

[表示 -> ステータスバー]の選択により、ステータスバーの表示/非表示の切り替えが可能です。

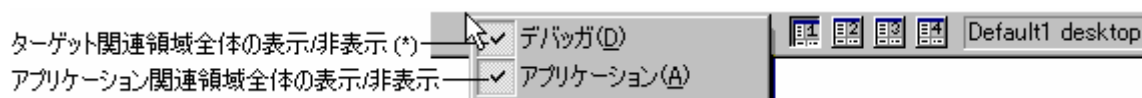
ステータスバー領域の一部でも表示している場合に選択すると、ステータスバー領域をすべて非表示にします。

ステータスバー領域を表示していない場合に選択すると、アプリケーション関連領域を表示します。

ステータスバーのポップアップメニューの選択により、ステータスバーを非表示にできます。

ポップアップメニューは、アプリケーションまたはターゲット関連領域にある個々の項目も含めたステータスバー項目のオン/オフの切り替えが可能です。

ステータスバー領域上で右マウスボタンをクリックするとポップアップメニューを表示します。



#### 注：

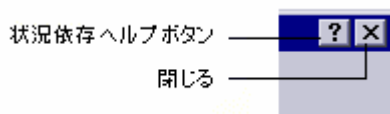
\*. ステータスバーのターゲット関連領域は、ターゲットによっては表示されません。

すべてのターゲット関連のステータスバー項目をオフにすると、ステータスバーのターゲット関連領域全体が非表示になります。ターゲット関連のステータスバー項目のいずれかをオンにすることにより再び表示できます。

### 1.3 ヘルプ機能

[ヘルプ]メニューは、High-performance Embedded Workshop メニューバーの右端にあります。[ヘルプ]メニューには[トピック]があります。[トピック]を選択すると、High-performance Embedded Workshop ヘルプウィンドウのメイン画面を表示します。

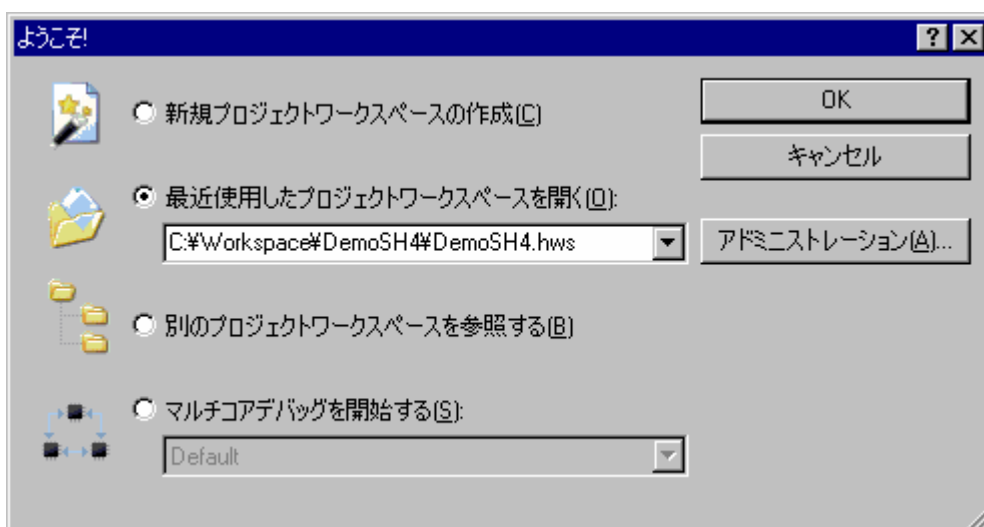
特定のダイアログボックスに関するヘルプを参照したいときは、各ダイアログボックスの右上端にある状況依存ヘルプボタンをクリックしてください（下の図を参照）。



状況依存ヘルプボタンをクリックすると、マウスポインタが?(クエスチョンマーク)付きのポインタに変わります。この状態で、ダイアログボックスの一部をクリックすると、その部分に関するヘルプを表示します。また、ダイアログボックスの一部を選択し F1 キーを押すと、そのヘルプを表示します。

#### 1.4 High-performance Embedded Workshopを起動する

High-performance Embedded Workshop を起動するには、Windows® の[スタート]メニューを開き、[すべてのプログラム -> Renesas -> High-performance Embedded Workshop]を選択し、High-performance Embedded Workshop のショートカットを選択します。起動画面に続き[ようこそ!]ダイアログボックスが開きます。



High-performance Embedded Workshop を起動しワークスペースを使用したことがある場合は、デフォルトで[最近使用したプロジェクトワークスペースを開く]ボタンが選択されています。一度もワークスペースを使用していない場合は、[新規プロジェクトワークスペースの作成]ボタンが選択されています。

新規ワークスペースを作成するには、[新規プロジェクトワークスペースの作成]ボタンを選択し、[OK]ボタンをクリックしてください。

最近使用したワークスペースを開くには、[最近使用したプロジェクトワークスペースを開く]ボタンを選択し、ドロップダウンリストから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。このリストには、[ファイル->最近使ったワークスペース]のサブメニューに表示されているものと同じ内容を表示します。

ワークスペースファイル(".HWS"ファイル)を指定してワークスペースを開くには[別のプロジェクトワークスペースを参照する]ボタンを選択し、[OK]ボタンをクリックしてください。

[同期デバッグ]ダイアログボックスを開くには、[マルチコアデバッグを開始する]ボタンを選択し、ドロップダウンリストから同期デバッグ設定を選択し、[OK]ボタンをクリックしてください。同期デバッグ機能を使用していない場合、この項目は表示されません。



ツールを登録したり、ツールの登録を取り消したりするには、[アドミニストレーション]ボタンをクリックしてください。

ワークスペースを開かずに High-performance Embedded Workshop を使うには[キャンセル]ボタンをクリックしてください。

次回起動する時に[ようこそ!]ダイアログボックスを開かないようにするには、[基本設定 -> オプション]を選択し、表示される[オプション]ダイアログボックスの[確認]タブで[ようこそダイアログボックスの表示]チェックボックスをオフにしてください。

High-performance Embedded Workshop を起動する時に起動画面を表示しないようにするには、[オプション]ダイアログボックスの[確認]タブで[起動画面]チェックボックスをオフにしてください。

## 1.5 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

1. [ようこそ!]ダイアログボックスから[新規プロジェクトワークスペースの作成]オプションを選択し、[OK]ボタンをクリックするか、[ファイル -> 新規ワークスペース]を選択してください。[新規プロジェクトワークスペース]ダイアログボックスを開きます。
2. [ワークスペース名]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、半角下線のみ使用してください。特に、日本語文字、空白、または半角マイナス記号は使用しないでください。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]が追加されます。これは、必要に応じて変更できます。ワークスペースとプロジェクトは別名にできます。[参照]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択したり、[ディレクトリ]にワークスペースを作成するディレクトリを入力できます。
3. ワークスペースの基本となる[CPU 種別]および[ツールチェーン]を選択してください。
4. 新規ワークスペースを作成するとき、High-performance Embedded Workshop は自動的に[プロジェクト名]で指定し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。プロジェクトタイプリストには、使用可能なプロジェクトの種類 (Application、Library 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択してください。表示するプロジェクトの種類は、現在の[CPU 種別]および[ツールチェーン]の組み合わせに有効な全種類となります。ツールチェーン専用、デバッグ専用 ([Debugger only - xxxxxx])、または High-performance Embedded Workshop のデバッグおよびツールチェーンの両方を設定するフルプロジェクトがあります。
5. 新規ワークスペースとプロジェクトを作成するには、[OK]ボタンをクリックしてください。選択したプロジェクトの作成をガイドするウィザードが開きます。

注：

同一ディレクトリに既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。

## 1.6 ワークスペースを開く

ワークスペースを開くには

1. [ようこそ!]ダイアログボックスから[別のプロジェクトワークスペースを参照する]オプションを選んで[OK]ボタンをクリックするか、[ファイル -> ワークスペースを開く]を選択してください。[ワークスペースを開く]ダイアログボックスを開きます。
2. 開きたいワークスペースファイル (".HWS"ファイル) を選択します。
3. [選択]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している場合、[ワークスペースプロパティ]ダイアログボックスを開きます。設定していない場合、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を開くかどうかは[ワークスペースプロパティ]ダイアログボックスの[ワークスペースを開いたときにワークスペース情報の表示]チェックボックス、ま

たは、[オプション]ダイアログボックスの[ワークスペース]タブの[ワークスペースを開いたときにワークスペース情報の表示]チェックボックスのオンオフの有無によります。[オプション]ダイアログボックスは、[基本設定->オプション]から開くことができます。[ワークスペースプロパティ]ダイアログボックスで[OK]ボタンをクリックするとワークスペースを開きます。[キャンセル]ボタンをクリックするとワークスペースを開きません。

ワークスペースを開いたときにワークスペース情報の表示(D)

最近開いたファイルを[ファイル]メニューの[最近使ったワークスペース]サブメニューに表示します。最近使ったワークスペースを再び開きたいときに使用してください。

### 最近使ったワークスペースを開くには

以下のいずれかの操作を選択してください。

- [ようこそ!]ダイアログボックスから[最近使用したプロジェクトワークスペースを開く]を選択し、ドロップダウンリストからワークスペース名を選択し、[OK]ボタンをクリックします。
- [ファイル->最近使ったワークスペース]を選択し、そのサブメニューからワークスペース名を選択します。

### 注：

High-performance Embedded Workshop 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、既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、既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く必要があります。

## 1.7 古いワークスペースを使用する

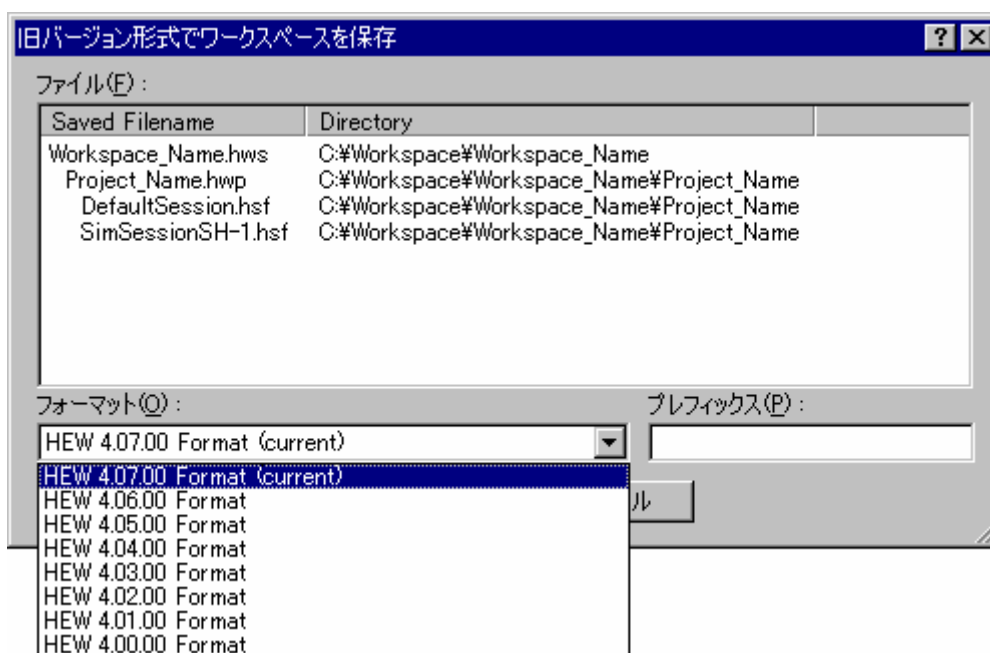
High-performance Embedded Workshop では、以前のバージョンの High-performance Embedded Workshop で作成したワークスペースも開くことができます。ワークスペースを開いたときに自動的にアップグレードします。アップグレードしたファイルの現在のディレクトリに、ワークスペースまたはプロジェクトのファイルのバックアップを保存します。

## 1.8 ワークスペースを保存する

ワークスペース（およびプロジェクト）の情報は、High-performance Embedded Workshop のバージョン毎に増えています。そのため、ワークスペースを保存する形式が High-performance Embedded Workshop のバージョン毎に少しずつ異なります。

現在のバージョン形式のワークスペースとして保存するには、[ファイル->ワークスペースの保存]を選択してください。

旧バージョン形式のワークスペースとして保存するには、[ファイル->旧バージョン形式でワークスペースを保存]を選択してください。前回ワークスペースを保存してからワークスペースまたはプロジェクトの情報を変更している場合は、現在のファイルとバージョンで保存するか否かを確認するメッセージを表示します。続いて[旧バージョン形式でワークスペースを保存]ダイアログボックスが開きます。



このダイアログボックスには、現在のワークスペースに関連するファイルのリスト、選択可能なバージョンをリストアップしたドロップダウンリスト（デフォルトで現在のバージョンを選択）、関連するファイル名を一括して変更するために短いプレフィックスを入力できるエディットボックスがあります。プレフィックスを変更すると、[ファイル]リストの表示を新しい名前でも更新します。

**例として High-performance Embedded Workshop V.4.02.00 で読み込み可能な形式としてワークスペースを保存するには**

1. [フォーマット]ドロップダウンリストで「HEW 4.02.00 Format」を選択します。
2. [プレフィックス]に「402\_」を入力します。
3. OK をクリックします。

現在のディレクトリ（ソースファイルなどそのまま使用可能）に、入力したプレフィックス付きファイル名でワークスペースを保存します。

**例:**

```
C:\Workspace\Workspace_Name\402_Workspace_Name.hws
C:\Workspace\Workspace_Name\DemoSH4\402_Project_Name.hwp
C:\Workspace\Workspace_Name\DemoSH4\402_DefaultSession.hsf
C:\Workspace\Workspace_Name\DemoSH4\402_SimSessionSH-1.hsf
```

現在のディレクトリ（ソースファイルなどそのまま使用可能）に、入力したプレフィックス付きファイル名でワークスペースを保存します。

## 1.9 ワークスペースを閉じる

High-performance Embedded Workshop のワークスペースを閉じるには、[ファイル->ワークスペースを閉じる]を選択してください。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択してください。

## 1.10 High-performance Embedded Workshopを終了する

High-performance Embedded Workshop を終了するには [ファイル->アプリケーションの終了]を選択するか、"Alt+F4"キーを押すか、システムメニューから[閉じる]オプションを選択してください。(システムメニューは High-performance Embedded Workshop 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。)

## 1.11 ツールシステム概要

ツールを追加することによって、High-performance Embedded Workshop の機能を拡張できます。これを行うには、[ツールアドミニストレーション]ダイアログボックスでツールを登録しておくことが必要です。これらのツールを用いて、ウィンドウ、メニュー、およびツールバーを High-performance Embedded Workshop システムに追加できます。ツールの例としては、High-performance Embedded Workshop のデバッガおよびビルダツールがあります。デバッガツールはデバッガに関連するすべてのメニューとツールバーを追加し、ビルダツールも同じようにビルド機能に関するすべてのメニューとツールバーを追加します。システムに登録したツールによって、High-performance Embedded Workshop の使い方が変わります。そのため、このユーザーズマニュアルに記載されたメニューのうち使用できないものもあります。例えば、ビルダツールがインストールされていない場合、[ビルド]メニューにはオプション設定のメニューオプションがありません。

## 1.12 High-performance Embedded Workshop管理情報ファイル

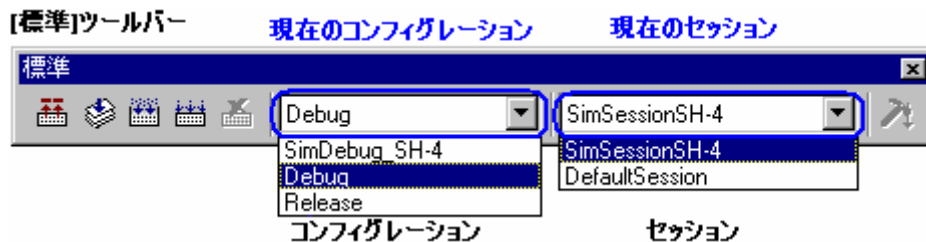
High-performance Embedded Workshop は、動作するために以下のファイルに管理情報を保持します。

- アプリケーションレベルの設定情報ファイル
- ウィンドウのデフォルトポジションファイル
- High-performance Embedded Workshop データベースファイル
- コンポーネントデータファイル\*
- キーボードショートカットレイアウトファイル
- マクロ情報ファイルおよび新規登録したマクロファイル
- カスタムプロジェクトタイプウィザード
- シンタックスハイライトファイル
- テンプレートファイル
- 同期デバッグオプション設定ファイル\*

\*. 機能のサポートはデバッガに依存します。

これらのファイルは、コンピュータ上のユーザアカウント別に保存します。High-performance Embedded Workshop をマルチインストールした場合は、ユーザアカウント別に High-performance Embedded Workshop 毎に保存します。

## 1.13 コンフィグレーションおよびセッションの概要



[標準]ツールバーに現在のコンフィグレーションおよび現在のセッションをドロップダウンリストボックスで表示します。

この例は、プロジェクト作成時に Renesas SuperH Standard 用ツールチェーンを選択し、ターゲットとするデバッグに SH-4 Simulator を選択した場合のコンフィグレーションとセッションを示しています。

ドロップダウンリストボックスを選択し現在のコンフィグレーションおよび現在のセッションから別のコンフィグレーションおよび別のセッションへ切り替えることができます。

必要に応じてコンフィグレーションおよびセッションを追加したり削除したりできます。

- コンフィグレーション

#### コンフィグレーションとは

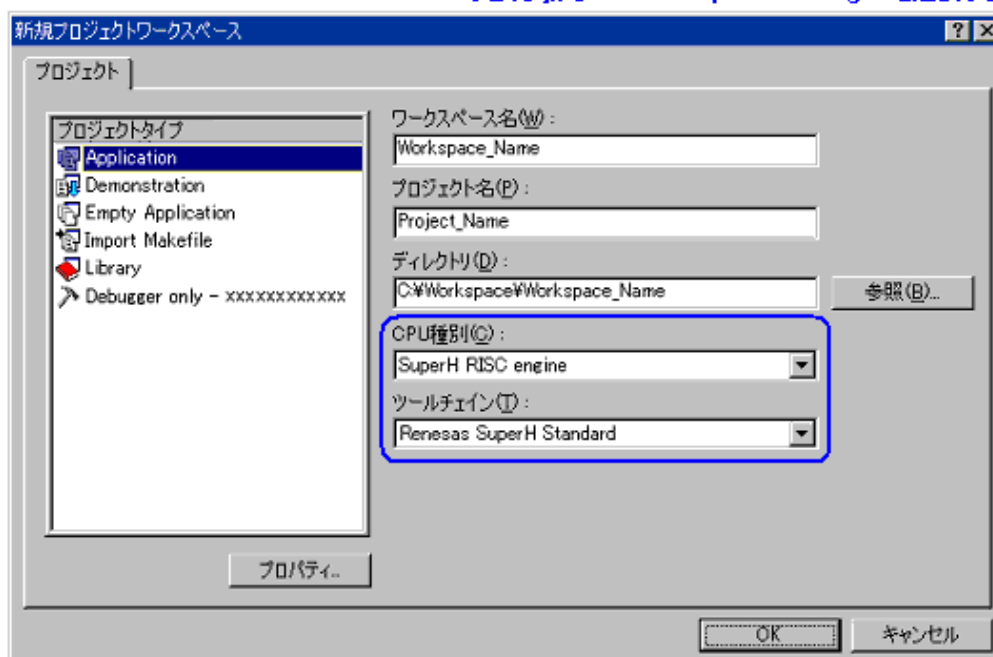
デバッグ情報の出力の有無、および最適化の有無といったビルドのオプションの設定内容に名前を付けて保持しているものです。"ビルドコンフィグレーション"ともいいます。

[標準]ツールバーの図の例では、SimDebug\_SH-4 コンフィグレーション、Debug コンフィグレーション、および Release コンフィグレーションを表示します。

#### High-performance Embedded Workshop が作成するコンフィグレーション

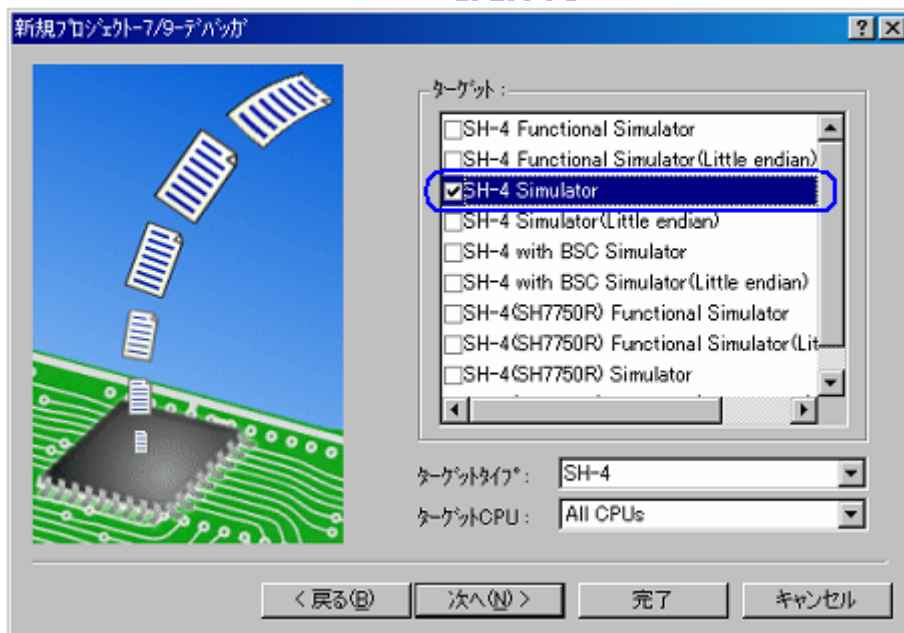
ツールチェーンを選択しプロジェクトを作成した場合、Debug コンフィグレーションおよび Release コンフィグレーションを自動的に作成します。

[CPU種別]からSuperH RISC engineを選択し、[ツールチェーン]からRenesas SuperH RISC engineを選択する



プロジェクトの作成時にターゲットとするデバッガをチェックした場合は、チェックしたデバッガ用のコンフィグレーションが作成されます。プロジェクトの作成時に[ターゲット]で"SH-4 Simulator"をチェックした場合は SimDebug\_SH-4 コンフィグレーションが作成されます。コンフィグレーションの名前はプロジェクトの作成時に変更できます。

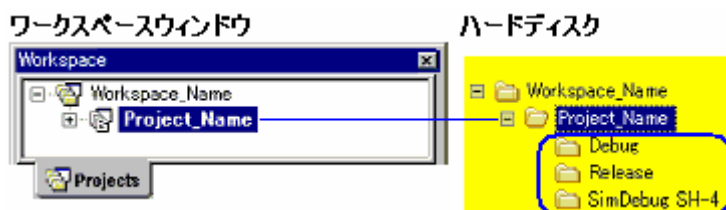
**[ターゲット]でHEWと接続するSH-4 Simulator  
をチェックする**



また、プロジェクトディレクトリ\* 下に各コンフィグレーション用のディレクトリがコンフィグレーションと同じ名前で作成されます。

**注：**

\*. プロジェクトディレクトリは、新規ワークスペースプロジェクト作成時のワークスペースディレクトリ下にプロジェクト名と同じ名前で作成されます。



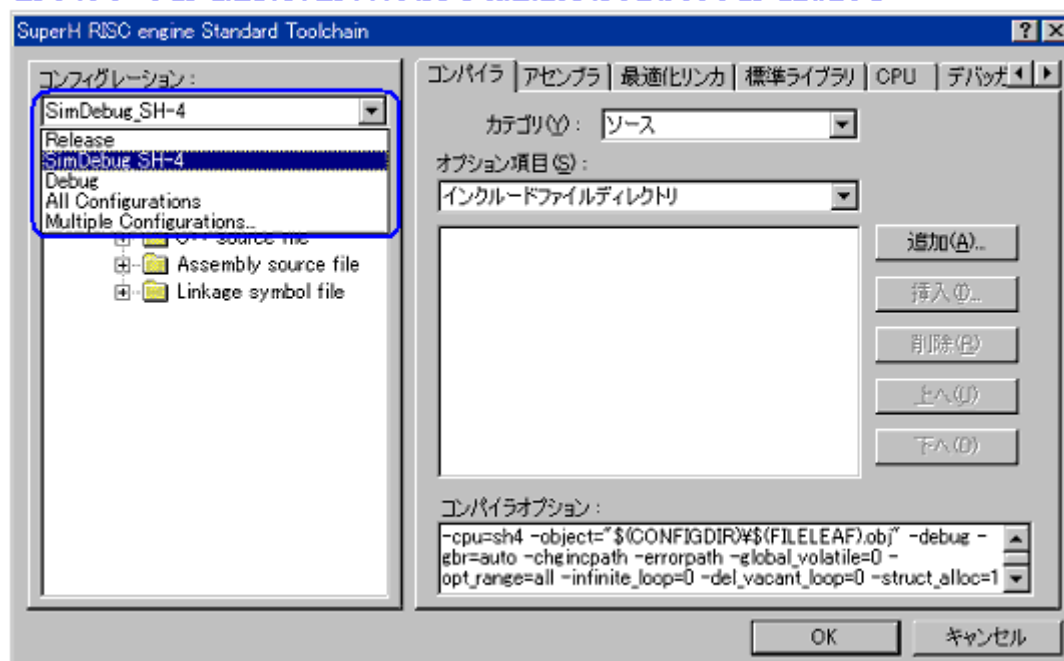
**ビルドオプションの設定方法**

コンフィグレーションに保持されるビルドのオプションを設定するには、[ビルド]メニューのビルドのオプションを設定するダイアログボックスでドロップダウンリストボックスからコンフィグレーションを選択し、コンパイラ、アセンブラ、および最適化リンカなどの各オプションを設定します。下の図は、Renesas SuperH Standard 用ツールチェーンのビルドのオプションを設定するダイアログボックスです。



## ビルドオプションを設定するダイアログボックス

コンフィグレーションを選択し、コンパイラおよび最適化リンカなどのオプションを設定する



コンフィグレーションの詳細は、「2.7 ビルドのコンフィグレーション」を参照してください。

- セッション

### セッションとは

おもにデバッガに接続してデバッグするときに使用する **High-performance Embedded Workshop のデバッグオプションの設定\***、デバッガ固有の様々なオプションの設定、およびメモリウィンドウ、またはレジスタウィンドウなどのウィンドウとその位置の情報などに名前を付け保持しているものです。"デバッグセッション"ともいいます。

[標準]ツールバーの図の例では、SimSessionSH-4 セッションおよび DefaultSession セッションを表示しています。各セッションの情報は、High-performance Embedded Workshop プロジェクトでそれぞれ別のファイルに保存されます。

### 注：

\*. High-performance Embedded Workshop と接続するデバッガ、コア、デバッグフォーマット、ダウンロードモジュールの設定、デバッガへ High-performance Embedded Workshop が接続するタイミング、およびコマンドバッチを High-performance Embedded Workshop が実行するタイミングなどのオプション設定があります。

### High-performance Embedded Workshop が作成するセッション

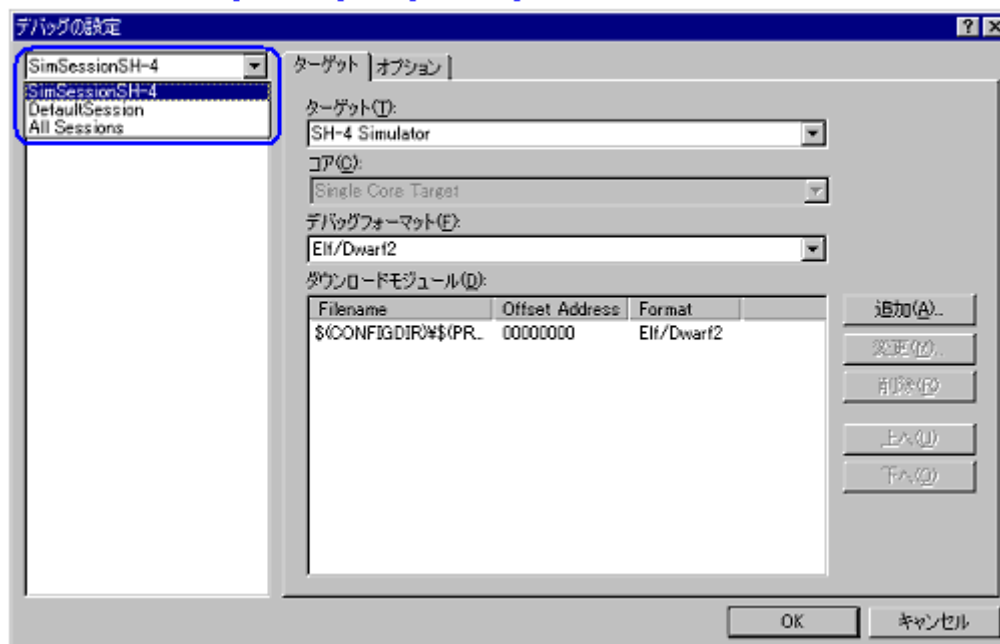
ツールチェーンを選択しプロジェクトを作成した場合、DefaultSession セッションを自動的に作成します。プロジェクトの作成時にターゲットとするデバッガをチェックした場合は、チェックしたデバッガを High-performance Embedded Workshop と接続するセッションを自動的に作成します。

例えば、プロジェクトの作成時に[ターゲット]で"SH-4 Simulator"をチェックした場合は SimSessionSH-4 セッションが作成されます。

## High-performance Embedded Workshop のデバッグオプションの設定方法

セッションに保持される High-performance Embedded Workshop のデバッグオプションを設定するには、[デバッグ]メニューの[デバッグの設定]ダイアログボックスでドロップダウンリストボックスからセッションを選択し、[ターゲット]タブおよび[オプション]タブで各オプションを設定します。

デバッグオプションを設定するダイアログボックス  
セッションを選択し、[ターゲット]および[オプション]を設定する



デバッグセッションの詳細は、「17.1.6 デバッグセッション」を参照してください。

### 1.14 マクロ生成支援機能とテスト支援機能の概要

プログラムのコーディングが完了したらテストを実施します。テストで問題が見つかり問題の箇所を修正して再テストを行います。プログラム開発において、テストに掛かる工数は非常に大きな割合を占めています。

High-performance Embedded Workshop では、このテスト工程の支援のために、マクロ生成支援機能およびテスト支援機能を提供しています。

- マクロ生成支援機能

マクロ生成支援機能は、High-performance Embedded Workshop システムのアプリケーション関連 (\*1)、ビルド関連 (\*2)、およびデバッグ関連 (\*3) などの一部の操作を High-performance Embedded Workshop コマンドラインのコマンドとして記録、または記録したコマンドを実行する機能です。

記録されるファイル (High-performance Embedded Workshop マクロファイル) は、拡張子が hdc のコマンドバッチファイルであり編集可能なテキストファイルです。このファイルは、各ユーザプロファイルのアプリケーションデータフォルダにある High-performance Embedded Workshop 管理情報フォルダ下の "Macros" フォルダに作成されます。デフォルトのファイル名は Default.hdc です。

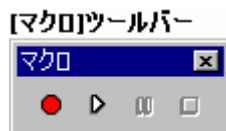
High-performance Embedded Workshop コマンドラインのコマンドに対応する High-performance Embedded Workshop システムのすべての操作が記録できるわけではありません。各機能のメニューオプション一覧には "マクロの記録" 欄にマクロの記録マーク (●) を表示し High-performance Embedded Workshop マクロファイルへ記録可能かどうかを示しています。



## 注：

- \*1. プロジェクトの変更、セッションの変更、コンフィグレーションの変更など
- \*2. コンパイル、ビルドなど
- \*3. モジュールのダウンロード、メモリ値の変更、レジスタ値の変更、ソフトウェアブレークの設定 / 解除、プログラム実行など

マクロ生成支援機能は、[ツール]メニューと[マクロ]ツールバーで利用できます。



[マクロ]ツールバーが表示されていない場合は、[基本設定 -> カスタマイズ]を選択し[カスタマイズ]ダイアログボックスを開きます。[カスタマイズ]ダイアログボックスの[ツールバー]タブで、[マクロ]チェックボックスをオンにしてください。

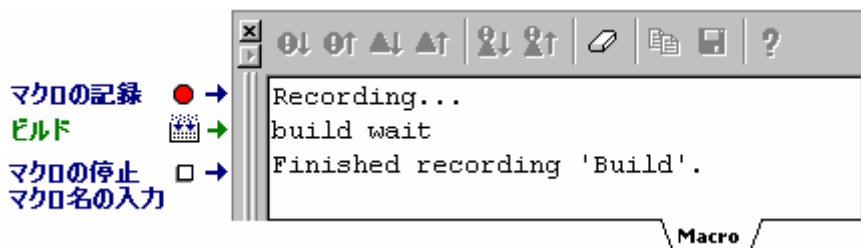
[ツール]メニュー	[マクロ]ツールバー	機能
マクロの設定	—	[マクロの設定]ダイアログボックスを開きます。下記の機能を備えており、選択した High-performance Embedded Workshop のマクロファイル毎に記録されたマクロ名がリストに表示されています。 <ul style="list-style-type: none"> <li>・マクロファイルの新規作成</li> <li>・マクロファイルのインポート</li> <li>・マクロの記録</li> <li>・マクロの実行</li> <li>・マクロの編集</li> <li>・マクロの割り当て</li> <li>・マクロの削除</li> </ul>
マクロの記録	●	マクロの記録を開始します。
マクロの実行	▶	[マクロの選択]ダイアログボックスが開きます。実行するマクロを選択します。
—	⏸	マクロの記録または実行を一時停止します。
マクロの停止	□	マクロの記録または実行を停止します。

[マクロの記録](●)を選択してから[マクロの停止](□)を選択するまでが1つの"マクロ"で、1つの High-performance Embedded Workshop マクロファイルには複数のマクロ ( High-performance Embedded Workshop コマンドラインのコマンドの集まり ) が記録できます。マクロには、複数の High-performance Embedded Workshop コマンドラインのコマンドが含まれます。

下記は High-performance Embedded Workshop マクロファイルとマクロの例です。

HEWマクロファイル	
マクロの記録 ● →	#Macro Operations -Start
CPUリセット [CPU] →	reset
ステップイン [Step] →	step 0x1
マクロの停止 □ →	#Macro Operations -End
マクロ名の入力	
マクロの記録 ● →	#Macro Build -Start
ビルド [Build] →	build wait
マクロの停止 □ →	#Macro Build -End
マクロ名の入力	

アウトプットウィンドウの[Macro]タブには現在のマクロの記録状況を出力します。上の図の「Build」マクロは下記のように出力します。



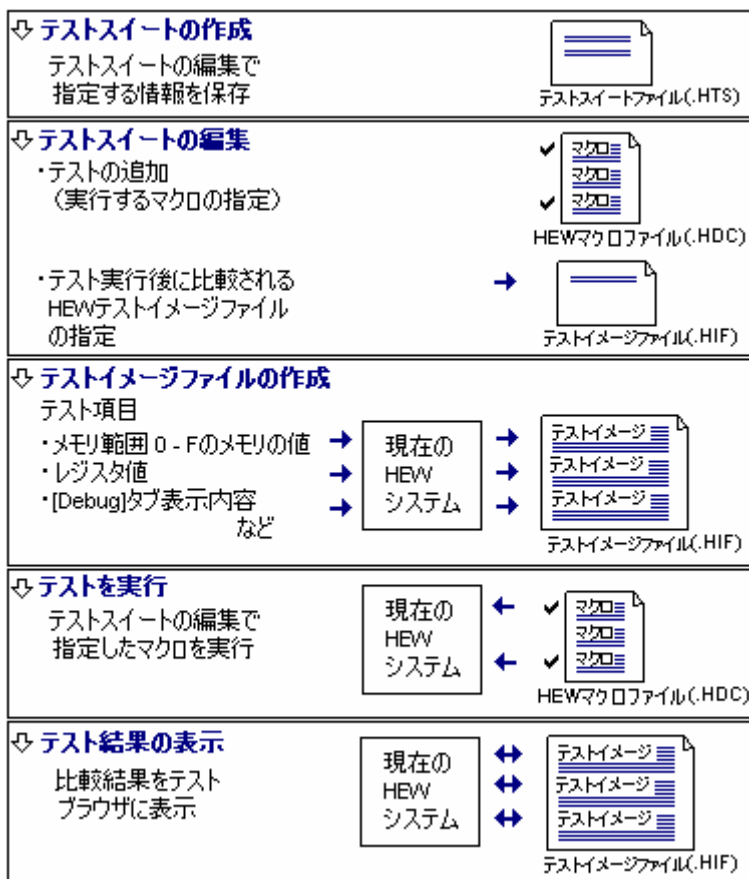
操作手順については「1.14.1 テストシナリオの例」も参照してください。

• テスト支援機能

テスト支援機能は、比較したいテスト項目（テストイメージデータ）でテストイメージファイルを作成し、マクロ生成支援機能で生成したマクロまたは従来の High-performance Embedded Workshop のコマンドバッチを実行して、テストイメージファイルと High-performance Embedded Workshop システムを比較できる機能です。テストイメージデータはテストイメージファイル（.HIF）に保存します。

High-performance Embedded Workshop システムのすべての機能のテストイメージデータを保存できるわけではありません。テストイメージデータを保存できる機能については、「16.6 テストイメージファイルにテストイメージデータとして保存できる機能」に記載していますので参照してください。

テスト支援機能の主な操作手順を下記に示します。操作手順については「1.14.1 テストシナリオの例」も参照してください。



テスト支援機能は、[テスト]メニューとワークスペースウィンドウの[Test]タブのポップアップメニューで利用できます。

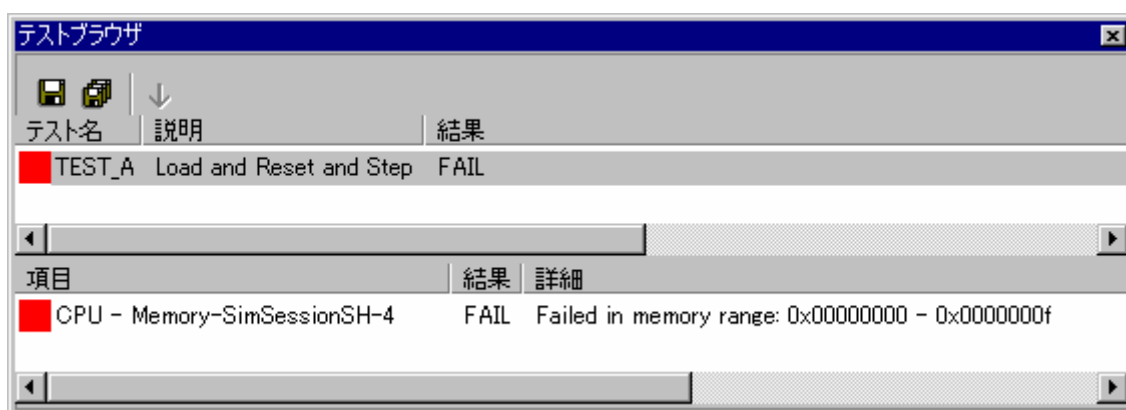
テスト支援機能の最初のステップは、テストスイート（テストの集まり）の作成です。[テスト->テストスイートの作成]で作成できます。テストスイートの作成後は、ワークスペースウィンドウの[Test]タブでも操作可能です。

テストスイートを編集し、テスト中に実行するマクロおよびテストベースのテストイメージファイルを指定します。

次にテスト内容として、テストイメージファイルに取得するためのいくつかのテスト項目を設定します。テストベースとしたい High-performance Embedded Workshop の状態で、設定したテスト項目に基づきテストイメージデータを取得しテストイメージファイルに保存します。

テスト（マクロの集まり）を実行し、テストが完了すると指定したテスト項目が比較されます。結果がテストブラウザウィンドウに表示されます。

比較した結果、不一致があるとテスト項目の結果カラムに"FAIL"を表示します。この場合、テスト項目をダブルクリックすると詳細情報が確認可能です。



### 1.14.1 テストシナリオの例

マクロ生成支援機能とテスト支援機能を使用した主なビルド、デバッグ操作の流れをテストシナリオの一例として紹介します。

このテストシナリオでは、SuperH ファミリー用シミュレータデバッガなどで"Demonstration"タイプを指定したプロジェクトやエミュレータデバッガなどで Tutorial プロジェクトに含まれているような C 言語で記述された 10 個のランダムデータを昇順/降順にソートするプログラムを使用します。配列に格納されるデータに差異が発生したケースを確認します。

ここでは、プログラムコード内の配列"a"をローカル変数からグローバル変数に編集してから使用します。

#### • テストの前に

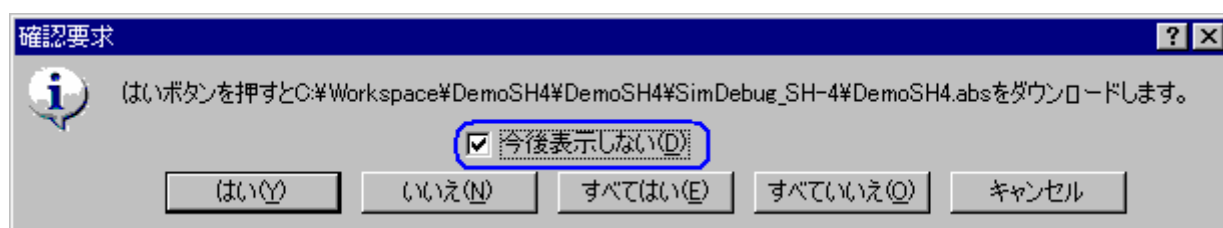
はじめにワークスペースを開きデバッガに接続します。

1. 新規にプロジェクトワークスペースを作成するか既にある場合はこれを開きます。プロジェクトワークスペースの作成については「17.1.2 デバッガを選択する」を参照してください。
2. [標準]ツールバーのセッションリストからデバッガに接続するセッションを選択します。セッションの選択については「17.1.6.1 セッションを選択する」を参照してください。

操作により確認要求ダイアログボックスが表示される場合があります。例えば、プログラムをすべてビルドした場合、デフォルトでダウンロードするかどうかの確認要求ダイアログボックスが開きます。プログラムをダウンロードする場合は「はい」ボタンをクリックします。

確認要求ダイアログボックスのボタンのクリックはマクロに記録されないため、次回マクロを実行すると、確認要求ダイアログボックスは表示されてしまいます。マクロ実行処理を継続するためにはいずれかのボタンをクリックしなくてはなりません。テスト支援機能を使用する場合、一連の操作を繰り返しテストするので処理を継続する必要があります。そのため、確認要求ダイアログボックスが表示されないよう次のように設定してください。

1. 確認要求ダイアログボックスの[今後表示しない]チェックボックスをオンにしてください。



テスト開始時の High-performance Embedded Workshop の環境をテスト開始前に毎回同じ環境にするため、テストに使用したい High-performance Embedded Workshop 環境を保存してください。

1. [ファイル->ワークスペースの保存]を選択してください。
2. [ファイル->セッションの保存]を選択してください。

テスト開始時に保存した High-performance Embedded Workshop の環境を読み込むことで、毎回同じ環境でテストを開始できます。

#### ● 本テストシナリオで使用するテストスクリプトの準備

このテストシナリオでは、マクロ生成支援機能を使用し、次の一連の操作を High-performance Embedded Workshop コマンドラインのコマンドとしてマクロに記録します。

このマクロをテストスクリプトとしてテスト実行時に使用します。

操作の分類	操作
プログラム実行前の準備 ・各設定値の初期化 ・プログラムの準備	セッションの再読み込み (ステップ 1(2)参照)
	レジスタ値の初期化 * (ステップ 1(3)参照)
	デモンストレーション用プログラムのビルド * (ステップ 1(4)参照)
プログラムダウンロード	デモンストレーション用プログラムのダウンロード (ステップ 1(5)参照)
変数の登録、プログラム実行、および実行結果の確認	メモリフィル機能で配列 a のメモリの値を 0 に設定 (ステップ 1(6)参照)
	CPU のリセット (ステップ 1(7)参照)
	main 関数内の任意のカーソル行まで GO 実行 (ステップ 1(9)参照)
	ランダムデータを格納している配列をウォッチウィンドウへ登録 * (ステップ 1(10)参照)

注：

\*. 機能のサポートはデバッガに依存します。

## • テストシナリオの例

テスト支援機能を使用し、配列のデータ（配列のアドレスのメモリ内容）をテスト対象とします。

テストベースとなる High-performance Embedded Workshop の状態のテストイメージデータをテストイメージファイルに保存した後、配列にデータを格納する回数を減らすようプログラムを変更した後にテストを実行し、配列に格納されているデータに差異が生じることをテストにより確認します。続いて格納する回数を元に戻し同じテストを実行します。最初のテストベースのテストイメージファイルと比較し配列に格納されているデータ値に差異が生じていないことを確認します。

ステップ 1：マクロの記録

ステップ 2：マクロを編集する（記録内容を見る）

ステップ 3：マクロを実行する

ステップ 4：テストスイートを作成する

ステップ 5：テストスイートを編集する

ステップ 6：テストイメージファイルを作成する

ステップ 7：プログラムを変更しテストを実行する

ステップ 8：テスト結果を確認する（不一致の場合）

ステップ 9：プログラムを元に戻しテストを実行する

ステップ 10：テスト結果を確認する（一致した場合）

### 1.14.2 ステップ 1：マクロの記録

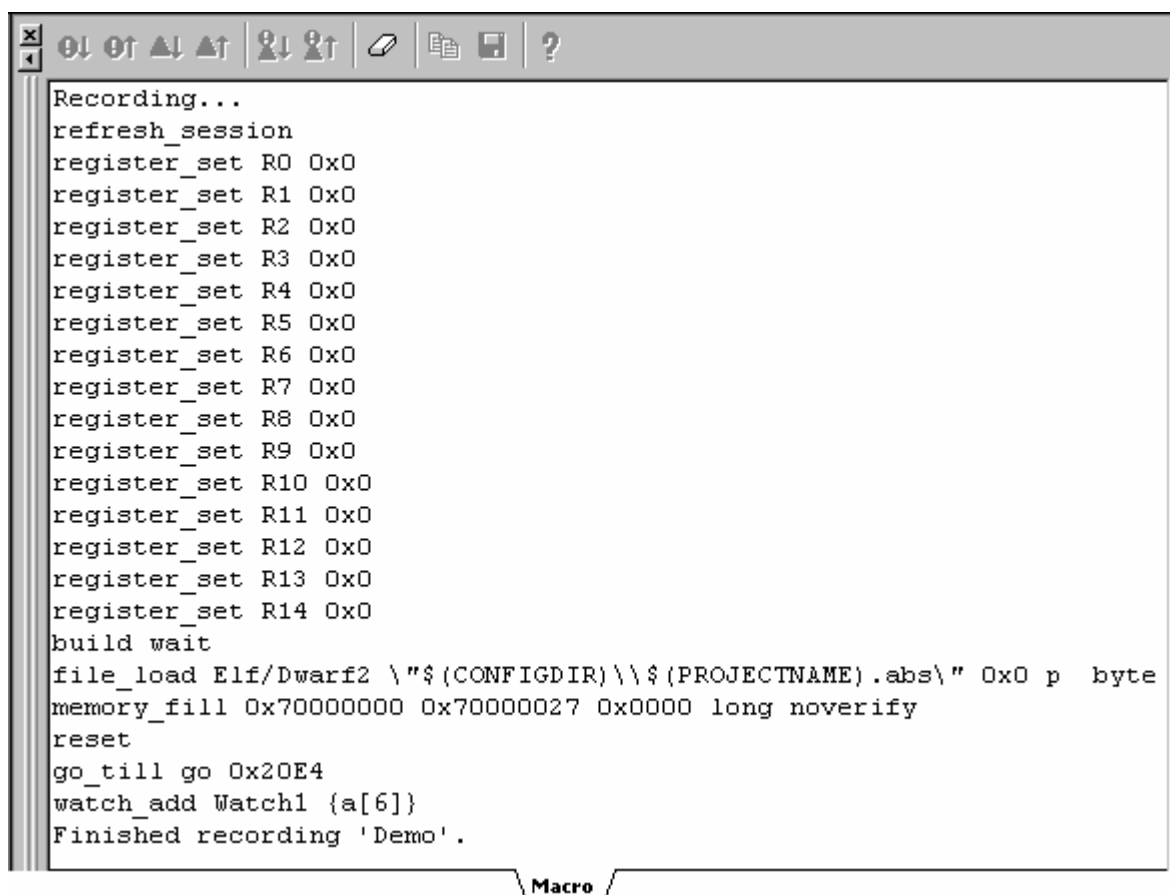
次の一連の操作でマクロ生成支援機能を使用し High-performance Embedded Workshop コマンドラインのコマンドを記録します。

記録した High-performance Embedded Workshop コマンドラインのコマンドは、アウトプットウィンドウの [Macro] タブに順次出力されます。ただし、操作によってはアウトプットウィンドウのタブが切り替わります。

1. [マクロ] ツールバーの [マクロの記録] ボタン (●) をクリックします。このときマウスポインタに [マクロの記録] アイコンが付きます (●)。アウトプットウィンドウの [Macro] タブに「Recording...」メッセージが出力されます。
2. [ファイル -> セッションのリフレッシュ] を選択します。  
☰ （refresh\_session コマンドが記録されます）
3. [CPU] ツールバーの [レジスタ] ボタン (R1) をクリックしレジスタウィンドウを表示します。値を変更するレジスタ上をダブルクリックし [レジスタ値設定] ダイアログボックスを開きます。[値] に 0 を入力します。  
☰ （register\_set コマンドが記録されます）
4. [標準] ツールバーの [ビルド] ボタン (●) をクリックしデモンストレーション用プログラムをビルドします。  
☰ （build コマンドが記録されます）
5. ワークスペースウィンドウの [Projects] タブの [Download modules] フォルダ下のデモンストレーション用プログラムをダブルクリックしダウンロードします。  
☰ （file\_load コマンドが記録されます）
6. [CPU] ツールバーの [メモリ] ボタン (●) をクリックし [表示開始アドレス] ダイアログボックスを開きます。[表示開始アドレス] に配列 a のアドレスを入力しメモリウィンドウを開きます。メモリウィンドウ上で充填する配列 a のアドレス領域をドラッグして選択してください。ウィンドウ内で右クリックしポップアップメニューから [フィル] を選択します。[メモリフィル] ダイアログボックスのフィルデータに 0 を入力します。  
☰ （memory\_fill コマンドが記録されます）

7. [デバッガ]ツールバーの[CPUリセット]ボタン(🔄)をクリックしCPUをリセットします。  
📄 (reset コマンドが記録されます)
8. main 関数を含むソースファイルをエディタウィンドウに表示します。
9. main 関数の最終行上で右クリックしポップアップメニューから[カーソル位置まで実行]を選択します。  
📄 (go\_till コマンドが記録されます)
10. ランダムデータを格納している配列要素"a[6]"をウォッチウィンドウへ追加します。  
📄 (watch\_add コマンドが記録されます)
11. 最後に[マクロ]ツールバーの[マクロの停止]ボタン(⏹)をクリックします。マウスポインタは通常の矢印に戻ります。
12. マクロ名を入力する[マクロの新規追加]ダイアログボックスが開きますので、マクロ名(例:"Demo")を入力してください。アウトプットウィンドウの[Macro]タブに「Finished recording 'Demo'.」が出力されます。

アウトプットウィンドウの[Macro]タブを選択してください。記録した High-performance Embedded Workshop コマンドラインのコマンドが確認できます。



```

Recording...
refresh_session
register_set R0 0x0
register_set R1 0x0
register_set R2 0x0
register_set R3 0x0
register_set R4 0x0
register_set R5 0x0
register_set R6 0x0
register_set R7 0x0
register_set R8 0x0
register_set R9 0x0
register_set R10 0x0
register_set R11 0x0
register_set R12 0x0
register_set R13 0x0
register_set R14 0x0
build wait
file_load Elf/Dwarf2 \"$(CONFIGDIR)\\$(PROJECTNAME).abs\" 0x0 p byte
memory_fill 0x70000000 0x70000027 0x0000 long noverify
reset
go_till go 0x20E4
watch_add Watch1 {a[6]}
Finished recording 'Demo'.
  
```

### 1.14.3 ステップ 2：マクロを編集する（記録内容を見る）

1. [ツール->マクロの設定]を選択し[マクロの設定]ダイアログボックスを開きます。マクロを記録した High-performance Embedded Workshop マクロファイル"Default"が[使用中のマクロファイル]リストに表示されています。

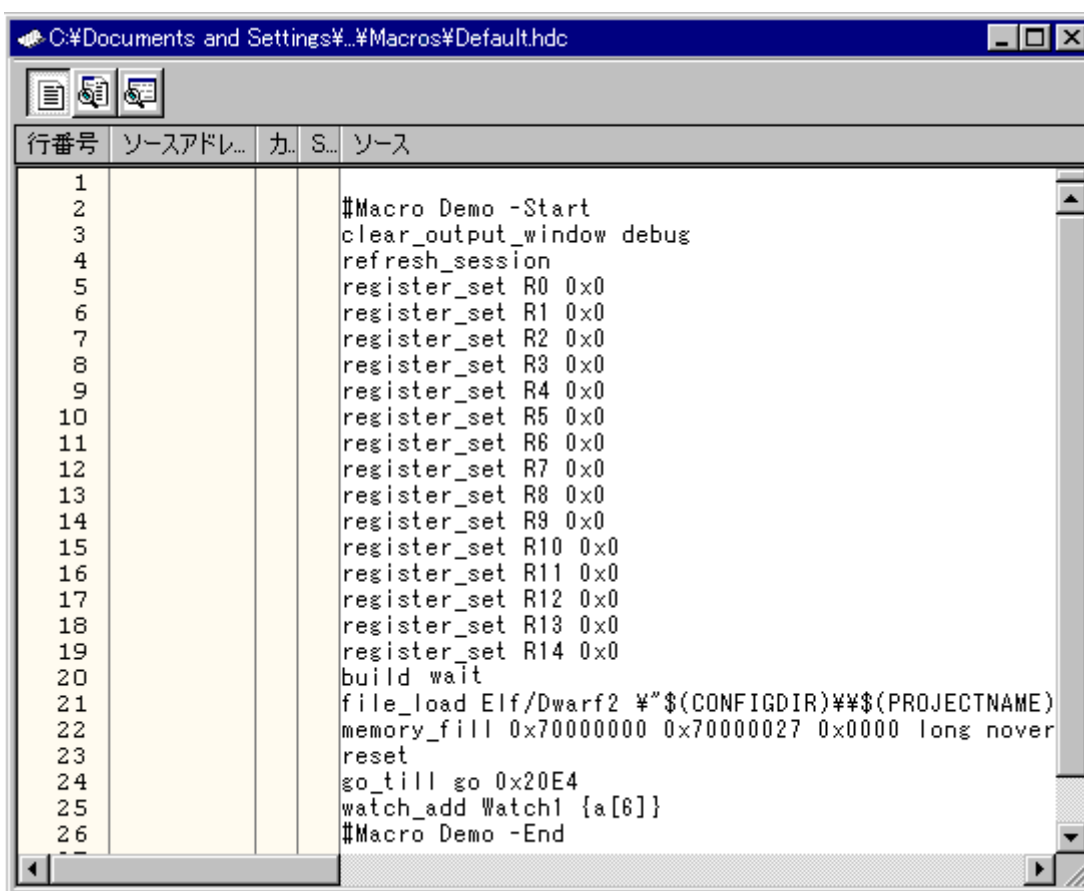


2. [マクロ一覧]リストから"Demo"マクロを選択し、[編集]ボタンをクリックします。エディタウィンドウに"Demo"マクロが記録された High-performance Embedded Workshop のマクロファイル"Default.hdc"が表示されます。

High-performance Embedded Workshop のマクロファイルには、マクロ名を識別する情報行の間に High-performance Embedded Workshop コマンドラインのコマンドが記録されています。

プログラム実行などを含むマクロの場合、アウトプットウィンドウの[Debug]タブには停止要因などの情報が表示されます。そのため、テスト結果を比較する際に差異が生じる場合があります。アウトプットウィンドウの各タブの表示内容をクリアする High-performance Embedded Workshop のコマンドラインの `clear_output_window` コマンドがあります。あらかじめ"#Macro Demo -Start"の次の行にアウトプットウィンドウの[Debug]タブの情報をクリアする以下の行を追加することも可能です。

```
clear_output_window debug
```



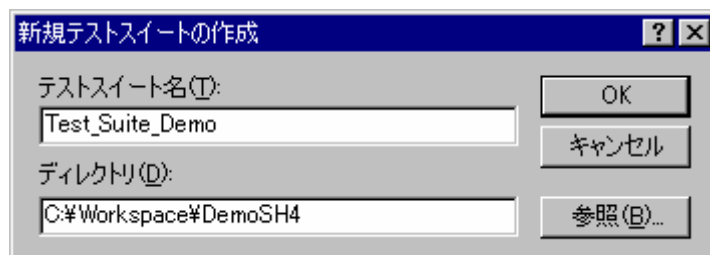
必要に応じて内容を編集後、[エディタ]ツールバーの[ファイルの保存]ボタン(📁)をクリックしてください。High-performance Embedded Workshop のマクロファイル"Default.hdc"を閉じてください。

#### 1.14.4 ステップ 3：マクロを実行する

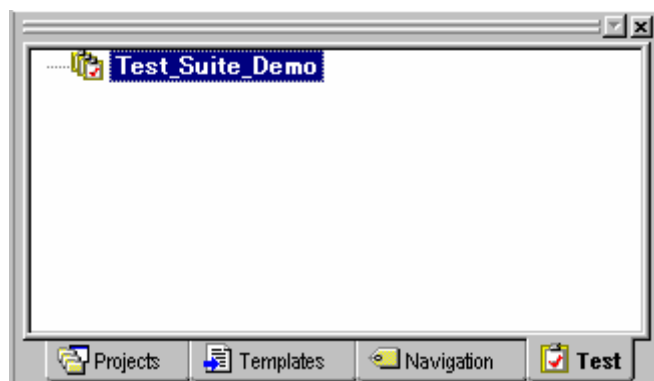
1. [マクロ]ツールバーの[マクロの実行]ボタン(▶)をクリックします。[マクロの選択]ダイアログボックスが開きます。
2. "Demo"マクロを選択し OK ボタンをクリックすると、記録したコマンドの実行を開始します。このときマウスポインタに[マクロの実行]アイコンが付きます(▶)。
3. マクロの実行が終了するとマウスポインタは通常の矢印に戻ります。

### 1.14.5 ステップ 4： テストスイートを作成する

1. [テスト->テストスイートの作成]を選択し、[新規テストスイートの作成]ダイアログボックスを開きます。



2. [テストスイート名]に"Test\_Suite\_Demo"を入力します。
3. [ディレクトリ] はあらかじめワークスペースディレクトリが表示されています。必要に応じて編集してください。
4. [OK]ボタンをクリックします。
5. ワークスペースウィンドウの[Test]タブを表示してください。"Test\_Suite\_Demo"テストスイートアイコンが表示されています。

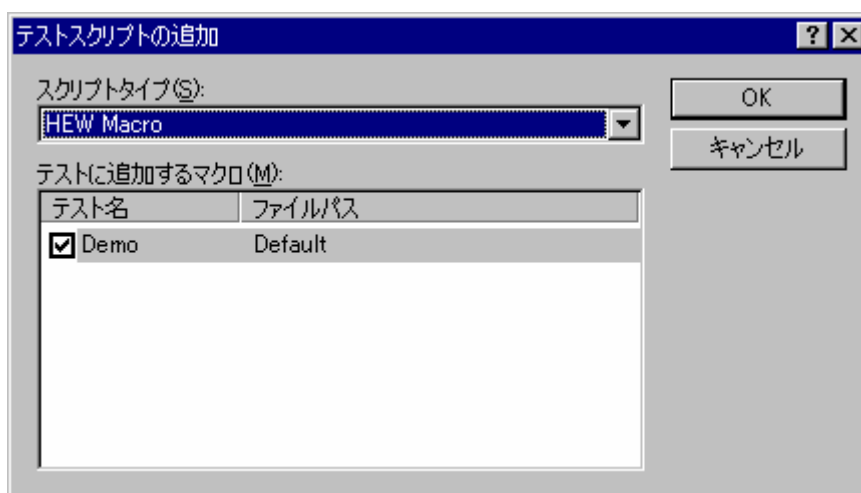


### 1.14.6 ステップ 5： テストスイートを編集する

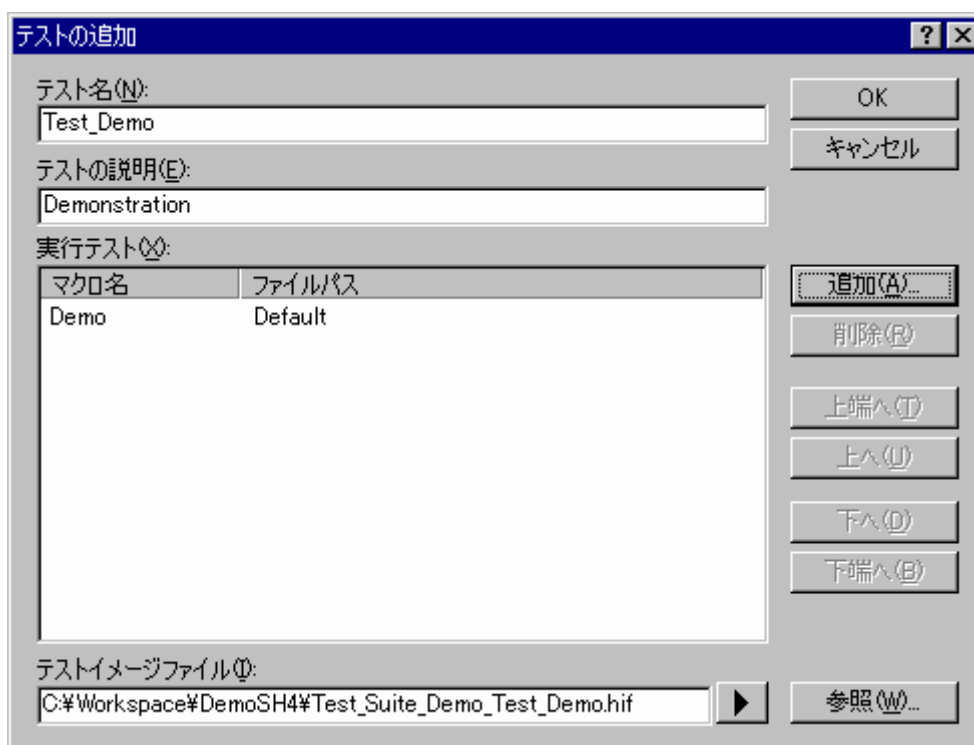
テストを実行するために「ステップ 1」で記録したマクロを選択します。

1. "Test\_Suite\_Demo"テストスイートアイコン上で右クリックしポップアップメニューを表示します。
2. [テストスイートの編集]を選択し、[テストスイートの編集]ダイアログボックスを開きます。
3. [追加]ボタンをクリックし[テストの追加]ダイアログボックスを開きます。
4. [テスト名]に"Test\_Demo"と入力します。
5. [テストの説明]に"Demonstration"と入力します。
6. [追加]ボタンをクリックし[テストスクリプトの追加]ダイアログボックスを開きます。
7. [テストに追加するマクロ]で"Demo"チェックボックスをオンにします。
8. OK ボタンをクリックします。





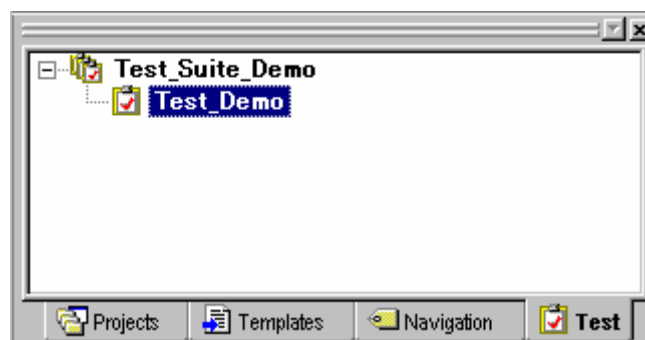
9. [テストイメージファイル]には、あらかじめ「ステップ4」で入力したテストスイート名に'\_'が付加されて表示されています。 '\_'に続いて[テスト名]に入力したテスト名"Test\_Demo"が表示されます ("テストスイート名"+"\_"+"テスト名")。必要に応じて編集してください。



10. OK ボタンをクリックします。[テストケース]リストに"Test\_Demo"テストが追加されます。



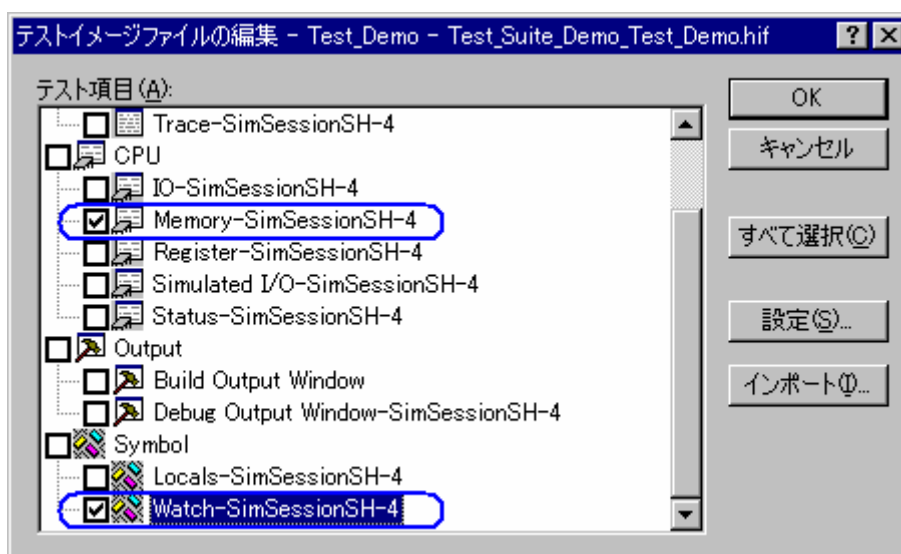
11. OK ボタンをクリックします。"Test\_Suite\_Demo"テストスイートアイコン下に"Test\_Demo"テストアイコンが表示されました。



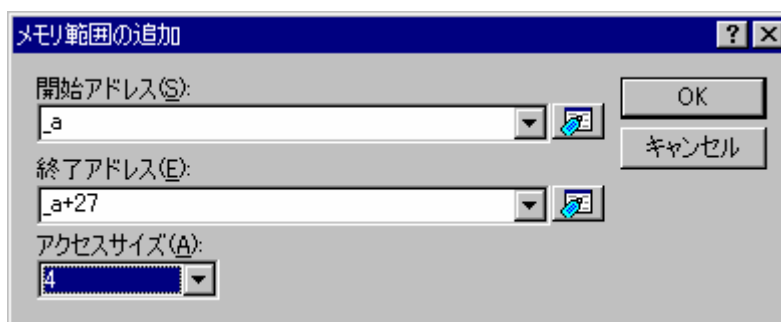
#### 1.14.7 ステップ 6： テストイメージファイルを作成する

テスト項目として配列のアドレスのメモリ内容、およびウォッチウィンドウの内容を比較するため、それらのテストイメージデータの取得範囲を設定します。ステップ 5 で指定したテストイメージファイル "Test\_Suite\_Demo\_Test\_Demo.hif" にテスト項目のテストイメージデータが保存されます。

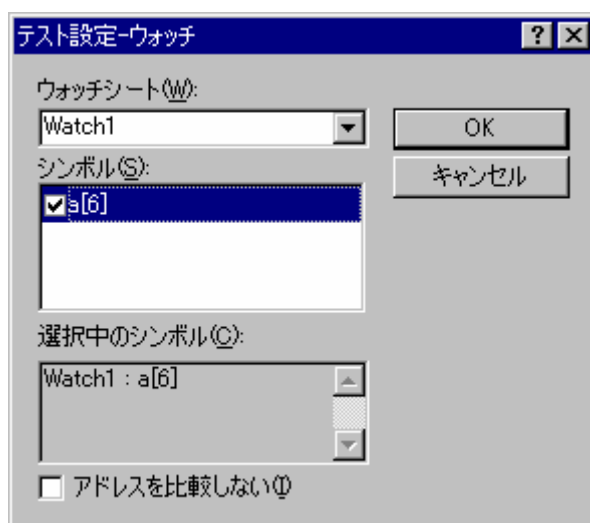
1. "Test\_Demo"テストアイコン上で右クリックしポップアップメニューを表示します。
2. [テストイメージファイルの編集]を選択し[テストイメージファイルの編集]ダイアログボックスを開きます。ダイアログボックスのタイトルには、テスト名"Test\_Demo"およびテストイメージファイル名"Test\_Suite\_Demo\_Test\_Demo.hif"が表示されます。
3. [テスト項目]の[CPU]下の[Memory-xxxxxx]および[Symbol]下の[Watch-xxxxxx]のチェックボックスをオンにします。



4. 配列のアドレスのメモリ内容をテストイメージファイルに保存します。[テストイメージファイルの編集]ダイアログボックスで、[テスト項目]の[CPU]下の[Memory-xxxxxx]をダブルクリックし、[テスト設定-メモリ]ダイアログボックスを開きます。
5. [追加]ボタンをクリックし、[メモリ範囲の追加]ダイアログボックスを開きます。
6. [開始アドレス]に配列の先頭アドレス、[終了アドレス]に配列の終了アドレスを入力し、[アクセスサイズ]を選択します。
7. OK ボタンをクリックします。



8. [テスト設定-メモリ]ダイアログボックスで OK ボタンをクリックします。
9. ウォッチウィンドウの Watch1 シートの配列要素"a[6]"の内容をテストイメージファイルに保存します。[テストイメージファイルの編集]ダイアログボックスで、[テスト項目]の[Symbol]下の[Watch-xxxxxx]をダブルクリックし、[テスト設定-ウォッチ]ダイアログボックスを開きます。
10. [シンボル]の"a[6]"のチェックボックスをオンにします。
11. OK ボタンをクリックします。

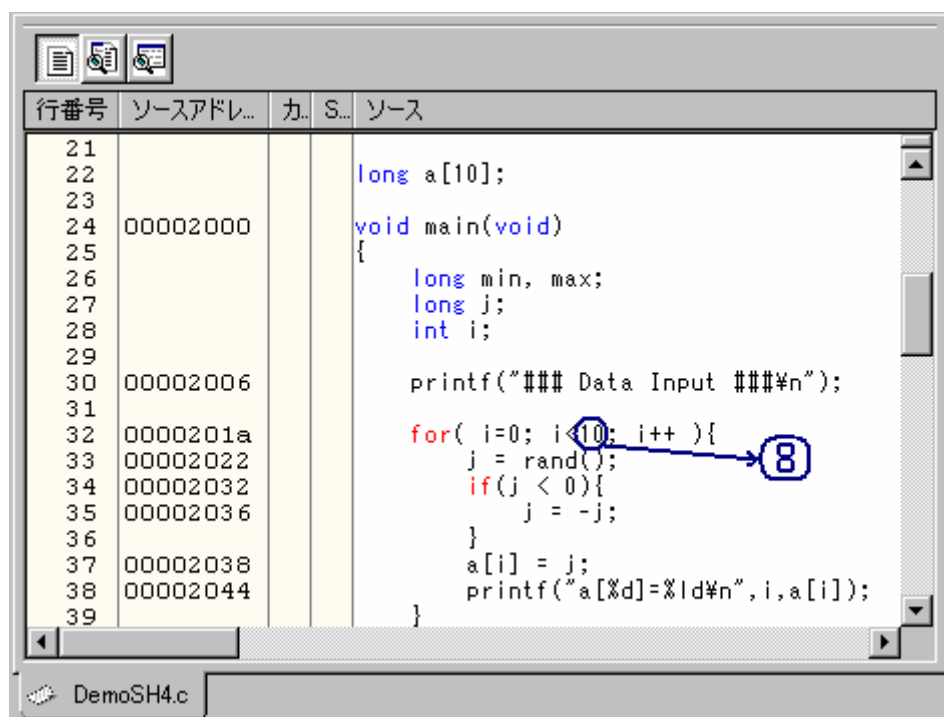


12. [テストイメージファイルの編集]ダイアログボックスで OK ボタンをクリックします。

この時点の High-performance Embedded Workshop システムの状態では、テスト項目の指定内容に基づいてテストイメージを取得し、ステップ 5 で生成したテストイメージファイルに保存されます。

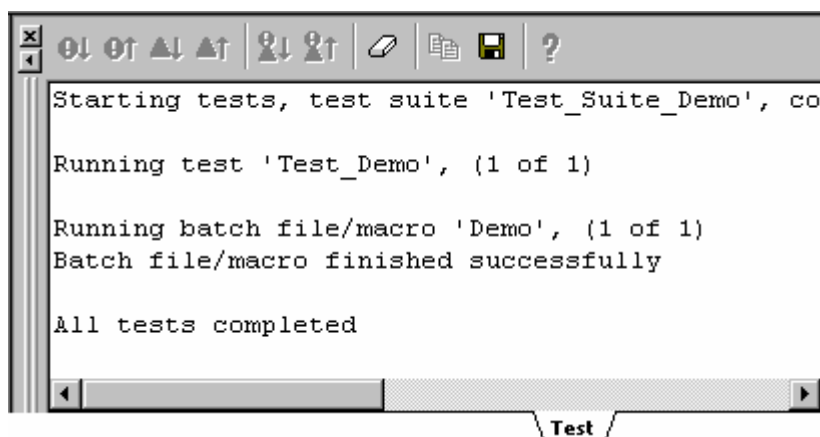
#### 1.14.8 ステップ 7: プログラムを変更しテストを実行する

この例では配列のメモリ領域の値をテスト対象としています。配列にデータを格納する回数を減らすようにプログラムを変更し、保存されたテストイメージと実際のメモリの値が不一致になる状態にします。



1. エディタウィンドウに main 関数を含むソースファイルを表示し配列にデータを格納する回数を減らすようにプログラムを変更します。

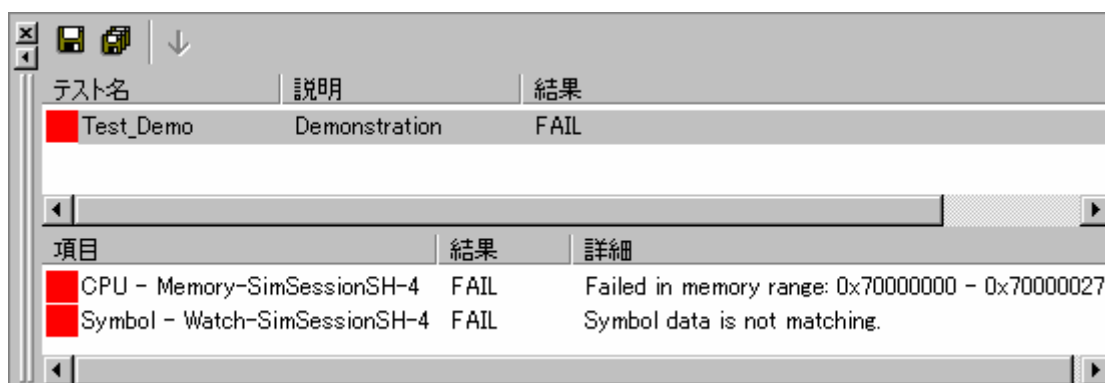
2. [エディタ]ツールバーの[ファイルの保存]ボタン(📁)をクリックします。
3. "Test\_Demo"テストアイコン上で右クリックしポップアップメニューを表示します。
4. [テストケースの実行]を選択しテストを開始します。テストの進捗状況および結果がアウトプットウィンドウの[Test]タブに表示されます。



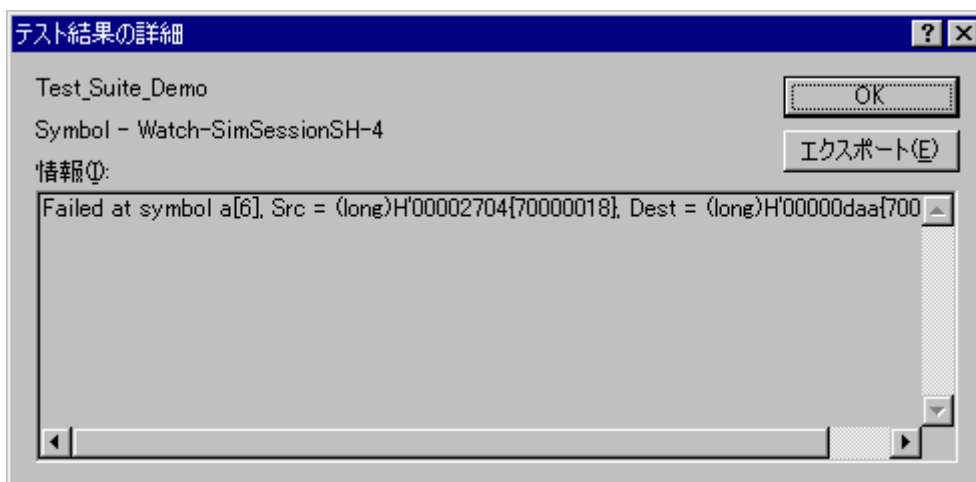
5. テストが完了すると、テストブラウザウィンドウを表示します。

#### 1.14.9 ステップ 8：テスト結果を確認する（不一致の場合）

テスト結果はテストブラウザウィンドウで確認できます。



1. 下側のペインで赤いマークの付いたテスト項目をダブルクリックすると不一致データの詳細を確認できます。
2. 例えば、[Symbol - Watch-xxxxxxx]行をダブルクリックすると、次のようなテスト結果の詳細ダイアログボックスを開きます。配列要素のテストイメージファイルのテストイメージの値 (Src=xxxxxxxx) と今回のテストのテストイメージの値 (Dest=xxxxxxxx) を表示します。



3. [エクスポート]ボタンをクリックすると、テスト項目の結果をテキストファイルへ保存できます。
4. OK ボタンをクリックします。

#### 1.14.10 ステップ 9：プログラムを元に戻しテストを実行する

配列にデータを格納する回数を元に戻すようプログラムを変更し、テストベースとテスト実行結果のメモリが一致する状態にします。

1. 配列にデータを格納する回数を 8 から 10 にプログラムを戻します。
2. [エディタ]ツールバーの[ファイルの保存]ボタン(📁)をクリックします。
3. "Test\_Demo"テストアイコン上で右クリックしポップアップメニューを表示します。
4. [テストケースの実行]を選択しテストを開始します。
5. テストが完了すると、テストブラウザウィンドウを表示します。

#### 1.14.11 ステップ 10：テスト結果を確認する（一致した場合）

テスト結果はテストブラウザウィンドウで確認できます。保存されているテストイメージデータとテスト結果が一致しました。

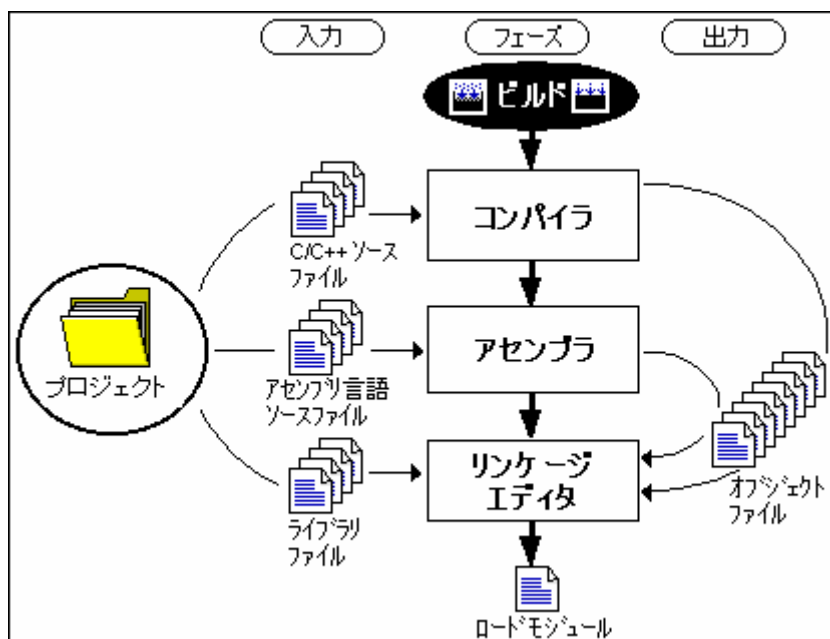


## 2. ビルドの基本

この章では High-performance Embedded Workshop 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. ビルドの応用」を参照してください。

### 2.1 ビルド処理

ビルド処理の一般的な流れを以下の図に示します。High-performance Embedded Workshop 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異なるため、以下の図とも異なる場合があります（例えば、コンパイラが無いなど）。ビルドの各フェーズにおいて、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、次のフェーズに移ります。

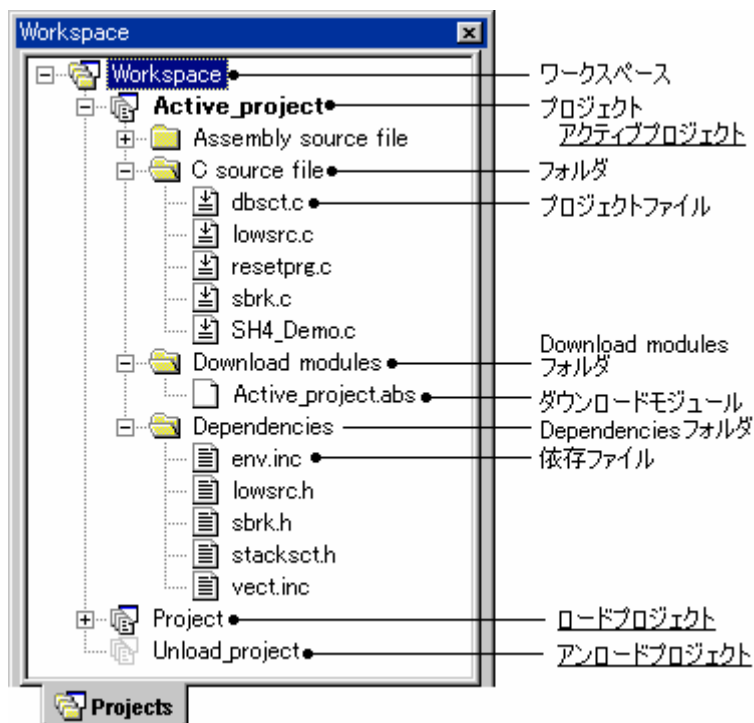


上記の例では、第一のフェーズがコンパイラ、第二のフェーズがアセンブラ、そして最後のフェーズがリンケージエディタです。コンパイラのフェーズでは、プロジェクトのC/C++ソースファイルを順次コンパイルします。アセンブラのフェーズでは、アセンブリ言語のソースファイルを順次アセンブルします。リンケージエディタのフェーズでは、すべてのライブラリファイルと、コンパイラフェーズとアセンブラフェーズからの出力ファイルをリンクして、ロードモジュールを作成します。

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. ビルドの応用」を参照してください。

### 2.2 ワークスペースウィンドウのProjectsタブの構成

ワークスペースウィンドウの[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。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。



ワークスペース上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能
プロジェクトの挿入	-	ワークスペースにプロジェクトを追加します。
全プロジェクトの読み込み	-	すべてのプロジェクトを読み込みます。
ツリーの縮小	-	ワークスペースアイコン下のツリーを縮小します。
クリーン 全プロジェクト	●	すべてのプロジェクトでコンフィグレーションの中間ファイルおよび出力ファイルを削除します。
バージョン管理	-	バージョン管理システムを実行します。
表示の構成	-	ワークスペースの表示の構成を設定します。
プロパティ	-	ワークスペースのプロパティを表示します。

プロジェクト上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能	
ビルド	ビルド	●	プロジェクトをビルドします。
	すべてをビルド	●	プロジェクトを全ビルドします。
	クリーン アクティブプロジェクト	●	現在のコンフィグレーションの中間ファイルおよび出力ファイルを削除します。
	すべての依存関係を更新	-	プロジェクトのファイル依存関係を更新します。
アクティブプロジェクトに設定	●	プロジェクトをアクティブにします。	
プロジェクトの削除	-	ワークスペースからプロジェクトを削除します。	
プロジェクトのアンロード または プロジェクトの読み込み	-	プロジェクトをアンロードします。 または プロジェクトを読み込みます。	
ファイルの追加	-	プロジェクトにファイルを追加します。	
ファイルの削除	-	プロジェクトからファイルを削除します。	
フォルダの追加	-	プロジェクトにフォルダを追加します。	
ツリーの展開/縮小	-	プロジェクト下のツリーを展開または縮小します。	
バージョン管理	-	バージョン管理システムを実行します。	
表示の構成	-	ワークスペースの表示の構成を設定します。	
プロパティ	-	プロジェクトのプロパティを表示します。	



フォルダ（Download module フォルダ、および Dependencies フォルダを除く）上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能
フォルダの追加	-	フォルダを追加します。
フォルダ名の変更	-	フォルダ名を変更します。
フォルダの削除	-	フォルダを削除します。
表示の構成	-	ワークスペースの表示の構成を設定します。

プロジェクトファイル上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能
開く <ファイル名>	-	High-performance Embedded Workshop エディタでファイルを開きます。
開く <ファイル名>（外部エディタ）*	-	外部エディタでファイルを開きます。
コンパイル <ファイル名>	●	1つのファイルをコンパイルします。
ビルドオプション	-	ファイルのビルド方法を設定します。
ファイルの追加	-	プロジェクトにファイルを追加します。
ファイルの削除	-	プロジェクトからファイルを削除します。
ビルドから除外 <ファイル名> または ビルドから除外の解除 <ファイル名>	-	ビルドからプロジェクトファイルを除外します。 または、 除外したプロジェクトファイルをビルドに再び入れます。
バージョン管理	-	バージョン管理システムを実行します。
表示の構成	-	ワークスペースの表示の構成を設定します。
ファイルの比較	-	ファイルを比較します。
プロパティ	-	ファイルのプロパティを表示します。

注：

\*. [基本設定 -> オプション]を選択すると表示される[オプション]ダイアログボックスの[エディタ]タブで、[外部エディタの使用]チェックボックスをオンにすると表示されます。

[Download module]フォルダ上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。（デバッガ接続時に使用可能）

ポップアップメニューオプション	マクロの記録	機能
全てダウンロード	●	[デバッグの設定]ダイアログボックスの[ターゲット]タブの[ダウンロードモジュール]リストに登録されているすべてのモジュール（ワークスペースウィンドウの[Projects]タブの[Download Modules]フォルダ下に表示されているダウンロードモジュールと同じです）をダウンロードします。
ダウンロードモジュールの追加	●	[ダウンロードモジュール]ダイアログボックスを表示し、ダウンロードモジュールを追加します。このモジュールをダウンロードします。
デバッグの設定	-	[デバッグの設定]ダイアログボックスを表示し、デバッグ設定を編集します。
表示の構成	-	ワークスペースの表示の構成を設定します。

ダウンロードモジュール上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。（デバッガ接続時に使用可能）

ポップアップメニューオプション	マクロの記録	機能
ダウンロード	●	モジュールを手動でダウンロードします。
ダウンロード(debug 情報のみ)	●	debug 情報のみモジュールを手動でダウンロードします。
アンロード	●	モジュールをアンロードします。
ダウンロードモジュールの追加	●	[ダウンロードモジュール]ダイアログボックスを表示し、ダウンロードモジュールを追加します。このモジュールをダウンロードします。
削除	-	選択したモジュールを削除します。
デバッグの設定	-	[デバッグの設定]ダイアログボックスを表示し、デバッグ設定を編集します。
表示の構成	-	ワークスペースの表示の構成を設定します。

ポップアップメニューオプション	マクロの記録	機能
モジュールの検索 *	-	ダウンロードモジュールから取得したすべてのファイルを別のディレクトリで再検索するためのダイアログボックスを開きます。
プロパティ	-	選択したダウンロードモジュールの設定内容を[ダウンロードモジュール]ダイアログボックスに表示します。選択したダウンロードモジュールの設定内容を編集します。ダウンロード済みの場合は、アンロードされます。

**注：**

\*. デバッグ専用プロジェクト ([Debugger only - xxxxxx]) を使用している場合のみ表示されます。

ダウンロードモジュールから取得したファイル上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。(デバッグ専用プロジェクト ([Debugger only - xxxxxx]) の場合に使用可能)

ポップアップメニューオプション	マクロの記録	機能
開く <ファイル名>	-	High-performance Embedded Workshop エディタでファイルを開きます。
ファイルの追加	-	プロジェクトにファイルを追加します。
表示の構成	-	ワークスペースの表示の構成を設定します。
ファイルの検索	-	ダウンロードモジュールから取得した同じファイル名を別のディレクトリで選択するためのダイアログボックスを開きます。
プロパティ	-	ファイルのプロパティを表示します。

[Dependencies]フォルダ上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能
表示の構成	-	ワークスペースの表示の構成を設定します。

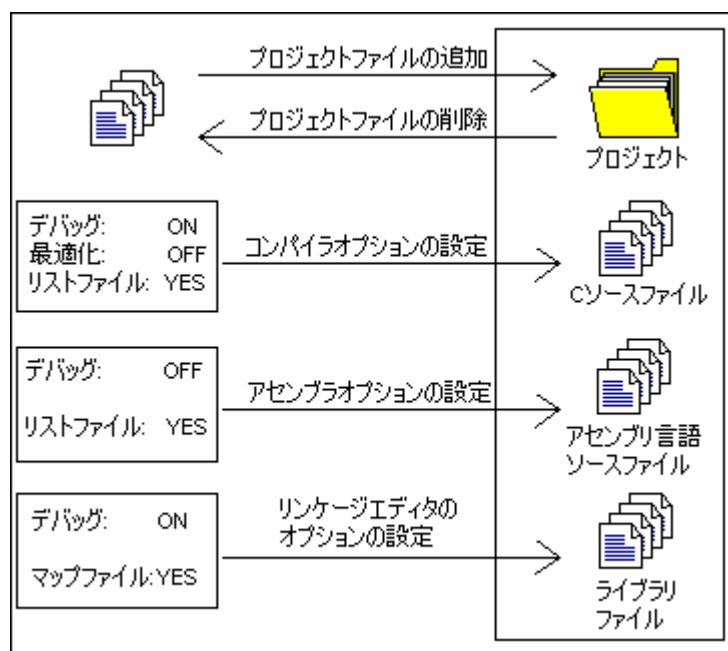
依存ファイル上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能
バージョン管理	-	バージョン管理システムを実行します。
表示の構成	-	ワークスペースの表示の構成を設定します。
ファイルの比較	-	ファイルを比較します。
プロパティ	-	ファイルのプロパティを表示します。

アクティブプロジェクト、ロードプロジェクト、アンロードプロジェクトについては「2.12 アクティブプロジェクトを設定する」を参照してください。

## 2.3 プロジェクトファイル

High-performance Embedded Workshop を使ってアプリケーションをビルドするには、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すべきかを指定しなければなりません(以下の図を参照)。



### 2.3.1 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ます。

#### プロジェクトにファイルを追加するには

1. アクティブプロジェクトに対してのみファイルが追加できます。以下のいずれかの操作を選択し[<プロジェクト名>'プロジェクトにファイルを追加']ダイアログボックスを開きます。
  - [プロジェクト->ファイルの追加]を選択する。
  - ワークスペースウィンドウの[Projects]タブのプロジェクト上で右クリックし、ポップアップメニューから[ファイルの追加]を選択する。
  - ワークスペースウィンドウの[Projects]タブが選択されているときに"Insert"キーを押す。
2. 追加するファイルを選択してください。
3. ダイアログボックスの一番下にある[相対パス]チェックボックスをオンにすると、ファイルを「相対プロジェクトファイル」にできます。これにより、プロジェクトファイルがワークスペースファイルに相対的になり、ワークスペース構造の外に置くことも可能です。ソースツリー全体を再配置すると、High-performance Embedded Workshop はファイルの相対位置をチェックしファイルを検索できます。このチェックボックスのデフォルトはオンです。
4. ダイアログボックスの一番下にある[登録済みファイルを非表示]チェックボックスをオンにすると、現在のプロジェクトに追加されていないファイルだけを表示します。このチェックボックスのデフォルトはオフです。
5. [追加]ボタンをクリックしてください。

プロジェクトにファイルを追加するには、他にも方法があります。これを以下に説明します。

- 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ポップアップメニューを表示します。ファイルが既にプロジェクト内にある場合、[プロジェクトにファイルの追加]は使用できません。[プロジェクトにファイルの追加]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。

- High-performance Embedded Workshop では、Windows® Explorer からワークスペースウィンドウの [Projects] タブにファイルおよびフォルダをドラッグアンドドロップできます。次節の「2.3.2 ファイルおよびフォルダをドラッグアンドドロップする」を参照してください。

**注：**

High-performance Embedded Workshop の認識できない形式のファイルを指定しても、プロジェクトに追加します。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を Windows® operating system に受け渡します。ファイルを Windows® Explorer 内で開いたときと同じように、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、[ファイル拡張子] ダイアログボックスをご覧ください。詳細は、「2.5 ファイル拡張子とファイルグループ」を参照してください。

### 2.3.2 ファイルおよびフォルダをドラッグアンドドロップする

ワークスペースウィンドウの [Projects] タブに Windows® Explorer からファイル（あるいは複数のファイル）およびフォルダをドラッグできます。この場合、アクティブプロジェクトのプロジェクトまたはアクティブプロジェクトのユーザフォルダ上のみドロップできます。複数のフォルダはサポートしていません。

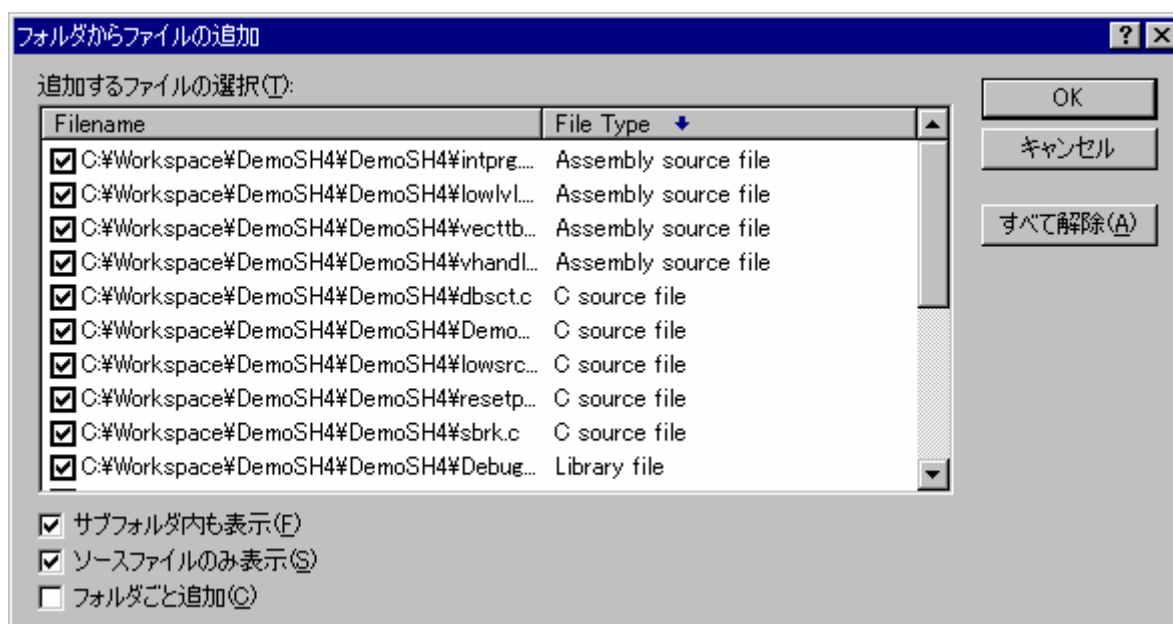
#### ファイルをドラッグアンドドロップする

ファイルのドロップ先により次のようになります。

- ファイルをアクティブプロジェクト上にドロップすると、ファイルは、そのファイルタイプのグループフォルダの下に追加されます。グループフォルダが存在しない場合は作成されます。例えば、ファイル「test.c」をアクティブプロジェクト上にドロップすると、「C source file」グループフォルダに追加されます。
- ファイルをユーザフォルダ上にドロップすると、ファイルはそのフォルダ下に直接追加されます。同じ名前のファイルでも異なるパスのファイルは追加されます。
- High-performance Embedded Workshop の認識できない形式のファイルをアクティブプロジェクト上にドロップすると、プロジェクト下に直接追加されます。

#### フォルダをドラッグアンドドロップする

Windows® Explorer のフォルダをドロップすると、[フォルダからファイルの追加] ダイアログボックスを開きます。どのファイルを追加するかを選択できます。このダイアログボックスには、ドロップしたフォルダに含まれていたすべてのファイルのファイル名を表示します。各リスト項目にはファイルのフルファイル名とファイルタイプを表示します。



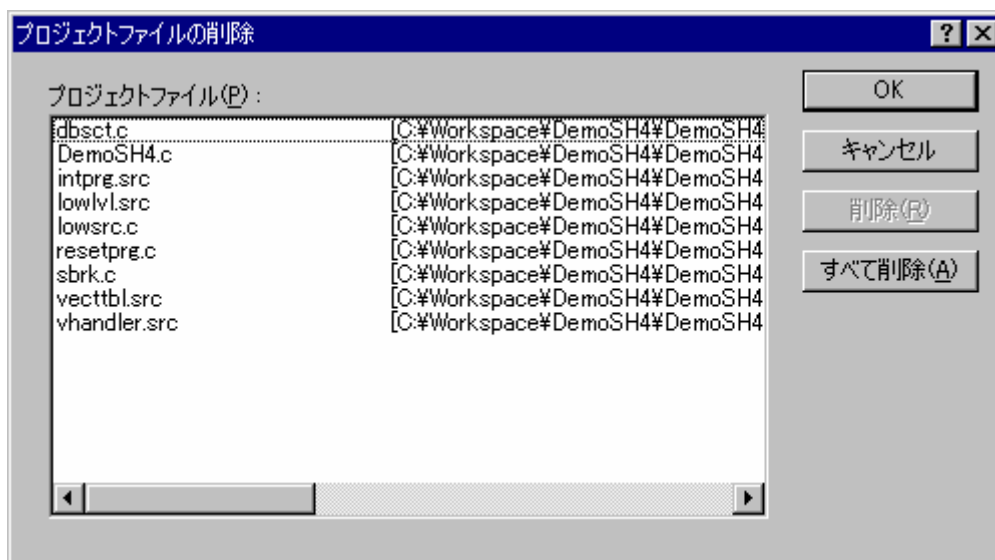
1. リストのカラムヘッダをクリックすると、そのカラムによってリスト項目をソートします。ダイアログボックスを表示したときは、ファイルタイプ（昇順）、ファイル名（昇順）の順にソートしています。
2. リスト項目は複数選択できます。選択した状態のリスト項目のチェックボックスをクリックまたは Enter キーを入力すると、選択した状態のすべての項目のオン、オフが可能です(クリックされたチェックボックスの元の状態に依存します)。
3. [サブフォルダ内も表示]チェックボックスをオフにすると、ドロップしたフォルダのファイルだけが表示されます。  
オンにすると、ドロップしたフォルダに含まれるサブフォルダのファイルも表示します。このチェックボックスのデフォルトはオンです。
4. [ソースファイルのみ表示]チェックボックスをオンにすると、このプロジェクトのソースファイルとして登録した拡張子のファイルだけ(例えば「C source file」など)を表示します。  
オフにするとファイルタイプをすべて表示します。High-performance Embedded Workshop が認識できない形式のファイルタイプの場合は、ダイアログボックスのファイルタイプカラムに[Unknown type]を表示します。このチェックボックスのデフォルトはオンです。
5. [フォルダごと追加]チェックボックスをオンにすると、ドロップしたツリーの下にツリー構造でファイルを追加します。例えば、ドロップする「data」フォルダ下にファイルがあると、「data」ユーザフォルダをプロジェクトに追加します。そのユーザフォルダ下にファイルを追加します。  
オフにすると、ユーザフォルダ上にフォルダをドロップした場合に、そのフォルダ下にすべてのファイルを追加します。  
アクティブプロジェクトのツリー上にフォルダをドロップすると、ファイルタイプのグループフォルダ下にファイルを追加します(ファイル「main.c」の場合は、アクティブプロジェクト下の「C source file」グループフォルダに追加します)。グループフォルダがない場合は生成されます。このチェックボックスのデフォルトはオフです。
6. [すべて解除]ボタンをクリックするとすべてのチェックボックスがオフになります。このとき[すべて選択]ボタンをクリックすると、すべてのチェックボックスがオンになります。
7. [OK]ボタンをクリックすると、チェックボックスがオンの状態のファイルをすべてプロジェクトに追加します。このダイアログボックスで[OK]ボタンをクリックし、次回このダイアログボックスを開くと、チェックボックスは前回のチェック状態を保存しています。

### 2.3.3 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、1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。

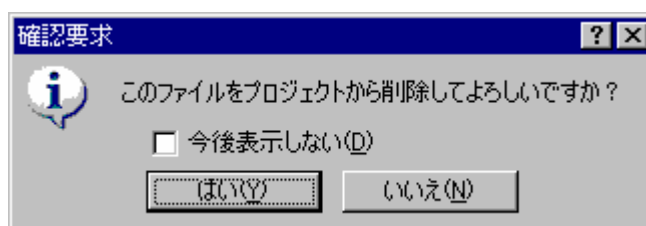
#### ファイルをプロジェクトから削除するには

- 以下のいずれかの操作を選択し[プロジェクトファイルの削除]ダイアログボックスを開きます。
  - [プロジェクト -> ファイルの削除]を選択する。
  - ワークスペースウィンドウの[Projects]タブのプロジェクト上で右クリックし、ポップアップメニューから[ファイルの削除]を選択する。
- [プロジェクトファイル]リストから削除したいファイルを選択してください。複数のファイルを選択することもできます。
- 選択したファイルを削除するには[削除]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[すべて削除]ボタンをクリックしてください。
- [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。



#### ワークスペースウィンドウの[Projects]タブで選択したファイルをプロジェクトから削除するには

- 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択してください。複数のファイルを選択するときは"Shift"キーまたは"Ctrl"キーを押してください。
- "Delete"キーを押してください。
- ファイルをプロジェクトから削除するかを確認するダイアログボックスが開きます。ファイルをプロジェクトから削除する場合は「はい」を選択してください。





確認要求のダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。確認要求のダイアログボックスを再び表示するには、[基本設定 -> オプション]を選択し[オプション]ダイアログボックスを開きます。[確認]タブの[プロジェクトからファイルを削除]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。

### 2.3.4 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ビルドから除外できます。

#### 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クリックしてください。  
次のようにマウスを使用したり、またはキー入力で複数のファイルを選択することもできます。
  - **複数のファイルを選択するには**  
Ctrl キーを押しながら対象のファイルをクリックします。
  - **複数のファイルを範囲で選択するには**  
範囲選択を開始するファイルをクリックし、Shift キーを押しながら範囲選択を終了するファイルをクリックします。
2. 以下のいずれかの操作を選択してください。ファイルのアイコンに赤いバツ印が付き(❌)、ビルドからファイルを除外します。
  - 右クリックしポップアップメニューから[ビルドから除外 <ファイル名>]を選択します。複数のファイルを選択した場合は<ファイル名>は表示されません。
  - [ビルド -> ビルドから除外/除外の解除]を選択します。

### 2.3.5 除外したプロジェクトファイルをビルドに再び入れる

除外したプロジェクト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]タブでファイルをクリックしてください。  
以下のようにマウスを使用したり、またはキー入力で複数のファイルを選択することもできます。
  - **複数のファイルを選択するには**  
Ctrl キーを押しながら対象のファイルをクリックします。
  - **複数のファイルを範囲で選択するには**  
範囲選択を開始するファイルをクリックし、Shift キーを押しながら範囲選択を終了するファイルをクリックします。
2. 以下のいずれかの操作を選択してください。ファイルのアイコンの赤いバツ印が外れ(☑)、ファイルがビルド可能になります。
  - 右クリックしポップアップメニューから[ビルドから除外の解除 <ファイル名>]を選択します。複数のファイルを選択した場合は<ファイル名>は表示しません。
  - [ビルド -> ビルドから除外/除外の解除]を選択します。



## 2.4 ワークスペースウィンドウでユーザフォルダを使う

ワークスペースウィンドウの[Projects]タブにユーザのフォルダを追加できます。これにより、プロジェクト内の特定の領域にファイルを論理的なグループにできます。フォルダ名として任意の名前をダイアログボックスで入力します。自動的に表示される[Download modules]フォルダおよび[Dependencies]フォルダに対して下記は操作できません。

### ユーザフォルダを追加するには

1. ワークスペースウィンドウの[Projects]タブでプロジェクトを選択してください。
2. 右クリックし、[フォルダの追加]を選択してください。
3. [フォルダの追加]ダイアログボックスの[フォルダ]に名前を入力してください。
4. [OK]ボタンをクリックしてください。
5. ファイルをフォルダ上にドラッグアンドドロップして、フォルダの分類を変更できます。

### フォルダにサブフォルダを追加するには

1. ワークスペースウィンドウの[Projects]タブでフォルダを選択してください。
2. 右クリックし、[フォルダの追加]を選択してください。
3. [フォルダの追加]ダイアログボックスの[フォルダ]に名前を入力してください。
4. [OK]ボタンをクリックしてください。
5. ファイルをフォルダ上にドラッグアンドドロップして、フォルダの分類を変更できます。

### フォルダをドラッグアンドドロップするには

ワークスペースウィンドウの[Projects]タブでアクティブプロジェクトのフォルダをドラッグしてください。

- フォルダをアクティブプロジェクト上にドロップすると、フォルダはそのプロジェクト下にドロップされます。
- フォルダをアクティブプロジェクトのユーザフォルダ上にドロップすると、フォルダはそのフォルダ下にドロップされます。

Windows® Explorer のフォルダをワークスペースウィンドウの[Projects]タブにドラッグアンドドロップできます。詳細は「2.3.2 ファイルおよびフォルダをドラッグアンドドロップする」を参照してください。

### ユーザフォルダを削除するには

1. ワークスペースウィンドウの[Projects]タブでフォルダを選択してください。
2. 以下のいずれかの操作を選択してください。このときフォルダは空である必要があります。
  - "Delete"キーを押してください。
  - 右クリックし、[フォルダの削除]を選択してください。

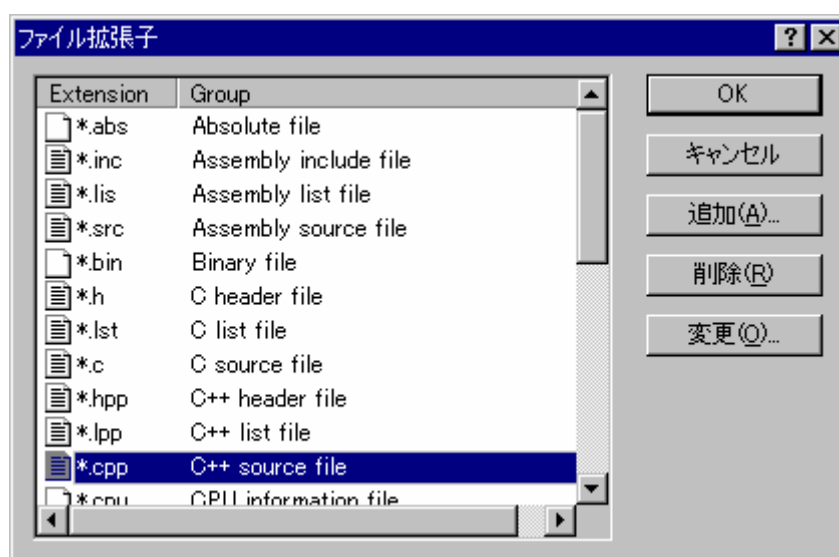
### ユーザフォルダ名を変更するには

1. ワークスペースウィンドウの[Projects]タブでフォルダを選択してください。
2. 右クリックし、[フォルダ名の変更]を選択してください。
3. [フォルダ名の変更]ダイアログボックスの[フォルダ]に新しい名前を入力してください。
4. [OK]ボタンをクリックしてください。

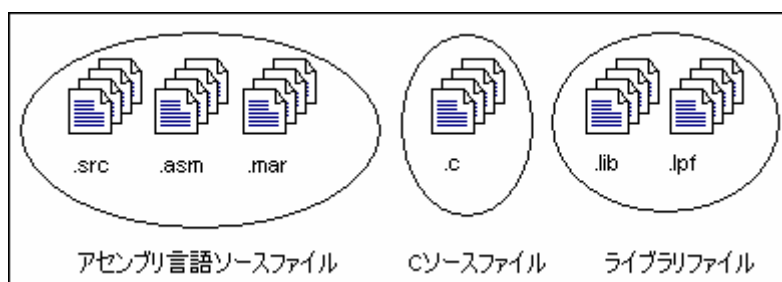
## 2.5 ファイル拡張子とファイルグループ

High-performance Embedded Workshop 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します。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ります。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指定できます。

ファイルの拡張子は、[ファイル拡張子]ダイアログボックスで表示、変更できます。このダイアログボックスを開くには[プロジェクト -> ファイルの拡張子]を選択してください。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しているすべての拡張子とファイルグループを表示します。



ファイル拡張子リストは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。下記の図に示すように、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)。



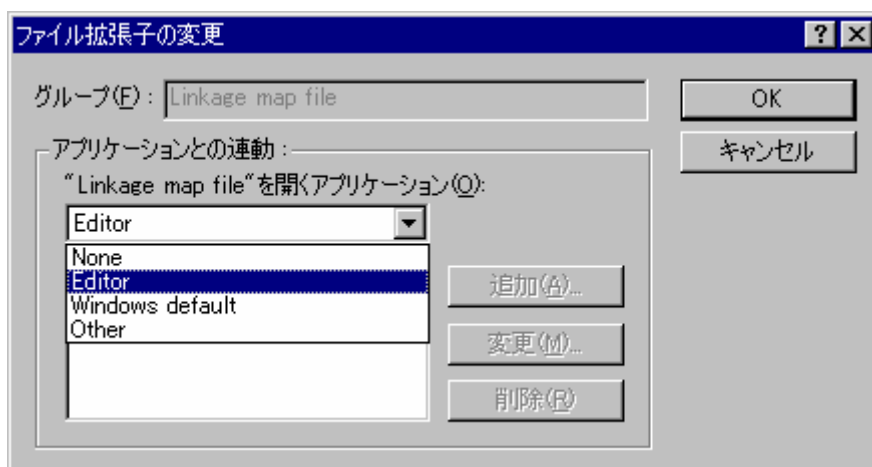
### 2.5.1 アプリケーションとファイルグループを関連付ける

[ファイル拡張子]ダイアログボックスでは、エディタでファイルを開くための設定だけでなく、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。

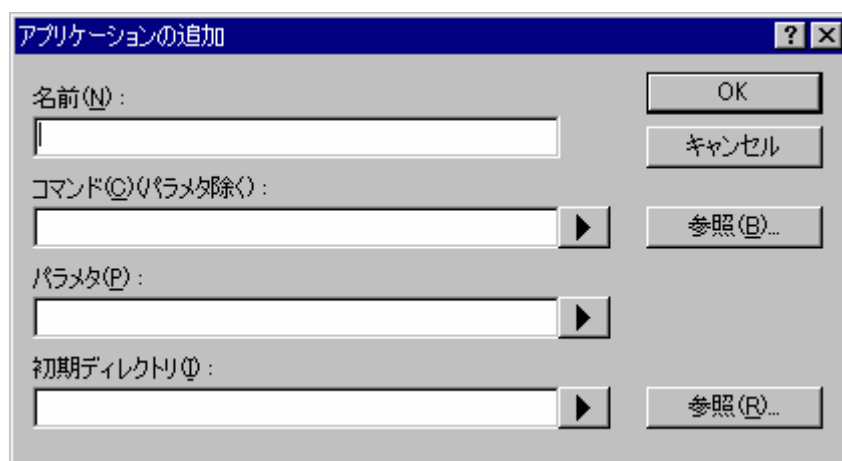


#### アプリケーションとファイルグループを関連付けるには

1. [プロジェクト -> ファイルの拡張子]を選択してください。[ファイル拡張子]ダイアログボックスを開きます。
2. [ファイル拡張子]ダイアログボックスで関連付けるファイルグループをリストから選択してください。
3. [変更]ボタンをクリックしてください。[ファイル拡張子の変更]ダイアログボックスを開きます。



4. 関連付けをしない場合は[None]を選択します。High-performance Embedded Workshop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、既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択するか、または、新しいアプリケーションを定義できます。新しいアプリケーションを定義する場合は[追加]ボタンをクリックしてください。[アプリケーションの追加]ダイアログボックスを開きます。



[名前]にアプリケーション名を入力します。[コマンド]にアプリケーションの絶対パスを入力します(パラメータは含めません)。[パラメタ]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)プレー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、「リファレンス 5. プレースホルダ」を参照してください。)[初期ディレクトリ]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を定義します。アプリケーションを変更するには[変更]ボタンをクリックしてください。[アプリケーションの変更]ダイアログボックスを開きます。このダイアログボックスは、[名前]が変更できないことを除いて[アプリケーションの追加]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。

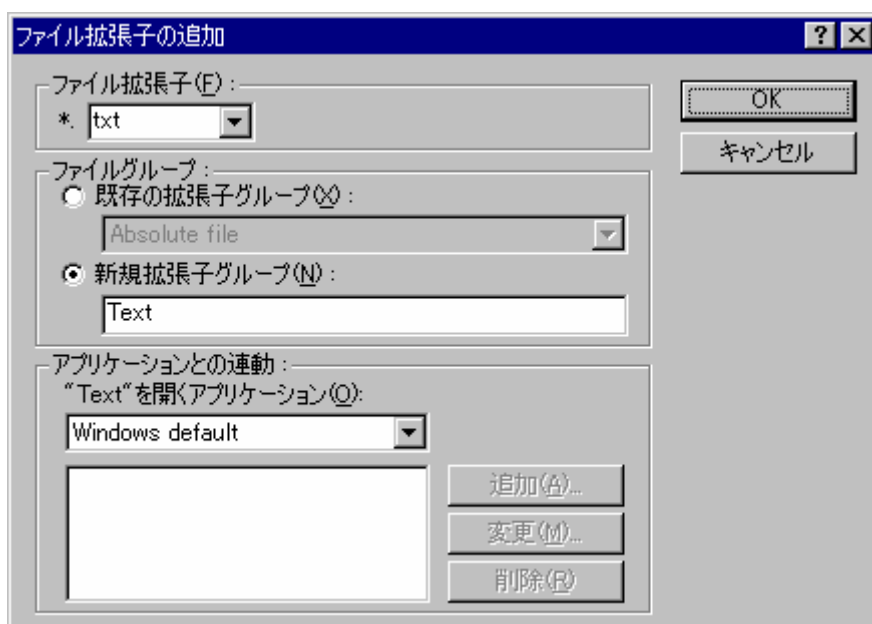
5. [OK]ボタンをクリックすると選択したファイルグループに対するアプリケーションを関連付けます。

## 2.5.2 新規ファイルグループに新規ファイル拡張子を登録する

新しい拡張子を作成するときは、その拡張子が既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か考慮してください。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登録するには

1. [プロジェクト -> ファイルの拡張子]を選択してください。[ファイル拡張子]ダイアログボックスを開きます。
2. [追加]ボタンをクリックしてください。[ファイル拡張子の追加]ダイアログボックスを開きます。



3. [ファイル拡張子]に定義する拡張子を入力してください。拡張子には半角英数字と半角下線のみ使用してください。ドロップダウンリストには、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つを選択すると、ファイル拡張子フィールドにテキストを自動的に表示します。
4. [新規拡張子グループ]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。[<拡張子グループ>を開くアプリケーション]ドロップダウンリストは以下から選択できます。
  - [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 High-performance Embedded Workshop エディタ内で開くことができます。[None]を選択した場合、High-performance Embedded Workshop でファイルを開くことを抑止します。[Other]を選択すると、ファイルを開くための他のツールを指定できます。詳細は、「2.5.1 アプリケーションとファイルグループを関連付ける」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、開いたファイルを Windows® operating system に受け渡します。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。

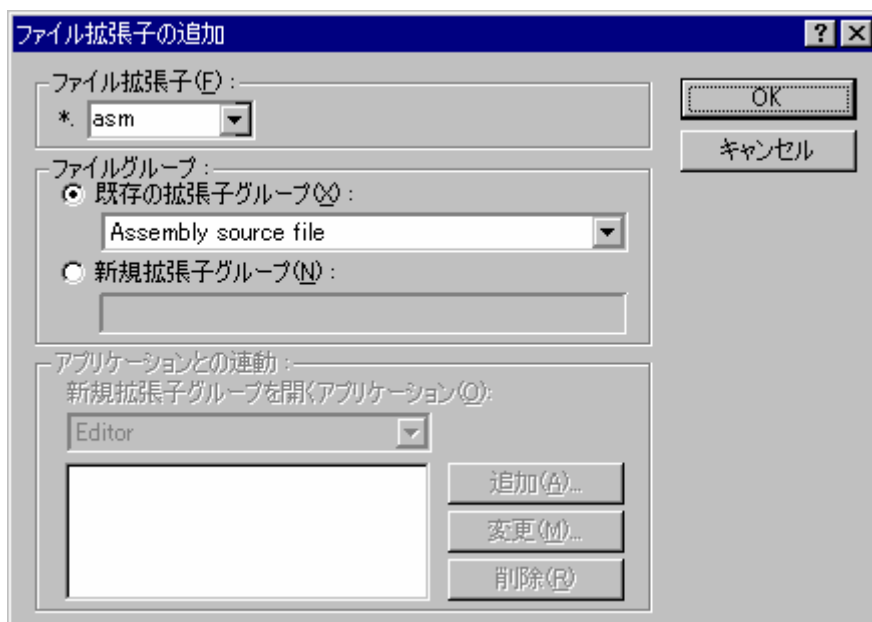
6. [OK]ボタンをクリックするとファイル拡張子リストに拡張子を追加します。

### 2.5.3 新規ファイル拡張子を作成する

High-performance Embedded Workshop に登録されてい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。追加方法を以下に説明します。

### 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. [プロジェクト -> ファイルの拡張子]を選択してください。[ファイル拡張子]ダイアログボックスを開きます。
2. 追加]ボタンをクリックしてください。[ファイル拡張子の追加]ダイアログボックスを開きます。



3. [ファイル拡張子]に定義する拡張子を入力してください。ドロップダウンリストには、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つを選択すると、ファイル拡張子フィールドにテキストを自動的に表示します。
4. [既存の拡張子グループ]オプションを選択して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ファイル拡張子リストに拡張子を追加します。

## 2.6 ビルドオプションを設定する

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 High-performance Embedded Workshop に各ファイルのビルド方法を指示します。このためには、[ビルド]メニューからメニューオプションを選択します。このメニューの名前は使用するツールによって異なります。

### ビルドフェーズにオプションを設定するには

1. [ビルド]メニューからオプションを変更するフェーズのメニューを選択すると、オプションを指定するダイアログボックスを開きます。
2. 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。

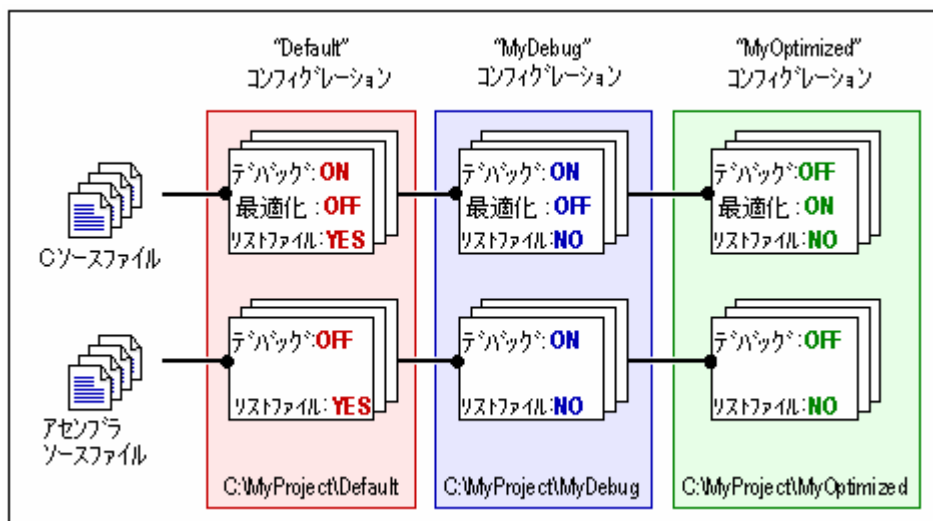
詳細は、状況依存のヘルプボタンをクリックするか、ヘルプを参照したい場所を選択し F1 キーを押してください。

## 2.7 ビルドのコンフィグレーション

High-performance Embedded Workshop 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。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択すると、各ビルドフェーズの各オプションを復歸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グ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して使用できます。

以下の図に"**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択できます。



### 2.7.1 ビルドコンフィグレーションを選択する

ビルドコンフィグレーションを選択するには

1. [ビルド -> ビルドの構成]を選択してください。[ビルドコンフィグレーション]ダイアログボックスを開きます。
2. [現在のコンフィグレーション]ドロップダウンリストから使用するビルドコンフィグレーションを選択してください。
3. [OK]ボタンをクリックしてください。



また、[標準]ツールバーのドロップダウンリストから異なるビルドコンフィグレーションを選択することもできます。



## 2.7.2 新規ビルドコンフィグレーションを追加する

### 新規ビルドコンフィグレーションを追加するには

1. [ビルド -> ビルドの構成]を選択してください。[ビルドコンフィグレーション]ダイアログボックスを開きます。
2. [追加]ボタンをクリックしてください。[コンフィグレーションの追加]ダイアログボックスを開きます。
3. [コンフィグレーション名]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。



4. [ベースコンフィグレーション]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。
5. 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を作成します。

## 2.7.3 ビルドコンフィグレーションを削除する

### ビルドコンフィグレーションを削除するには


1. [ビルド -> ビルドの構成]を選択してください。[ビルドコンフィグレーション]ダイアログボックスを開きます。
2. 削除するビルドコンフィグレーションを選択し[削除]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[ビルドコンフィグレーション]ダイアログボックスを閉じます。

## 2.8 プロジェクトをビルドする

### 2.8.1 ファイルをコンパイルする

プロジェクトにある1つのファイルをコンパイルできます。

#### 1つのファイルをコンパイルするには

1. ワークスペースウィンドウの[Projects]タブからビルドするファイルを選択してください。
2. 以下のいずれかの操作を選択してください。
  - [コンパイル]ツールバーボタン()をクリックする。
  - ポップアップメニューから[コンパイル <ファイル名>]を選択する。
  - [ビルド -> コンパイル]を選択する。
  - "Ctrl+F7"キーを押す。


すべての出力は、アウトプットウィンドウの[Build]タブに表示します。

## 2.8.2 プロジェクトをビルドする

[ビルド]オプションでは前回のビルド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"が前回のビルド以後に変更された場合、ファイル"TEST.C"を再コンパイルします。

### ビルドするには


以下のいずれかの操作を選択してください。

- [ビルド]ツールバーボタン()をクリックする。
- "F7"キーを押す。
- [ビルド -> ビルド]を選択する。
- ワークスペースウィンドウの[Projects]タブのプロジェクト上で右クリックし、ポップアップメニューから[ビルド -> ビルド]を選択する。

[すべてをビルド]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。

### "すべてをビルド"を実行するには

以下のいずれかの操作を選択してください。

- [すべてをビルド]ツールバーボタン()をクリックする。
- [ビルド -> すべてをビルド]を選択する。
- ワークスペースウィンドウの[Projects]タブのプロジェクト上で右クリックし、ポップアップメニューから[ビルド -> すべてをビルド]を選択する。

ビルド、あるいは全ビルドからのすべての出力は、アウトプットウィンドウの[Build]タブに表示します。プロジェクトファイルのどれかにエラーが起ると、ビルド処理も全ビルド処理も終了します。

## 2.8.3 複数のプロジェクトをビルドする

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するには

1. [ビルド -> 複数ビルド]を選択してください。[複数のビルド]ダイアログボックスを開きます。
2. [複数のビルド]ダイアログボックスでは、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を選択するには、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オンにします。例えば、プロジェクト"hewtest2"全体をビルド実行したい場合、"hewtest2-Debug"、"hewtest2-Release"を選択してオンにし、その他のチェックボックスをオフにしてください。




3. 項目を選択したら、[ビルド]ボタンをクリックしてください。選択したプロジェクトやコンフィグレーションのビルドを開始します。
4. 選択した項目に対して、全ビルド実行したい場合、[すべてをビルド]ボタンをクリックしてください。
5. 選択した項目の中間ファイルおよび出力ファイルを削除したい場合、[クリーン]ボタンをクリックしてください。このときビルドは実行されません。
6. 通常のビルドと同じように、ビルドの結果をアウトプットウィンドウの[Build]タブに表示します。
7. 再びダイアログボックスを開くと、前回のチェック状態を回復します。

## 2.8.4 ツールの実行を中止する

ビルド中のツールの実行を中止できます。

### ツールの実行を中止するには

以下のいずれかの操作を選択してください。

- [ツールの実行の中止]ツールバーボタン()をクリックする。
- [ビルド -> ツールの中止]を選択する。

ツールの実行が中止されると、アウトプットウィンドウの[Build]タブに"Build Stopped by User"というメッセージが表示されます。

### 注：

ビルド実行を中止したときに実行されていたツールによっては、無効なファイルを出力ファイルとして残していることがあります。ビルドを再実行する場合は、中止したツールの出力ファイルをすべて削除してください。

## 2.8.5 ビルドの中間ファイルおよび出力ファイルを削除する

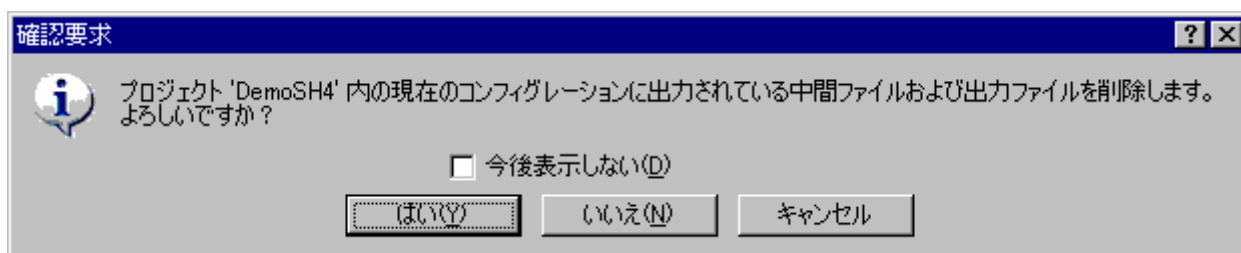
ワークスペースに含まれるプロジェクトのコンフィグレーションにあるビルドの中間ファイルおよび出力ファイルを削除できます。

### 現在のコンフィグレーションにあるビルドの中間ファイルおよび出力ファイルを削除するには

以下のいずれかの操作を選択してください。

- [ビルド -> クリーン アクティブプロジェクト]を選択する。
- ワークスペースウィンドウの[Projects]タブのプロジェクト上で右クリックする。表示されるポップアップメニューから[ビルド -> クリーン アクティブプロジェクト]を選択する。

すべてのビルドの中間ファイルおよび出力ファイルを削除するかを確認するダイアログボックスが開きます。削除するには「はい」を選択してください。削除しない場合は「いいえ」または「キャンセル」を選択してください。




確認要求ダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。

### 確認要求ダイアログボックスを再び表示するには

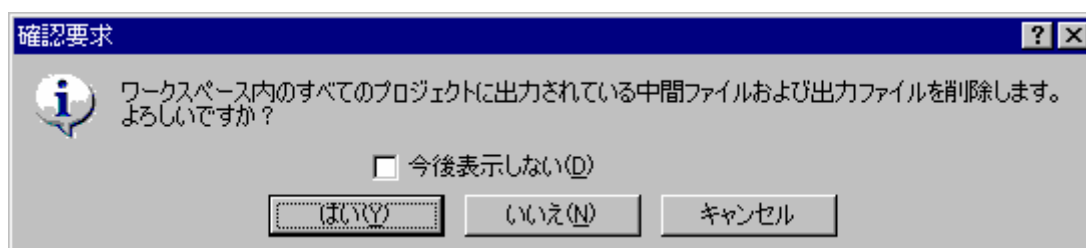
1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [確認]タブを選択してください。
3. [クリーン アクティブプロジェクト]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. OK をクリックしてください。

### すべてのプロジェクトのコンフィグレーションにあるビルドの中間ファイルおよび出力ファイルを削除するには

以下のいずれかの操作を選択してください。

- [クリーン 全プロジェクト]ツールバーボタン()をクリックする。
- [ビルド -> クリーン 全プロジェクト]を選択する。
- ワークスペースウィンドウの[Projects]タブのワークスペース上で右クリックする。表示されるポップアップメニューから[クリーン 全プロジェクト]を選択する。

すべてのビルドの中間ファイルおよび出力ファイルを削除するかを確認するダイアログボックスが開きます。削除するには「はい」を選択してください。削除しない場合は「いいえ」または「キャンセル」を選択してください。



確認要求ダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。

#### 確認要求ダイアログボックスを再び表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスが開きます。
2. [確認]タブを選択してください。
3. [クリーン 全プロジェクト]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. OK をクリックしてください。

中間ファイルおよび出力ファイルを削除したプロジェクト名およびコンフィグレーション名をアウトプットウィンドウの[Build]タブに表示します。

コンフィグレーションを複数選択しビルドの中間ファイルおよび出力ファイルを削除できます。詳細は、「2.8.3 複数のプロジェクトをビルドする」を参照してください。

#### 注：

リロケータブルファイルおよびアブソリュートファイルのみを削除します。リンケージリストファイル（".MAP"）、アセンブリプログラム用リストファイル（".LIS"）などは削除されません。アブソリュートファイルの拡張子は指定したフォーマットにより異なります。

### 2.8.6 アウトプットウィンドウのBuildタブの構成

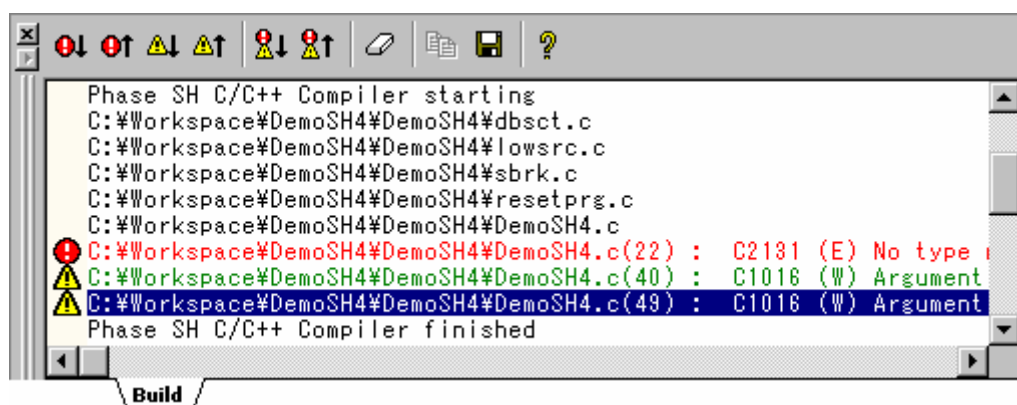
ビルド実行（コンパイラ、アセンブラなど）の出力結果を表示します。ソースファイルにエラーがある場合、アウトプットウィンドウの[Build]タブにアイコン、およびエラーメッセージをソースファイル名、行番号とともに出力します。

エラーメッセージの表示に関するツールバーボタンまたはポップアップメニューオプションを選択すると、エラーメッセージ出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。

エラーメッセージ出力行をダブルクリックすると、該当ソースがある場合はエディタにエラーメッセージの該当ソース行を表示します。

エラーメッセージの表示に関する操作を行った場合は、ステータスバーにエラーメッセージを表示します。

エラーメッセージ行の文字列を他の行の文字列と異なる色で強調表示されるようにカスタマイズが可能です。下の図はエラー行の文字列に赤色、ウォーニング行の文字列に緑色を設定した例です。



ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
-		次のエラーの出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。
-		前のエラーの出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。
-		次のウォーニングの出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。
-		前のウォーニングの出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。
次エラー/ウォーニング/情報の表示		次のエラー、ウォーニング、またはインフォメーションの出力行を強調表示し、該当ソース行がある場合はエディタに該当ソース行を表示します。
前エラー/ウォーニング/情報の表示		前のエラー、ウォーニング、またはインフォメーションの出力行を強調表示し、該当ソースがある場合はエディタに該当ソース行を表示します。
ヘルプ		この出力行に関するヘルプを表示します。
ジャンプ	-	この出力行に関するソース行をエディタに表示します。
クリア		ウィンドウ内容をクリアします。
保存		ウィンドウ内容をテキストファイルに保存します。
コピー		ウィンドウ内容の選択部分を Windows® クリップボードにコピーします。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。

[Build]タブではビルド実行により出力されたエラーメッセージ行の左側の余白（カラム）にエラーメッセージのエラーレベルに対応したアイコンを表示します。

アイコン名	アイコン	エラーメッセージレベル
Build Error		エラー
Build Warning		ウォーニング
Information		インフォメーション

アウトプットウィンドウの[Build]タブでは、エラーメッセージ行の文字列を他の行の文字列と異なる色で強調表示されるようにカスタマイズが可能です。

### 既存の色をカスタマイズするには

1. [基本設定 -> 表示の形式]を選択してください。[表示形式]ダイアログボックスを開きます。
2. [Output]アイテムをダイアログボックスの左側のツリーから選択し展開します。
3. 色を変更したいカテゴリを選択します。次の表に示すカテゴリが選択できます。

カテゴリ	カラータブの前面の色 (デフォルト)	カラータブの背景の色 (デフォルト)	カスタマイズされるタブ	カスタマイズされる出力内容
Text	SYSTEM	SYSTEM	すべてのタブ	すべて
Build Error Text	黒	白	Build	エラー行
Build Warning Text	黒	白	Build	ウォーニング行
Information Text	黒	白	Build	インフォメーション行

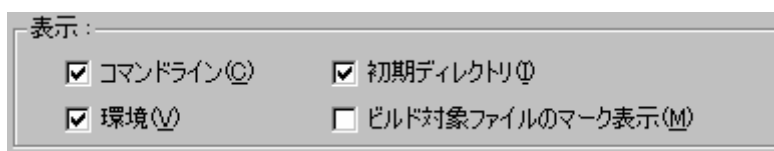
4. [カラー]タブの[前面]リストと[背景]リストの選択を変更します。
5. [OK]ボタンをクリックします。

### 2.8.7 アウトプットウィンドウのBuildタブの内容の制御

ビルド中にビルド実行情報（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。High-performance Embedded Workshop では、"ビルド"、"すべてをビルド"、"コンパイル"処理中、アウトプットウィンドウの[Build]タブにそのオプションを表示するかどうかを[オプション]ダイアログボックスで指定できます。

#### ビルド中にビルド情報の表示の有無を指定するには

1. [基本設定 -> オプション]を選択すると[オプション]ダイアログボックスを開きます。
2. [ビルド]タブを選択してください。
3. [表示]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。
  - [コマンドライン]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。
  - [環境]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。
  - [初期ディレクトリ]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。

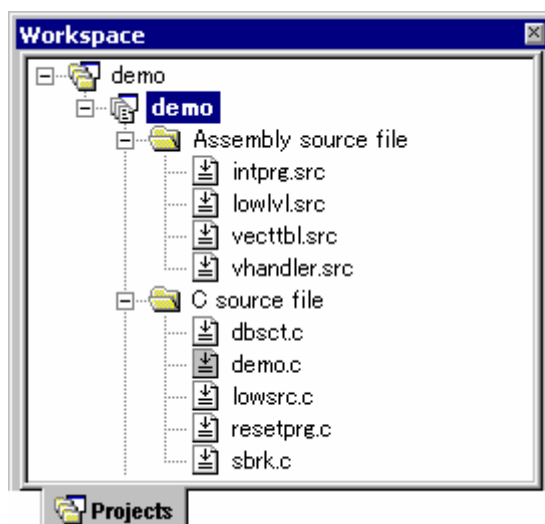


### 2.8.8 ビルド対象ファイルをワークスペースウィンドウにマーク表示する

前回のビルドで生成されたファイルより後に更新されたファイル（ビルド対象ファイル）をワークスペースウィンドウ上にマーク表示します。下記ウィンドウでは、ビルド対象ファイルは"demo.c"です。

次に[ビルド]をクリックしたとき、このファイルをビルドします。アクティブプロジェクトの依存プロジェクトのファイルにもマーク表示します。

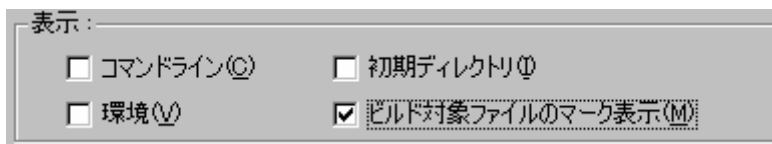
これらファイルのマーク表示は、ビルドに影響することが起こるたびに更新します。例えば、オプションの変更、ファイルの追加、依存関係の変更、ファイルの修正などです。





### ビルド対象ファイルをワークスペースウィンドウにマーク表示するには

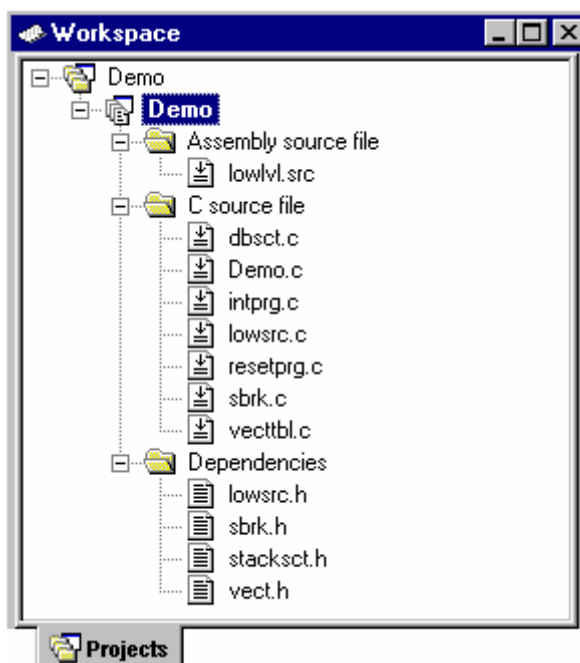
1. [基本設定 -> オプション]を選択し[オプション]ダイアログボックスを開きます。
2. [ビルド]タブを選択してください。
3. [ビルド対象ファイルのマーク表示]チェックボックスをオンにしてください。
4. [OK]ボタンをクリックしてください。



## 2.9 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、High-performance Embedded Workshop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ワークスペースウィンドウの[Projects]タブに表示します。

最初に、すべての依存ファイルは[Dependencies]フォルダに表示されています。



#### 注:

インクルードファイルからさらにファイルがインクルードされている場合、依存ファイルとして表示するインクルードファイルのネストレベルは31までです。

プロジェクトのファイル依存関係は、以下の操作タイミングで自動更新されます。

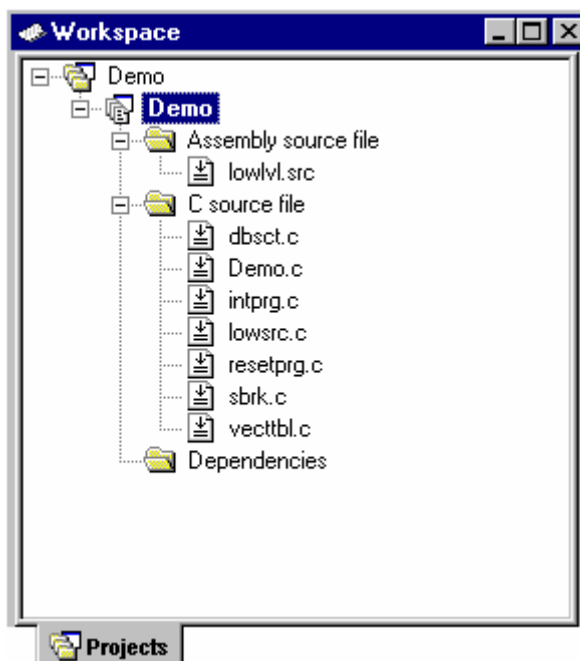
操作タイミング	主な操作
ワークスペースを開く	[ようこそ]ダイアログボックスのオプションの選択 [ファイル -> ワークスペースを開く]の選択 [ファイル -> 新規ワークスペースの作成]の選択
ワークスペースにプロジェクトを挿入する	[プロジェクト -> プロジェクトの挿入]の選択
アクティブプロジェクトに設定する	[プロジェクト -> アクティブプロジェクトに設定]の選択
ファイルを追加する	[プロジェクト -> ファイルの追加]の選択
ファイルを保存する	[ファイル -> 上書き保存]の選択
ビルドする	[ビルド -> ビルド]の選択
依存関係を自動更新する	[依存関係の自動更新をしない]チェックボックスをオフにする

### プロジェクトのファイル依存関係を自動更新しないようにするには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ビルド]タブを選択します。
3. [依存関係の自動更新をしない]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオフです。
4. OK をクリックしてください。

#### 注：

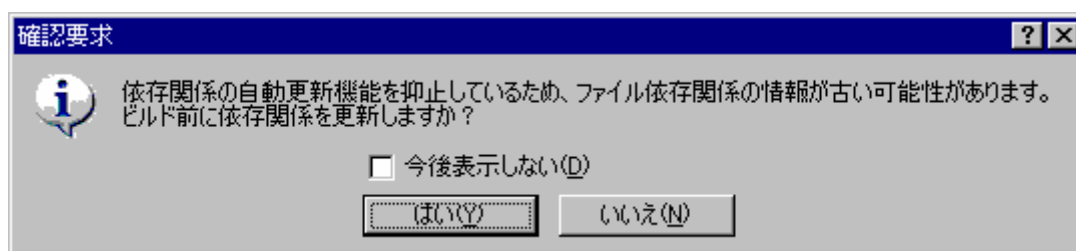
この設定を行うと、プロジェクトのファイル依存関係はすべてクリアされます。プロジェクトのファイル依存関係を手動で更新するには、「プロジェクトのファイル依存関係を手動で更新するには」を参照してください。



また、[ビルド -> ビルド]を選択した場合、ビルド前に依存関係を更新するかを確認するダイアログボックスが開きます。

依存関係の自動更新機能を抑止しているため、ファイル依存関係の情報が古い可能性があります。

ビルド前に依存関係を更新する場合は、「はい」を選択してください。「いいえ」を選択した場合、現在の依存関係の情報でビルドを実行します。



確認要求ダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。

#### 確認要求ダイアログボックスを再び表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスが開きます。
2. [確認]タブを選択してください。
3. [ビルド前に依存関係を更新]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. OK をクリックしてください。

#### プロジェクトのファイル依存関係を手動で更新するには

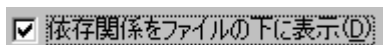
以下のいずれかの操作を選択してください。

- [ビルド -> すべての依存関係を更新]を選択する。
- ワークスペースウィンドウの[Projects]タブのプロジェクト上で右クリックし、ポップアップメニューから[ビルド -> すべての依存関係を更新]を選択する。

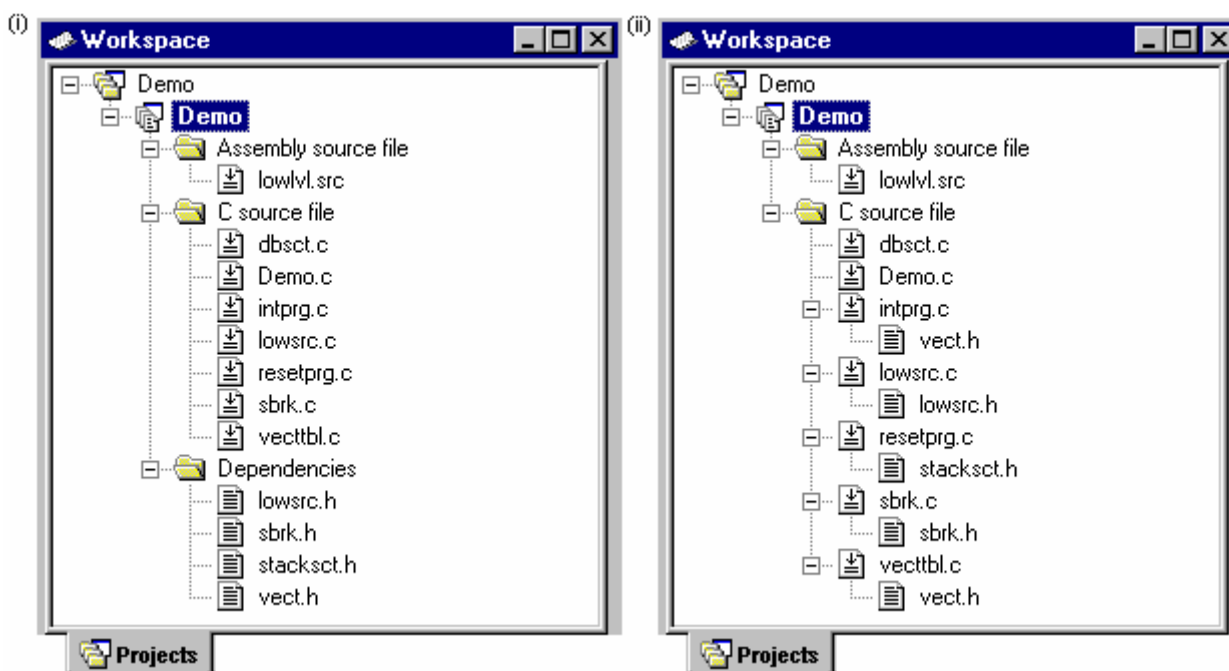
## 2.10 表示方法を指定する

ワークスペースウィンドウの[Projects]タブの中で右クリックすると、ポップアップメニューを表示します。その中から[表示の構成]を選択し、以下の情報の表示方法を設定してください。以下に[表示の構成]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。

#### 各ファイルの下に依存ファイルを表示する



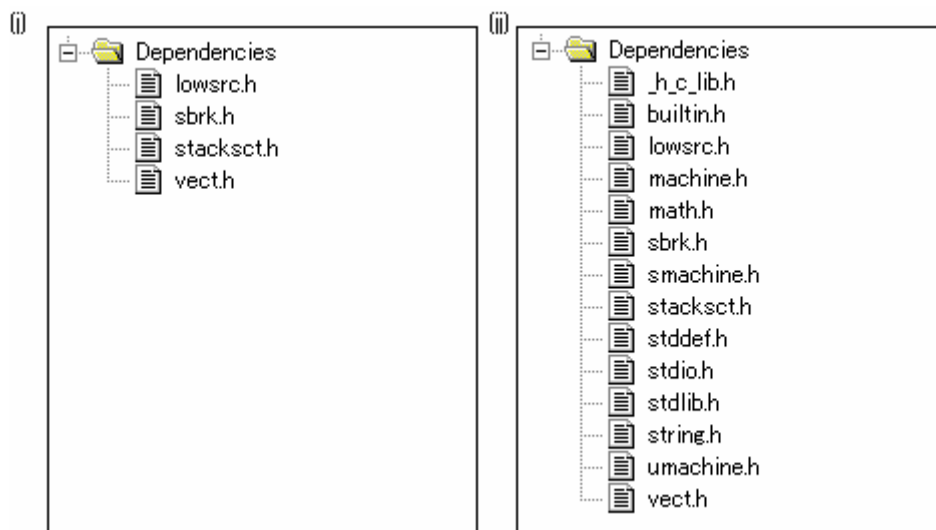
[依存関係をファイルの下に表示]チェックボックスをオンにすると、依存ファイルがそれをインクルードするソースファイルの下に平坦な構造で表示され、ファイル自体がフォルダになります（図(ii)を参照）。このオプションを選択しないと[Dependencies]フォルダにすべてのファイル依存を示します（図(i)を参照）。



標準ライブラリファイルのインクルードを表示する



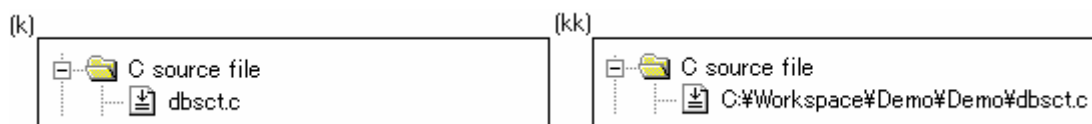
デフォルトでは、標準インクルードパスの依存ファイルは表示しません（図(j)を参照）。例えば、Cコードで#include <stdio.h>などのインクルード文を書くと、stdio.hは依存ファイルとして表示しません。そのようなシステムインクルードファイルを表示するには、[インクルードする標準ライブラリの表示]チェックボックスをオンにしてください（図(jj)を参照）。



## ファイルのパスを表示する

ファイルパスの表示(F)

デフォルトでは、ファイルのパスは表示されません（図(k)を参照）。[ファイルパスの表示]チェックボックスをオンにすると、[Projects]タブのすべてのファイルを絶対パス（ドライブ名からのパス）で表示します（図(kk)を参照）。



## ファイルをタイムスタンプの順序でソートする

ファイルのタイムスタンプによるソート(T)

[ファイルのタイムスタンプによるソート]チェックボックスをオンにすると、ワークスペース内のファイルをタイムスタンプの順序で並べることができます。最新のファイルがリストの先頭、最も古いファイルがリストの最後になります。

このオプションを設定した後でファイルを更新した場合は、手動で表示順を更新しなければいけません。

### 表示順を手動で更新するには

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[リフレッシュ]を選択してください。

## 2.11 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1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

新規プロジェクト(N)

1. [プロジェクト -> プロジェクトの挿入]を選択してください。[プロジェクトの挿入]ダイアログボックスを開きます。
2. [新規プロジェクト]ラジオボタンを選択してください。
3. [OK]ボタンをクリックしてください。[新規プロジェクトの挿入]ダイアログボックスを開きます。
4. [プロジェクト名]にプロジェクト名を入力してください。プロジェクト名は32文字以内で、半角英数字、半角下線のみ使用してください。特に、日本語文字、空白、または半角マイナス記号を使用しないでください。プロジェクト名を入力すると、High-performance Embedded Workshop 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. [参照]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択してください。または、[ディレクトリ]にディレクトリを入力できます。

6. プロジェクトタイプリストには使用可能なプロジェクトの種類を示します (Application、Library など)。このリストから作成するプロジェクトの種類を選択してください。
7. [OK]ボタンをクリックすると、プロジェクトを作成しワークスペースに追加します。

#### 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. [プロジェクト -> プロジェクトの挿入]を選択してください。[プロジェクトの挿入]ダイアログボックスを開きます。
2. [既存プロジェクト]ラジオボタンを選択してください。
3. [参照]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 (".HWP"ファイル) を検索してください。
4. [OK]ボタンをクリックするとそのプロジェクトをワークスペースに追加します。

## 2.12 アクティブプロジェクトを設定する

プロジェクトには 3 つの状態があります。アクティブプロジェクト、ロード (読み込まれた) プロジェクト、およびアンロードプロジェクトです。

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が、アクティブなプロジェクトは1つだけです。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を実行します。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します。

#### プロジェクトをアクティブにするには

以下の操作のいずれかを選択します。

- [プロジェクト -> アクティブプロジェクトに設定]サブメニューからアクティブにしたいプロジェクトを選択してください。
- ワークスペースウィンドウの[Projects]タブからアクティブでないプロジェクトを選択してください。右クリックしてポップアップメニューを表示し、[アクティブプロジェクトに設定]オプションを選択してください。

プロジェクトが読み込まれていると、プロジェクトのディレクトリを開き、ファイルを見ることができます。またプロジェクトのビルダおよびデバッグオプションを変更することもできます。読み込みプロジェクトに対して[ツール]メニューのツールを実行することもできます。

#### プロジェクトをアンロードにするには

1. ワークスペースウィンドウの[Projects]タブからアクティブでないプロジェクトを選択してください。
2. 右クリックしてポップアップメニューを表示し、[プロジェクトのアンロード]を選択してください。一度にいくつかのプロジェクトを選択することもできます。

アンロードしたプロジェクトは、ワークスペースウィンドウの[Projects]タブのアイコンがグレー表示です。アンロードしたプロジェクトがある場合は以下の操作によりプロジェクトを読み込むことができます。

### ワークスペースのすべてのプロジェクトを読み込むには

1. ワークスペースウィンドウの[Projects]タブのワークスペースを選択してください。
2. 右クリックしポップアップメニューを表示し、[全プロジェクトの読み込み]を選択してください。

### ワークスペースのプロジェクトを読み込むには

1. ワークスペースウィンドウの[Projects]タブからアンロードしたプロジェクトを選択してください。
2. 右クリックしてポップアップメニューを表示し、[プロジェクトの読み込み]を選択してください。一度にいくつかのプロジェクトを選択することもできます。

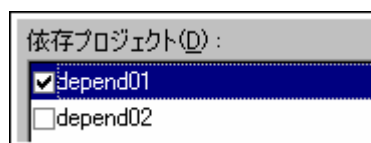
## 2.13 プロジェクト間の依存関係を指定する

ワークスペースのプロジェクトは、他のプロジェクトと依存関係を持たせることができます。ビルド処理をすると、依存プロジェクトを最初にビルドします。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使用します。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、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を正確にビルドし、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を先にビルドします。

依存プロジェクトをビルドするとき、High-performance Embedded Workshop 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、High-performance Embedded Workshop 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、High-performance Embedded Workshop 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指定するには

1. [プロジェクト -> 依存関係]を選択してください。[依存プロジェクト]ダイアログボックスを開きます。
2. 依存関係を指定するプロジェクトを選択してください。[依存プロジェクト]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を表示します。



3. [依存プロジェクト]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。選択した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オンにしてください。
4. [OK]ボタンをクリックしてください。

アクティブプロジェクトに対し依存プロジェクトを設定している場合、ビルド時、アクティブプロジェクトより先に依存プロジェクトのビルドが必要か High-performance Embedded Workshop は判定します。

この依存プロジェクトの判定は、アクティブプロジェクトを親とした場合、子にあたるプロジェクトまで実施します。子にあたるプロジェクトに対して依存プロジェクトを設定していても、その依存プロジェクトは判定しません。



## 2.14 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]タブで削除したいプロジェクトを選択してください。
2. 右クリックしポップアップメニューを表示してください。
3. [プロジェクトの削除]を選択してください。

#### 注：

ワークスペースからアクティブプロジェクトを削除することはできません。

## 2.15 ワークスペースのプロジェクト相対パス

High-performance Embedded Workshop 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選択できます。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、High-performance Embedded Workshop 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、High-performance Embedded Workshop 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。

High-performance Embedded Workshop 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ていました。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、High-performance Embedded Workshop の標準的な動作です。

### プロジェクト相対パスを設定するには

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. 右クリックしてポップアップメニューを表示してください。
3. [プロパティ]を選択してください。
4. [.HWS ファイルからプロジェクトファイルへの相対パス]チェックボックスをオンにしてください。
5. [OK]ボタンをクリックしてください。

.HWSファイルからプロジェクトファイルへの相対パス(P)

### 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。

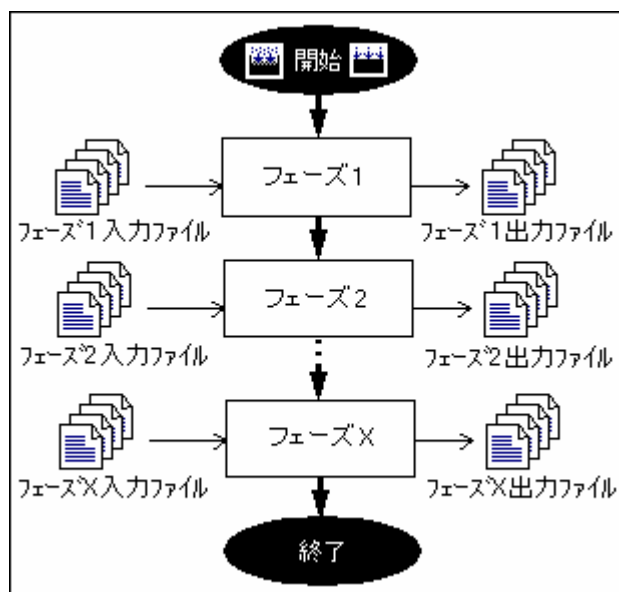
#### 3.1 ビルド実行の復習

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、リンカージェディタを用いて説明しました。High-performance Embedded Workshop 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必要があります。

##### 3.1.1 ビルドとは？

プロジェクトのビルドとは、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/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。以下の図にビルド処理を示します。



High-performance Embedded Workshop ではビルド処理を変更できます。[ビルド -> ビルドフェーズ]を選択し [ビルドフェーズ]ダイアログボックスを開きます。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。



この章では[ビルドフェーズ]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。

## 3.2 カスタムビルドフェーズを作成する

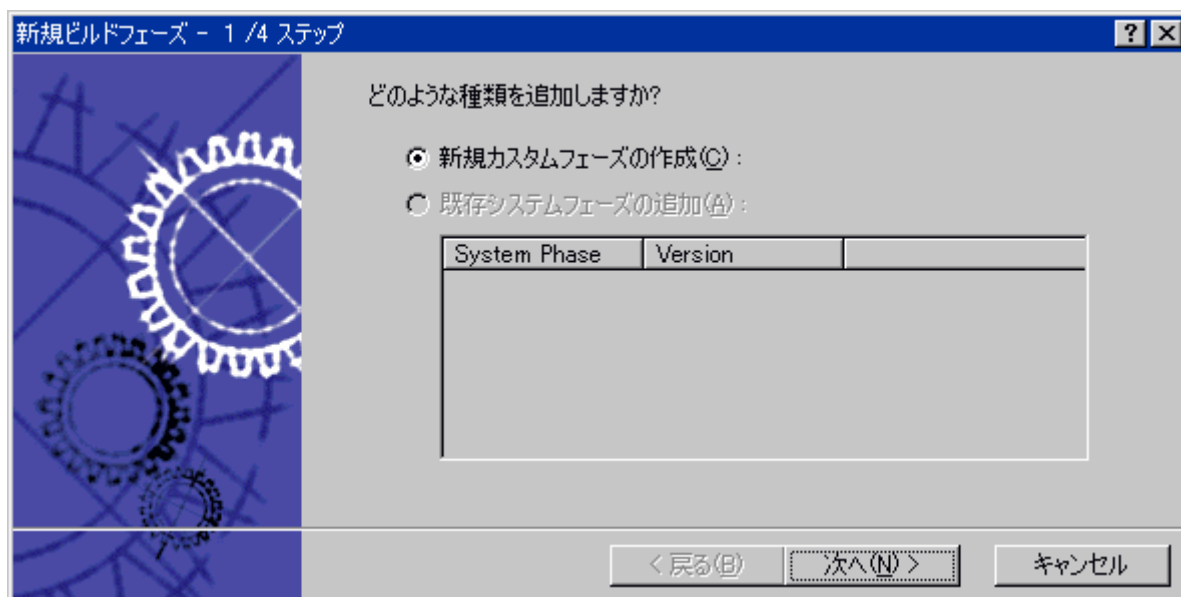
標準のビルドの前後または途中で他のツールを実行する場合、独自のビルドフェーズ（カスタムビルドフェーズ）を作成します。カスタムビルドフェーズを略してカスタムフェーズといいます。

### 新しいカスタムフェーズを作成するには

1. [ビルド->ビルドフェーズ]を選択すると[ビルドフェーズ]ダイアログボックスを開きます。
2. [追加]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[新規ビルドフェーズ]ダイアログボックスを開きます。
3. 以下の4つのステップにしたがってください。ステップを進めるには[次へ>]ボタンを、戻るには[戻る]ボタンをクリックしてください。

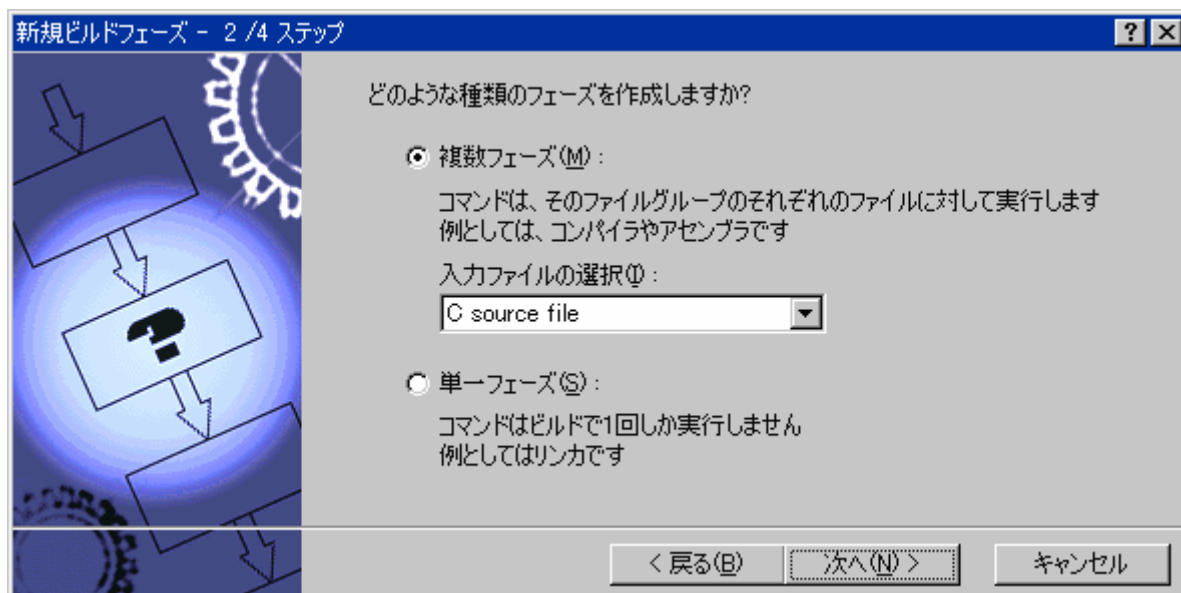
#### ステップ1

[新規カスタムフェーズの作成]または[既存システムフェーズの追加]を選択します。システムビルドフェーズ（略してシステムフェーズといいます）は、使用しているツールチェーン（コンパイラ、アセンブラ、リンカージェネレータ、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ない場合、[既存システムフェーズの追加]ボタンが非アクティブになります。[新規カスタムフェーズの作成]を選んでカスタムフェーズを作成してください。

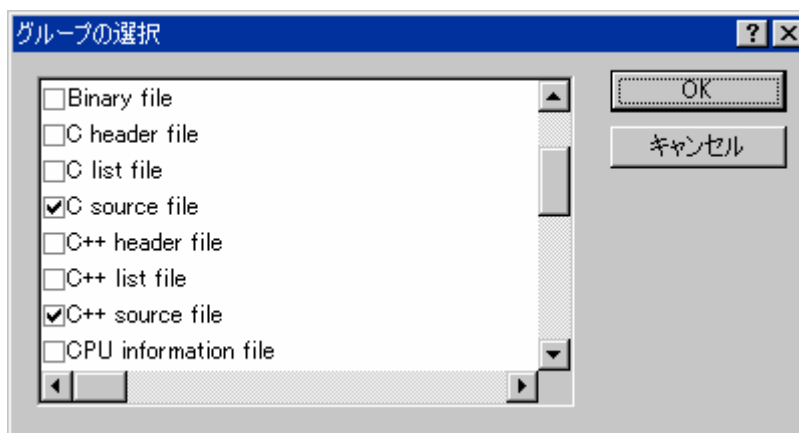


## ステップ 2

作成するフェーズの種類を選択します。2つの選択肢（[複数フェーズ]または[単一フェーズ]）があります。複数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を適用します。[入力ファイルの選択]フィールドに[C source file]を選択すると、プロジェクト内のファイル毎にコンパイラを実行します。単一フェーズを選択するとビルド実行中に一度だけ実行します。



[入力ファイルの選択]ドロップダウンリストは、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。[入力ファイルの選択]ドロップダウンリストの中の[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できる[グループの選択]ダイアログボックスを開きます。



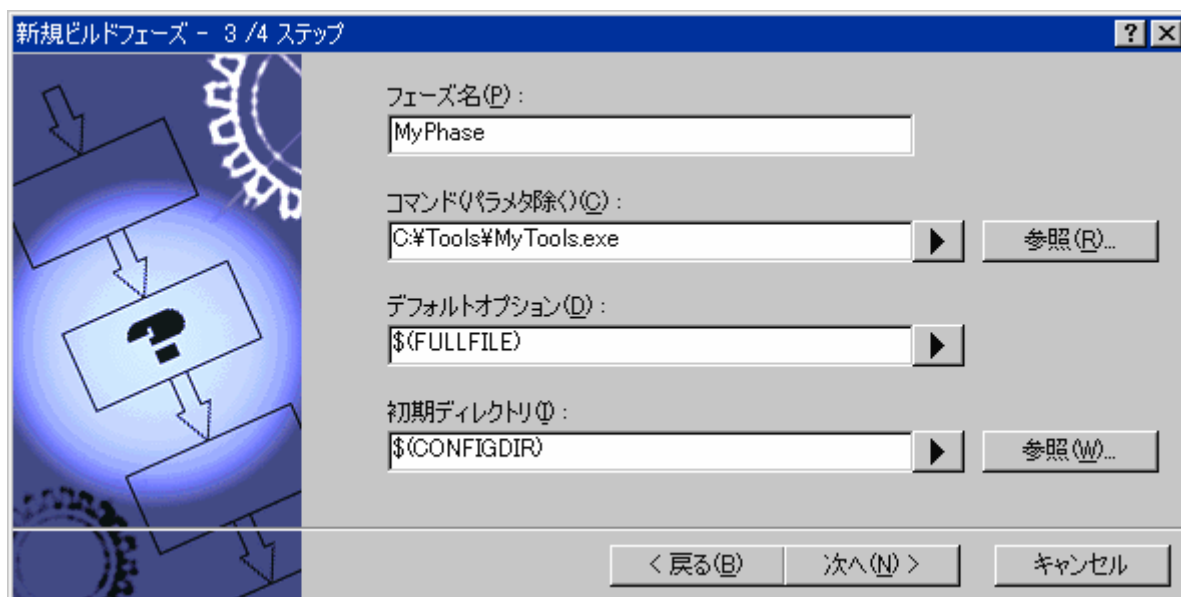
複数選択すると、[入力ファイルの選択]ドロップダウンリストは[Multiple Groups]と表示します。プロジェクトに追加されているカスタムフェーズ用に、複数の入力ファイルグループをこのダイアログボックスで選択できます。ファイルグループを選択するには、ファイルグループ名の隣にあるチェックボックスをオンにしてください。このダイアログボックスでは、1つ以上のファイルグループが選択できます。

### ステップ 3

フェーズのコマンドについての情報を入力します。[フェーズ名]にフェーズ名を入力します。[コマンド]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 High-performance Embedded Workshop のメニューバーの[ビルド]メニューで指定します）。[デフォルトオプション]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加えられるオプションです。[初期ディレクトリ]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。

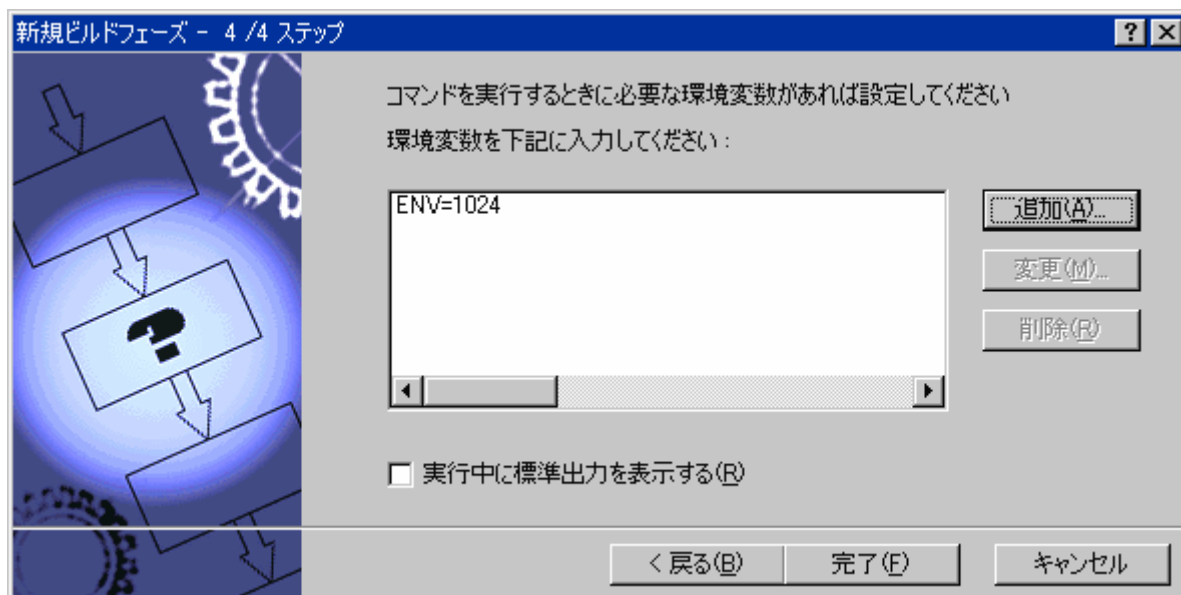
#### 注：

[フェーズ名]には日本語文字（全角英数字も含む）を使用しないでください。また、High-performance Embedded Workshop V.1.x で作成したプロジェクトを High-performance Embedded Workshop V.2.X 以上で使用する場合、日本語名のカスタムフェーズについては、半角英数字等（半角下線、半角マイナス、空白は使用可能）の名称で設定し直してください。

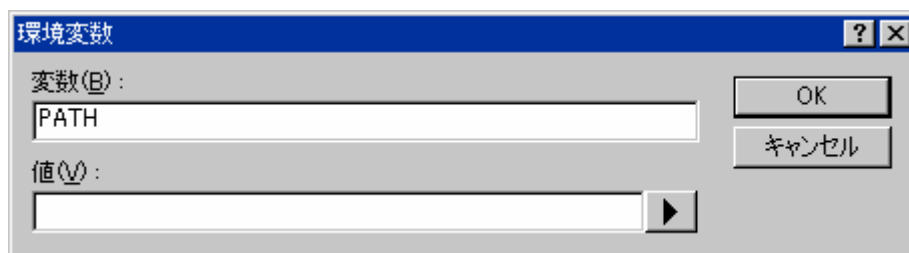


#### ステップ 4

フェーズに必要な環境変数を指定します。



新しい環境変数を追加するには、[追加]ボタンをクリックしてください。[環境変数]ダイアログボックスを開きます。新しい環境変数を追加するには[変数]に環境変数名を入力して[値]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、[変更]ボタンをクリックします。[変数]と[値]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を追加します。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、[削除]ボタンをクリックします。



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[実行中に標準出力を表示する]オプションを使用してください。出力が発生するご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、High-performance Embedded Workshop 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アウトプットウィンドウに表示します。

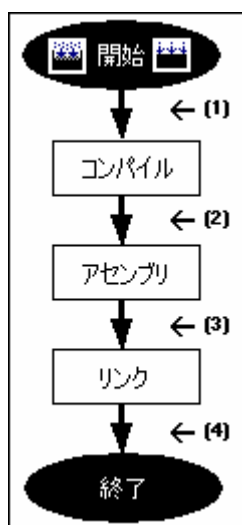
#### 注：

[実行中に標準出力を表示する]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 High-performance Embedded Workshop 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、[実行中に標準出力を表示する]オプションのチェックを外してください。

[完了]ボタンをクリックすると、指定した内容で新しいフェーズを作成します。デフォルトでは[ビルドフェーズ]ダイアログボックスの[ビルド順序]タブの[ビルドフェーズの順序]リストの最後に新しいフェーズを追加します。

### 3.3 ビルドフェーズの順序

標準的ビルド処理（下の図）では、(1)コンパイラの前、(2)アセンブラの前、(3)リンカの前、(4)リンカの後、の4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。



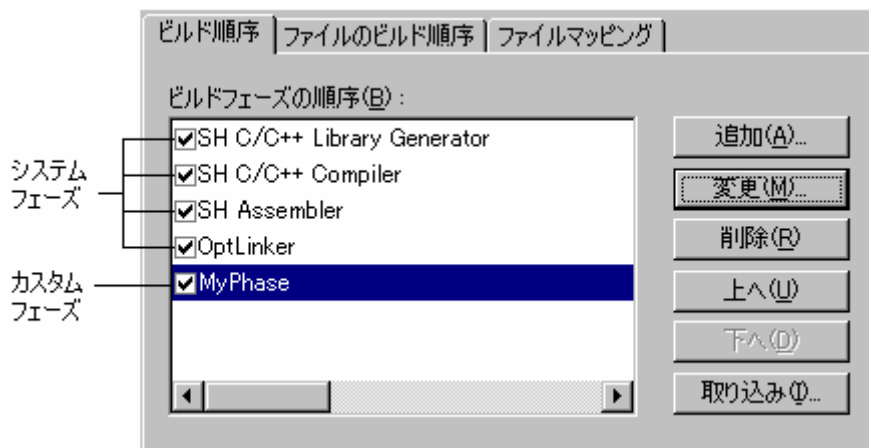
[ビルド -> ビルドフェーズ]を選択すると[ビルドフェーズ]ダイアログボックスを開きます。[ビルドフェーズ]ダイアログボックスでビルドフェーズの順序を変更できます。このダイアログボックスにはフェーズの順序



に関するタブが2つあります。[ビルド順序]タブと[ファイルのビルド順序]タブです。変更後[OK]ボタンをクリックしてください。

### 3.3.1 ビルド順序タブ

[ビルド順序]タブは、[ビルド]ボタン(🏗️)または[すべてをビルド]ボタン(🏗️)操作で実行されるフェーズの順序を示します。各フェーズの左にあるチェックボックスのオン/オフの有無はそのフェーズの有効/無効を示します。ビルド時はこのチェックボックスがチェックされているフェーズを実行します。



[ビルド]または[すべてをビルド]でシステムフェーズ/カスタムフェーズの順序を変更するには

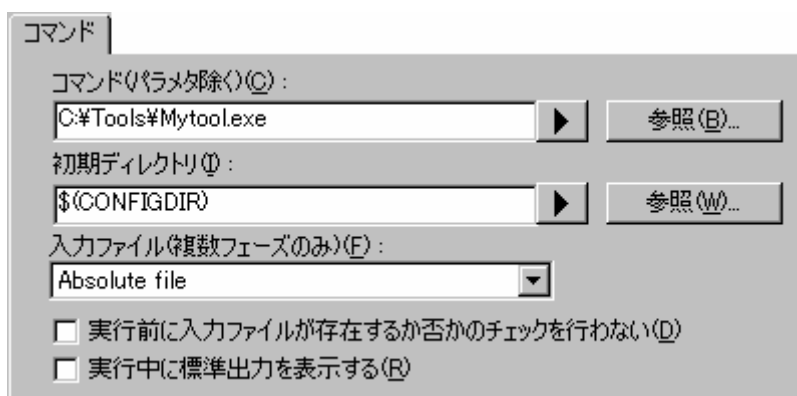
1. 移動するフェーズを選択してください。[上へ]または[下へ]ボタンをクリックすると上下に移動します。
2. [OK]ボタンをクリックしてください。

システムフェーズのプロパティを参照するには

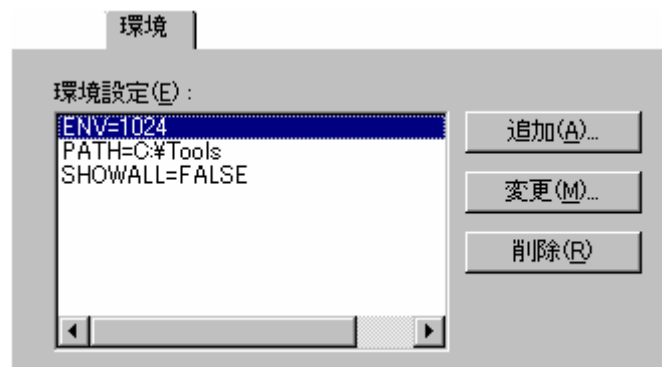
1. プロパティを参照するシステムフェーズを選択してください。
2. [変更]ボタンをクリックしてください。
3. [コマンド]タブではシステムフェーズに関する情報を表示します。コピーライト、バージョンなどです。
4. システムフェーズの環境設定を参照するには[環境]タブを選択してください。
5. [OK]ボタンをクリックしてください。

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択してください。
2. [変更]ボタンをクリックしてください。Phaseの変更ダイアログボックスの[コマンド]タブを表示します。



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. 複数フェーズの場合、フェーズを実行する入力ファイルタイプを[入力ファイル(複数フェーズのみ)]ドロップダウンリストから選択してください。
5. 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、[実行前に入力ファイルが存在するか否かのチェックを行わない]チェックボックスをオンにしてください。
6. 標準出力への出力をコマンド実行中に表示したい場合は、[実行中に標準出力を表示する]チェックボックスをオンにしてください。
7. [環境]タブを選択しフェーズの環境設定を行ってください。



8. 環境変数の追加は[追加]ボタン、変更は[変更]ボタン、削除は[削除]ボタンを使用してください。
9. [出力シンタックス]タブを選択し、カスタムフェーズのエラー、ウォーニング、およびインフォメーションを定義してください。

本機能で解析できるメッセージは、ファイル名を含んだメッセージのみです。したがって、定義するシンタックスにはファイル名の指定が必要です。ファイル名を指定する際には、必ずプレースホルダを使用してください。



アウトプットウィンドウの[Build]タブに出力されるエラーメッセージが[出力シンタックス]の定義と一致する場合、そのエラーメッセージの左側にアイコンを表示したり、該当するソースコードをエディタで参照したりできます。

例えば、Renesas SuperH C/C++ Compiler の場合は、アウトプットウィンドウの[Build]タブに以下のようなエラーメッセージを出力します。

```
C:\¥Workspace¥SH¥SH¥resetprg.c(70) : C2225 (E) Undeclared name "a"
```

このエラーを定義するには、以下のシンタックスを設定します。

```
$(FULLFILE)$(LINE))*(E)*
```

エラーメッセージとシンタックスとの対応を以下に示します。

エラーメッセージ例	シンタックス例	説明
C:\¥Workspace¥SH¥SH¥resetprg.c	\$(FULLFILE)	絶対パスファイル名のプレースホルダ
(70)	\$(LINE))	行番号のプレースホルダ (丸括弧付き)
: C2225	*	":"およびエラーメッセージ番号用のワイルドカード
(E)	(E)	エラーを示す(E)
Undeclared name "a"	*	エラーメッセージ用のワイルドカード

この正規表現フォーマットにはいくつかの制限があります。2つの互いに曖昧な文字をすぐ隣に定義することはできません。例えば、"\*\*\*"、"\*\$(FILEDIR)"、または"\$\$(FULLFILE)\*"のような任意の組み合わせは定義できません。

また、ファイル名、ディレクトリ名で使用される":", "\", および"."は、ファイル名、ディレクトリ名のプレースホルダとそれ以外の区切りとして定義することはできません。例えば、"\$\$(FULLFILE):\$(LINE)"、または"\$\$(FILENAME).\*"のような任意の組み合わせは定義できません。

- シンタックスタイプとして"Error"を選択し、出力メッセージがシンタックスの定義と一致する場合  
ビルド実行により出力されるメッセージの左側の余白にエラーアイコン(❗)を表示します。その行をダブルクリックすると、エラーメッセージの該当ソース行をエディタに表示します。
- シンタックスタイプとして"Warning"を選択し、出力メッセージがシンタックスの定義と一致する場合  
ビルド実行により出力されるメッセージの左側の余白にウォーニングアイコン(⚠)を表示します。その行をダブルクリックすると、エラーメッセージの該当ソース行をエディタに表示します。
- シンタックスタイプとして"Other"を選択し、出力メッセージがシンタックスの定義と一致する場合  
ビルド実行により出力されるメッセージの左側の余白にインフォメーションアイコン(ℹ)を表示します。その行をダブルクリックすると、エラーメッセージの該当ソース行をエディタに表示します。

ビルドフェーズの終了時にエラーおよびウォーニング数の合計をアウトプットウィンドウの[Build]タブに表示します。

10. 出力シンタックスの追加は[追加]ボタン、変更は[変更]ボタン、削除は[削除]ボタンを使用してください。
11. 変更後[OK]ボタンをクリックしてください。

**注：**

システムフェーズは、[ツールアドミニストレーション]ダイアログボックスで[環境]のみ変更できます。

### カスタムフェーズを削除するには

1. 削除するフェーズを選択し、[削除]ボタンをクリックしてください。
2. 確認ダイアログボックスが表示された場合は、[はい]ボタンをクリックしてください。

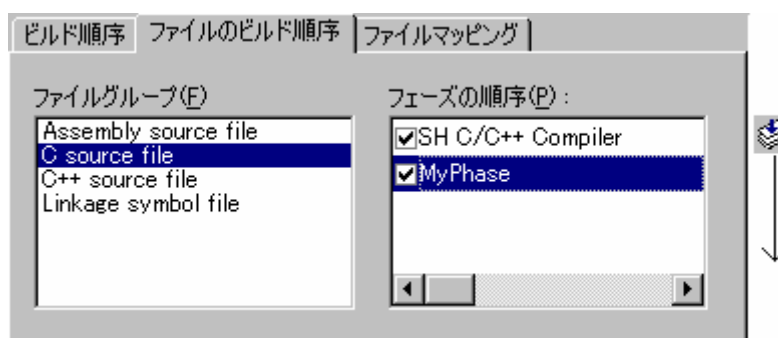
### カスタムフェーズを取り込むには

1. [取り込み]ボタンをクリックしてください。[カスタムフェーズ導入]ダイアログボックスを表示し、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択してください。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[フェーズの導入]ダイアログボックスを開きます。
3. 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強調表示し、[OK]ボタンをクリックしてください。そのフェーズはビルドフェーズ]ダイアログボックスの[ビルドフェーズの順序]リストの最後に追加されます。

## 3.3.2 ファイルのビルド順序タブ

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[ビルド -> コンパイル]を選択するか、[ファイルのコンパイル]ツールバーボタン(📄)をクリックすると、ファイルをコンパイルします。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択し[ビルド -> コンパイル]を実行すると、ファイルをアセンブルします。ファイルグループと実行するフェーズの関係は[ビルドフェーズ]ダイアログボックスの[ファイルのビルド順序]タブで管理されています。

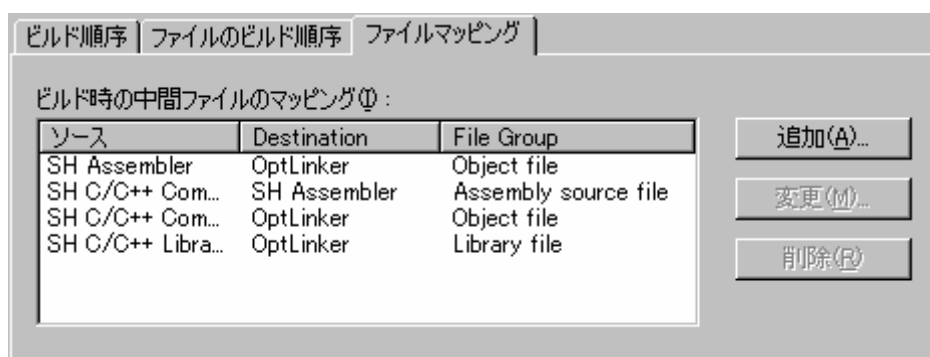
リストには、[ファイルグループ]リストボックスのファイルグループに対してビルドファイル操作を選択したとき実行されるすべての現在のフェーズを表示します。下の図では"C source file"ファイルグループを選択しており、"Compiler"フェーズと"MyPhase"フェーズが関連付けられています。



[ビルド順序]タブのリストに新しいエントリを追加すると、自動的に[ファイルのビルド順序]タブの[フェーズの順序]リストに新しいエントリを追加します。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、"ファイルのビルド"操作を"C source file"に適用する時に実行するフェーズのリストに自動的に追加します。[ビルド -> コンパイル]を選んだときに実行したくないフェーズがある場合、[フェーズの順序]リストのフェーズ名の左にあるチェックボックスをオフにしてください。

### 3.3.3 ファイルマッピングタブ

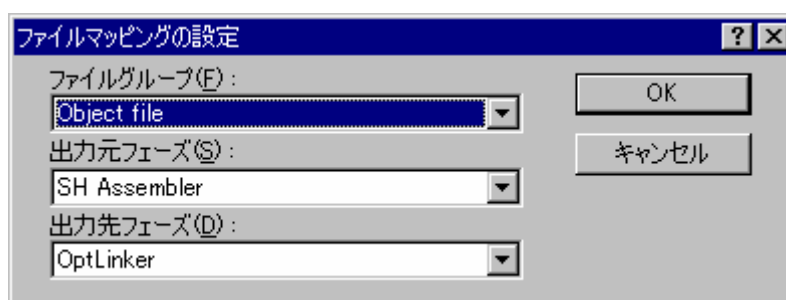
デフォルトでは、フェーズに入力できるファイルはプロジェクトに含まれるものだけです。つまり、[新規ビルドフェーズ]ダイアログボックスの[入力ファイルの選択]ドロップダウンリストに指定した種類のプロジェクトファイルだけです。[ビルド -> ビルドフェーズ]を選択すると[ビルドフェーズ]ダイアログボックスを開きます。もし前のフェーズから出力されたファイル（中間ファイル）をフェーズで使いたい場合は、[ビルドフェーズ]ダイアログボックスの[ファイルマッピング]タブで定義してください。



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します。

#### ファイルのマッピングを追加するには

1. [追加]ボタンをクリックしてください。[ファイルマッピングの設定]ダイアログボックスを開きます。
2. [ファイルグループ]ドロップダウンリストボックスから新しいマッピングのためのファイルグループを選択してください。
3. [出力元フェーズ]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）を選択してください。
4. [出力先フェーズ]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）を選択してください。
5. [OK]ボタンをクリックすると新しいマッピングを追加します。



### 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択してください。
2. [変更]ボタンをクリックしてください。[ファイルマッピングの設定]ダイアログボックスを開きます。
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。

### ファイルマッピングを削除するには

1. 削除するマッピングを選択してください。
2. [削除]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変更が有効になります。

## 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[ビルド]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択してください。起動するダイアログボックスは、選択したカスタムフェーズが単一フェーズか複数フェーズかによって異なります（新規ビルドフェーズウィザードでフェーズが作成されたときのフェーズ種類の選択によります）。

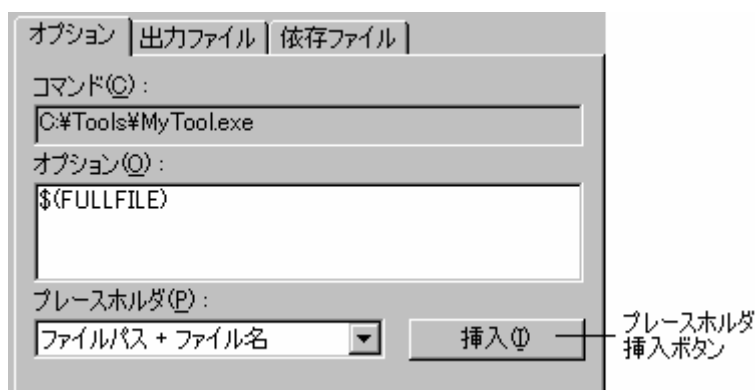


選択したフェーズが複数の場合、ダイアログボックスの左側にはプロジェクトファイルのリストが表示され、ファイル上でビルドオプションを指定できます。選択したフェーズが単一の場合、プロジェクトファイルのリストは表示されません。

右側には3つのオプションタブがあります。選択した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트리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 オプションタブ

[オプション]タブ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。[コマンド]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。[オプション]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[プレースホルダ]ドロップダウンリストボックスから選択し、[挿入]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、「リファレンス 5. プレースホルダ」を参照してください。



注：

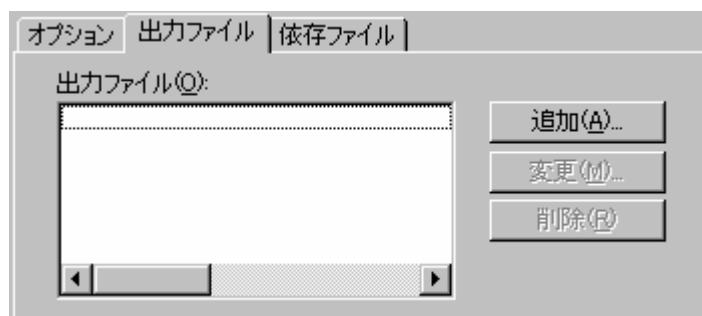
コマンドラインオプションの合計文字数が 256 文字を超える場合は、エラーになります。

### 3.4.2 出力ファイルタブ

[出力ファイル]タブではフェーズで作成される出力ファイルを指定します。High-performance Embedded Workshop 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を実行します。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を実行しません。

注：

出力ファイルを指定しない場合フェーズを常に実行します。



出力ファイルを追加するには

1. [追加]ボタンをクリックしてください。[出力ファイルの追加]ダイアログボックスを開きます。
2. ファイル名を入力するか、[参照]ボタンで選択してください。
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。



### 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択してください。
2. [変更]ボタンをクリックすると[出力ファイルの変更]ダイアログボックスを開きます。
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。変更した項目をリストに追加します。

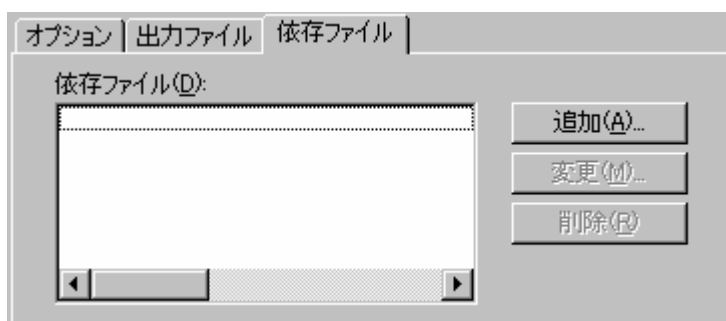
### 出力ファイルを削除するには

1. 削除する出力ファイルを選択してください。
2. [削除]ボタンをクリックしてください。

出力ファイルのパスにはブレースホルダを使用できます。

### 3.4.3 依存ファイルタブ

[依存ファイル]タブではフェーズに必要な依存ファイルを指定します。High-performance Embedded Workshop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を実行します。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しません。



### 依存ファイルを追加するには

1. [追加]ボタンをクリックしてください。[依存ファイルの追加]ダイアログボックスを開きます。
2. ファイル名を入力するか、[参照]ボタンで選択してください。
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。

### 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択してください。
2. [変更]ボタンをクリックすると[依存ファイルの変更]ダイアログボックスを開きます。
3. フィールドを変更して[OK]ボタンをクリックすると、変更した項目をリストに追加します。

### 依存ファイルを削除するには

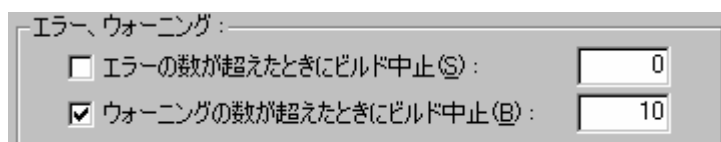
1. 削除するファイルを選択してください。
2. [削除]ボタンをクリックしてください。

依存ファイルのパスにはブレースホルダを使用できます。

### 3.5 ビルドを管理する

ビルドの中止条件の初期状態は「すべてのフェーズを実行し、途中で致命的なエラーが発生したときだけ中止する」です。ビルドの中止条件は[オプション]ダイアログボックスの[ビルド]タブで変更できます。

[基本設定 -> オプション]を選択し[オプション]ダイアログボックスの[ビルド]タブを表示してください。ある一定の回数を超えるエラーが発生した後にビルドを中止したいとき、[エラーの数が超えたときにビルド中止]チェックボックスをオンにして右のフィールドにエラー数を入力してください。指定した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、[ウォーニングの数が超えたときにビルド中止]チェックボックスをオンに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。



[ビルド]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できます。表示するには、それぞれのチェックボックスをオンにしてください。

#### 注：


- 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。
- 依存関係を設定したプロジェクトのビルドでは、子プロジェクトでエラーが発生すると親プロジェクトはビルドを行いません。
- [エラーの数が超えたときにビルド中止]をチェックして数値を指定したとき、次に示す項目に注意してください。
  - i. エラー数が指定した数を超えた場合、ビルドは直ちに停止します。該当フェーズの完了までビルドを続けません。
  - ii. 1回のツールの実行でエラー数が指定した数を超えた場合、指定したエラー数に1を加えた数のエラーメッセージまでをアウトプットウィンドウに表示しますが、それ以上のエラーメッセージは表示されません。また、エラー数が指定した数値を超えた旨のメッセージもアウトプットウィンドウに表示されませんのでご注意ください。

なお、[ウォーニングの数が超えたときにビルド中止]についても[エラーの数が超えたときにビルド中止]と同様です。また、[ウォーニングの数が超えたときにビルド中止]と[エラーの数が超えたときにビルド中止]とは相互の関係はなく、それぞれ独立した機能です。

### 3.6 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できます。

#### ログファイルを指定するには

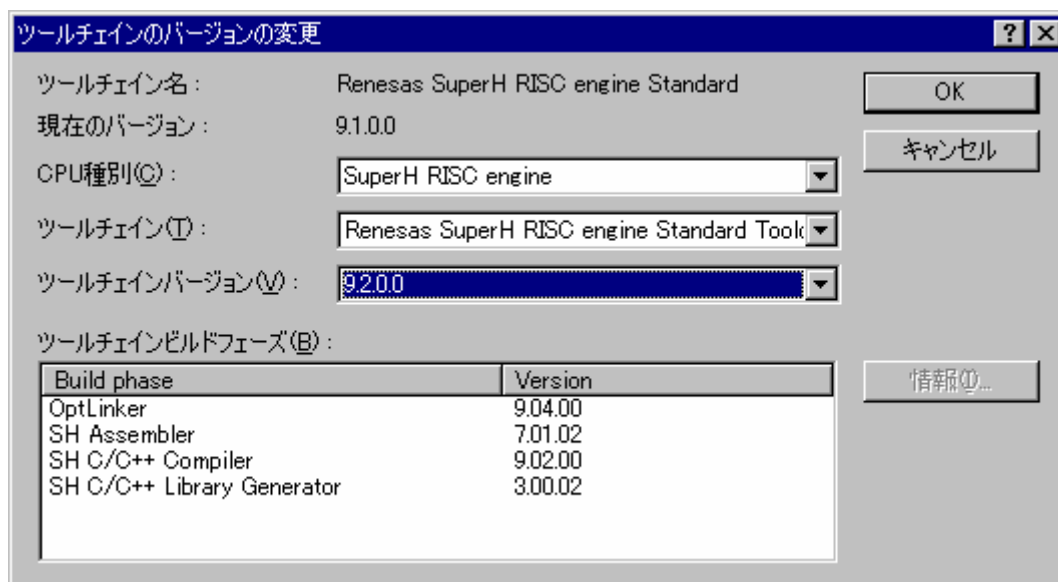
1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [ログ]タブを選択してください。
3. [ビルドログファイル生成]チェックボックスをオンにしてください。
4. [ディレクトリ]フィールドにログファイルの絶対パスを入力するか、[参照]ボタンをクリックしてパスを選択するか、またはプレースホルダボタン(  )をクリックし表示されるポップアップメニューからプレースホルダを選択してください。

5. [OK]ボタンをクリックしてください。



### 3.7 ツールチェーンのバージョンを変更する

同じツールチェーンの2つ以上のバージョンが High-performance Embedded Workshop に登録されているとき [ツールチェーンのバージョンの変更] 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開くには [ツール -> ツールチェーンバージョンを変更] を選択してください。このダイアログボックスの [ツールチェーンバージョン] 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。



ツールチェーンを構成するツールの情報を表示するには [ツールチェーンのバージョンの変更] ダイアログボックスの [ツールチェーンビルドフェーズ] ドロップダウンリストからツールを1つ選択し、[情報]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にそのツールの情報を表示します。このダイアログボックスを閉じるには [閉じる]ボタンをクリックしてください。

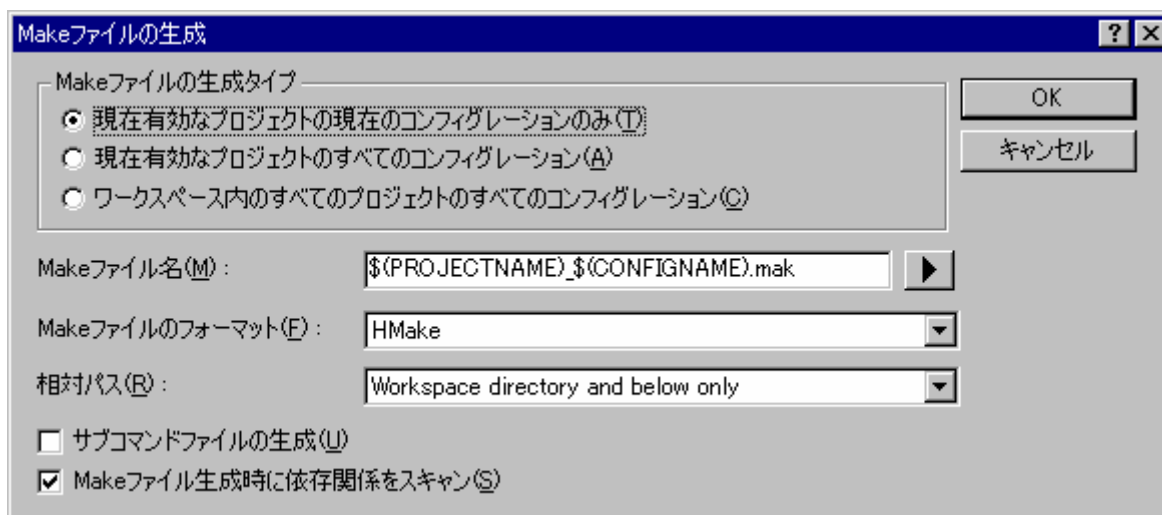
### 3.8 Makeファイルの生成

High-performance Embedded Workshop では、ワークスペースに関連した Make ファイルを生成できます。生成可能な Make ファイルのフォーマットは、HMake、NMake、および GNUMake です。Make ファイルを実行する Make ツールは、DOS コマンドプロンプト上で動作するツールであり、生成した Make ファイルを使用することで、High-performance Embedded Workshop を起動しなくてもプロジェクトをビルドできます。

生成した Make ファイルは、High-performance Embedded Workshop 上で実行できます。この設定は、「3.9 Make ファイルを使ったビルド」を参照してください。

### Make ファイルを生成するには

1. Make ファイルを生成するワークスペースを開いてください。  
(ワークスペースに複数のプロジェクトを登録している場合は、Make ファイルを生成するプロジェクトをアクティブプロジェクトとしてください)
2. Make ファイルを生成するコンフィグレーションを選択してください。
3. [ビルド -> Make ファイルの作成]を選択してください。[Make ファイルの生成]ダイアログボックスが開きます。



4. [Make ファイルの生成タイプ]から生成したい Make ファイルのタイプを選択してください。
  - 現在有効なプロジェクトの現在のコンフィグレーションのみ (デフォルト)
  - 現在有効なプロジェクトのすべてのコンフィグレーション
  - ワークスペース内のすべてのプロジェクトのすべてのコンフィグレーション
5. [Make ファイル名]エディットボックスに Make ファイルのファイル名を入力してください。プレースホルダを挿入する場合は、挿入する箇所にカーソルを移動して、プレースホルダボタン(▶)をクリックし、ポップアップメニューからプレースホルダを選択してください。Make ファイルのファイル名のデフォルト値は以下の通りです。
  - [現在有効なプロジェクトの現在のコンフィグレーションのみ]を選択した場合  
\$(PROJECTNAME)\_\$(CONFIGNAME).mak
  - [現在有効なプロジェクトのすべてのコンフィグレーション]を選択した場合  
\$(PROJECTNAME).mak
  - [ワークスペース内のすべてのプロジェクトのすべてのコンフィグレーション]を選択した場合  
\$(WORKSPNAME).mak
6. [Make ファイルのフォーマット]ドロップダウンリストから生成する Make ファイルのフォーマットを選択してください。生成可能な Make ファイルのフォーマットは、HMake、NMake、および GNUMake です。
7. [相対パス]ドロップダウンリストから Make ファイル内のディレクトリ表現方法を選択してください。デフォルト値は、"Workspace directory and below only"です。

選択可能なディレクトリ表現方法は以下の通りです。

オプション	ワークスペースディレクトリ およびそれ以下のディレクトリ	ワークスペースディレクトリ の外のディレクトリ
None	絶対パス	絶対パス
Workspace directory and below only	相対パス	絶対パス
All	相対パス	相対パス

8. [サブコマンドファイルの生成]チェックボックスは、オンにすると Make ファイルが出力されるディレクトリ内にコマンドファイルを生成します。デフォルトはオフです。

**注：**

以下に示すコンパイラパッケージに含まれるツールチェーンを使用している場合に[Make ファイルのフォーマット]で GNU Make を選択すると、[サブコマンドファイルの生成]チェックボックスはオンに固定されています。

- SuperH ファミリ用 C/C++コンパイラパッケージ
- H8SX, H8S, H8 ファミリ用 C/C++コンパイラパッケージ
- RX ファミリ用 C/C++コンパイラパッケージ

9. [Make ファイルの生成時に依存関係をスキャン]チェックボックスは、オンにすると Make ファイルを生成する前に依存関係の更新を行います。デフォルトはオンです。
10. [OK]ボタンをクリックしてください。

High-performance Embedded Workshop は、ワークスペースディレクトリ内に"make"ディレクトリを作り、その下に Make ファイルを生成します。[Make ファイルのフォーマット]で HMake を選択して作成した Make ファイルは、High-performance Embedded Workshop インストールディレクトリにある HMAKE.EXE で実行できます。

Make ファイルを編集する場合は、「リファレンス 13. HMAKE ユーザガイド」を参照してください。

**注：**

High-performance Embedded Workshop をインストールしたディレクトリに空白がある場合、Make ファイル生成機能で Make ファイルのフォーマットに"GNU Make"を選択すると、"GNU Make"コマンドが正しく動作しないことがあります。

#### Make ファイル (HMake 用) を実行するには

1. DOS コマンドプロンプトウィンドウを開き、Make ファイルが生成されたディレクトリに移動してください。
2. HMAKE を実行してください。コマンドラインは、  
HMAKE.EXE <Make ファイル名>  
です。

**注：**

生成した Make 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。例えば、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絶対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、失敗する可能性があります。一般的に、できるだけプレースホルダを使用して、絶対パス、または特定のパスの使用はなるべく避けてください。

### 3.9 Makeファイルを使ったビルド

High-performance Embedded Workshop は、High-performance Embedded Workshop 標準のビルドのほかに、特定の Make ツール (hmake、nmake、gmake) を使用したビルドを実行できます。

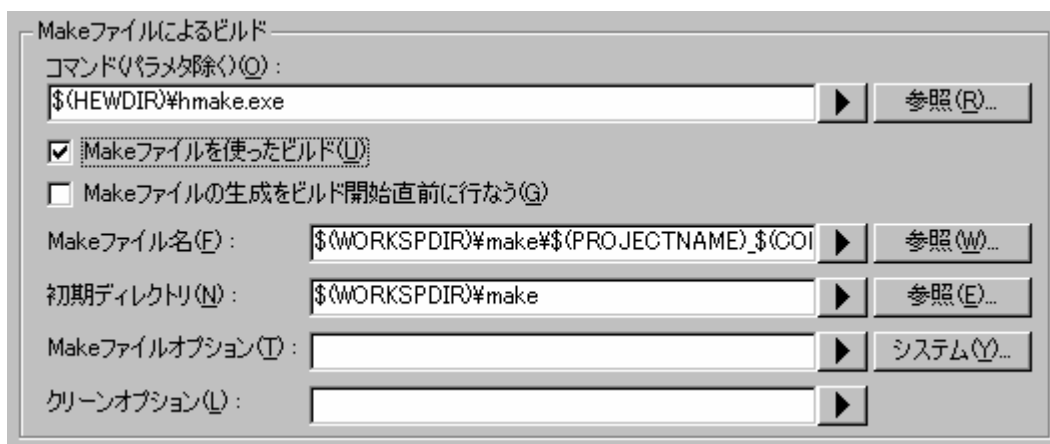
Make ファイルを使ったビルド方法について以下に説明します。

nmake、gmake は、お客様ご自身でご用意ください。

hmake は、High-performance Embedded Workshop インストールディレクトリにインストールされます。

#### Make ファイルの実行制御を設定するには


1. High-performance Embedded Workshop のワークスペースを作成してください。
2. High-performance Embedded Workshop 生成の Make ファイルを使用したい場合、[ビルド -> Make ファイルの生成]を選択し、Make ファイルを生成してください。Make ファイル生成の詳細は、前節の「3.8 Make ファイルの生成」を参照してください。
3. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスが開きます。
4. [ビルド]タブを選択してください。



5. [コマンド (パラメタ除く)]エディットボックスに Make ファイルを実行する Make ツールを入力してください。デフォルトは、hmake ( \$(HEWDIR)\hmake.exe ) です。
6. [Make ファイルを使ったビルド]チェックボックスをオンにしてください。これにより標準のビルドではなく Make ファイルを使ったビルドを実行します。
7. [Make ファイルの生成をビルド開始直前に行なう]チェックボックスは、オンにするとビルド開始直前に毎回 Make ファイルを生成し直します。Make ファイルが、常に最新の High-performance Embedded Workshop プロジェクトの状態に対応するようになります。
8. [Make ファイル名]エディットボックスに Make ファイルを入力してください。デフォルトは、\$(WORKSPDIR)\make\\$(PROJECTNAME)\_\$(CONFIGNAME).mak です。  
**注：** コマンドラインで Make ファイルを指定する必要がない場合、このエディットボックスは空にできます。
9. [初期ディレクトリ]エディットボックスに Make ファイルを実行するときのカレントディレクトリを入力してください。このエディットボックスが空の場合は、\$(WORKSPDIR)をデフォルトとします。
10. [Make ファイルオプション]エディットボックスに Make ファイルのビルド用のオプションを入力してください。
11. [クリーンオプション]エディットボックスに Make ファイルのクリーン用のオプションを入力してください。
12. [OK]ボタンをクリックしてください。



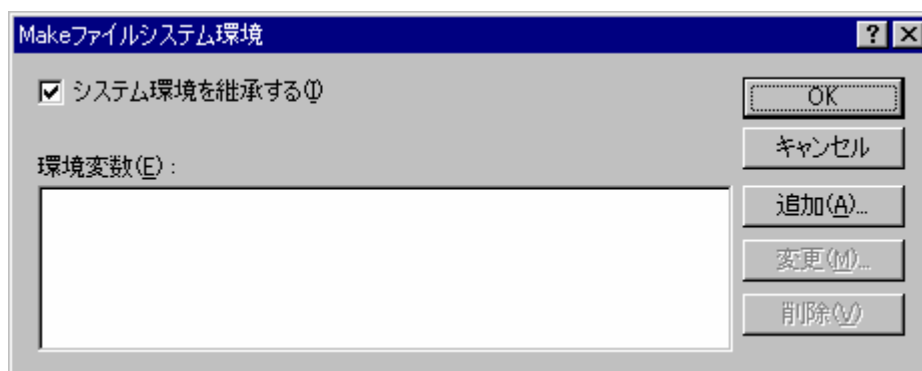
Make ファイルを使ったビルドの設定を行うと、[ビルド]ボタン、メニュー、およびキーボードのショートカットは、Make ファイルの実行制御に切り替わります。[クリーン 全プロジェクト]ボタン、メニュー、およびキーボードショートカットもまた、Make ファイルの実行制御に切り替わります。

標準のビルドと同様に、アウトプットウィンドウの[Build]タブに結果をすべて出力します。High-performance Embedded Workshop がサポートしているビルドフェーズが実行された場合、エラーメッセージ出力行をダブルクリックするとソースファイルにジャンプできます。ヘルプリンクをサポートしている場合もあります。HMAKE.exe を使用している場合に[すべてをビルド]ツールバーボタン(  )をクリックすると、HMAKE へすべての再ビルドを実行するコマンドを渡します。

### Make ファイルのシステム環境を設定するには

デフォルトでは、High-performance Embedded Workshop 内で使用されているシステム環境で Make ファイルを実行します。これを変更したい場合、下記のように環境を構成できます。

1. [ビルド]タブで、[Make ファイルオプション]の右側の[システム]ボタンをクリックしてください。[Make ファイルシステム環境]ダイアログボックスが開きます。



2. [システム環境を継承する]チェックボックスは、システム環境の継承を切り替える場合に使用します。チェックがオンの場合、Make ファイルを標準の環境で実行します（デフォルト）。チェックがオフの場合、Make ファイルを新規の環境で実行します。  
注：「SystemRoot」変数は、Windows® 上で動作するプログラムの許可に必要なため、新規の環境に継承されます。
3. [環境変数]などのコントロールを使用し新たな環境変数を設定できます。ここで設定された値は、現在のシステム環境の値より優先します。
4. 変更を保存するために、OK をクリックし[オプション]ダイアログボックスに戻ってください。このダイアログボックスの OK ボタンがクリックされるまで値はワークスペースに渡されません。

### 3.10 リンク順序をカスタマイズする

High-performance Embedded Workshop では、通常、モジュールはファイル名のアルファベット順にリンクしますが、リンクする順序をカスタマイズできます。

#### リンク順序をカスタマイズするには

1. [ビルド -> リンク順の指定]を選択してください。[リンク順序のカスタマイズ]ダイアログボックスが開きます。
2. [リンク順序のカスタマイズの使用]チェックボックスをオンにします。



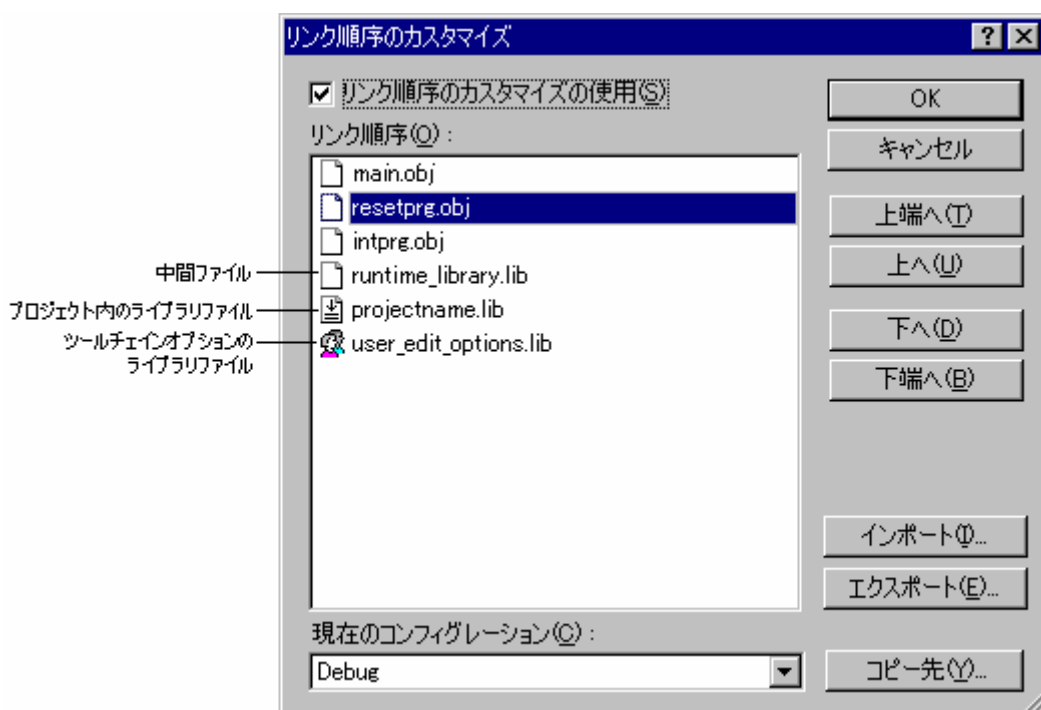
- [リンク順序]リストボックスで、モジュールをリンクする際の順序を設定します。各モジュールはその属性によって異なるアイコンを表示しています。これを下図に示します。

**注：**

以下に示すコンパイラパッケージに含まれるツールチェーンを使用している場合に[リンク順序]リストボックスにオブジェクトファイルと一緒にライブラリファイルが表示されます。

- SuperH ファミリ用 C/C++コンパイラパッケージ
- H8SX, H8S, H8 ファミリ用 C/C++コンパイラパッケージ
- RX ファミリ用 C/C++コンパイラパッケージ

リンク順序に関係するのはオブジェクトファイルのみです。ライブラリファイルの順序は、未定義シンボルを検索する場合に使用されます。

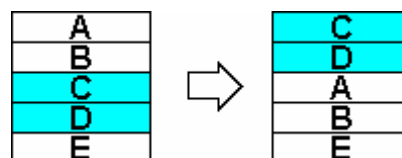


リスト項目は並び替えができます。以下にその方法を示します。

- 1つまたは連続した複数のモジュールを選択して移動させる

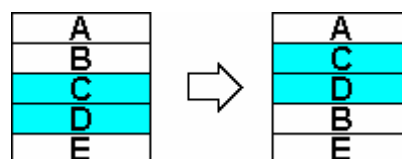
[上端へ]ボタン

選択したモジュールをすべて先頭に移動します。



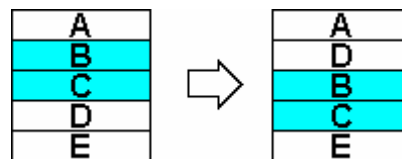
[上へ]ボタン

選択したモジュールと1つ上のモジュールを入れ替えます。



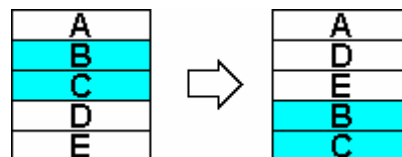
[下へ]ボタン

選択したモジュールと1つ下のモジュールを入れ替えます。



[下端へ]ボタン

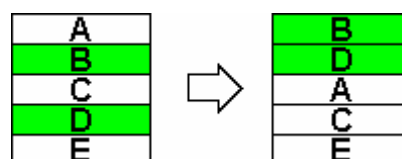
選択したモジュールをすべて最後に移動します。



### • 連続しない複数のモジュールを選択して移動させる

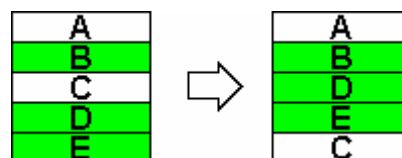
[上端へ]ボタン

選択したモジュールをすべて先頭に移動します。



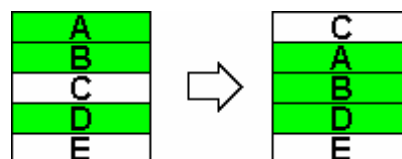
[上にグループ化]ボタン

選択したモジュールの中で先頭のモジュールの位置に残りの選択モジュールを集めます。



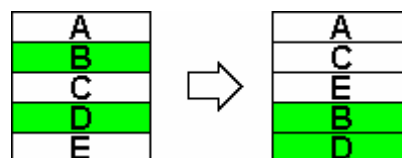
[下にグループ化]ボタン

選択したモジュールの中で最後のモジュールの位置に残りの選択モジュールを集めます。



[下端へ]ボタン

選択したモジュールをすべて最後に移動します。



上端のモジュールを選択した場合、[上端へ]ボタン、[上へ]ボタンは選択できません。また、下端のモジュールを選択した場合、[下端へ]ボタン、[下へ]ボタンは選択できません。

現在のリンク順序の設定をテキスト形式で保存できます。保存したファイルでリンク順序を編集し、再び読み込めば、ファイルの情報のリンク順序を設定します。

### リンク順序をテキストファイルから読み込むには

1. [ビルド -> リンク順の指定]を選択してください。[リンク順序のカスタマイズ]ダイアログボックスを開きます。
2. [リンク順序のカスタマイズの使用]チェックボックスをオンにしてください。
3. [インポート]ボタンをクリックすると[リンク順序のインポート]ダイアログボックスを開きます。
4. ファイルを指定し、[選択]ボタンをクリックしてください。[リンク順序のインポートレポート]ダイアログボックスを開きます。

5. ファイルに保存されているリンク順序を読み込む場合は、[続行]ボタンをクリックしてください。新しいリンク順序でリストを更新します。

**注：**

現在のリンク順序の設定が上書きされるので注意してください。

6. 現在のリンク順序を変更しない場合は、[キャンセル]ボタンをクリックしてください。

**リンク順序をテキストファイルに保存するには**

1. [ビルド -> リンク順の指定]を選択してください。[リンク順序のカスタマイズ]ダイアログボックスを開きます。
2. [リンク順序のカスタマイズの使用]チェックボックスをオンにしてください。
3. [エクスポート]ボタンをクリックすると[リンク順序のエクスポート]ダイアログボックスを開きます。
4. ディレクトリ、ファイル名を指定し、[保存]ボタンをクリックしてください。現在のリストボックスの内容を指定したファイルに保存します。

**例：**

```
# Exported linkage order for project "Project_name" - configuration "Debug"
c:\workspace\workspace_name\project_name\debug\main.obj
c:\workspace\workspace_name\project_name\debug\resetprg.obj
c:\workspace\workspace_name\project_name\debug\intprg.obj
c:\workspace\workspace_name\project_name\debug\runtime_library.lib
c:\workspace\workspace_name\project_name\debug\projectname.lib
c:\workspace\workspace_name\project_name\debug\user_edit_options.lib
```

エディタなどでファイルを開き、リンク順序を編集後、テキスト形式で保存してください。

「リンク順序をテキストファイルから読み込むには」を実行すると、編集したリンク順序を読み込みます。

複数のコンフィグレーションを使用する場合、リンク順序を同じにする場合があります。現在のリンク順序の設定を他のコンフィグレーションにコピーする方法を以下に説明します。

**他のコンフィグレーションにリンク順序をコピーするには**

1. [ビルド -> リンク順の指定]を選択してください。[リンク順序のカスタマイズ]ダイアログボックスを開きます。
2. [リンク順序のカスタマイズの使用]チェックボックスをオンにしてください。
3. [現在のコンフィグレーション]ドロップダウンリストで、コピー元のコンフィグレーションを選択してください。現在のコンフィグレーションをデフォルトで表示します。
4. [コピー先]ボタンをクリックすると[コンフィグレーションのコピー先選択]ダイアログボックスを開きます。現在のリンク順序を現在のプロジェクトのどのコンフィグレーションにコピーするかを指定します。
5. コンフィグレーションを選択し、[OK]ボタンをクリックしてください。

## 4. エディタ

この章では High-performance Embedded Workshop が提供するエディタの使い方を説明します。

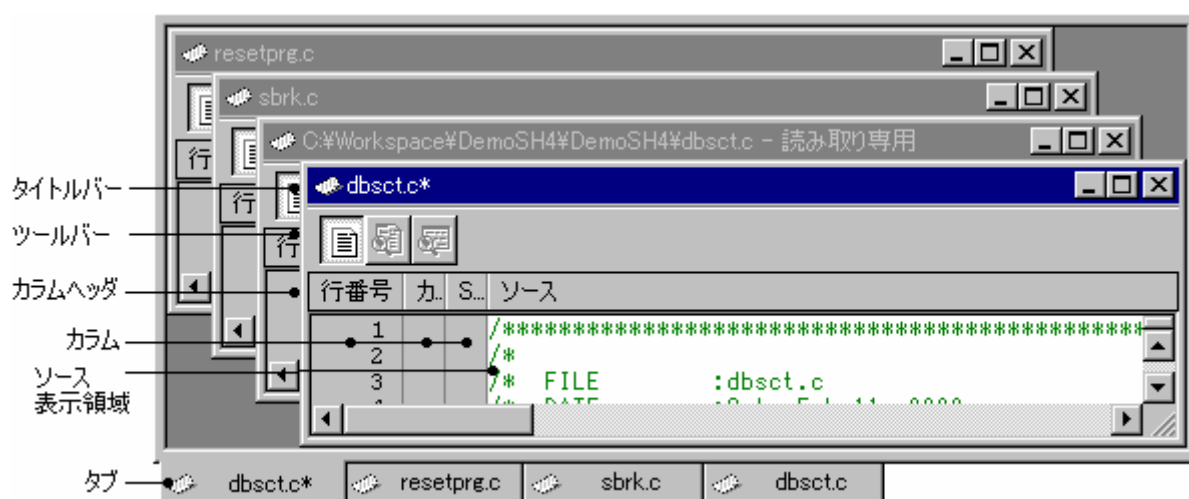
### 4.1 エディタウィンドウ

エディタウィンドウには、表示中や編集中のファイルがあります。

アクティブウィンドウのタイトルバーは他のウィンドウとは色が異なります。(図では"dbstc.c"がアクティブウィンドウです。)文字入力やテキスト貼り付けなどのテキスト操作はアクティブウィンドウでのみ行うことができます。

アクティブウィンドウから他のソースファイルのウィンドウへ切り替えるには(他のウィンドウをアクティブウィンドウにするには)、以下の方法があります。

- そのウィンドウが表示されていればそれをクリックする。
- "Ctrl+Tab"キーまたは"Ctrl+Shift+Tab"キーを押して順次ウィンドウをアクティブにする。
- [ウィンドウ]メニューからウィンドウ名を選択する。
- エディタウィンドウの下の該当するタブを選択する。



**タイトルバー** ファイルの情報(ファイル名、編集中的のアスタリスク(\*)、および読み取り専用属性)を表示します。

アクティブプロジェクトのディレクトリ下に含まれているファイルの場合は、エディタウィンドウのタイトルバーにファイル名を表示します。アクティブプロジェクトのディレクトリ下に含まれていないファイルの場合は、エディタウィンドウのタイトルバーに絶対パスを含むファイル名を表示します。

**ツールバー** ツールバーは、デバッガに接続されている場合のみ表示します。各表示モードが有効である場合、ツールバーのボタンをクリックすると表示モードを切り替えます。各ボタン上にマウスポインタを置くとボタン名をポップアップ表示します。

**カラムヘッダ** カラムヘッダ上のマウスのドラッグで各カラムの幅を調節できます。また、カラムヘッダ上で右クリックするとポップアップメニューを表示します。カラムが利用できる場合は、エントリの横にはチェックマークがあります。エントリをクリックすると、各カラムの表示、非表示を切り替えます。

各カラムヘッダ上にマウスポインタを置くと、カラム名をポップアップ表示します。カラムヘッダ自体の表示、非表示を切り替えることもできます。

カラム	ソースカラムの左側にはいくつかの余白（カラム）があります。 各カラム上にマウスポインタを置くと、カラム名をポップアップ表示します。
ソースカラム	シンタックスを強調表示したコードを含みます。エディタの1行の最大文字数（半角）は、8,192文字です。 この領域内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。
タブ	ファイルの情報（ファイル名、および編集中的のアスタリスク（*））を表示します。 各タブ上にマウスポインタを置くと、絶対パスを含むファイル名をポップアップ表示します。 この領域内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

## 4.2 複数のファイルを扱う

プロジェクトのファイル进行操作するときにはエディタウィンドウを使います。エディタによって、一度に複数のファイルを開いたり、ファイル間の切り替えをしたり、異なる構成に並べ替えたり、任意の順序に編集できます。ウィンドウ上での操作は Windows®のアプリケーションの一般的な機能であり、以下の[ウィンドウ]メニューから選択できます。

メニューオプション	動作
[ウィンドウ -> 重ねて表示]	開いているすべてのウィンドウの左上が見えるように重ねて並べます。
[ウィンドウ -> 上下に並べて表示]	開いているすべてのウィンドウが互いに重ならずエディタウィンドウの全体を占めるように並べます（垂直方向）。
[ウィンドウ -> 左右に並べて表示]	開いているすべてのウィンドウが互いに重ならずエディタウィンドウの全体を占めるように並べます（水平方向）。
[ウィンドウ -> アイコンの整列]	すべての最小化したウィンドウをエディタウィンドウの下に並べます。
[ウィンドウ -> すべて閉じる]	開いているエディタウィンドウをすべて閉じます。

また、エディタのファイルはブック形式で表示できます。つまり、各ファイルにタブが付いていてファイル間の行き来を容易にできます。

### ファイルをブック形式で表示するには

ブック形式でファイルを表示 (H)


1. [基本設定 -> オプション] を選択してください。[オプション]ダイアログボックスを開きます。
2. [エディタ]タブを選択してください。
3. [ブック形式でファイルを表示]チェックボックスをオンにしてください。
4. [OK]ボタンをクリックしてください。

## 4.3 標準のファイル操作

### 4.3.1 新規ファイルを作成する

#### 新しいエディタウィンドウを作成するには

以下のいずれかの操作を選択してください。




- [新規ファイル]ツールバーボタン()をクリックする。
- "Ctrl+N"キーを押す。
- [ファイル -> 新規作成]を選択する。

新しいウィンドウにはデフォルトで名前が付きます。ファイルを保存するときに名前を変更できます。

### 4.3.2 ファイルを編集する

High-performance Embedded Workshop エディタは標準的な編集機能をサポートします。通常の方法（メニュー、ツールバー、キーボードのショートカット）で編集できるほか、各エディタウィンドウにあるポップアップメニューでも編集できます。ポップアップメニューを表示するには、開いたウィンドウ上で右クリックしてください。

以下の表に編集の基本操作を示します。

名称	機能	操作
元に戻す	最新の編集操作を取り消す。	[編集 -> 元に戻す]を選択する。 Ctrl+Z キーを押す。
やり直し	最新の取り消した編集操作をやり直す。	[編集 -> やり直し]を選択する。 Ctrl+Y キーを押す。
切り取り	選択部分を削除し Windows® クリップボードに貼り付ける。	[切り取り]ツールバーボタン(  )をクリックする。 Ctrl+X キーを押す。 [編集 -> 切り取り]を選択する。 [切り取り] - ポップアップメニューを選択する。
コピー	選択部分をコピーし Windows® クリップボードに貼り付ける。	[コピー]ツールバーボタン(  )をクリックする。 Ctrl+C キーを押す。 [編集 -> コピー]を選択する。 [コピー] - ポップアップメニューを選択する。
貼り付け	Windows® クリップボードの内容をコピーしてアクティブウィンドウのカーソル位置に貼り付ける。	[貼り付け]ツールバーボタン(  )をクリックする。 Ctrl+V キーを押す。 [編集 -> 貼り付け]を選択する。 [貼り付け] - ポップアップメニューを選択する。
削除	選択部分を削除する (Windows® クリップボードに貼り付けない)。	[編集 -> 削除]を選択する。 Delete キーを押す。
すべて選択	アクティブウィンドウの内容すべてを選択する。	[編集 -> すべて選択]を選択する。 Ctrl+A キーを押す。

ファイルを編集すると、エディタウィンドウのタイトルバーにアスタリスク (\*) を表示します。(例: filename.c\*)


アスタリスクはファイルを保存するまで表示します。現在のウィンドウですべての編集が戻された場合には、アスタリスクの表示が消えます。

### エディタでテキストを選択する

すべてのエディタと同じ方法でテキストを選択できます。ただし、列を選択するには、Alt キーを押した状態でマウスによりテキストを選択してください。

### 4.3.3 ファイルを保存する

#### エディタウィンドウの内容を保存するには


1. 内容を保存するウィンドウがアクティブであることを確認してください。
2. 以下のいずれかの操作を選択してください。
  - [ファイルの保存]ツールバーボタン()をクリックする。
  - "Ctrl+S"キーを押す。
  - [ファイル -> 上書き保存]を選択する。

3. ファイルを新規に保存する場合、名前を付けて保存するダイアログボックスを開きます。ファイル名とディレクトリを指定して、[保存]ボタンをクリックしてください。
4. 既に保存したことのあるファイルの場合は、ダイアログボックスを表示せずにファイルを更新します。

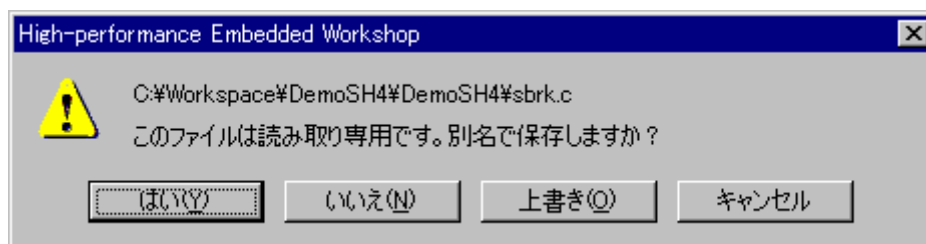
#### エディタウィンドウの内容を新しい名前で保存するには

1. 内容を保存するウィンドウがアクティブであることを確認してください。
2. [ファイル->名前を付けて保存]を選択してください。
3. 名前を付けて保存するダイアログボックスを開きます。ファイル名とディレクトリを指定して[保存]ボタンをクリックしてください。

#### すべての開いているエディタウィンドウの内容を保存するには

1. 以下のいずれかの操作を選択してください。
  - [すべて保存]ツールバーボタン()をクリックする。
  - [ファイル->すべて保存]を選択する。
2. 新規にファイルを保存する場合、名前を付けて保存するためのダイアログボックスを開きます。ファイル名とディレクトリを指定してください。既に保存したことのあるファイルの場合は、ダイアログボックスを表示せずにファイルを更新します。


ファイルが読み取り専用属性の場合にファイルを保存しようとする、別名で保存するか否かを確認するダイアログボックスが開きます。



- 「はい」ボタンをクリックするとファイルを保存するダイアログボックスが開きます。ファイル名を変更して保存してください。
- 「いいえ」ボタンをクリックするとファイルを保存しません。
- 「上書き」ボタンをクリックするとファイルを上書き保存します。ファイルは読み取り専用ではありません。
- 「キャンセル」ボタンをクリックするとファイルを保存しません。

#### 4.3.4 ファイルを開く

##### [ファイルを開く]ダイアログボックスでファイルを開くには

1. 以下のいずれかの操作を選択し[ファイルを開く]ダイアログボックスを開きます。
  - [ファイルを開く]ツールバーボタン()をクリックする。
  - "Ctrl+O"キーを押す。
  - [ファイル->開く]を選択する。



2. 右のディレクトリブラウザを使って、開きたいファイルのあるディレクトリに移動してください。  
[ファイルの種類]コンボボックスでファイルの種類を選択してください。("All Files" (\*.\*) を指定すると、そのディレクトリのすべてのファイルを表示します。)
3. ファイルを選択し[開く]ボタンをクリックしてください。

High-performance Embedded Workshop のメインウィンドウ上に Windows® Explorer からファイルをドラッグアンドドロップして開きます。

また、ワークスペースウィンドウの[Projects]タブのソースファイルは、以下のいずれかの方法で個々のファイルが開けます。

- ファイルをダブルクリックする。
- ファイルを選択し、右クリックしてポップアップメニューから[開く <ファイル名>]を選択する。
- ファイルが選択されフォーカスがある場合に Enter キーを押す。

ファイルは、High-performance Embedded Workshop エディタ以外のエディタ（外部エディタと呼びます）で開くこともできます。詳細は、「6.9 外部エディタを使う」を参照してください。

最近開いたファイルを[ファイル]メニューの[最近使ったファイル]サブメニューに追加します。この機能は最近開いたファイルを再び開きたいときに便利です。

#### 最近利用したファイルを開くには

[ファイル->最近使ったファイル]を選択し、このサブメニューから開きたいファイルを選択してください。

### 4.3.5 ファイルを閉じる

#### 個別にファイルを閉じるには

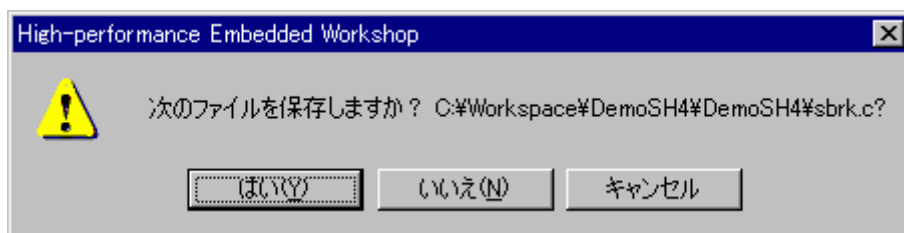
以下のいずれかの操作を選択してください。

- エディタウィンドウのシステムメニュー（[最大化]表示されていないときの各ウィンドウの左上）をダブルクリックする。
- エディタウィンドウのシステムメニュー（[最大化]表示されていないときの各ウィンドウの左上）をクリックして[閉じる]を選択する。
- アクティブウィンドウであることを確認後"Ctrl+F4"キーを押す。
- アクティブウィンドウであることを確認後[ファイル->閉じる]を選択する。
- "閉じる"ボタン（[最大化]表示されていないときの各ウィンドウの右上）をクリックする。

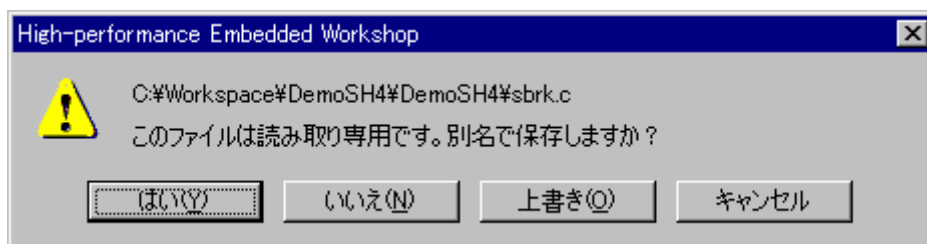
#### すべてのファイルを閉じるには

[ウィンドウ->すべて閉じる]を選択してください。

ファイルを編集し保存していない場合にファイルを閉じようとする、保存するか否かを確認するダイアログボックスが開きます。



- 「いいえ」ボタンをクリックするとファイルを保存せずにファイルを閉じます。
- 「キャンセル」ボタンをクリックするとエディタウィンドウに戻ります。
- 「はい」ボタンをクリックするとファイルを上書き保存しファイルを閉じます。ファイルが読み取り専用属性の場合に「はい」ボタンをクリックすると、別名で保存するか否かを確認するダイアログボックスが開きます。



- 「はい」ボタンをクリックするとファイルを保存するダイアログボックスが開きます。ファイル名を変更して保存してください。
- 「いいえ」ボタンをクリックするとファイルを保存せずにファイルを閉じます。
- 「上書き」ボタンをクリックするとファイルを上書き保存しファイルを閉じます。ファイルは読み取り専用ではなくなります。
- 「キャンセル」ボタンをクリックするとエディタウィンドウに戻ります。

#### 4.3.6 ウィンドウを閉じるポップアップメニュー



High-performance Embedded Workshop のエディタウィンドウでは、タブ上のポップアップメニューにより編集ファイルを上書き保存したり、ウィンドウを閉じたりできます。

エディタウィンドウのタブ上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	機能
上書き保存	編集中のウィンドウの内容を保存します。
閉じる	ファイルを閉じます。

逆アセンブリウィンドウでは、タブ上のポップアップメニューによりウィンドウを閉じることが可能です。


逆アセンブリウィンドウのタブ上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	機能
閉じる	逆アセンブリウィンドウを閉じます。

## 4.4 検索とファイル内の移動

### 4.4.1 文字列を検索する

現在のファイルで文字列を検索するには

1. 内容を検索するウィンドウがアクティブウィンドウであることを確認してください。
2. 検索開始位置にカーソルを置いてください。
3. 以下のいずれかの操作を選択し[検索]ダイアログボックスを開きます。
  - [検索]ツールバーボタン(  )をクリックする。
  - "Ctrl+F"キーを押す。
  - [編集 -> 検索]を選択する。
  - エディタウィンドウのポップアップメニューから[検索]を選択する。




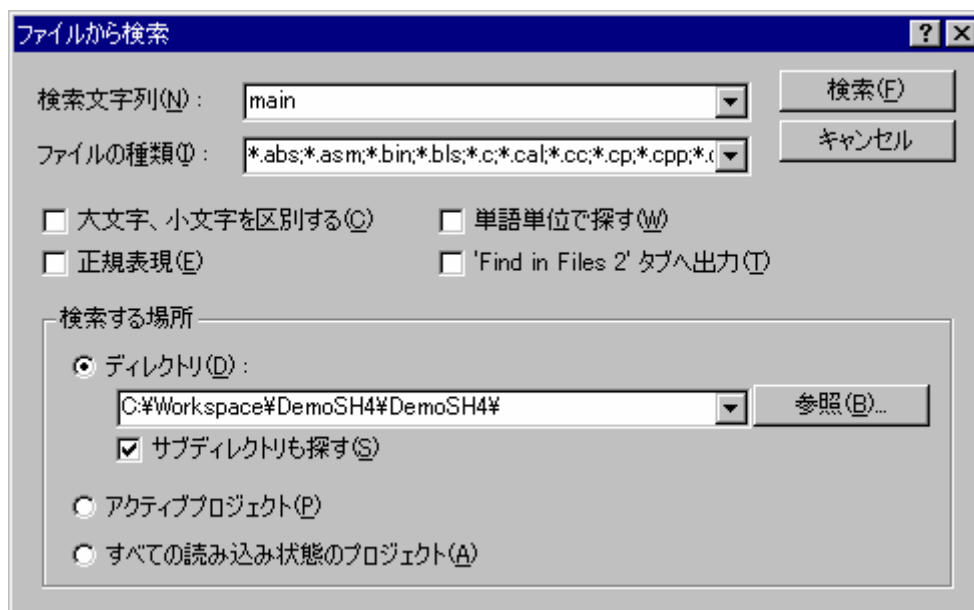
4. [検索文字列]に検索する文字列を入力するか、またはドロップダウンリストボックスから以前検索した文字列を選択してください。文字列を選択してから[検索]ダイアログボックスを開くと、その文字列を自動的に[検索文字列]に表示します。
5. 単語単位で検索する場合は、[単語単位で探す]チェックボックスをオンにしてください。このオプションを選択しない場合は、一部でも一致する文字列を検索します。
6. 検索文字列の大文字と小文字を区別する場合は、[大文字、小文字を区別する]チェックボックスをオンにしてください。
7. 正規表現を使って文字列を検索する場合は、[正規表現]チェックボックスをオンにしてください。詳細は「リファレンス 4. 正規表現」を参照してください。
8. [検索する方向]ラジオボタンにより、検索する方向を指定できます。[下へ]を選択すると、カーソル位置からファイルの下の方向に検索します。[上へ]を選択すると、カーソル位置から上の方向に検索します。
9. [次を検索]ボタンをクリックすると検索を始めます。

また、複数のファイルで文字列を検索できます。

#### 4.4.2 複数のファイルで文字列を検索する

##### 複数のファイルで文字列を検索するには

- 以下のいずれかの操作を選択し[ファイルから検索]ダイアログボックスを開きます。
  - [ファイルの中から検索]ツールバーボタン(  )をクリックする。
  - "F4"キーを押す。
  - [編集 -> ファイルから検索]を選択する。



- [検索文字列]に検索する文字列を入力するか、またはドロップダウンリストボックスから以前検索した文字列を選択してください。文字列を選択してから[ファイルから検索]ダイアログボックスを開くと、その文字列を自動的に[検索文字列]に表示します。
- [ファイルの種類]に検索するファイルの拡張子を指定してください。複数の拡張子を指定するときは、セミコロンで区切ってください(例: \*.c;\*.h)。
- 検索文字列の大文字と小文字を区別する場合は、[大文字、小文字を区別する]チェックボックスをオンにしてください。
- 単語単位で検索する場合(一部でも一致する文字列ではありません)は、[単語単位で探す]チェックボックスをオンにしてください。  
注: [単語単位で探す]と[正規表現]は同時にオンとするような検索機能はサポートしていません。
- 正規表現を使って文字列を検索する場合は、[正規表現]チェックボックスをオンにしてください。詳細は「リファレンス 4. 正規表現」を参照してください。
- アウトプットウィンドウの[Find in Files 1]タブではなく[Find in Files 2]タブに検索結果を表示する場合は、[Find in Files 2]タブへ出力]チェックボックスをチェックしてください(アウトプットウィンドウで2つの異なる検索結果を得られます)。
- 検索する場所の種類を選択してください。
  - [ディレクトリ]  
このオプションを選択すると、検索文字列を探すためにディレクトリ(ディレクトリ構成)を検索します。[サブディレクトリも探す]チェックボックスをオンにすると、指定ディレクトリのすべてのサ

ブディレクトリを検索します。[ディレクトリ]で指定したディレクトリのみを検索する場合は、このチェックボックスがオフであることを確認してください。

- [アクティブプロジェクト]  
このオプションを選択すると、ワークスペースのアクティブプロジェクトに登録されているファイルを検索します。
- [すべての読み込み状態のプロジェクト]  
このオプションを選択すると、ワークスペースのすべての読み込み状態のプロジェクトに登録されているファイルを検索します。

[アクティブプロジェクト]オプションおよび[すべての読み込み状態のプロジェクト]オプションは、プロジェクト依存ファイル（インクルードファイル）を検索しないので注意してください。

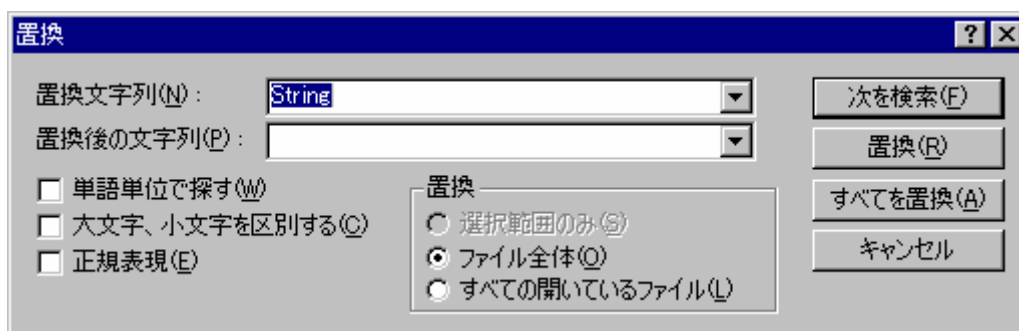
9. [検索]ボタンをクリックすると検索を始めます。検索結果はアウトプットウィンドウの[Find in Files 1]タブまたは[Find in Files 2]タブに表示します。検索結果をダブルクリックすると、ファイルの該当文字列にジャンプします。実行中の[ファイル内から検索]操作を止めるには、[編集 -> ファイルから検索を中止]を選択します。ファイル内から検索操作が完了したらアウトプットウィンドウの文字列をダブルクリックすると、ファイルの該当文字列にジャンプできます。

#### 4.4.3 文字列を置換する

文字列の置換は前節で説明した文字列の検索に似ています。異なる点は、文字列検索後、他の文字列に置き換える点です。

##### ファイルの文字列を置換するには

1. 内容を置換するウィンドウがアクティブであることを確認してください。
2. 検索を始める位置にカーソルを置いてください。
3. 以下のいずれかの操作を選択し[置換]ダイアログボックスを開きます。
  - "Ctrl+H"キーを押す。
  - [編集 -> 置換]を選択する。
  - エディタウィンドウのポップアップメニューから[置換]を選択する。



4. [置換文字列]に置換前の文字列を入力するか、またはドロップダウンリストボックスから以前検索した文字列を選択してください。文字列を選択してから[置換]ダイアログボックスを開くと、その文字列を自動的に[置換文字列]に表示します。
5. [置換後の文字列]に置換後の文字列を入力するか、またはドロップダウンリストボックスから以前検索した文字列を選択してください。
6. 単語単位で検索する場合は、[単語単位で探す]チェックボックスをオンにしてください。このオプションを選択しない場合は、一部でも一致する文字列を検索します。

7. 置換文字列の大文字と小文字を区別する場合は、[大文字、小文字を区別する]チェックボックスをオンにしてください。
8. 正規表現を使って文字列を検索する場合は、[正規表現]チェックボックスをオンにしてください。詳細は「リファレンス 4. 正規表現」を参照してください。
9. [次を検索]をクリックすると、検索文字列、条件に合致したものの次を検索します。置換したい場合は[置換]をクリックしてください。[すべてを置換]をクリックすると、該当するすべての文字列を置換します。[キャンセル]をクリックすると、置換動作を停止します。なお、[置換]で[選択範囲のみ]を選択している場合は文字列の選択範囲が置換対象となり、[ファイル全体]を選択している場合はファイル全体が置換対象となります。[すべての開いているファイル]を選択している場合は、エディタで現在開いているファイルがすべて置換対象となります。

#### 4.4.4 指定した行にジャンプする

##### ファイルの指定した行にジャンプするには

1. ファイルがアクティブであることを確認してください。
2. 以下のいずれかの操作を選択し[ジャンプ]ダイアログボックスを開きます。
  - "Ctrl+G"キーを押す。
  - [編集 -> ジャンプ]を選択する。
  - エディタウィンドウのポップアップメニューから [ジャンプ]を選択する。



3. 指定する行番号を入力して、[OK] ボタンをクリックしてください。カーソルが指定した行の先頭に移ります。


#### 4.5 ブックマーク

一度に多くの大容量のファイルを扱うとき、必要な行や領域を見つけるのが難しくなります。エディタでブックマークをあらかじめ特定の行に設定しておくことで、後でその行にジャンプできます。例えば、行数の多いC言語ファイルの各関数の定義位置にブックマークを設定すると便利です。ブックマークは、解除するまでか、設定したファイルを閉じるまで有効です。


##### ブックマークを設定するには

1. ブックマークを設定する行にカーソルを置いてください。
2. 以下のいずれかの操作を選択してください。
  - [ブックマークの挿入/削除]ツールバーボタン(🔖)をクリックする。
  - "Ctrl+F2"キーを押す。
  - [編集 -> ブックマーク -> ブックマークの挿入/削除]を選択する。
  - ポップアップメニューから [ブックマーク -> ブックマークの挿入/削除]を選択する。
3. その行にブックマークアイコン(🔖)を表示します。


### ブックマークを解除するには

1. ブックマークを解除する行にカーソルを置いてください。
2. 以下のいずれかの操作を選択してください。
  - [ブックマークの挿入/削除]ツールバーボタン()をクリックする。
  - "Ctrl+F2"キーを押す。
  - [編集 -> ブックマーク -> ブックマークの挿入/削除]を選択する。
  - ポップアップメニューから[ブックマーク -> ブックマークの挿入/削除]を選択する。
3. その行からブックマークアイコンが削除されます。


### 次のブックマークにジャンプするには

1. カーソルが検索するファイルの中にあることを確認してください。
2. 以下のいずれかの操作を選択してください。
  - [次のブックマークへ]ツールバーボタン()をクリックする。
  - "F2"キーを押す。
  - [編集 -> ブックマーク -> 次のブックマーク]を選択する。
  - ポップアップメニューから[ブックマーク -> 次のブックマーク]を選択する。

### 同じファイルの1つ前のブックマークに戻るには


1. カーソルが検索するファイルの中にあることを確認してください。
2. 以下のいずれかの操作を選択してください。
  - [前のブックマークへ]ツールバーボタン()をクリックする。
  - "Shift+F2"キーを押す。
  - [編集 -> ブックマーク -> 前のブックマーク]を選択する。
  - ポップアップメニューから[ブックマーク -> 前のブックマーク]を選択する。

### ファイル内のすべてのブックマークを解除するには

1. すべてのブックマークを解除するファイルのウィンドウをアクティブにしてください。
2. 以下のいずれかの操作を選択してください。
  - [すべてのブックマークの削除]ツールバーボタン()をクリックする。
  - [編集 -> ブックマーク -> すべてのブックマークの削除]を選択する。
  - ポップアップメニューから[ブックマーク -> すべてのブックマークの削除]を選択する。

## 4.6 ファイルを印刷する

### ファイルを印刷するには

1. 印刷するファイルのウィンドウをアクティブにしてください。
2. 以下のいずれかの操作を選択してください。
  - [印刷]ツールバーボタン()をクリックする。
  - "Ctrl+P"キーを押す。



- [ファイル->印刷]を選択する。

## 4.7 テキストのレイアウト

### 4.7.1 ページ設定

High-performance Embedded Workshop エディタからファイルを印刷するとき、印刷ダイアログボックスの設定により、印刷方法が変わります（例えば、片面印刷、両面印刷など）。また、[ページ設定]ダイアログボックスでは、印刷するテキストの余白（上下左右）を指定できます。プリンタによっては、A4サイズの端まで印刷できない場合があるので、この指定が必要です。また、用紙の左端にとじしろを残したい場合などにも便利です。

#### ページの余白を設定するには

1. [ファイル->ページレイアウトの設定]を選択してください。[ページ設定]ダイアログボックスを開きます。
2. [左]、[右]、[上]、[下]フィールドに必要な余白を入力してください。
3. [インチ]または[ミリメートル]ラジオボタンを選択してください。
4. [OK]ボタンをクリックしてください。



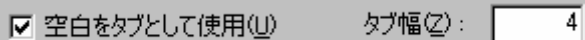
#### ページのヘッダとフッタを設定するには

1. [ファイル->ページレイアウトの設定]を選択してください。[ページ設定]ダイアログボックスを開きます。
2. [ヘッダ]、[フッタ]フィールドに表示するテキストを入力してください。ページ番号、テキスト配置、日付の各フィールドとともに通常のプレースホルダが利用できます。プレースホルダはページが印刷される前に実際の値に展開されます。
3. [OK]ボタンをクリックしてください。

#### 印刷の折り返しを設定するには

1. [ファイル->ページレイアウトの設定]を選択してください。[ページ設定]ダイアログボックスを開きます。
2. [折り返し]チェックボックスをオンにしてください。折り返し機能が有効になり、テキストをすべて印刷します。
3. [OK]ボタンをクリックしてください。

### 4.7.2 タブを変更する



#### タブの文字数を変更するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [エディタ]タブを選択してください。
3. [タブ幅]に必要なタブの文字数を指定してください。
4. [OK]ボタンをクリックしてください。

"Tab"キーを押すと、通常、タブ文字をファイルに入力します。しかし、タブ文字のかわりに空白文字列を入力したほうがよい場合があります。[オプション]ダイアログボックスの[エディタ]タブで、タブ文字を空白文字列として指定できます。

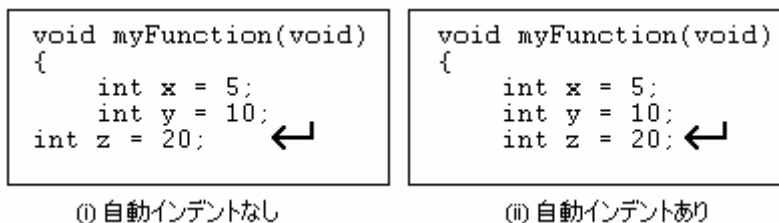
#### タブ文字の代わりに空白文字列を使用するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [エディタ]タブを選択してください。
3. タブ文字の代わりに空白文字列を使用する場合は[空白をタブとして使用]チェックボックスをオンにしてください。
4. [OK]ボタンをクリックしてください。

### 4.7.3 自動インデント

標準のエディタで Enter キーを押すと、カーソルは次の行の左端に移ります。自動インデントを設定すると、カーソルは、改行時、前行の最初の文字の下に移ります。したがって、自分でタブや空白を入力することなく、C/C++言語またはアセンブリ言語のコードをより速く見やすく入力できます。

以下に2つの例を示します。(i)は自動インデントの設定をしなかったときに Enter キーを押した場合の例を示します。カーソルは次の行の左端に移ります。"int z=20"の行は、上の2行と揃っていません。(ii)は自動インデントを設定して Enter キーを押した場合の例を示します。カーソルは前行の"int"の"i"の下に移ります。したがって、"int z=20"行を入力すると、テキストは自動的に前の行と整列(自動インデント)します。



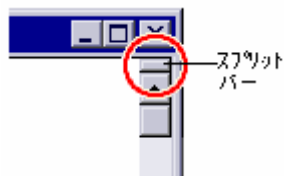
#### 自動インデントを設定するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [エディタ]タブを選択してください。
3. [自動インデント]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. [OK]ボタンをクリックすると、自動インデントを設定します。



## 4.8 ウィンドウを分割する

High-performance Embedded Workshop では、エディタウィンドウを2つに分割できます。以下に、エディタウィンドウの右上にある"閉じる"ボタンのすぐ下にあるスプリットバーボタンを示します。



### ウィンドウを分割するには

スプリットバーボタンをダブルクリックしてウィンドウを2つに分割するか、スプリットバーボタンを押したままマウスを下に移動して分割したい位置でマウスボタンを離してください。

### スプリットバーの位置を調節するには

スプリットバーボタンを押したままマウスを下に移動して、分割したい位置でマウスボタンを離してください。

### ウィンドウの分割を解除するには

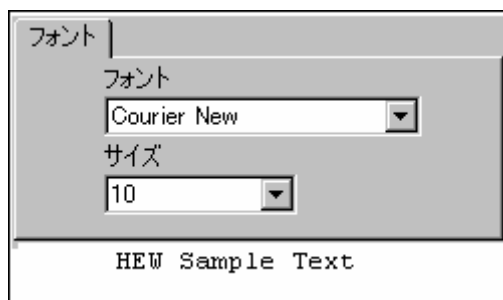
スプリットバーボタンをダブルクリックするか、スプリットバーボタンをウィンドウの一番上か一番下に移動してください。

## 4.9 エディタのフォントを変更する

High-performance Embedded Workshop ではエディタのフォントを指定できます。ファイルの種類にかかわらず、エディタウィンドウはすべて同じフォントを使用します。

### エディタのフォントを変更するには

1. [基本設定 -> 表示の形式]を選択してください。[表示形式]ダイアログボックスを開きます。
2. ツリーの中の[Source]アイコンを選択してください。[表示形式]ダイアログボックスの右側の[フォント]タブが使用可能です。
3. [フォント]リストからフォントの種類を選択してください。
4. [サイズ]リストからフォントの大きさを選択してください。
5. フォントを変更するとそのフォントがどう見えるか下にサンプルテキストを表示します。
6. [OK]ボタンをクリックしてください。



## 4.10 シンタックスを色付けする

コードをより読みやすくするため、指定した文字列（キーワードなど）を異なる色で表示できます。例えば、C ソースコードのコメントを緑色で、C 言語の型（int など）を青色で表示できます。

色付け方法は、ファイルグループ単位で指定できます。例えば、C source file、C header file、Assembly source file、Assembly include file でそれぞれ異なった色付け方法を定義できます。

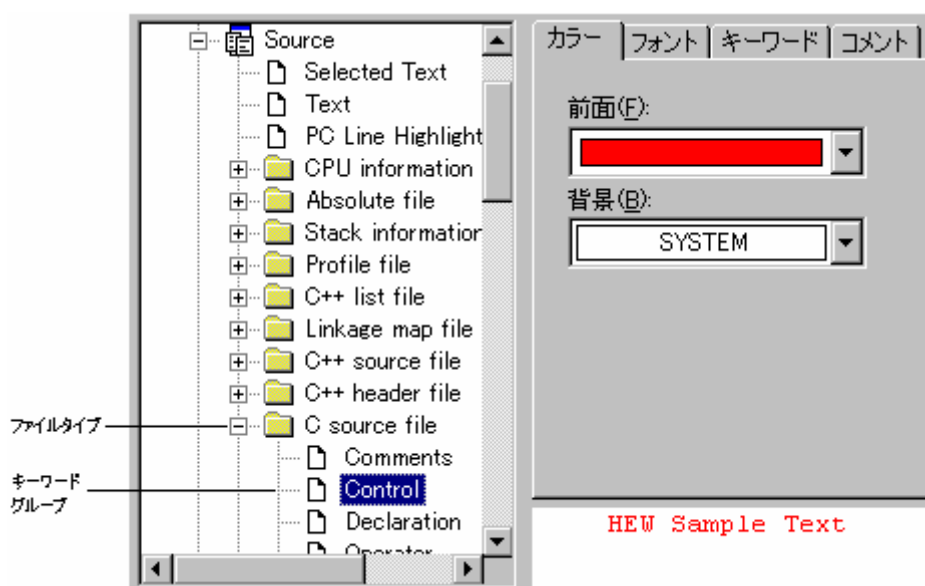
### 注：

新規ファイルを作成すると、デフォルトではファイルの拡張子がないためシンタックスの色付けは行いません。（エディタが自動的に新規ファイルに付ける名前には拡張子がありません）。新規ファイルにシンタックスの色付けをするには、上記の既存のファイルグループの拡張子を持つ名前でファイルを保存してください。ファイル拡張子の情報については「2.5 ファイル拡張子とファイルグループ」を参照してください。

### 4.10.1 テキストの色を変更する

#### 既存の色を変更するには

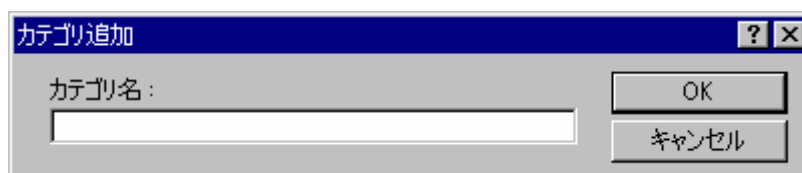
1. [基本設定 -> 表示の形式]を選択してください。[表示形式]ダイアログボックスを開きます。
2. 色を変更したいアイテムを選択します。エディタの場合は、ダイアログボックスの左側のツリーにある"Source"を展開してください。
3. シンタックスの色を編集したいファイルタイプを展開してください。
4. 対象となるキーワードグループを選択してください。ダイアログボックスの右側のタブは、選択した項目によって変わります。
5. [カラー]タブを選択してください。
6. [前面]リストと[背景]リストの色を選択してください。[SYSTEM]を選択するとコントロールパネルで現在設定してある文字色と背景色になります。
7. [OK]ボタンをクリックしてください。



## 4.10.2 新規キーワードを作成する

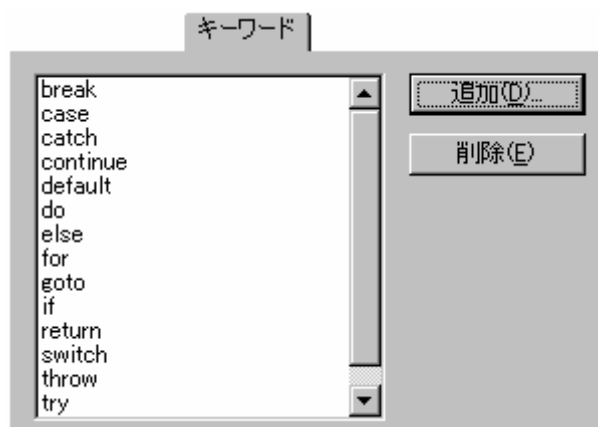
### 新規のキーワードグループを作成するには

1. [基本設定 -> 表示の形式]を選択してください。[表示形式]ダイアログボックスを開きます。エディタの場合は、ダイアログボックスの左側のツリーにある[Source]を展開してください。
2. キーワードグループを追加したいファイルタイプをツリーから選択してください。
3. ツリー下の[追加]ボタンをクリックすると[カテゴリ追加]ダイアログボックスを開きます。[カテゴリ名]にキーワードグループを入力し、[OK]ボタンをクリックすると新しいキーワードグループを追加します。キーワードグループ名を変更するには、キーワードグループを選択してツリー下の[変更]ボタンをクリックしてください。[カテゴリの変更]ダイアログボックスを開きます。[カテゴリ名]にキーワードグループを入力し、[OK]ボタンをクリックするとキーワードグループ名を変更します。キーワードグループを削除するには、キーワードグループを選択してツリー下の[削除]ボタンをクリックしてください。

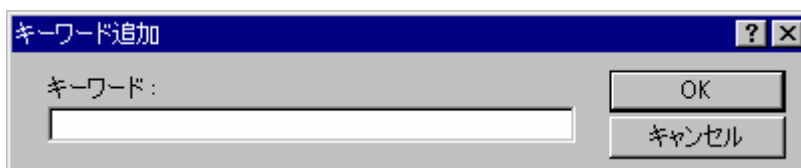


### 新規のキーワードを作成するには

1. [基本設定 -> 表示の形式]を選択してください。[表示形式]ダイアログボックスを開きます。
2. シンタックスの強調表示を変更したいアイテムを、ツリーの中のアイコンの下から選択してください。このアイテムはファイルタイプ（例：C言語のソースファイル）および適切なキーワードグループ（例：識別子またはプリプロセッサ）となります。
3. [キーワード]タブを選択してください。



4. キーワードを追加するには[追加]ボタンをクリックしてください。[キーワード追加]ダイアログボックスを開きます。[キーワード]にキーワードを指定し[OK]ボタンをクリックしてダイアログボックスを閉じてください。キーワードを削除するには、キーワードを選択して[削除]ボタンをクリックしてください。

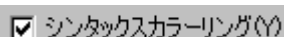
**注：**

[キーワード追加]ダイアログボックスの[キーワード]には半角英数字と半角下線、半角#のみ入力してください。特に、日本語文字やその他の文字を入力しないでください。

### 4.10.3 シンタックスの色付けを有効/無効にする

#### シンタックスの色付けを有効/無効にするには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [エディタ]タブを選択してください。
3. [シンタックスカラーリング]チェックボックスをオンにすると有効になり、チェックしないと無効になります。設定後、[OK]ボタンをクリックしてください。



### 4.11 テンプレート

ソフトウェア開発時、同じテキストを繰り返し入力する場合があります。例えば、関数定義、ループ、関数の機能のコメント欄などです。High-performance Embedded Workshopでは、現在アクティブなエディタウィンドウに、定型テキスト（テンプレート）を挿入できます。テンプレート設定後は、手作業で入力するかわりに定型テキストを簡単に挿入できるようになります。

テンプレート一覧を示します。この一覧はワークスペースウィンドウの[Templates]タブにあります。



High-performance Embedded Workshop で追加した新しいテンプレートを[Templates]フォルダの下に表示します。[Toolchain Templates]フォルダには、現在のツールチェーンによって High-performance Embedded Workshop システムで使用するために提供される読み取り専用のテンプレートを表示します。


この一覧にあるテンプレートはドラッグアンドドロップしてエディタファイルに挿入できます。さらに、エディタからテキスト領域を[Templates]フォルダにドラッグアンドドロップして、簡単にテンプレートを作成することも可能です。

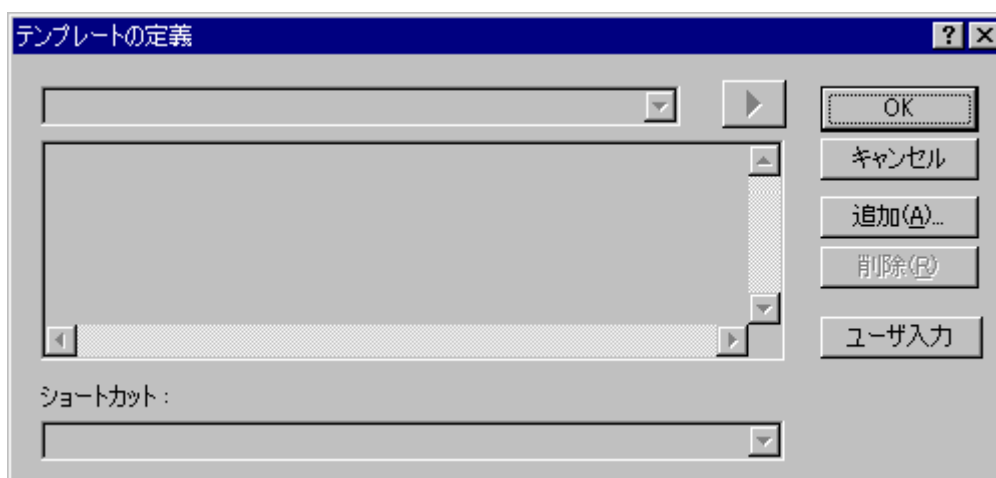
[Templates]フォルダ上で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	機能
追加	新しいテンプレートを追加します。
編集	現在選択しているテンプレートを編集します。
削除	現在選択しているテンプレートを削除します。

### 4.11.1 テンプレートを設定する

#### テンプレートを設定するには

- 以下のいずれかの操作を選択し[テンプレートの定義]ダイアログボックスを開きます。
  - [テンプレートの定義]ツールバーボタン(  )をクリックする。
  - [編集 -> テンプレート -> テンプレートの定義]を選択する。
  - ワークスペースウィンドウの[Templates]タブの[Templates]フォルダ上で右クリックし、ポップアップメニューから[編集]を選択する。
  - エディタウィンドウ内で右クリックし、ポップアップメニューから[テンプレート -> テンプレートの定義]を選択する。



- [追加]ボタンをクリックしてください。[テンプレートの追加]ダイアログボックスが表示され、選択したテンプレート名を尋ねます。この場合、独自のテンプレート名を選択してください。複製したテンプレート名のメッセージが表示され、テンプレートは追加されません。
- 既存のテンプレートを変更したいときは、[テンプレート名]ドロップダウンメニューから変更するテンプレートを選択してください。
- テンプレートテキストエリアにテキストを入力してください。他のエディタウィンドウからテキストをコピーし、"Ctrl+V"キーを押してこのダイアログボックスに貼り付けることができます。
- テンプレートを設定するとき、次のプレースホルダを使うことができます。

メニューエントリ	プレースホルダ	説明
ファイルパス+ファイル名	\$(FULLFILE)	ファイル名 (絶対パスを含む)
ファイル名	\$(FILENAME)	ファイル名 (パスを除き拡張子を含む)
ファイルリーフ	\$(FILELEAF)	ファイル名 (パスと拡張子を除く)
ワークスペース名	\$(WORKSPNAME)	ワークスペース名
プロジェクト名	\$(PROJECTNAME)	プロジェクト名
行番号	\$(LINE)	テンプレートを挿入する最初の行番号
時間	\$(TIME)	テンプレートが挿入される時間
日付, テキスト	\$(DATE_TEXT)	現在の年月日をテキスト表示
日付, 日/月/年	\$(DATE_DMY)	現在の日/月/年
日付, 月/日/年	\$(DATE_MDY)	現在の月/日/年
日付, 年/月/日	\$(DATE_YMD)	現在の年/月/日
ユーザ名	\$(USER)	現在のユーザ
カーソル位置	\$(CURSOR)	挿入カーソル: テンプレートを挿入した後カーソルをこの位置に設定



6. '\$(CURSOR)'を入力すると、テンプレートが挿入された後のカーソルはこの位置になります。'\$(CURSOR)'を設定しないと、テンプレートが挿入された後のカーソルはテンプレート最後の文字の後ろになります（通常の貼り付け操作と同じ）。
7. テンプレート用に確保されているショートカットキーは 10 個あります。ショートカットキーを使用する場合、[テンプレートの編集]ダイアログボックス下部にあるドロップダウンリストで選択してください。"Ctrl+0" ~ "Ctrl+9"の中から選択できます。

## ユーザ入力


テンプレートを定義する際、ユーザ入力フィールドも定義できます。以下のプレースホルダを使って指定します。

```
$(USERINPUT<n:1-10>|"<some text>")
```

'n'はユーザ入力識別子を表す数字です。これらのプレースホルダは手動で追加することも可能ですが、[テンプレートの定義]ダイアログボックスの[ユーザ入力]ボタンを使うとプレースホルダを自動的に追加できます。テンプレートをファイルに挿入するとダイアログボックスが表示され、ユーザは各フィールドにカスタムのテキストを入力できます。その後、このテキストをプレースホルダの代わりに挿入します。ユーザはこのようなユーザ入力フィールドを 10 個定義できます。

### 4.11.2 テンプレートを削除する

#### テンプレートを削除するには

1. 以下のいずれかの操作を選択し[テンプレートの定義]ダイアログボックスを開きます。
  - [テンプレートの定義]ツールバーボタン()をクリックする。
  - [編集 -> テンプレート->テンプレートの定義]を選択する。
  - ワークスペースウィンドウの[Templates]タブの[Templates]フォルダ上で右クリックし、ポップアップメニューから[編集]を選択する。
  - エディタウィンドウ内で右クリックし、ポップアップメニューから[テンプレート -> テンプレートの定義]を選択する。
2. [テンプレート名]ドロップダウンリストから削除したいテンプレート名を選択し、[削除]ボタンをクリックしてください。
3. 確認ダイアログボックスが表示された場合は、[はい]ボタンをクリックしてください。
4. [OK]ボタンをクリックしてください。

テンプレートを削除するには、他にも方法があります。これを以下に説明します。


#### ワークスペースウィンドウの[Templates]タブで選択したテンプレートを削除するには

1. ワークスペースウィンドウの[Projects]タブで削除したいファイルを選択してください。複数のテンプレートを選択するときは"Shift"キーまたは"Ctrl"キーを押してください。
2. ワークスペースウィンドウの[Templates]タブの[Templates]フォルダ上で右クリックし、ポップアップメニューから[削除]を選択する。

### 4.11.3 テンプレートを挿入する

#### テンプレートを挿入するには

1. 以下のいずれかの操作を選択し[テンプレートの定義]ダイアログボックスを開きます。


- ツールバーでテンプレートを選択し[テンプレートの挿入]ツールバーボタン(  )をクリックする。
  - [編集->テンプレート->テンプレートの挿入]を選択する。
  - エディタウィンドウ内で右クリックし、ポップアップメニューから[テンプレート->テンプレートの挿入]を選択する。
2. [テンプレート名]ドロップダウンリストで挿入するテンプレートを選択し、OK ボタンをクリックします。選択したテンプレートが現在のエディタウィンドウに追加されます。

"Alt"キーとテンプレート番号を押すことでテンプレートを挿入する方法もあります(例えばテンプレート4を挿入するには"Alt+4")。これらのショートカットは[テンプレートの定義]ダイアログボックスで定義できます。ダイアログボックスの下部のドロップダウンリストで選択可能です。

## 4.12 括弧の組み合わせ

複雑なソースコードは扱いにくいことがあります。特に、C言語のブロックが互いに深いネスト構造になっている場合や、if文で複雑な論理文が表現されている場合などです。High-performance Embedded Workshopでは、そのような場合のために、括弧の種類 { }, ( ), [ ]ごとに括弧の間のテキストを強調表示できます。

### 括弧の組み合わせを見つけるには

1. 括弧の始めを強調表示するか、カーソルを括弧の前に置いてください。
2. 以下のいずれかの操作を選択してください。
  - [括弧の呼応]ツールバーボタン(  )をクリックする。
  - "Ctrl+Shift+M"キーを押す。
  - [編集->括弧の呼応]を選択する。
  - ポップアップメニューから[括弧の呼応]を選択する。

ファイル全体の構造をチェックするために、カーソルをファイルの始めに置いて、括弧の組み合わせの操作を繰り返し行ってください。組み合わせがなくなるまで、括弧の組み合わせごとに次々と強調表示します。

## 4.13 ファイルを読み取り専用属性にする

High-performance Embedded Workshopではファイルを読み取り専用属性にできます。読み取り専用属性のファイルを保存するときは、別名で保存するかどうかを確認するダイアログボックスが開きます。

### ファイルを読み取り専用属性にするには

1. 以下のいずれかの操作を選択してください。
  - ワークスペースウィンドウの[Projects]タブで、ファイルを右クリックしポップアップメニューを表示してください。複数のファイルを選択できます。
  - エディタウィンドウを右クリックしポップアップメニューを表示してください。
2. [プロパティ]を選択すると[プロパティ]ダイアログボックスが開きます。
3. [読み取り専用]チェックボックスをオンにしてください。複数のファイルを選択し、読み取り専用および書き込み可能なファイルが混在する場合は、チェックボックスはグレー(中間状態)です。チェックボックスのデフォルトはオフです。
4. エディタウィンドウのタイトルバーに「読み取り専用」を表示します。(例: filename.c - 読み取り専用)

**注：**

エディタウィンドウでファイルを開いている状態で、Windows® Explorer からそのファイルの読み取り専用属性のオンオフを切り替えた場合、エディタウィンドウのタイトルバーの表示内容は、更新されないためファイルの属性と一致しません。編集するか、ファイルを閉じてから再度開くまでエディタウィンドウのタイトルバーの表示内容を更新しません。

#### 4.14 デバッグ時にファイルの編集を抑止する

デバッグ時にファイルの編集を抑止したい場合があります。

High-performance Embedded Workshop ではデバッガに接続しデバッグしている時にファイルの編集を抑止する機能があります。編集を抑止するファイルは High-performance Embedded Workshop エディタで開いているすべてのファイルです。

##### デバッグ時にファイルの編集を抑止するには

1. [基本設定->オプション]を選択し、[オプション]ダイアログボックスを開きます。
2. [エディタ]タブを選択します。
3. [ターゲット接続中はファイルの編集を許可しない]チェックボックスをオンにすると、デバッガに接続している時に、ファイルの編集を抑止できます。デバッガに接続していないときはファイルの編集は可能です。このチェックボックスのデフォルトはオフです。
4. [OK]ボタンをクリックします。

#### 4.15 エディタカラムの管理

エディタのカラムは、デバッガの機能に応じて追加されます。また、カラムの表示、非表示を選択できます。

##### すべての High-performance Embedded Workshop エディタでカラムをオフにするには

1. 以下のいずれかの操作を選択し[エディタ全体のカラム状態]ダイアログボックスを開きます。
  - [編集 -> 表示カラムの設定]を選択します。
  - エディタウィンドウ内で右クリックし、ポップアップメニューから[表示カラムの設定]を選択する。



2. チェックボックスは、そのカラムが有効か無効かを示します。チェックしている場合は有効です。チェックボックスがグレー表示の場合、一部のエディタではカラムが有効で、別のエディタでは無効であることを意味します。
3. [OK]ボタンをクリックしてください。

### 1つの High-performance Embedded Workshop エディタでカラムをオフにするには

1. 削除したいカラムのあるエディタウィンドウを開きます。
2. 以下のいずれかの操作を選択してください。
  - High-performance Embedded Workshop エディタのヘッダ内で右クリックするとポップアップメニューが表示されます。カラムが利用できる場合は、エントリの横にはチェックマークがあります。エントリをクリックすると、カラムの表示、非表示を切り替えます。
  - エディタウィンドウ内で右クリックし[カラム]を選択します。カスケードされたメニューオプションを表示します。各カラムがこのポップアップメニューに表示されます。カラムが有効である場合、名前の横にチェックマークがあります。エントリをクリックすると、カラムの表示、非表示を切り替えます。

High-performance Embedded Workshop エディタのヘッダが表示されている場合は、カラムの幅をマウスのドラッグにより調節できます。

### 4.16 カラムのヘッダの表示/非表示を切り替える

エディタウィンドウはカラムのヘッダがあります。カラムのヘッダ自体の表示、非表示の切り替えが可能です。

#### カラムのヘッダの表示、非表示を切り替えるには

1. エディタウィンドウでソースコードまたは逆アセンブリを表示している領域内を右クリックしてください。
2. 表示されるポップアップメニューから[カラムヘッダの表示/非表示]を選択してください。

### 4.17 エディタ内からファイルを開く

エディタウィンドウに表示しているファイル名を元にそのファイルをエディタウィンドウで開くことができます。

#### エディタウィンドウに表示されているファイルを開くには

1. エディタウィンドウの[ソース]領域で表示したいファイル名の上を右クリックしてください。



2. ポップアップメニューから[開く 'ファイル名']を選択してください。

#### 4.18 ツールチップウォッチ

プログラムに定義された変数の値を参照するための機能です。値を参照したい変数を含むプログラムをエディタウィンドウまたは逆アセンブリウィンドウ（ソースモード）に表示し、変数名の上にマウスカーソルを置くと、ツールチップとしてポップアップウィンドウにウォッチ情報を表示します。

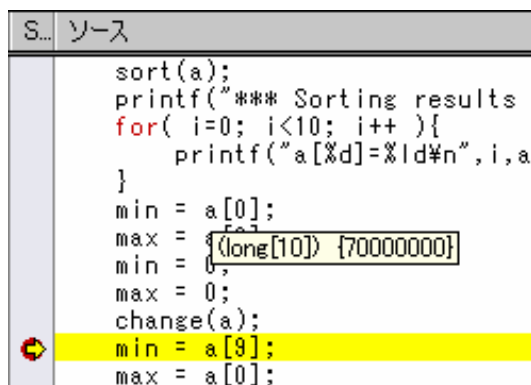
##### ツールチップウォッチ機能を使用するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [エディタ]タブを選択してください。
3. [ツールチップウォッチ]チェックボックスをオンにしてください。
4. [OK]ボタンをクリックしてください。

ツールチップウォッチ

##### エディタウィンドウまたは逆アセンブリウィンドウ（ソースモード）にツールチップウォッチを表示するには

1. 参照したい変数を表示しているエディタウィンドウまたは逆アセンブリウィンドウ（ソースモード）を開きます。
2. 参照したい変数名の上にマウスカーソルを置きます。ツールチップとしてポップアップウィンドウに変数のウォッチ情報を表示します。



```

S... ソース
sort(a);
printf("*** Sorting results
for( i=0; i<10; i++ ){
    printf("a[%d]=%ld\n", i, a
}
min = a[0];
max = {long[10]} {700000000}
min = 0;
max = 0;
change(a);
min = a[9];
max = a[0];

```

#### 4.19 式の評価

[式の評価]ダイアログボックスを開きます。例えば、 $205 \times 2$  のような数式を入力して、その結果をデバッガが現在サポートしているすべての基数で表示します。

##### 式評価するには

1. [編集 -> 式の評価]を選択してください。[式の評価]ダイアログボックスを開きます。
2. 式を入力後、[評価]ボタンをクリックしてください。

簡単な式、そして括弧、基数とシンボルを持つ複雑な式を評価できます。

すべての演算子は同じ優先順位を持っていますが、括弧は評価の順序を変更できます。演算子は C/C++ と同じ意味を持っています。式もまた数値が要求されるコマンドで使うことができます。結果は現在サポートしているすべての基数で表示します。

#### 有効な演算子:

加算 (+)	減算 (-)	乗算 (*)	除算 (/)
論理 AND (&&) *	論理 OR (  ) *	論理 NOT (!) *	等しい (==)
ビットごとの AND (&)	ビットごとの OR ( )	ビットごとの NOT (~)	等しくない (!=)
左算術シフト (<<)	右算術シフト (>>)	より小さい (<)	より大きい (>)
剰余 (%) *	ビットごとの排他的 OR (^)	以下 (<=)	以上 (>=)

#### 注:

\*. 機能のサポートはデバッガに依存します。

#### レジスタ名:

値を入力するときにレジスタ値を使用できます。"#PC"または"%PC"のように、レジスタ名のプレフィックスに"#"記号または"% "記号を付けます。サポートするプレフィックスはデバッガに依存します。

#### 文字定数:

文字定数として、シングルクォーテーション(')で囲まれた文字が使用できます。例えば、'A'等です。これらは、ASCII コードに変換され、1 バイトの即値として使用されます。

#### 文字列リテラル:

文字列リテラルとして、ダブルクォーテーション(")で囲まれた文字列が使用できます。例えば、"abc"等です。

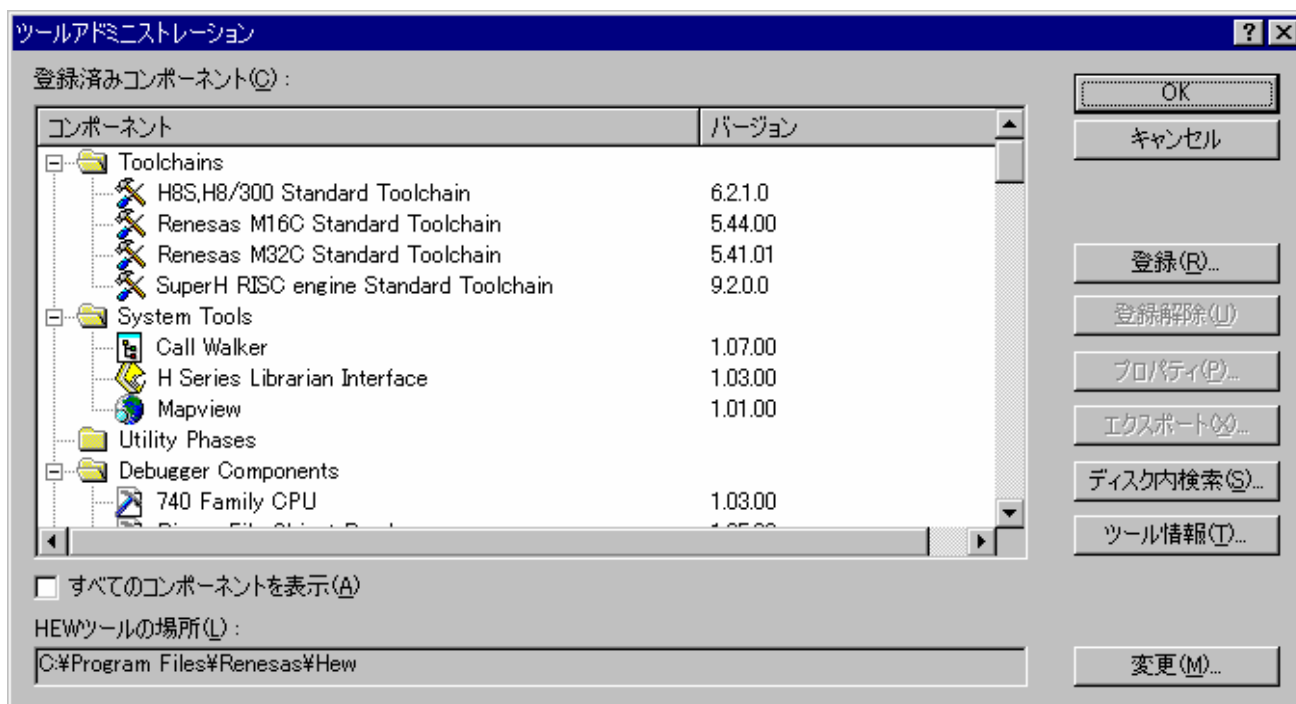
#### 注:

機能のサポートはデバッガに依存します。

## 5. ツール管理

[ツールアドミニストレーション]ダイアログボックスで、High-performance Embedded Workshop で使うツールを管理します。

このダイアログボックスは[ツール -> アドミニストレーション]で開きます。すべてのワークスペースが閉じているときは変更可能で、ワークスペースが開いているときは参照のみ可能です。



High-performance Embedded Workshop には 5 種類の標準ツールがあります。

- [Toolchains] 一連のビルドフェーズ（例:コンパイラ、アセンブラ、リンケージエディタ、ライブラリアン）。ビルド機能を実現します。
- [System Tools] [ツール]メニューからを選択することのできるアプリケーションです（.EXE）。ツールチェインをサポートする追加のアプリケーションです（例:ライブラリアンインタフェース）。
- [Utility Phases] 特定のビルド機能をサポートする、あらかじめ用意されたビルドフェーズです（例:ソースコードの複雑度解析、ソースコードの行カウントなど）。特定のツールチェインに依存しない追加のビルド機能です。
- [Debugger Components] 特定のデバッグ機能をサポートするツールです（例: CPU DLL、デバッガ、オブジェクトリーダなど）。
- [Extension Components] High-performance Embedded Workshop システムのある領域における特定のキー機能をサポートするツールです。これらのツールはインストールすると、必ず登録します（例: RTOS Debug Interface、フラッシュサポートなど）。

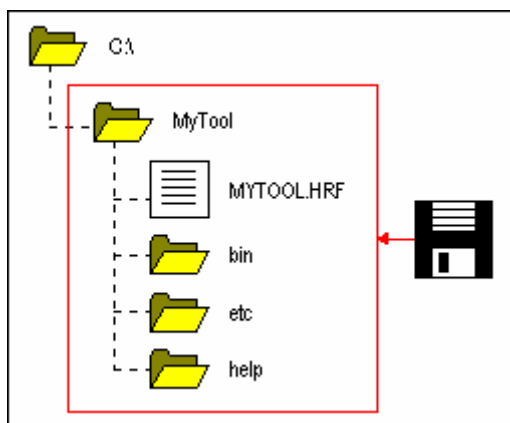
### 5.1 ツールの位置

High-performance Embedded Workshop では、新しいツールがインストールされるたびに High-performance Embedded Workshop との連動に必要なツールの位置を自動的に保持します。インストール後、High-performance Embedded Workshop はそのツールに関する情報（位置を含む）を保持します。これを登録と呼びます。初期登録は自動で行いますが、開発の途中で、プロジェクトのツールをより効率良く利用するためにツールを登録することが必要になることがあります。

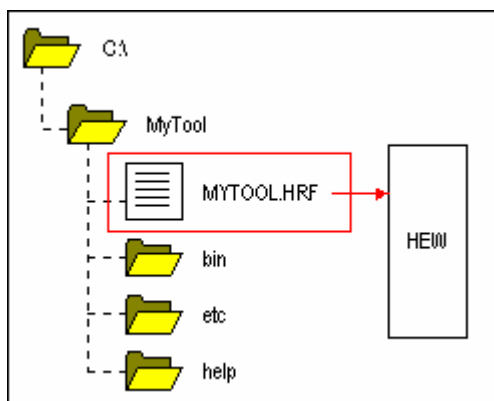


## 5.2 High-performance Embedded Workshop登録ファイル

High-performance Embedded Workshop と互換性のあるツール（ツールチェイン、システムツール、またはユーティリティフェーズ）をインストールすると、拡張子.HRF のファイルもインストールします（以下の図を参照）。.HRF ファイルには High-performance Embedded Workshop への登録に必要な情報を記述しています。



登録するには、そのツールの.HRF ファイルを[ツールアドミニストレーション]ダイアログボックスにロードします（以下の図を参照）。



High-performance Embedded Workshop でツールを使うには、まず登録が必要です。[ツールアドミニストレーション]ダイアログボックスは現在登録されているツールを表示します。このダイアログボックスを開くには、ワークスペースがすべて閉じていることを確認して[ツール -> アドミニストレーション]を選択してください。ワークスペースを開いた状態で[ツールアドミニストレーション]ダイアログボックスにアクセスすると、ダイアログが開きませんが変更はできません。High-performance Embedded Workshop がデフォルトでインストールされると、新しいツールを自動的に登録します。

High-performance Embedded Workshop はツールデータベースファイルに、ツール情報を格納します。デフォルトでは、このファイルを High-performance Embedded Workshop アプリケーションディレクトリに作成します。しかし、ネットワーク環境で作業を行っている場合は、このディレクトリは他の場所に設定されることがあります。ツールディレクトリの場所は変更が可能です。

### ツールの場所を変更するには

1. [ツール -> アドミニストレーション]を選択してください。
2. [HEW ツールの場所]の[変更]ボタンをクリックしてください。

3. 新しいツールがあるディレクトリのルートディレクトリを選択し、[OK]ボタンをクリックしてください。
4. ディレクトリが切り替わり、ツールの場所が新しいディレクトリに移ります。この場所にある新しいツールはスキャンする必要があります。スキャンには、[ディスク内検索]ボタンまたは[登録]ボタンを使用します。

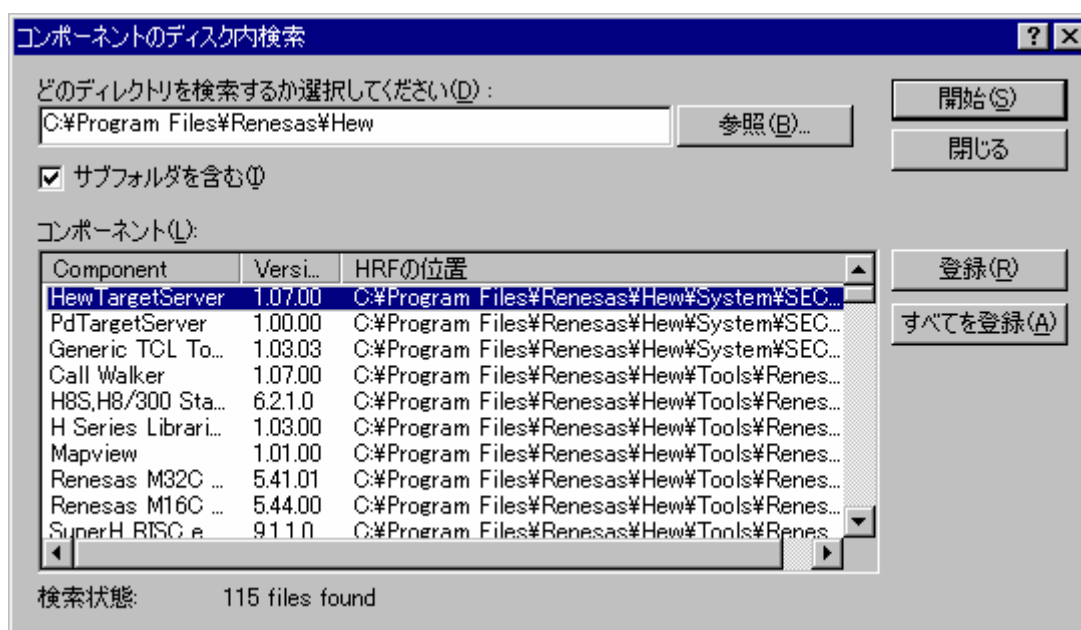
### 5.3 ツールを登録する

High-performance Embedded Workshop は起動後、インストールしたすべてのツールを自動的に登録します。しかし、場合によってはツールを手動で登録する必要があります。

ドライブ内を検索し High-performance Embedded Workshop に互換性のあるツールを見つけることができます。High-performance Embedded Workshop のツールインストール情報が削除または破壊されたときに再びツール情報を作成する場合に有用です。

#### ツールを検索して登録するには

1. [ツールアドミニストレーション]ダイアログボックスの[ディスク内検索]ボタンをクリックすると[コンポーネントのディスク内検索]ダイアログボックスを開きます。
2. [どのディレクトリを検索するか選択してください]に、検索するディレクトリを入力してください。または、[参照]ボタンをクリックしディレクトリを選択してください。
3. [サブフォルダを含む]チェックボックスをオンにすると、指定したディレクトリとその下のディレクトリをすべて検索します。
4. [開始]ボタンをクリックすると検索を始めます。検索中、[開始]ボタンは[中断]ボタンに変わります。検索を途中で止めるには[中断]ボタンをクリックしてください。
5. [コンポーネント]リストに検索結果を表示します。個別にツールを登録するにはそのツールを選択し[登録]ボタンをクリックしてください。すべてのツールを登録するには[すべてを登録]ボタンをクリックしてください。
6. [閉じる]ボタンをクリックしてください。



### ツールを1つ登録するには

1. [登録]ボタンをクリックすると標準のファイルを開くダイアログボックスが開きます。フィルタが[HEW Registration Files (\*.hrf)]に設定されています。
2. 登録するファイルの.HRF ファイルを選択し、[選択]ボタンをクリックしてください。
3. 選択したツールに関する情報を示すダイアログボックスを開きます。ツールを登録するには[登録]ボタンをクリックしてください。登録しない場合は[閉じる]ボタンをクリックしてください。

High-performance Embedded Workshop 登録ファイルはツールがインストールされたルートディレクトリにあります。

### 5.4 ツールの登録を取り消す

新しいプロジェクト作成時に、登録された互換性のあるすべてのシステムツールがツールメニューに追加されます。ときにはユーザにとっては、これによって効率が下がり、使いにくいかもしれません。そのようなときは、登録を取り消すことができます。[ツールアドミニストレーション]ダイアログボックスでツールを選択し[登録解除]ボタンをクリックしてください。確認のダイアログボックスを開きます。登録を取り消す場合は[はい]をクリックしてください。

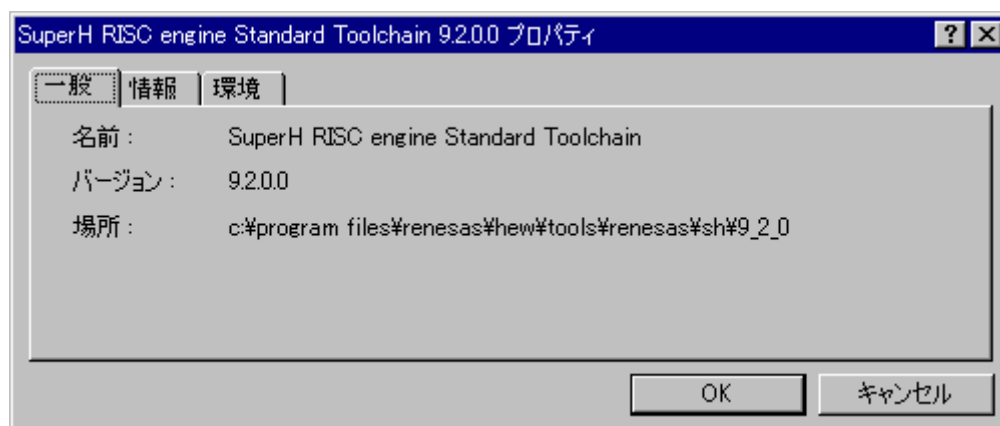
#### 注：

ツールの登録を取り消しても、ハードディスクからツールがなくなることはありません。単に、High-performance Embedded Workshop に格納されているそのツールに関する情報を削除するだけです（High-performance Embedded Workshop から切断します）。この動作はいつでも元に戻すことができ、ツールを再登録できます（「5.3 ツールを登録する」参照）。

### 5.5 ツールのプロパティの参照と編集

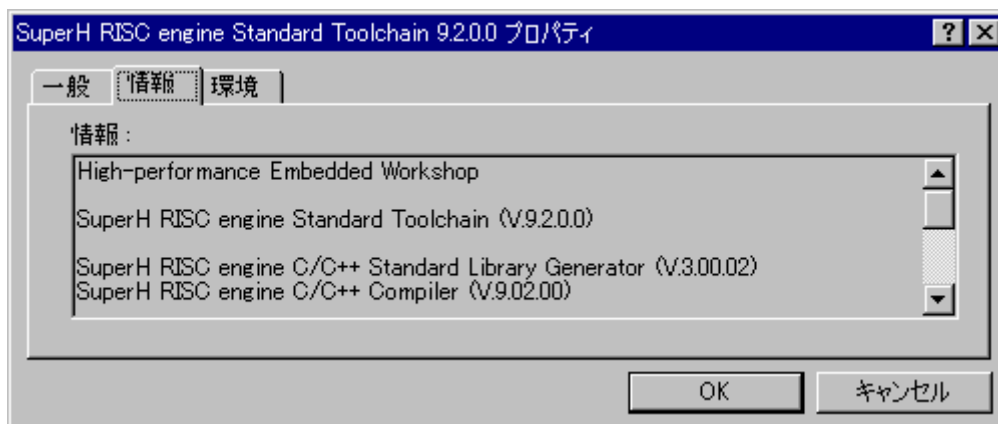
ツールに関する情報を参照するには、ツールを[登録済みコンポーネント]リストから選択し[プロパティ]ボタンをクリックしてください。

[一般]タブでは名前、バージョン、位置を示します。



### ツールについての情報を参照するには

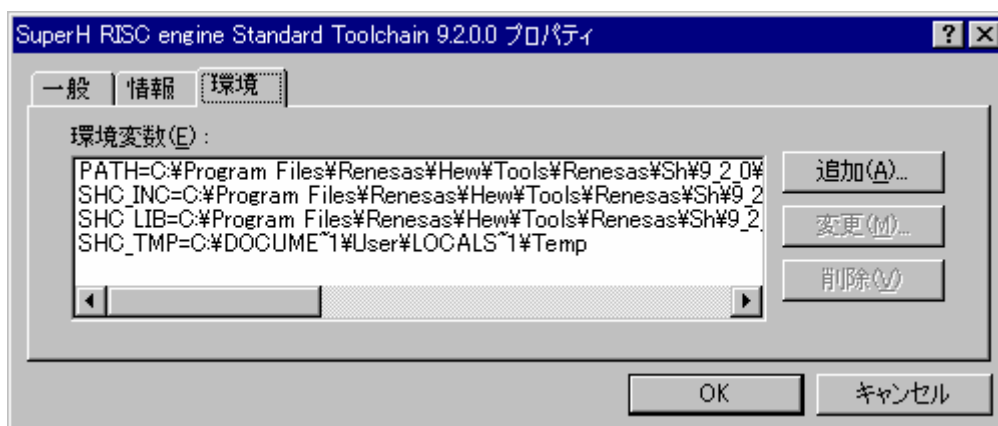
[情報]タブをクリックしてください。例えば、著作権、履歴などを表示します。



ツールの互換性に問題がある場合、この[情報]エディットフィールドで問題を報告することがあります。

### ツールの環境設定を参照、編集するには

[環境]タブを選択してください。このタブは、ツールチェーンの環境を変更するとき最も一般的に使用します。



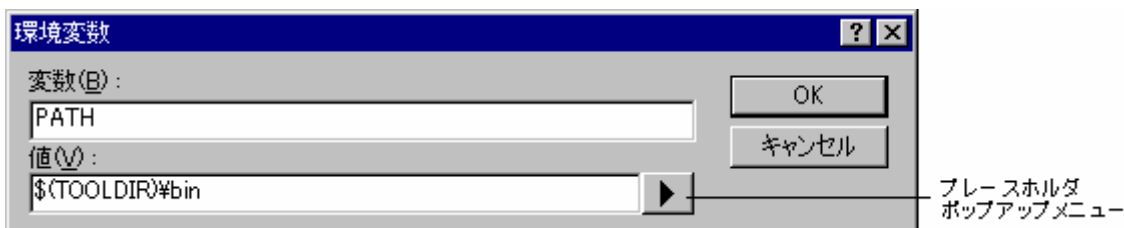
### 新しい環境変数を追加するには

1. [追加]ボタンをクリックしてください。[環境変数]ダイアログボックスを開きます。
2. [変数]に変数名を入力してください。
3. [値]に変数の値を入力してください。
4. [OK]ボタンをクリックすると、[環境]タブに新しい変数が加わります。

### 環境変数を変更するには

1. [環境]タブで変更する環境変数を選択してください。
2. [変更]ボタンをクリックしてください。[環境変数]ダイアログボックスを開きます。
3. [変数]と[値]を必要に応じて変更してください。

4. [OK]ボタンをクリックすると、環境変数を変更します。



### 環境変数を削除するには

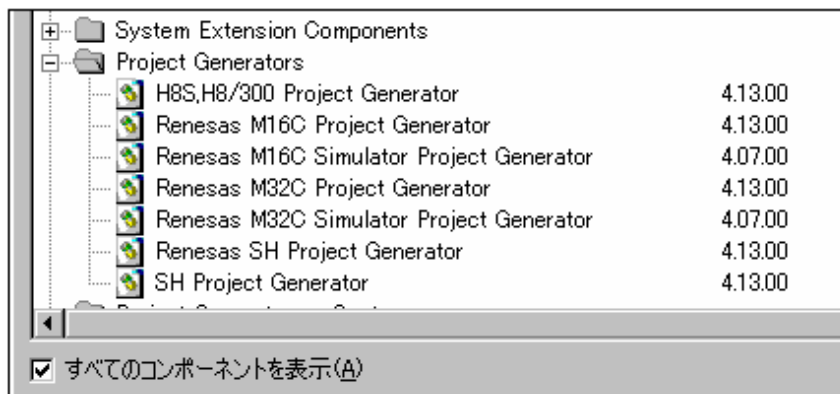
1. [環境]タブから削除する環境変数を選択してください。
2. [削除]ボタンをクリックしてください。

### 注：

環境変数の値を柔軟に指定できるようにするため、プレースホルダポップアップメニューがあります。プレースホルダの詳細は、「リファレンス 5. プレースホルダ」を参照してください。

## 5.6 テクニカルサポートについて

[ツールアドミニストレーション]ダイアログボックスでは、システムツールに関する情報を表示できます。これらのツールは High-performance Embedded Workshop の一部であり、手動で登録または登録取り消しできません。[ツールアドミニストレーション]ダイアログボックスの[すべてのコンポーネントを表示]チェックボックスをオンにすると、隠れていたコンポーネントフォルダを表示します（下の図を参照）。






テクニカルサポートを受ける際、ツールに関する詳細をお尋ねすることがあります。その場合、そのツールのフォルダを開き、ツールを選択し、[プロパティ]ボタンをクリックしてください。ここで表示されるダイアログボックスはこの章の前半で説明したものと同じように動作します。ただし、[環境]タブはありません。

High-performance Embedded Workshop にはまた、登録されたツールに関する情報をファイルに出力する機能があるので、High-performance Embedded Workshop システム全体の情報を得ることができます。もし High-performance Embedded Workshop で問題がありましたら、テクニカルサポート担当宛にこの情報を送付してください。

### ツール情報を出力するには

1. [ツール->アドミニストレーション]を選択してください。
2. [ツール情報]ボタンをクリックすると、[ツール情報ファイルの保存]ダイアログボックスを開きます。
3. ファイルの場所を選択し、[保存]ボタンをクリックしてください。
4. High-performance Embedded Workshop に現在登録されているツールの情報をファイル出力します。

また、問題のあるコンポーネントは、[ツールアドミニストレーション]ダイアログボックスで確認できます。1つは登録済みコンポーネントが見つからない場合で、次のアイコンを表示します。

	H8S,H8/300 Series Simulator Target Platform	3.0
	HMon Embedded Monitor Platform	1.0
	Intel Hex Record Object Reader	1.0

[Component not found]アイコン

もう1つは、コンポーネントは見つかったが、古いバージョンであったり他の依存するコンポーネントが使用不可であったりして使用できない場合で、次のアイコンを表示します。



[Incompatible component found]アイコン

#### 注：

ツールにいずれかのエラーがある場合、以下の方法でさらに情報を得ることができます。

### ツールエラーのフィードバックを取得するには

1. [ツール->アドミニストレーション]を選択してください。
2. 問題のあるツールをリストから選択してください。
3. [プロパティ]ボタンをクリックしてください。
4. [情報]タブを選択し、エディットフィールドを下までスクロールしてください。この領域に問題の原因を表示します。

## 5.7 オンデマンドのコンポーネント

High-performance Embedded Workshop バージョン 3.0 以降にはオンデマンドのコンポーネントという概念があります。これらのコンポーネントはアプリケーションやデバッガコンポーネントが自動的にロードするわけではありません。プロジェクトを生成する過程でユーザがロードするものです。

### オンデマンドのコンポーネントをロードまたはアンロードするには

1. [プロジェクト -> コンポーネント]を選択してください。[コンポーネントギャラリー]ダイアログボックスを開きます。
2. ロードしたいコンポーネントを選択して[ロード]ボタンをクリックしてください。コンポーネントのイメージはロードされた状態に変わります。
3. コンポーネントをアンロードしたい場合は、コンポーネントを選択して[アンロード]ボタンをクリックしてください。コンポーネントのイメージはアンロードの状態に変わります。
4. [OK]ボタンをクリックし、変更を確認してください。



#### 注：

ユーザのワークスペースにある各プロジェクトには、ロード/アンロードされた異なるコンポーネントを格納できます。複数のプロジェクトがある場合は、[Multiple projects]および[All projects]を使うことで、複数のプロジェクトにわたってコンポーネントのロード状態を変更できます。コンポーネントがあるプロジェクトでロードされ、他のプロジェクトではロードされない状態をユーザが選択した場合は、[中間状態]アイコンを表示します。

## 5.8 カスタムプロジェクトタイプ

High-performance Embedded Workshop の[プロジェクト -> プロジェクトタイプの作成]メニューで、現在のプロジェクトの設定を利用してプロジェクトのテンプレートを作成できます。このテンプレートをカスタムプロジェクトジェネレータと呼びます。新しいプロジェクトタイプの名称とプロジェクト生成ウィザードのスタイルを指定できます。作成したカスタムプロジェクトジェネレータは、[ツールアドミニストレーション]ダイアログボックスで表示できます。カスタムプロジェクトジェネレータを他のユーザが使用するには、[ツールアドミニストレーション]ダイアログボックスで該当するカスタムプロジェクトジェネレータを選択して、[エクスポート]ボタンをクリックしてください。カスタムプロジェクトジェネレータの実行環境がインストール可能な実行ファイルにまとめられます。このファイルを目的とする他のマシンで実行すると、カスタムプロジェクトジェネレータをインストールします。

一度カスタムプロジェクトタイプを作成すると、その後[新規プロジェクトワークスペース]ダイアログボックスに表示します。これを使用してプロジェクトの複製を作成できます。

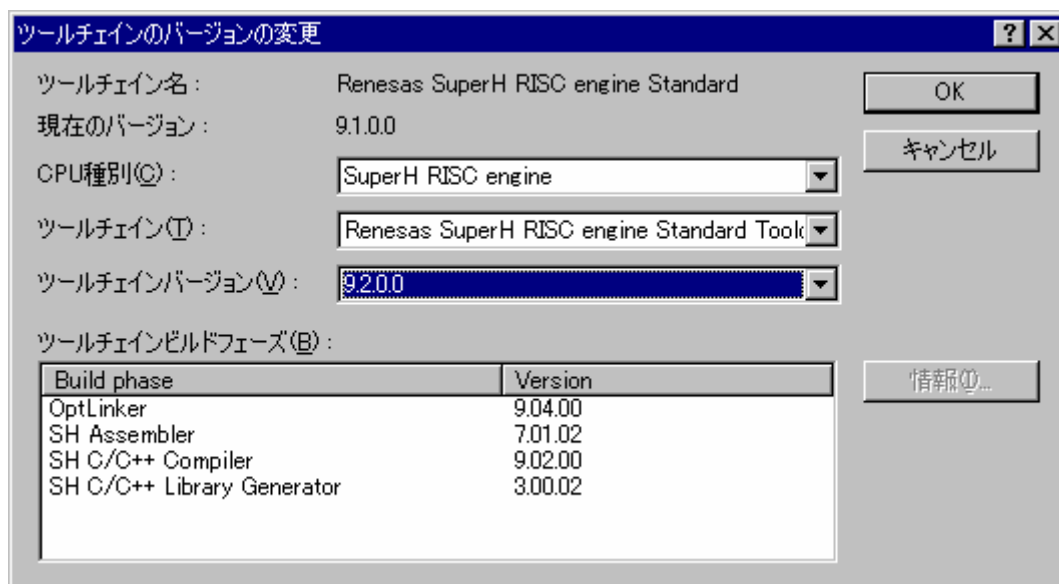
複製したプロジェクトを High-performance Embedded Workshop システムからエクスポートし、そのプロジェクトを使用する他のユーザに渡すことができます。

#### 注：

[プロジェクト -> プロジェクトタイプの作成]により使用中のプロジェクトをもとにしたプロジェクトのテンプレートを作成できます。このテンプレートにはツールチェーンのバージョン情報も含まれます。ご使用の



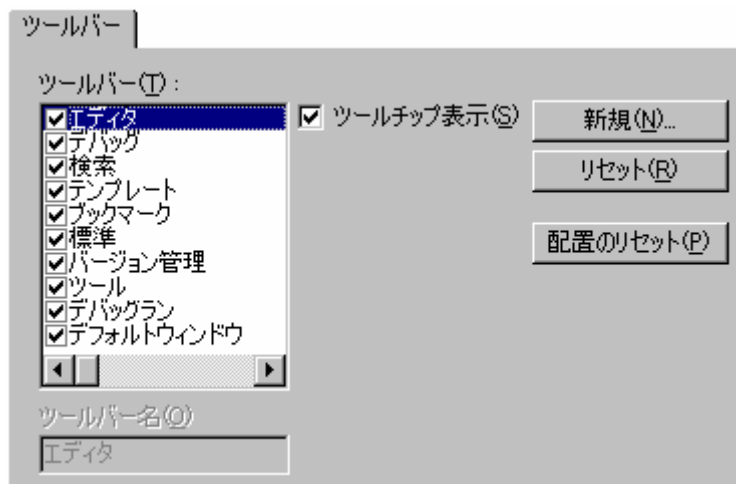
High-performance Embedded Workshop システムでツールチェーンをバージョンアップした後に、このテンプレートを使ってプロジェクトを作成した場合、作成したプロジェクトのツールチェーンバージョンが使用環境と一致しているかご確認ください。アップデート可能なツールチェーンが登録されている場合、[ツール->ツールチェーンバージョンを変更]により表示する[ツールチェーンのバージョンの変更]ダイアログボックスでツールチェーンバージョンを変更できます。



## 6. 環境のカスタマイズ

### 6.1 ツールバーのカスタマイズ

High-performance Embedded Workshop ではツールバーを標準で提供します。また、[カスタマイズ]ダイアログボックスを使用して、新しいツールバーを作成できます。



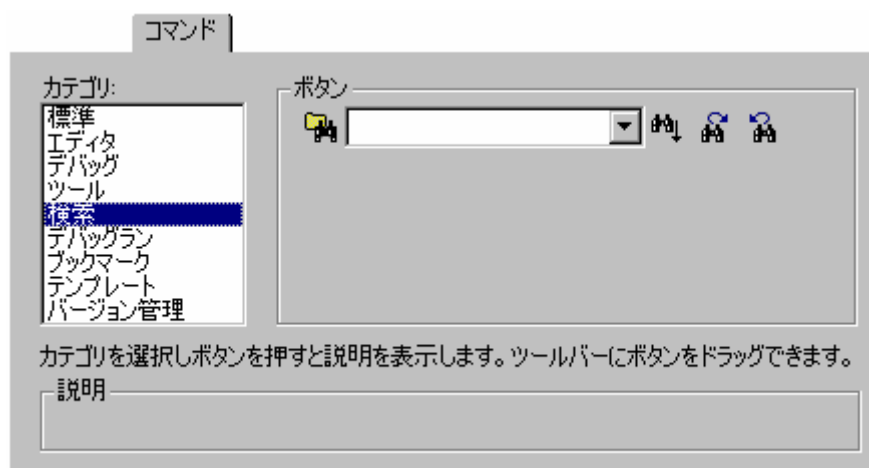
#### 新しいツールバーを作成するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [ツールバー]タブを選択してください。
3. [新規]ボタンをクリックしてください。[新規ツールバー]ダイアログボックスを開きます。
4. [ツールバー名]に新しいツールバー名を入力してください。
5. [OK]ボタンをクリックすると新しいツールバーを作成します。

作成した新しいツールバーにはボタンがなく、位置を固定せずに表示します。

#### ツールバーにボタンを追加するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [コマンド]タブを選択してください。
3. [カテゴリ]リストからボタンのカテゴリを選択し、使用できるボタンを参照してください。[ボタン]エリアからボタンを選択するとそのボタンの機能を表示します。
4. ボタンをクリックしてダイアログボックスからツールバーにドラッグしてください。



### ツールバーからボタンを削除するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [コマンド]タブを選択してください。
3. ボタンをツールバーから[ボタン]エリアへドラッグしてください。

### ユーザが定義したツールバー名を変えるには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [ツールバー]タブを選択してください。
3. [ツールバー]リストでユーザが定義したツールバーで名前を変更したいものを選択してください。
4. [ツールバー名]でツールバー名を変更してください。
5. [OK]ボタンをクリック新しいツールバー名を保存します。

### ユーザ定義のツールバーを削除するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [ツールバー]タブを選択してください。
3. [ツールバー]リストからユーザ定義のツールバーを選択すると[リセット]ボタンが[削除]ボタンに変化します。[削除]ボタンをクリックしてください。

### 標準のツールバーを初期状態に戻すには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [ツールバー]タブを選択してください。
3. [ツールバー]リストから標準のツールバーを選択し[リセット]ボタンをクリックしてください。

### ツールバーの配置を初期状態に戻すには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [ツールバー]タブを選択してください。
3. [配置のリセット]ボタンをクリックしてください。

すべてのツールバーをデフォルト位置に戻します。

セッション中に設定した配置は、すべてリセットされるので注意してください。

### ツールバーのツールチップを表示/非表示するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [ツールバー]タブを選択してください。
3. [ツールチップ表示]チェックボックスをオンにすると表示、チェックしないと非表示になります。

## 6.2 ツールメニューのカスタマイズ

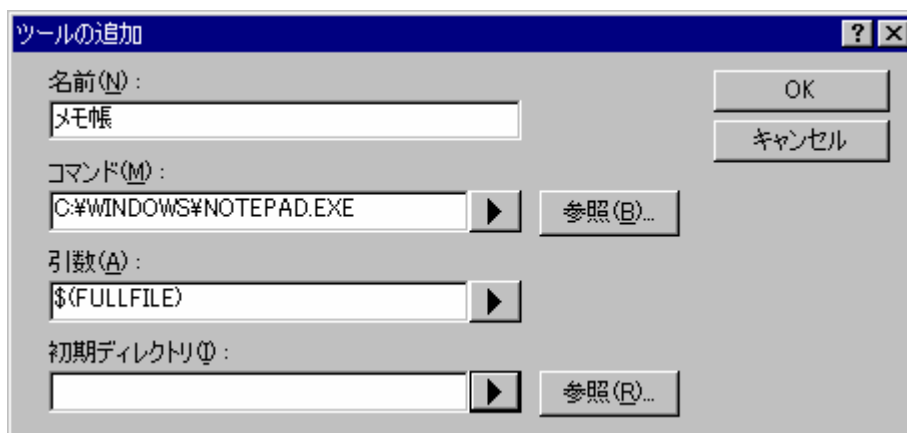
ツールメニューをカスタマイズして新しいメニューオプションを含めることができます。

### 新しいメニューオプションを追加するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [メニュー]タブを選択してください。最初に、全ワークスペースに適用できるグローバルアプリケーションワイドツールを追加する（[アプリケーション内有効:] に追加）か、または現在のワークスペースのみに適用できるワークスペースワイドツールを追加する（[ワークスペース内有効:] に追加）かどうかを決めてください。一度決定したら、ダイアログボックスの該当部分を選択するようにしてください。



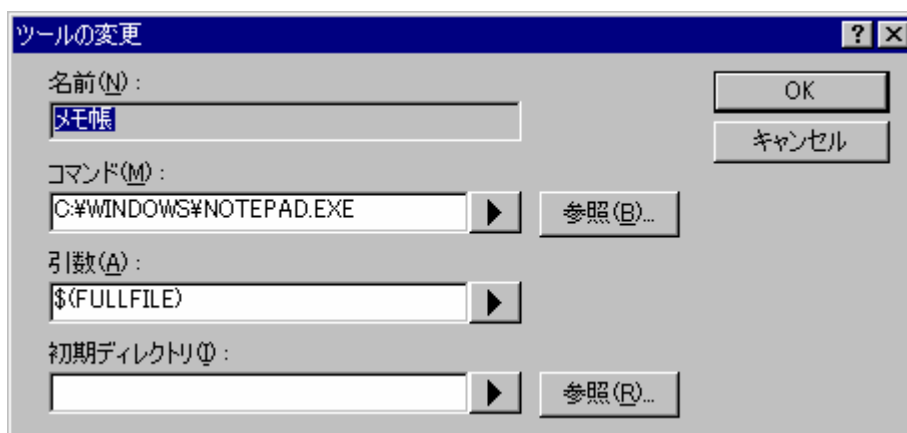
3. [追加]ボタンをクリックしてください。[ツールの追加]ダイアログボックスを開きます。
4. [名前]にツール名を入力してください。
5. [コマンド]にコマンドを入力してください。ただし、コマンドに渡す引数は入力しないでください。
6. [引数]にコマンドに渡す引数を入力してください。
7. [初期ディレクトリ]にツールを実行する初期ディレクトリを入力してください。
8. [OK]ボタンをクリックするとメニューオプションを[ツール]メニューに追加します。



新しいメニューオプションをリストの最後に追加します（ツールメニューの一番下）。


#### メニューオプションを変更するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [メニュー]タブを選択してください。
3. 変更するメニューオプションを選択し[変更]ボタンをクリックしてください。
4. [ツールの変更]ダイアログボックスを変更後、[OK]ボタンをクリックしてください。



#### メニューオプションを削除するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [メニュー]タブを選択してください。
3. 削除するメニューオプションを選択し[削除]ボタンをクリックしてください。

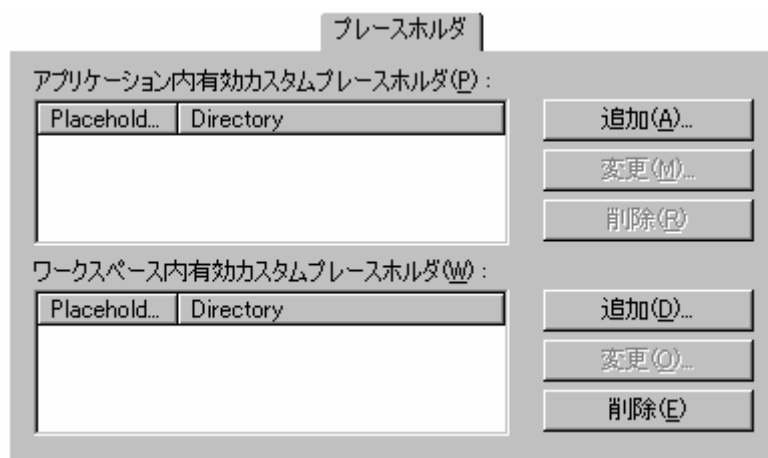
追加したツールは、[システムツール]ツールバーから利用できます。各ツール用にツールバーボタンがあります。ツールバーボタンをクリックすると、ツールメニューから選択したときと同様にツールを起動します。EXE コマンドツールの場合、ボタンのイメージは EXE 用の Windows® のアイコンと同じです。このイメージが利用できない場合は、デフォルトイメージ(  )を使用します。

### 6.3 カスタムプレースホルダを使う

High-performance Embedded Workshop では、ディレクトリを定義するときに、以前に定義したプレースホルダを指定できます。これによって、プロジェクトを再配置できます。

また、High-performance Embedded Workshop では、カスタムプレースホルダを定義することもできます。このことは、ユーザがカスタムプレースホルダを定義し、そのディレクトリの値を決められることを意味します。一度定義されると、このプレースホルダは High-performance Embedded Workshop の別の場所で有効になります。

[アプリケーション内有効プレースホルダ]に定義したプレースホルダは、High-performance Embedded Workshop で使用するすべてのワークスペースやプロジェクトに対して有効です。それに対して、[ワークスペース内有効プレースホルダ]に定義したプレースホルダは、現在のワークスペースにおいてのみ有効です。



#### カスタムプレースホルダを追加するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [プレースホルダ]タブを選択してください。
3. [アプリケーション内有効プレースホルダ]、あるいは[ワークスペース内有効ホルダ]のどちらを使いきたいか、選択してください。
4. リスト横にある[追加]ボタンをクリックしてください。[新規カスタムプレースホルダ]ダイアログボックスを開きます。
5. このフィールドでは、プレースホルダの適切な名前、およびプレースホルダが意味する内容の説明を選択してください。
6. 次に、このプレースホルダと関連するディレクトリを選択してください。

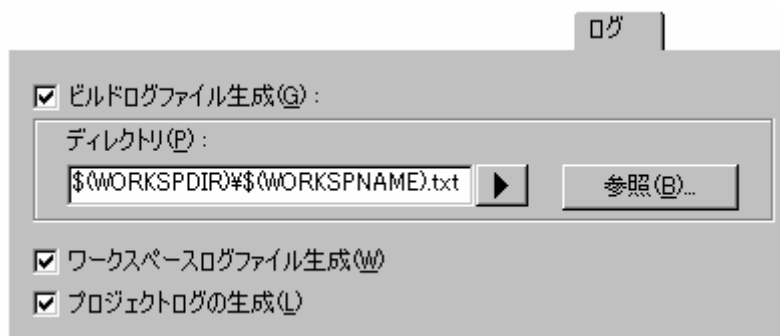
#### 注：

High-performance Embedded Workshop V.2.01 より、カスタムプレースホルダとしてユーザ定義のディレクトリを設定できます。このカスタムプレースホルダはツールチェインオプションの設定に使えます。ディレクトリパス指定の際は、下の図のように[ディレクトリ]に絶対パスを指定してください。



#### 6.4 ワークスペースやプロジェクトのログ機能を使う

High-performance Embedded Workshop では、ワークスペースとプロジェクトのログ機能をアプリケーションの中に統合しています。この機能は、[カスタマイズ]ダイアログボックスの[ログ]タブで設定できます。この設定により、ユーザ名や変更内容についてログファイルを作成できるため、ネットワークデータベースが動作しているときなどに特に便利です。



[ワークスペースログファイル生成]をクリックすると、ワークスペースのすべての変更箇所がワークスペースと同じ名前で拡張子が.logのファイルにログします。このファイルの位置は、ワークスペースファイルと同じディレクトリです。

[プロジェクトログの生成]をクリックすると、現在のワークスペースでのすべてのプロジェクトの変更箇所がプロジェクトと同じ名前で拡張子が.logのファイルにログします。このファイルの位置は、ワークスペースファイルと同じディレクトリです。

ワークスペースを保存するとログファイルを更新します。

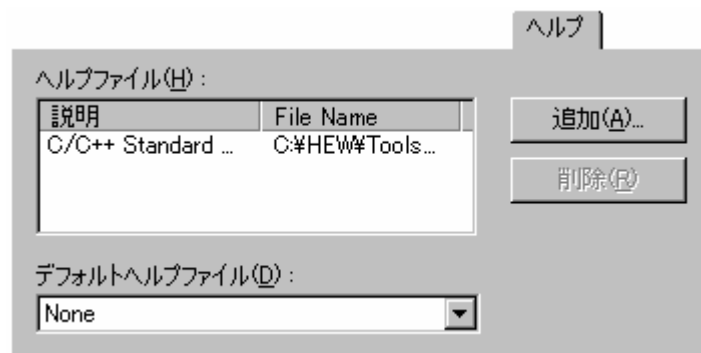
#### 6.5 ヘルプシステムを構築する

High-performance Embedded Workshop ではエディタウィンドウで状況依存ヘルプを提供します。エディタウィンドウでテキストを選択しF1キーを押すと、選択したテキストに関するヘルプを検索します。検索するヘルプファイルは[カスタマイズ]ダイアログボックスの[ヘルプ]タブに表示します。

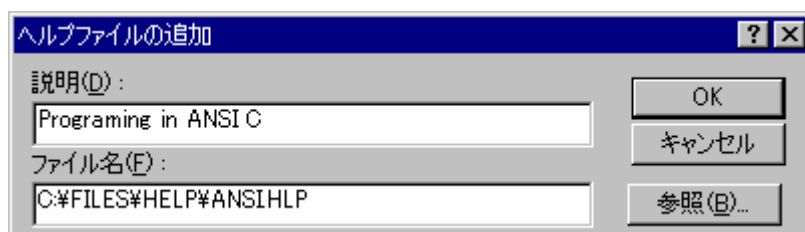
##### 新しいヘルプファイルを追加するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [ヘルプ]タブを選択してください。





2. [追加]ボタンをクリックしてください。[ヘルプファイルの追加]ダイアログボックスを開きます。



3. [説明]にヘルプファイルの説明を入力してください。
4. [ファイル名]にヘルプファイルへの絶対パスを入力してください (または[参照]ボタンをクリックしてファイルを選択してください)。
5. [OK]ボタンをクリックすると新しいヘルプファイルを定義します。

デフォルトのヘルプファイルを設定する場合は[デフォルトヘルプファイル]ドロップダウンリストからヘルプファイルを選択してください。F1 キーを押したときに任意のヘルプファイルを参照するには[(None)]を選択してください。

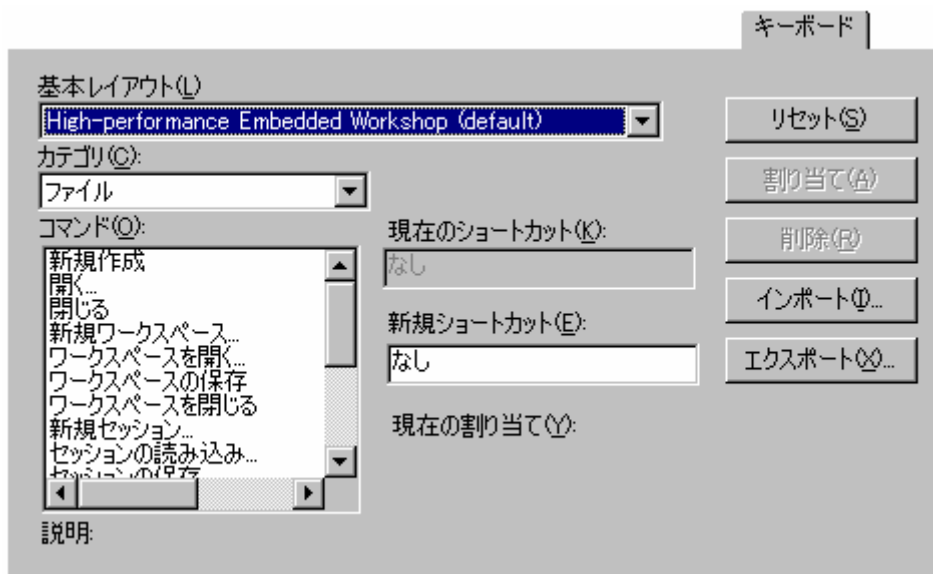
### ヘルプファイルを削除するには

1. [基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。
2. [ヘルプ]タブを選択してください。
3. 削除するファイルを選択して[削除]ボタンをクリックしてください。
4. [OK]ボタンをクリックしてください。

## 6.6 キーボードショートカットをカスタマイズする

High-performance Embedded Workshop では、キーボードショートカットを自由にカスタマイズできます。High-performance Embedded Workshop 以外のツールを以前使用していた場合、主な操作を特によく使うキーボードショートカットに変えることができます。

[基本設定 -> カスタマイズ]を選択してください。[カスタマイズ]ダイアログボックスを開きます。[キーボード]タブを選択してください。



このダイアログボックスで、High-performance Embedded Workshop のデフォルトのキーボードショートカット、あるいは、PD デバッガショートカットのどちらかを選択できます。キーボードのレイアウト全体を変更するには、[基本レイアウト] ドロップダウンリストから項目を 1 つ選んでください。デフォルトでは、High-performance Embedded Workshop 設定を使用しています。

このダイアログボックスでいろいろな操作ができます。

#### 新規キーボードショートカットを割り当てるには

1. 変更したいコマンドのメインメニューの[カテゴリ]を選択してください。メニューのあるコマンドのみ変更できます。いくつかの特殊なコマンド（カテゴリ名は"Global"）も変更できます。
2. ショートカットを割り当てたいコマンドを[コマンド]リストから選択してください。現在のショートカットを[現在のショートカット]に表示します。
3. [新規ショートカット]でキー入力してください。キーをいろいろな組み合わせることができます。例えば、[Ctrl+J]、[Ctrl+Shift+O]などです。選択したショートカットを既に使用している場合、[現在の割り当て]にコマンドを表示します。
4. [割り当て]ボタンをクリックしてください。
5. [OK]ボタンをクリックしてください。[OK]ボタンをクリックするまで変更は保存しません。

#### キーボードショートカットを削除するには

1. 変更したいコマンドのメインメニューの[カテゴリ]を選択してください。メニューのあるコマンドのみ変更できます。いくつかの特殊なコマンド（カテゴリ名は"Global"）も変更できます。
2. ショートカットを削除したいコマンドを[コマンド]リストから選択してください。現在のショートカットを[現在のショートカット]に表示します。
3. [削除]ボタンをクリックしてください。
4. [OK]ボタンをクリックしてください。

#### キーボードショートカットをすべてリセットするには

1. [リセット]ボタンをクリックしてください。すべてのショートカットが、現在選択しているキーボードの配置のデフォルト設定に戻ります。
2. [OK]ボタンをクリックしてください。[OK]ボタンをクリックするまで変更は保存しません。

[キーボード]タブでは、キーボードの設定を決められたファイルへインポートしたり、エクスポートしたりできます。これにより、あるマシンから別のマシンへ設定を転送できます。

#### キーボードショートカットをインポートするには

1. [基本レイアウト]リストから置き換えたいキーボードの配置を選択してください。
2. [インポート]ボタンをクリックしてください。[レイアウトのインポート]ダイアログボックスを開きます。
3. キーボードの配置をインポートするファイル名を選択してください。
4. [OK]ボタンをクリックしてください。

#### キーボードショートカットをエクスポートするには

1. [エクスポート]ボタンをクリックしてください。[レイアウトのエクスポート]ダイアログボックスを開きます。
2. 現在選択しているキーボードの配置の設定をエクスポートするファイル名を選択してください。
3. [OK]ボタンをクリックしてください。

## 6.7 基本設定の影響範囲

### 6.7.1 カスタマイズダイアログボックスの影響範囲

[基本設定 -> カスタマイズ]を選択すると[カスタマイズ]ダイアログボックスを開きます。[カスタマイズ]ダイアログボックスの設定内容は各タブの項目により影響範囲が異なります。下の表に各タブの項目の影響範囲を示します。

タブ	コントロール	影響範囲
ツールバー	すべて	ワークスペース単位
コマンド	すべて	システム全体
メニュー	アプリケーション内有効	システム全体
	ワークスペース内有効	ワークスペース単位
プレースホルダ	アプリケーション内有効カスタムプレースホルダ	システム全体
	ワークスペース内有効カスタムプレースホルダ	ワークスペース単位
デバッグ	外部デバッグツール	プロジェクト単位
	外部デバッグのディレクトリ	デフォルト（システム全体および各プロジェクト）
	コマンドラインオプション	プロジェクト単位
	セッションファイル	プロジェクト単位
	ダウンロードモジュール	プロジェクト単位
ログ	すべて	ワークスペース単位
ヘルプ	すべて	ワークスペース単位
キーボード	すべて	システム全体

### 6.7.2 オプションダイアログボックスの影響範囲

[基本設定 -> オプション]を選択すると[オプション]ダイアログボックスを開きます。[オプション]ダイアログボックスの設定内容はシステム全体に影響します。

## 6.8 ワークスペースオプションを指定する

High-performance Embedded Workshop では[オプション]ダイアログボックスでワークスペースの様々なオプションを設定できます。[基本設定 -> オプション]の[ワークスペース]タブを選択してください。

### 6.8.1 起動時に前回開いたワークスペースを開く

High-performance Embedded Workshop を終了すると最後に開いたワークスペースを保存します。次に High-performance Embedded Workshop を起動したとき、前回開いたワークスペースを自動的に開きます。

#### 起動時に前回開いたワークスペースを開くには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ワークスペース]タブを選択してください。
3. [起動時に前回開いたワークスペースを開く]チェックボックスをオンにしてください。
4. [OK]ボタンをクリックします。

起動時に前回開いたワークスペースを開く (Q)

### 6.8.2 ワークスペースを開いたときにファイルを表示する

High-performance Embedded Workshop は、ワークスペースを閉じるとき開いていたファイルを記憶します。そして、再びワークスペースを開くとき、High-performance Embedded Workshop は同じファイルを復帰させる（つまり開く）ことができます。これにより、中断したセッションを引き続き行うことができます。このチェックボックスをオンにすると、ワークスペースを開いたときに前回開いていたファイルを開きます。

#### ワークスペースを開いたときにファイルを表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ワークスペース]タブを選択してください。
3. [ワークスペースを開いたときにファイルを表示]チェックボックスをオンにしてください。
4. [OK]ボタンをクリックします。

ワークスペースを開いたときにファイル表示 (R)

### 6.8.3 ワークスペースを開いたときにワークスペース情報を表示する

多くのワークスペースを使うと、各ワークスペースの内容を覚えておくのは難しくなります。High-performance Embedded Workshop では各ワークスペースの説明を入力しておくことができます。ワークスペースを開くときは常にこの説明を表示します。

#### ワークスペースの説明を入力するには

1. ワークスペースウィンドウの[Projects]タブでワークスペースアイコンを選択してください。
2. 右クリックし、ポップアップメニューから[プロパティ]オプションを選択してください。[ワークスペースプロパティ]ダイアログボックスを開きます。
3. [情報]に説明を入力してください。
4. ワークスペースを開いたときに[ワークスペースプロパティ]ダイアログボックスを開くようにするには、[ワークスペースを開いたときにワークスペース情報の表示]チェックボックスをオンにしてください。このチェックボックスは[オプション]ダイアログボックスの[ワークスペース]タブの[ワークスペースを開いたときにワークスペース情報の表示]チェックボックスと同じ機能です。
5. [OK]ボタンをクリックします。

ワークスペースを開いたときにワークスペース情報の表示 (D)

#### 6.8.4 ツール実行前にワークスペースを保存する

このチェックボックスをオンにすると、ビルドフェーズを実行（[ビルド]、[すべてビルド]、[コンパイル]操作）する前や、バージョン管理コマンドを実行する前に、現在のワークスペースを保存できます。

##### ツール実行前にワークスペースを保存するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ワークスペース]タブを選択してください。
3. [ツール実行前にワークスペースの保存]チェックボックスをオンにしてください。
4. [OK]ボタンをクリックします。

ツール実行前にワークスペースの保存(S)

#### 6.8.5 ワークスペース保存前に確認する

[ツール実行前にワークスペースを保存する]機能を使用する場合、ワークスペースを保存する前に、High-performance Embedded Workshop は確認の画面を表示します。ツール実行前のワークスペースの保存について、詳細は「6.8.4 ツール実行前にワークスペースを保存する」を参照してください。

##### ワークスペース保存前に確認の画面を表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ワークスペース]タブを選択してください。
3. [ワークスペース保存前に確認]チェックボックスをオンにしてください。
4. [OK]ボタンをクリックします。

ワークスペース保存前に確認(P)

#### 6.8.6 セッション保存前に確認する

このチェックボックスをオンにすると、High-performance Embedded Workshop はセッションをディスクに保存する前にプロンプトを表示します。

##### セッションをディスクに保存する前にプロンプトを表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ワークスペース]タブを選択してください。
3. [セッション保存前に確認]チェックボックスをオンにしてください。
4. [OK]ボタンをクリックします。

セッション保存前に確認(V)

#### 6.8.7 自動バックアップ機能を有効にする

このチェックボックスをオンにすると、指定した時間間隔（デフォルト 10 分）で、ワークスペース、プロジェクト、およびセッションファイルをバックアップします。加えた変更はすべて、テンポラリファイルに保存します。

### 自動バックアップ機能を有効にするには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ワークスペース]タブを選択してください。
3. [自動バックアップ時間間隔]チェックボックスをオンにしてください。
4. 自動バックアップ機能を起動する時間（分）を選択してください（最大 60 分）。
5. [OK]ボタンをクリックします。



ワークスペースを開いて以下のダイアログボックスが表示される場合、前回そのワークスペースを使用したときに問題が発生したことを意味します。



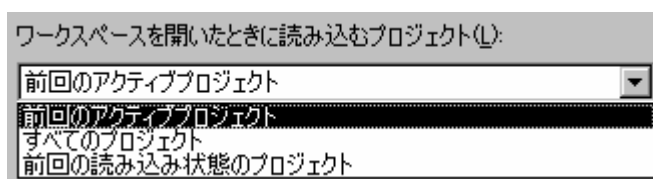
ファイルを回復するには、回復したいファイルの名前の横にあるチェックボックスをオンにし、[OK]ボタンをクリックしてください。[キャンセル]ボタンをクリックすると、自動バックアップファイルを削除し、元のファイルからロードを行います。

### 6.8.8 ワークスペースを開いたときに読み込むプロジェクトを設定する

ワークスペースを開いたときに読み込むプロジェクトを設定できます。この設定は、複数のプロジェクトを含むワークスペースに関連があり、アプリケーションシステム全体に影響します。

#### ワークスペースを開いたときに読み込むプロジェクトを設定するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ワークスペース]タブを選択してください。



3. [ワークスペースを開いたときに読み込むプロジェクト]リストからワークスペースを開くときの動作を設定します。以下のいずれかの操作を選択してください。



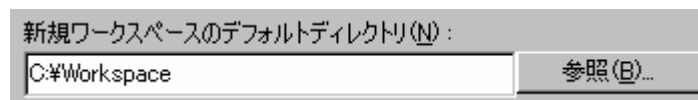
- [前回のアクティブプロジェクト]  
前回ワークスペースを閉じたときにアクティブだったプロジェクトだけを読み込みます。
  - [すべてのプロジェクト]  
ワークスペースに登録されているすべてのプロジェクトを読み込みます。
  - [前回の読み込み状態のプロジェクト]  
前回ワークスペースを閉じたときに読み込んでいたプロジェクトを読み込みます。
4. [OK]ボタンをクリックします。

### 6.8.9 新規ワークスペースのデフォルトディレクトリを変更する

新しいワークスペースを作成すると[新規プロジェクトワークスペース]ダイアログボックスが開きます。このダイアログボックスにはその新しいワークスペースが作成されるディレクトリを入力するフィールドがあります。デフォルトでは Workspace ディレクトリです。しかし、他のディレクトリにデフォルトを変更したい場合、このフィールドにそのディレクトリを入力するか、[参照]ボタンで位置を指定してください。

#### 新規ワークスペースのデフォルトディレクトリを変更するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ワークスペース]タブを選択してください。
3. 新しいワークスペースを作成するディレクトリを[新規ワークスペースのデフォルトディレクトリ]にを入力するか、[参照]ボタンをクリックして参照してください。
4. [OK]ボタンをクリックします。



### 6.9 外部エディタを使う

High-performance Embedded Workshop エディタ以外のエディタ（外部エディタ）も使うことができます。外部エディタを指定した場合、以下を操作すると外部エディタで開きます。

- [ファイル -> 開く]メニューを選択すると表示される[ファイルを開く]ダイアログボックスでファイルを選択したとき
- [ファイル -> 最近使ったファイル]メニューからファイルを選択したとき
- ワークスペースウィンドウの[Projects]タブのファイルをダブルクリックしたとき
- ワークスペースウィンドウの[Navigation]タブのエントリをダブルクリックしたとき
- アウトプットウィンドウの[Build]タブのエラーまたはウォーニングをダブルクリックしたとき
- アウトプットウィンドウの[Find in Files 1]タブのエントリをダブルクリックしたとき
- アウトプットウィンドウの[Find in Files 2]タブのエントリをダブルクリックしたとき
- ワークスペースウィンドウの[Projects]タブのファイル上で右クリックしポップアップメニューから[開く <ファイル名> (外部エディタ)]を選択したとき
- [編集 -> ソースブレイクポイント]メニューを選択すると表示される[ブレイクポイント]ダイアログボックスで、[コードの編集]ボタンをクリックしたとき
- Windows® Explorer から High-performance Embedded Workshop ウィンドウ上にファイルをドラッグアンドドロップしたとき（ただしワークスペースウィンドウの[Projects]タブのアクティブプロジェクトやユーザフォルダ上にファイルをドラッグアンドドロップしたときは、プロジェクトにファイルを追加しません。）



## 外部エディタを指定するには



1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [エディタ]タブを選択してください。
3. [外部エディタの使用]チェックボックスをオンにしてください。[外部エディタ]ダイアログボックスを開きます。



4. [コマンド]に実行可能ファイルのパス（引数をのぞいて）を入力してください。
5. [ファイルを開く時の引数]にファイルを開くために必要な引数を入力してください。開くファイルのパスには \$(FULLFILE) プレースホルダを使ってください。
6. [行番号を指定してファイルを開く時の引数]にファイルの特定の行を開くために必要な引数を入力してください。開くファイルのパスには \$(FULLFILE) プレースホルダを使ってください。また、カーソルを最初に置く行の番号には \$(LINE) プレースホルダを使ってください。
7. [詳細>>]ボタンをクリックすると、次のファイルを開く操作の場合に外部エディタまたは内蔵（High-performance Embedded Workshop）エディタのどちらでファイルを開くかを選択できます。
  - [ファイル -> 開く]メニューから  
[ファイルを開く]ダイアログボックスでファイルを選択したとき  
デフォルトは"内蔵エディタ"が選択されています。
  - ワークスペースウィンドウ上でダブルクリック  
ワークスペースウィンドウの[Projects]タブのファイルをダブルクリックしたとき  
ワークスペースウィンドウの[Navigation]タブのエントリをダブルクリックしたとき  
デフォルトは"外部エディタ"が選択されています。
  - アウトプットウィンドウ上でダブルクリック  
アウトプットウィンドウの[Build]タブのエラーまたはウォーニングをダブルクリックしたとき  
アウトプットウィンドウの[Find in Files 1]タブのエントリをダブルクリックしたとき  
アウトプットウィンドウの[Find in Files 2]タブのエントリをダブルクリックしたとき  
デフォルトは"外部エディタ"が選択されています。
  - HEW ウィンドウにドラッグアンドドロップ  
Windows® Explorer から High-performance Embedded Workshop ウィンドウ上にファイルをドラッグアンドドロップしたとき（ただしワークスペースウィンドウの[Projects]タブのアクティブプロジェクトやユーザフォルダ上にファイルをドラッグアンドドロップしたときは、プロジェクトにファイルを追加します。このときファイルは開きません。）  
デフォルトは"内蔵エディタ"が選択されています。

- [OK]ボタンをクリックします。

[外部エディタの使用]チェックボックスをオンにした場合、以下を操作すると常に外部エディタで開きます。

- [編集 -> ソースブレークポイント]メニューを選択すると表示される[ブレークポイント]ダイアログボックスで、[コードの編集]ボタンをクリックしたとき
- [ファイル-> 最近使ったファイル]メニューからファイルを選択したとき

また、ワークスペースウィンドウの[Projects]タブのファイル上で右クリックしポップアップメニューを表示すると、[開く <ファイル名>]の下に[開く <ファイル名> (外部エディタ)]が追加されます。

#### 注：

外部エディタを使う場合、以下のことに注意してください。

- どのように起動しても、外部エディタを起動するたびに、エディタは新規に起動します。
- ビルド操作 ([ビルド]、[すべてビルド]、[コンパイル]) を行う前にファイルを保存してください。

## 6.10 表示するフォントのカスタマイズ

High-performance Embedded Workshop のコンポーネントは違った外観で表示させることが可能です。表示するフォントやテキストの色を変更できます。

### ウィンドウの外観を変更するには

- [基本設定 -> 表示の形式]を選択してください。[表示形式]ダイアログボックスを開きます。
- 外観を変更したい表示を選択してください。すべての表示を変更するには、[All views]のカテゴリを選択してください。
- ツリーの項目を展開し、外観の変更が可能なすべての項目を表示します。
- 項目を選択してください。本ダイアログボックスの右側にある、タブの変更に注意してください。
- [フォント]または[カラー]を変更してください。
- 変更を保存するには[OK]ボタンをクリックしてください。



各ウィンドウに表示されているテキストの色やフォントなどが[表示形式]ダイアログボックスでカスタマイズ可能です。

項目	カテゴリ名	カスタマイズ対象
Source	Selected Text	エディタウィンドウおよび逆アセンブリウィンドウ（ソースモード）に表示されているテキストを選択した部分
	PC Line Highlight	[オプション]ダイアログボックスの[エディタ]タブで、[PCカーソル行のハイライト表示]チェックボックスがオンの場合に、エディタウィンドウおよび逆アセンブリウィンドウ（ソースモード）のPC（プログラムカウンタ）の位置がある行
	その他、各々のファイルタイプのシンタックスに基づいたキーワードグループ	エディタウィンドウおよび逆アセンブリウィンドウ（ソースモード）に表示されているコメントまたはキーワード*1 *1. 制御文、式文、宣言文、プリプロセッサ制御文、データ文、データ群、ディレクティブ、値
差分	LHS Different Lines	差分ウィンドウの左側のペインに表示されている不一致テキスト行
	LHS Moved Lines	差分ウィンドウの左側のペインに表示されている移動されたテキスト行
	RHS Different Lines	差分ウィンドウの右側のペインに表示されている不一致テキスト行
	RHS Moved lines	差分ウィンドウの右側のペインに表示されている移動されたテキスト行
レジスタ	変更	レジスタウィンドウに表示されている値が変更された部分のテキスト
メモリ	アクセス履歴あり	メモリウィンドウに表示されているカバレッジ計測時のコード実行部分のテキスト
	アクセス履歴なし	メモリウィンドウに表示されているカバレッジ計測時のコード未実行部分のテキスト
	カバレッジ計測領域外	メモリウィンドウに表示されているカバレッジ計測領域外の部分のテキスト
	メモリ領域外	メモリウィンドウに表示されているメモリ領域外の部分のテキスト
	変更	メモリウィンドウに表示されている値が変更された部分のテキスト
Output	Build Error	アウトプットウィンドウの[Build]タブに表示されているエラーメッセージ行
	Build Warning	アウトプットウィンドウの[Build]タブに表示されているウォーニングメッセージ行
	Information	アウトプットウィンドウの[Build]タブに表示されているインフォメーションメッセージ行
逆アセンブリ	PC Line Highlight	[オプション]ダイアログボックスの[エディタ]タブで、[PCカーソル行のハイライト表示]チェックボックスがオンの場合にエディタウィンドウおよび逆アセンブリウィンドウ（混合モードおよび逆アセンブリモード）でPC（プログラムカウンタ）の位置がある行
	ソース行	エディタウィンドウおよび逆アセンブリウィンドウ（混合モード）に表示されているソース行
全般	Text	各ウィンドウに表示されているテキスト

## 6.11 バーチャルデスクトップを使う

High-performance Embedded Workshop はバーチャルデスクトップの概念を導入しています。ウィンドウ配置をボタンのクリックで切り替えることができます。ボタンをクリックすると、ウィンドウ配置の設定によりツールバーとウィンドウが表示または非表示になります。

最大4パターンのデスクトップ構成が可能です。セッションを保存すると、各構成のウィンドウ位置がセッションファイルに保存されます。構成を切り替えるだけで他のウィンドウへアクセスできます。ツールバーとウィンドウはバーチャルデスクトップ構成に依存します。ソースファイルは、バーチャルデスクトップのシステムから独立しているため表示します。

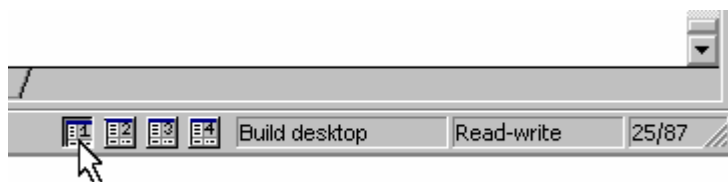
### デスクトップ構成に新しい名前を付けるには

1. [ウィンドウ -> バーチャルデスクトップ -> デスクトップマネージャ]を選択してください。[デスクトップマネージャ]ダイアログボックスを開きます。
2. 名前を変更するデスクトップ構成を選択してください。
3. [名前の変更]ボタンをクリックしてください。
4. 新しい名前を入力し、[OK]ボタンをクリックしてください。
5. [OK] ボタンをクリックしてください。

### デスクトップ構成を切り替えるには

デスクトップ構成の切り替えには、いくつかの方法があります。

もっとも簡単な方法は以下の通りです。ステータスバー上のバーチャルデスクトップボタンをクリックします。



この例では、選択されたデスクトップ番号は"1"で、"Build"という名前が付いています。ボタンの右側のボックスに"Build desktop"と表示しています。別のデスクトップボタンをクリックするとボタンの右側のボックスの表示も切り替わります。一度クリックすると、新しい構成の形式でウィンドウを読み込みます。

もう1つの方法は以下の通りです。

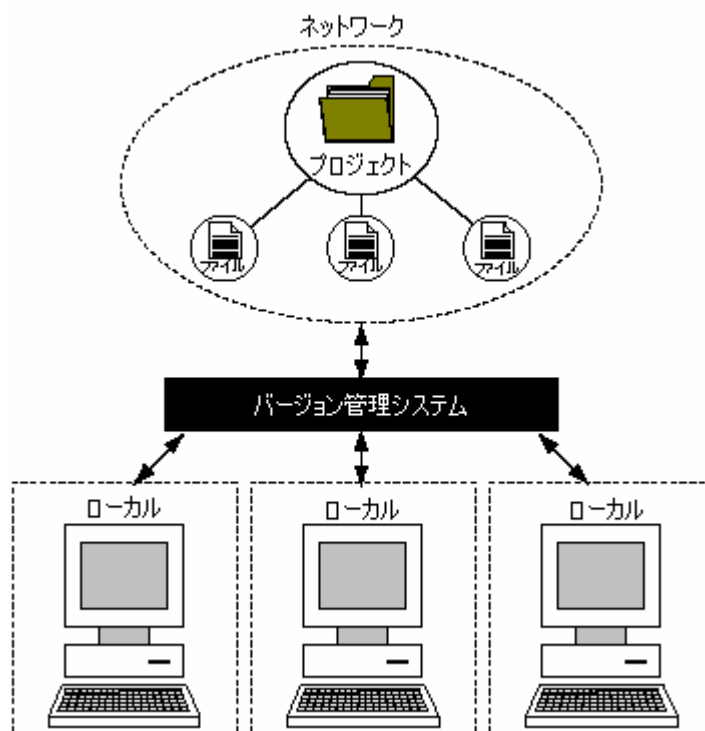
1. [ウィンドウ -> バーチャルデスクトップ]を選択してください。
2. サブメニューを選択してください。
3. 表示したいデスクトップ構成を選択してください。選択したデスクトップ構成のサブメニューにチェックマークが付きます。

## 7. バージョン管理

High-performance Embedded Workshop はバージョン管理システムと接続できます。プロジェクトでバージョン管理システムを使用する理由を以下に示します。

- プロジェクト開発環境の統合性を維持するため
- プロジェクトのバージョンを記録・保存するため
- ソースファイルに対するバージョン管理を行い、複数のユーザが1つのプロジェクトを共同開発できるようにするため

バージョン管理システムを使用するプロジェクトの一般例を示します。ここでは複数のユーザがソースコードを相互参照するために同じ共有ネットワークドライブを使用しています。バージョン管理システムはソースファイルの参照や更新を管理するために使用します。



- **High-performance Embedded Workshop のバージョン管理の対象ファイル**

High-performance Embedded Workshop のバージョン管理の対象ファイルは、標準では以下の種類があります。

ファイルの種類	拡張子グループ名	ファイルの拡張子	ファイルのサポート CPU 種別
ワークスペースファイル	HEW Workspaces	hws	共通
プロジェクトファイル	HEW Projects	hwp	共通
バージョン管理用ファイル	Configuration File	hvc	共通
ソースファイル	C source file	c	共通
	C++ source file	cpp	<ul style="list-style-type: none"> <li>• SuperH ファミリ</li> <li>• H8SX, H8S, H8 ファミリ</li> <li>• RX ファミリ</li> <li>• M32R ファミリ</li> </ul>
	Assembly source file	a30	<ul style="list-style-type: none"> <li>• M16C ファミリ</li> <li>• R8C ファミリ</li> </ul>

ファイルの種類	拡張子グループ名	ファイルの拡張子	ファイルのサポート CPU 種別
		ms	• M32R ファミリ
		a74	• 740 ファミリ
		asm	• SuperH ファミリ
		mar	• H8SX, H8S, H8 ファミリ
		src	• RX ファミリ
インクルードファイル	C header file	h	共通
	C++ header file	hpp	• SuperH ファミリ • H8SX, H8S, H8 ファミリ • RX ファミリ • M32R ファミリ
	Assembly include file	inc	共通

上記以外のファイルもプロジェクトに追加すると、バージョン管理の対象にできます。

#### • High-performance Embedded Workshop と接続可能なバージョン管理システム

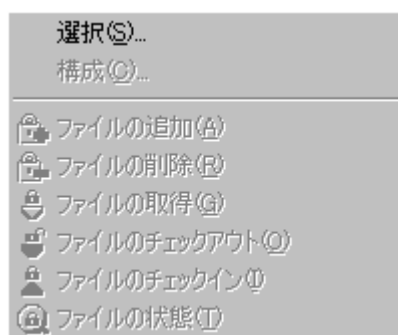
High-performance Embedded Workshop はバージョン管理システム自体を提供していません。あらかじめバージョン管理システムがマシンにインストールされている必要があります。High-performance Embedded Workshop の GUI インタフェースを使用しバージョン管理システムと接続が可能です。

はじめに High-performance Embedded Workshop と接続するバージョン管理システムを選択します。High-performance Embedded Workshop と接続可能なバージョン管理システムは下記の 2 種類のいずれかです。

分類	バージョン管理システム名	説明
カスタマイズ設定によるバージョン管理	Custom VCS	MS-DOS コマンドプロンプトからコマンドラインコマンドを実行するバージョン管理システムを対象としています。(RCS、CVS、PVCS など)バージョン管理システムのコマンドと High-performance Embedded Workshop の GUI インタフェースを関連付けし定義する必要があります。
Microsoft® Visual SourceSafe 5.0、6.0、または 2005 によるバージョン管理	Microsoft Visual SourceSafe	Microsoft® Visual SourceSafe 5.0、6.0、または 2005 を対象としています。主な機能をあらかじめ定義済みです。

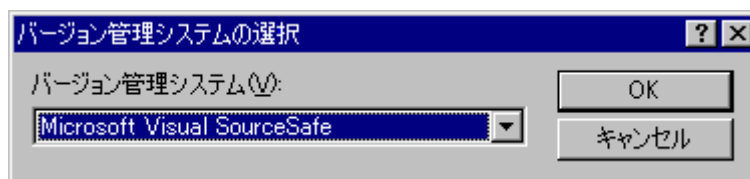
## 7.1 バージョン管理システムを選択する

最初に、[ツール -> バージョン管理]の 8 つのメニューオプションは以下のように表示します。このとき、まだ現在のワークスペースでバージョン管理システムを選択していないため、[ツール -> バージョン管理 -> 選択]だけが使用できます。



### バージョン管理システムを選択するには

1. [ツール -> バージョン管理 -> 選択]を選択します。[バージョン管理システムの選択]ダイアログボックスが開きます。このダイアログボックスにはサポートするバージョン管理システムをすべて表示します。



2. [バージョン管理システム]リストからバージョン管理システムを選択します。
  3. [OK]ボタンをクリックします。
- バージョン管理システムに"Custom VCS"を選択した場合は、[ツール -> バージョン管理 -> 構成]が使用できるようになります。「8. カスタムバージョン管理システム」を参照してください。
  - バージョン管理システムに"Microsoft Visual SourceSafe"を選択した場合は、[ツール -> バージョン管理 -> 選択]の他のメニューオプションが使用できるようになります。「9. Visual SourceSafe バージョン管理システム」を参照してください。

#### バージョン管理システムの選択を解除するには

1. [ツール -> バージョン管理 -> 選択]を選択します。[バージョン管理システムの選択]ダイアログボックスが開きます。
2. [バージョン管理システム]リストから"<None>"を選択します。
3. [OK]ボタンをクリックします。

## 7.2 バージョン管理設定内容のインポートとエクスポート

ワークスペースごとに異なるバージョン管理設定を行うことができます。High-performance Embedded Workshop ではそれぞれのバージョン管理設定を保存して他のワークスペースでインポートできます。これにより、複数のワークスペースで何度も同じバージョン管理設定を行う必要がなく、時間を節約できます。

#### バージョン管理設定をエクスポートするには

1. [ツール -> バージョン管理 -> 構成]を選択してください。[バージョン管理設定]ダイアログボックスが開きます。
2. [エクスポート]ボタンをクリックしてください。[現在のバージョン管理構成をエクスポート]ダイアログボックスが開きます。
3. 設定内容を保存するディレクトリを選択してください。
4. ファイル名を入力し[保存]ボタンをクリックしてください。

#### バージョン管理設定をインポートするには

1. [ツール -> バージョン管理 -> 構成]を選択してください。[バージョン管理設定]ダイアログボックスが開きます。
2. [インポート]ボタンをクリックしてください。[バージョン管理構成をインポート]ダイアログボックスが開きます。
3. インポートするファイル(\*.HVC)を選択してください。
4. [インポート]ボタンをクリックしてください。



## 8. カスタムバージョン管理システム

「7.1 バージョン管理システムを選択する」でバージョン管理システムに"Custom VCS"を選択した場合、High-performance Embedded Workshop と Custom VCS を接続するには次を定義していただく必要があります。

- バージョン管理メニューオプションを定義し、そのオプションに関連したカスタムバージョン管理のコマンド実行ファイル(.EXE)の位置、コマンドパラメータ、およびバージョン管理ツールコマンドを実行した結果の制御方法などの定義
- バージョン管理対象ファイルの位置(ディレクトリマッピング)、および環境変数の定義
- バージョン管理コマンド実行時の制御、およびユーザ設定など一般的なオプションの定義

これらを定義すると、High-performance Embedded Workshop のメニューオプションまたはツールバーボタンを選択したときにカスタムのバージョン管理システムのコマンドが実行でき、その実行結果を High-performance Embedded Workshop で見ることが可能になります。

バージョン管理システムのインストールや設定は、各バージョン管理システムのマニュアルを参照してください。

カスタムバージョン管理システムと High-performance Embedded Workshop との具体的な操作方法は、「8.11 カスタムバージョン管理システムの使用例」を参照してください。

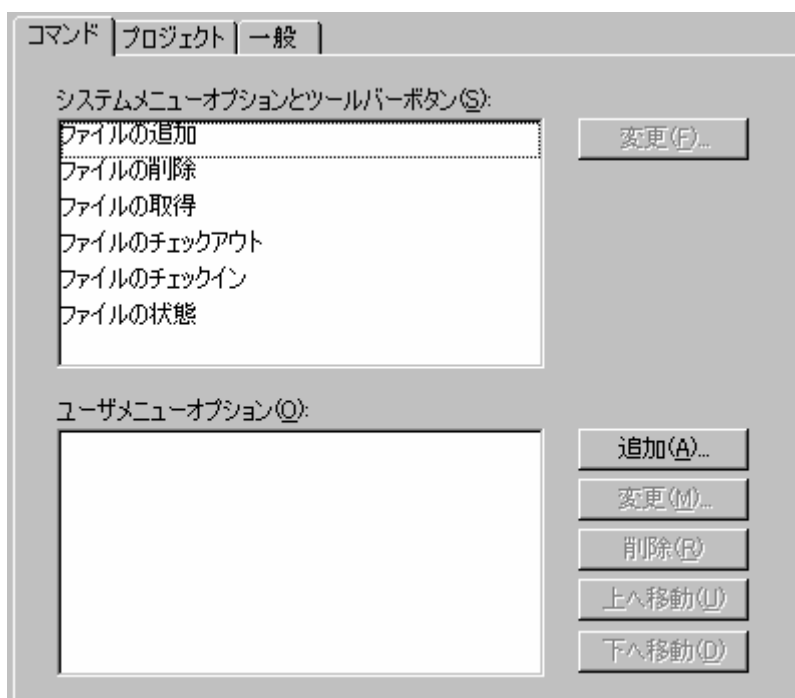
### 8.1 バージョン管理メニューオプションを定義する

カスタムのバージョン管理システムでは、[ツール->バージョン管理]サブメニューからオプションを選択するか、バージョン管理ツールバーボタンでバージョン管理コマンド実行ファイルを起動できます。関連するコマンドを実行し、出力をアウトプットウィンドウの[Version Control]タブに表示します。

**バージョン管理メニューオプションまたはツールバーボタンによりコマンドを実行するには**

1. ワークスペースウィンドウの[Projects]タブからバージョン管理コマンドを適用する項目(ワークスペース、プロジェクト、フォルダ、またはファイルなど)を選択してください。右クリックしてポップアップメニューを表示し、[バージョン管理]サブメニューから定義したメニューオプションを選択できます。メニューオプションを選択すると、ワークスペースウィンドウで選択した項目に応じたファイルに対し、メニューオプションに関連付けたコマンドが実行されます。
2. 例えば、ワークスペースアイコンを選択すると、そのワークスペースのプロジェクトのすべてのファイルがバージョン管理コマンドに渡ります。これには、High-performance Embedded Workshop のシステムファイルも含まれます。
3. [ツール->バージョン管理]サブメニューからメニューオプションを選択するか、[バージョン管理]ツールバーボタンを選択してください。

カスタムバージョン管理を利用すると、バージョン管理システムの指定を任意に定義できます。[ツール->バージョン管理->構成]を選択し[バージョン管理設定]ダイアログボックスを開きます。



[コマンド]タブには2つのメニューオプションのリストがあります。

- **システム定義メニューオプション**([システムメニューオプションとツールバーボタン])

1つめのリスト[システムメニューオプションとツールバーボタン]は、[ツール->バージョン管理]サブメニューに常に表示されているメニューオプションです。これに対応する[バージョン管理]ツールバーがあります。このリストには最もよく使用されるバージョン管理コマンドに関連した6種類のメニューオプションが表示されています。

- **ユーザ定義メニューオプション**([ユーザメニューオプション])

2つめのリスト[ユーザメニューオプション]は、ユーザ定義の追加メニューオプションで、定義すると[ツール->バージョン管理]サブメニューの最後に追加されます。システムメニューオプションに表示されていないメニューオプションを追加する場合はこのリストで定義してください。ユーザ定義のメニューオプションはバージョン管理ツールバーには表示されません。

### 8.1.1 システム定義メニューオプション

バージョン管理ツールバーボタンは6種類あります。これらのボタンは、最もよく使用されるバージョン管理コマンドへのショートカットとなります。最初にワークスペースを作成すると、これらのボタンはアクティブではありません。その理由は、まだバージョン管理コマンドがツールバーボタンに関連付けられていないためです。

ツールバーボタンは、[ツール->バージョン管理]サブメニューの6つのメニューオプションと同等です。[ツール->バージョン管理->ファイルの取得]はバージョン管理から[ファイルの取得]ツールバーボタンをクリックするのと同じ効果があります。ツールバーボタンの表示は固定されているので、ツールバーボタンをクリックしたときに実行するコマンドを定義してください。

ツールバーまたは[ツール->バージョン管理]サブメニューのシステム定義オプションからコマンドを起動するには、まず、バージョン管理システムを起動したときに実行する関連したコマンドを定義しなければなりません。オプション名とその説明を下記に示します。

オプション名	説明
ファイルの追加	バージョン管理システムに選択したファイルを追加する
ファイルの削除	バージョン管理システムから選択したファイルを削除する
ファイルの取得	バージョン管理システムから選択したファイルの読み取り専用ローカルファイルを取得する
ファイルのチェックアウト	バージョン管理システムから選択したファイルの書き込み可能なローカルファイルを取得する
ファイルのチェックイン	バージョン管理システムに選択したファイルのローカルコピーを戻し更新する
ファイルの状態	選択したファイルの状態を表示する

### システムメニューオプションやツールバーボタンから実行するコマンドを定義するには

1. [ツール -> バージョン管理 -> 構成]を選択します。[バージョン管理設定]ダイアログボックスが開きます。
2. [システムメニューオプションおよびツールバーボタン]リストからコマンドを定義するオプションを選択し[変更]ボタンをクリックすると、[定義コマンド]ダイアログボックスを開きます。
3. [追加]ボタンで選択したメニューオプションにコマンドを定義してください。詳細は「8.2 バージョン管理コマンドを定義する」を参照してください。
4. [OK]ボタンをクリックして[定義コマンド]ダイアログボックスを閉じてください。
5. [OK]ボタンをクリックして[バージョン管理設定]ダイアログボックスを閉じてください。

#### 8.1.2 ユーザ定義メニューオプション

ユーザ定義のメニューオプションは複数作成できます。メニューオプション名も自由に付けられます。既にユーザ定義のメニューオプションを作成している場合、コマンドの定義の変更、削除、または、メニューオプションの表示順序も指定できます。

ユーザ定義のメニューオプションはバージョン管理ツールバーには表示されません。

### 新しいバージョン管理メニューオプションを作成するには

1. [ツール -> バージョン管理 -> 構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [ユーザメニューオプション]リストの横にある[追加]ボタンをクリックしてください。[メニューオプションの追加]ダイアログボックスを開きます。
3. [オプション]にメニューオプション名を入力してください。
4. [追加]ボタンをクリックしてください。[コマンド追加]ダイアログボックスを開きます。詳細は「8.2 バージョン管理コマンドを定義する」を参照してください。
5. [OK]ボタンをクリックし、[コマンド追加]ダイアログボックスを閉じてください。
6. [OK]ボタンをクリックし、[メニューオプションの追加]ダイアログボックスを閉じてください。
7. [OK]ボタンをクリックし、[バージョン管理設定]ダイアログボックスを閉じてください。

### 既存のバージョン管理メニューオプションの定義コマンドを変更するには

1. [ツール -> バージョン管理 -> 構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [ユーザメニューオプション]リストから変更するメニューオプションを選択し、[変更]ボタンをクリックしてください。[定義コマンド]ダイアログボックスを開きます。
3. [コマンド]リストから変更するメニューオプションを選択し、[変更]ボタンをクリックしてください。[コマンド変更]ダイアログボックスを開きます。詳細は「8.2 バージョン管理コマンドを定義する」を参照してください。
4. [OK]ボタンをクリックし、[コマンド変更]ダイアログボックスを閉じてください。

5. [OK]ボタンをクリックし、[定義コマンド]ダイアログボックスを閉じてください。
6. [OK]ボタンをクリックし、[バージョン管理設定]ダイアログボックスを閉じてください。

#### 既存のバージョン管理メニューオプションを削除するには

1. [ツール->バージョン管理->構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [ユーザメニューオプション]リストから削除するメニューオプションを選択し、[削除]ボタンをクリックしてください。
3. [OK]ボタンをクリックし、[バージョン管理設定]ダイアログボックスを閉じてください。

#### バージョン管理メニューオプションの表示順序を変更するには

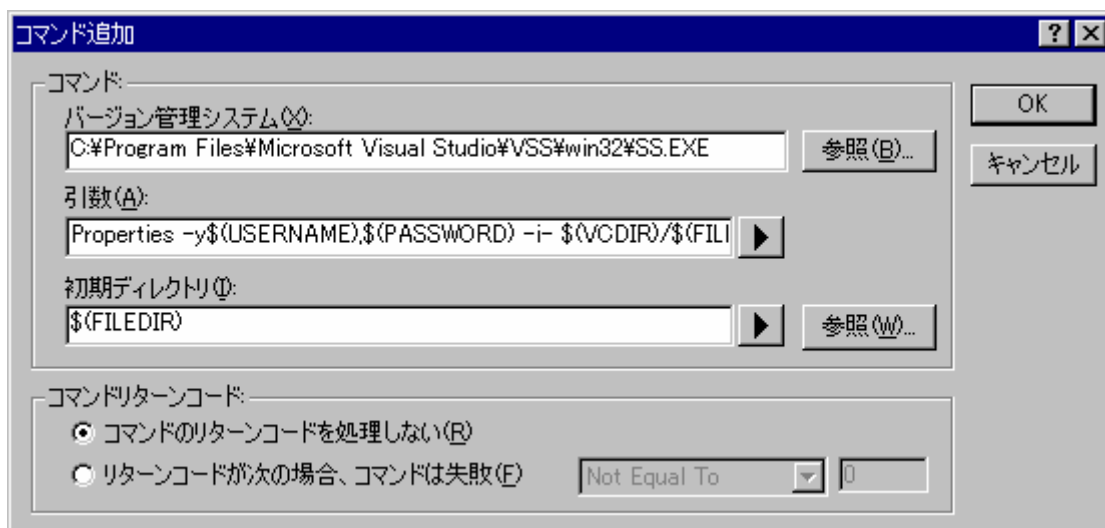
1. [ツール->バージョン管理->構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. 移動するメニューオプションを選択し、上に移動する場合は[上へ移動]ボタンを、下に移動する場合は[下へ移動]ボタンを必要に応じてクリックしてください。
3. [OK]ボタンをクリックし、[バージョン管理設定]ダイアログボックスを閉じてください。

## 8.2 バージョン管理コマンドを定義する

バージョン管理コマンドは、[定義コマンド]ダイアログボックスに表示されます。コマンドを定義する数には制限はなく、実行する順番を指定できます。既存のコマンドは変更または削除できます。

#### バージョン管理コマンドを定義するには

1. [定義コマンド]ダイアログボックスの[追加]ボタンをクリックしてください。[コマンド追加]ダイアログボックスを開きます。



2. [バージョン管理システム]にコマンド実行ファイルの絶対パスを入力するか、[参照]ボタンをクリックしてコマンド実行ファイルの位置を指定してください。
3. [引数]にコマンドラインパラメータを入力してください。
4. [初期ディレクトリ]に実行ファイルを起動するディレクトリを入力するか、[参照]ボタンをクリックして起動ディレクトリを指定してください。ただし、特に起動ディレクトリを指定する必要がない場

合は、\$(FILEDIR) プレースホルダを指定してファイルのディレクトリと同じディレクトリでコマンドを実行してください。

5. [コマンドリターンコード]オプションを選択してください。オプションの設定は「8.5 バージョン管理コマンドリターンコードオプション」を参照してください。
6. [OK]ボタンをクリックすると新しいコマンドを定義します。

#### 既存のバージョン管理コマンドを修正するには

1. 変更するコマンドを[定義コマンド]ダイアログボックスの[コマンド]リストから選択してください。
2. [変更]ボタンをクリックしてください。[コマンド変更]ダイアログボックスを開きます。
3. 必要に応じて修正を行い、[OK]ボタンをクリックしてください。

#### 既存のバージョン管理コマンドを削除するには

1. 削除するコマンドを[定義コマンド]ダイアログボックスの[コマンド]リストから選択してください。
2. [削除]ボタンをクリックしてください。

#### バージョン管理メニューオプションの順序を変更するには

1. 移動するメニューオプションを[定義コマンド]ダイアログボックスの[コマンド]リストから選択してください。
2. 必要に応じて[上へ移動]または[下へ移動]ボタンをクリックしてください。

### 8.3 引数を指定する

引数は正しく指定しなければなりません。もし間違っていると、バージョン管理システムが正しく動作しません。また、1つのバージョン管理コマンドを複数のファイルに適用する場合があるため、引数をフレキシブルな方法で指定することが重要です。これを行うために、[引数]にはプレースホルダボタン(▶)があります(プレースホルダの詳細は、「リファレンス 5. プレースホルダ」を参照してください)。プレースホルダボタンをクリックすると、使用できるプレースホルダをポップアップメニューで表示します。以下の表に各プレースホルダの説明と実際の値を示します。

ポップアップメニュー	プレースホルダ	実際の値
User login name	\$(USERNAME)	現在のユーザログイン ([一般]タブ)
User login password	\$(PASSWORD)	現在のユーザパスワード ([一般]タブ)
Version control directory	\$(VCDIR)	仮想バージョン管理マップ ([プロジェクト]タブ)
Comment	\$(COMMENT)	コマンド実行前に指定したコメント
File path + name	\$(FULLFILE)	操作するファイル名とその絶対パス
Filename	\$(FILENAME)	操作するファイル名 (拡張子を含む)
File leaf	\$(FILELEAF)	操作するファイル名 (拡張子を含まない)
File extension	\$(EXTENSION)	操作するファイルの拡張子
File directory	\$(FILEDIR)	操作するファイルのディレクトリ
Configuration directory	\$(CONFIGDIR)	現在のコンフィグレーションディレクトリ
Project directory	\$(PROJDIR)	現在のプロジェクトのディレクトリ
Workspace directory	\$(WORKSPDIR)	現在のワークスペースディレクトリ
Temp directory	\$(TEMPDIR)	テンポラリディレクトリ
Command directory	\$(EXEDIR)	バージョン管理実行ディレクトリ
Windows directory	\$(WINDIR)	Windows® がインストールされているディレクトリ
Windows system directory	\$(WINSYSDIR)	Windows® のシステムファイルがあるディレクトリ
Workspace name	\$(WORKSPNAME)	現在のワークスペース名
Project name	\$(PROJECTNAME)	現在のプロジェクト名
Configuration name	\$(CONFIGNAME)	現在のコンフィグレーション名



## 8.4 コメントを指定する

バージョン管理コマンドに"\$ (COMMENT)"プレースホルダを含む場合、コマンド実行時に[コメント入力]ダイアログボックスが表示されるのでコメントを入力してください。

コメントはコマンドごとに指定できます。また、[全てのファイルにコメントを適用]チェックボックスをオンにし[OK]ボタンをクリックすると、すべてのファイルに同じコメントが指定できます。

## 8.5 バージョン管理コマンドリターンコードオプション

バージョン管理コマンドが実行されると、その出力はアウトプットウィンドウの[Version Control]タブに表示します。

コマンドの実行が完了するとリターンコードを取得します。コマンドを定義する際、このリターンコードを使用し、コマンド実行中にエラーが発生した、残りのコマンドが実行されるべきでないことを示すことができます (例: abort)。[コマンドリターンコード]グループに含まれる管理機能を使用すれば、この動作を指定できます。

コマンドのリターンコードでエラーを示すことができるならば、[リターンコードが次の場合、コマンドは失敗]オプションを選択して必要に応じてドロップダウンリストおよびエディットボックスを設定してください。[リターンコードが次の場合、コマンドは失敗]オプションが選択されている場合、High-performance Embedded Workshop は各コマンドのリターンコードをチェックしてエラーが起こったかどうか判定します。エラーが起こった場合、コマンド実行は停止してそれに続くコマンド実行やコマンドの後に続く処理 (例: ビルド) は実行されません。

[コマンドのリターンコードを処理しない]オプションが選択されている場合、High-performance Embedded Workshop は各コマンドのリターンコードをチェックしません。したがって、すべてのコマンドを実行します。

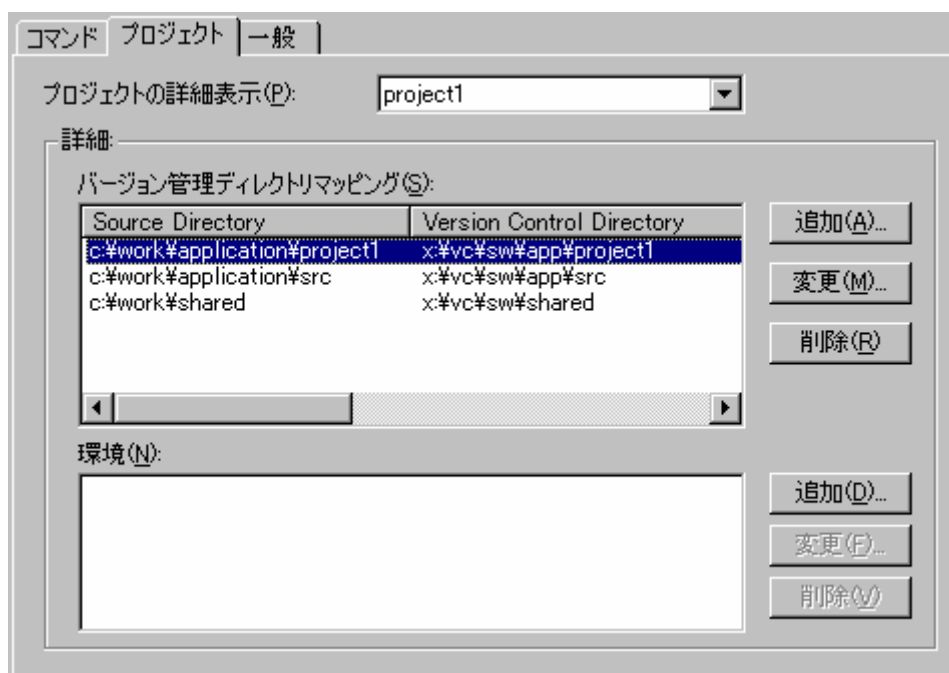
## 8.6 ディレクトリをマッピングする

ファイルの位置を指定するときには、プレースホルダを使用してください。そうでないと、そのコマンドは指定したファイルにしか適用できません。例えば、バージョン管理のアプリケーションに-GET コマンドが使われていて、ファイルの読み出し専用コピーを作成するとします。このとき、[引数]は以下のように指定できます。

```
-GET "c:¥vc¥files¥project¥main.c"
```

しかし、このコマンドを実行しても、ファイル MAIN.C しかコピーできません。この問題を解決するために、High-performance Embedded Workshop にはプレースホルダとディレクトリのマッピングというシステムがあります。マッピングにより、どの「作業中の」ディレクトリ (つまり、作業中のソースファイルがあるディレクトリ) がどの「管理」ディレクトリ (つまり、バージョン管理システムに保存されているソースファイルのディレクトリ) に対応するかを指定します。これら 2 つのディレクトリ間のマッピングは、[バージョン管理設定]ダイアログボックスの[プロジェクト]タブで指定できます。

マッピングを定義すると、Version control directory プレースホルダ \$(VCDIR)を使用してプロジェクトファイルを保存するディレクトリを示すことができます。



### 新しいマッピングを定義するには

1. [ツール -> バージョン管理 -> 構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [プロジェクト]タブを選択してください。
3. [バージョン管理ディレクトリマッピング]リストの横にある[追加]ボタンをクリックしてください。[バージョン管理ディレクトリマッピング]ダイアログボックスを開きます。
4. [ソースディレクトリ]にマッピングを作成したいディレクトリを設定してください。以下のいずれかの操作を選択してください。
  - ディレクトリを直接入力する。
  - プレースホルダボタンをクリックして表示されるメニューから、Project directory、または Workspace directory を選択する。
  - [参照]ボタンをクリックして表示される[ソースディレクトリを参照]ダイアログボックスで、ディレクトリを選び[選択]ボタンをクリックする。
5. [バージョン管理ディレクトリ]にソースディレクトリへマップするバージョン管理システムのディレクトリを入力してください。
6. [OK]ボタンをクリックしてください。

### 既存のマッピングを変更するには

1. [ツール -> バージョン管理 -> 構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [プロジェクト]タブを選択してください。
3. [バージョン管理ディレクトリマッピング]リストから変更するマッピングを選択し、[変更]ボタンをクリックしてください。[バージョン管理ディレクトリマッピング]ダイアログボックスを開きます。
4. 2つのディレクトリを変更して[OK]ボタンをクリックしてください。



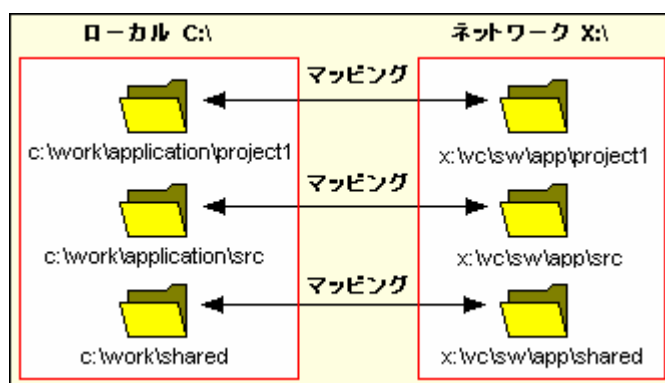


### 既存のマッピングを削除するには

1. [ツール->バージョン管理->構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [プロジェクト]タブを選択してください。
3. [バージョン管理ディレクトリマッピング]リストから削除するマッピングを選択し[削除]ボタンをクリックしてください。

## 8.7 ディレクトリマッピング例

マッピングを定義すると、Version control directory プレースホルダ\$(VCDIR)を使用してプロジェクトファイルを保存するディレクトリを示すことができます。以下に例を示します。ここではネットワークを共有したバージョン管理ドライブ(X:¥)と開発が行われているローカルドライブ(C:¥)からマッピングされた3つのディレクトリがあります。



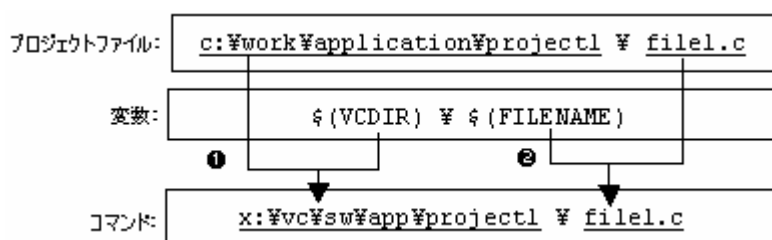
例えば、-GET コマンドが使われていてファイルの読み出し専用コピーを作成するバージョンコントロールのアプリケーションが選ばれているとします。プロジェクトのすべてのファイルを取得するには、次のコマンドを使用します。

```
-GET "$(VCDIR)¥$(FILENAME)"
```

そのプロジェクトファイルのコマンドを実行すると、\$(VCDIR)をファイルマッピングの中の対応するバージョン管理ディレクトリに置き換えます。

例えば、FILE1.C が C:¥work¥application¥project1¥file1.c にあるとします。FILE1.C に-GET コマンドが適用されると、次のようになります。

1. \$(VCDIR)が X:¥vc¥sw¥app¥project1 に置き換えられます。これは C:¥work¥application¥project1 のバージョン管理ディレクトリマッピングに対応しているからです。
2. \$(FILENAME)が FILE1.C に置き換えられます。

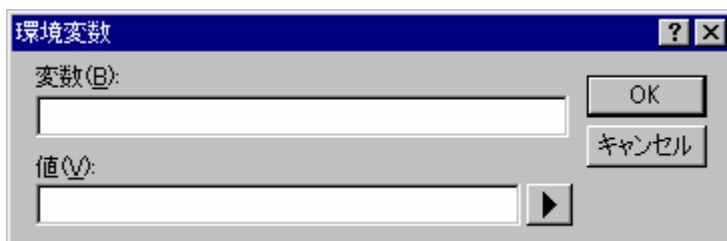


## 8.8 バージョン管理環境変数を設定する

現在の設定を確認するには[バージョン管理設定]ダイアログボックスの[プロジェクト]タブを表示してください。

### 新しい環境変数を追加するには

1. [環境]リストの横にある[追加]ボタンをクリックしてください。[環境変数]ダイアログボックスを開きます。



2. [変数]に環境変数名を入力してください。
3. [値]に環境変数の値を入力してください。
4. [OK]ボタンをクリックすると新しい変数を[環境]リストに追加します。

### 環境変数を変更するには

1. 変更する環境変数を[環境]リストから選択してください。
2. [変更]ボタンをクリックしてください。
3. [変数]と[値]を必要に応じて変更してください。
4. [OK]ボタンをクリックすると変更した環境変数をリストに追加します。

### 環境変数を削除するには

1. 削除する環境変数を[環境]リストから選択してください。
2. [削除]ボタンをクリックしてください。

## 8.9 バージョン管理システムの実行を制御する

[バージョン管理設定]ダイアログボックスの[一般]タブでは、使用するバージョン管理システムと High-performance Embedded Workshop を制御できます。また、現在のバージョン管理コンフィギュレーションファイルへの絶対パスを表示します。

- [コマンド実行前に確認]チェックボックス

このチェックボックスをオンにすると、バージョン管理コマンドを実行する前にダイアログボックスが開きます。このダイアログボックスではコマンドの実行の有無を確認できます。[OK]をクリックすると選んだコマンドを実行します。[キャンセル]をクリックするとコマンドは実行しません。

- [DOS ウィンドウで実行]チェックボックス

デフォルトでは、バージョン管理コマンドの出力はアウトプットウィンドウの[Version Control]タブに結果を出力します。

このチェックボックスをオンにすると、各コマンドを DOS ウィンドウで実行します。これは、バージョン管理システムがユーザ入力を要求するメッセージに対して入力が可能になります。実行終了後に DOS ウィンドウを閉じる場合は、[コマンド終了後 DOS ウィンドウを閉じる]チェックボックスをオンにしてください。

- [バージョン管理のディレクトリの区切りに'/'を使用]チェックボックス

デフォルトでは、High-performance Embedded Workshop がプレースホルダ\$(VCDIR)を置き換えるときバックスラッシュ文字\" (日本語 Windows®では円記号) を使ってディレクトリを区切ります。しかし、使用するバージョン管理システム(例: Visual SourceSafe)で、ディレクトリを区切るときにスラッシュ文字\"を使う場合は、[バージョン管理のディレクトリの区切りに\"を使用]チェックボックスをオンにしてスラッシュ文字\"でディレクトリを区切ってください。

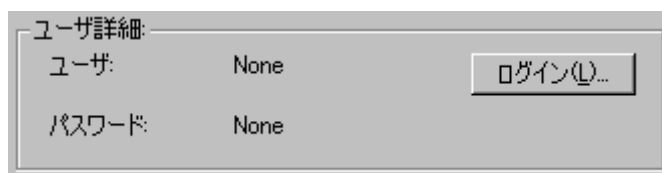
## 8.10 ユーザ名とパスワードを変更する

バージョン管理ツールでは一般的にユーザ名とパスワードの入力をコマンドラインで行う必要があります。これは、ファイルを保護し、どのファイルがどのユーザによって変更されたか記録するためです。バージョン管理システムでは2つのプレースホルダ"User login name", \$(USERNAME)と"User login password", \$(PASSWORD)をサポートします。コマンドを実行すると、これらのプレースホルダは[バージョン管理設定]ダイアログボックスの[一般]タブの現在の設定に置き換えます。

プレースホルダ\$(USERNAME)と\$(PASSWORD)に値を設定するには、最初にログインの指定をしておく必要があります。もしこれらのプレースホルダのうちのどちらかのプレースホルダを使うコマンドを実行する前にログイン指定をしていなかった場合、コマンド実行前にログインするよう要求します。

### ユーザ名とパスワードを設定するには (ログインの指定)

1. [ツール -> バージョン管理 -> 構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [一般]タブを選択してください。



3. [ログイン]ボタンをクリックしてください。[ログイン]ダイアログボックスを開きます。



4. [ユーザ名]にユーザ名を入力してください。

5. [パスワード]にパスワードを入力してください。
6. [確認のためパスワード再入力]にパスワードを再入力してください。
7. [OK]ボタンをクリックすると新しいユーザ名とパスワードを設定します。[パスワード]と[確認のためパスワード再入力]で異なるパスワードが入力された場合はもう一度パスワードを入力するよう要求します。

## 8.11 カスタムバージョン管理システムの使用例

カスタムバージョン管理システムは、MS-DOS コマンドプロンプトからコマンドラインコマンドを実行するバージョン管理システムを対象としています。

ここでは、High-performance Embedded Workshop と下記のバージョン管理システムを接続して使用する例を説明します。

- High-performance Embedded Workshop と RCS を接続して使用する
- High-performance Embedded Workshop と CVS を接続して使用する

### 8.11.1 High-performance Embedded WorkshopとRCSを接続して使用する

High-performance Embedded Workshop と RCS を接続して使用する例を紹介します。

- **RCS の環境**

バージョン管理システムは以下を使用します。

- GNU RCS
- GNU diff

この手順では、上記のバージョン管理システムが"C:¥RCS"にあると仮定します。

- **ワークスペース**

High-performance Embedded Workshop のワークスペースは、以下のフォルダに作成してください。

C:¥Workspace¥rcs\_sample

High-performance Embedded Workshop のワークスペース作成後、RCS 管理用フォルダを作成してください。

C:¥Workspace¥rcs\_sample¥rcs\_sample¥RCS

ワークスペースの作成、バージョン管理の設定、バージョン管理機能のチェックイン、比較、チェックアウト、ログの確認の使用手順を説明します。

ステップ 1：ワークスペースを作成する

ステップ 2：カスタムバージョン管理システムを選択する

ステップ 3：バージョン管理を設定する

ステップ 4：バージョン管理機能を使用する（チェックイン）

ステップ 5：バージョン管理機能を使用する（比較）

ステップ 6：バージョン管理機能を使用する（チェックアウト）

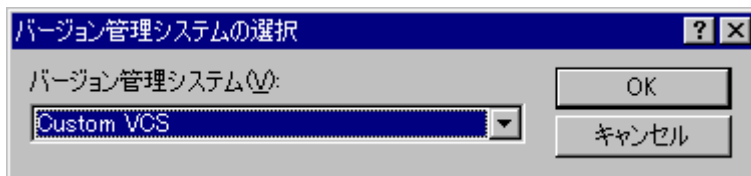
ステップ 7：バージョン管理機能を使用する（ログの確認）

### 8.11.1.1 ステップ 1：ワークスペースを作成する

1. [ファイル -> 新規ワークスペース]を選択し、[新規プロジェクトワークスペース]ダイアログボックスを開きます。
2. 下記を指定しワークスペースを作成してください。
  - ワークスペース名： rcs\_sample
  - プロジェクト名： rcs\_sample
  - ディレクトリ： C:\¥WorkSpace¥rcs\_sample
3. ワークスペース作成後、RCS 管理用フォルダを作成してください。  
C:\¥WorkSpace¥rcs\_sample¥rcs\_sample¥RCS

### 8.11.1.2 ステップ 2：カスタムバージョン管理システムを選択する

1. [ツール -> バージョン管理 -> 選択]を選択します。[バージョン管理システムの選択]ダイアログボックスが開きます。



2. [バージョン管理システム]リストから"Custom VCS"を選択します。
3. [OK]ボタンをクリックします。

### 8.11.1.3 ステップ 3：バージョン管理を設定する

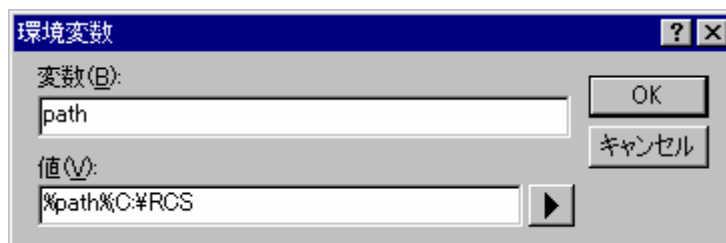
[ツール -> バージョン管理 -> 構成]を選択し、[バージョン管理設定]ダイアログボックスを開きます。

- 環境変数を設定する

1. [プロジェクト]タブを選択します。



2. [環境]にある[追加]ボタンをクリックします。[環境変数]ダイアログボックスが開きます。



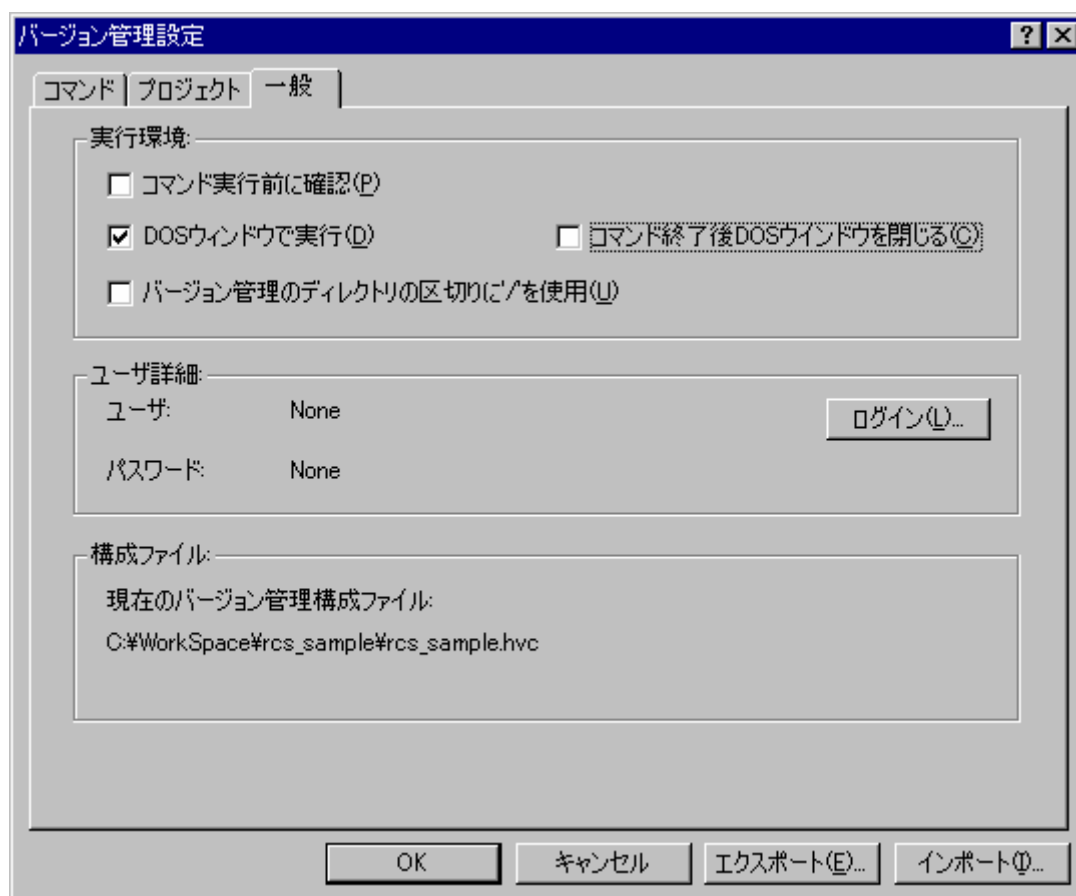
3. [変数]に"path"、[値]に"%path%C:¥RCS"を設定します。
4. [OK]ボタンをクリックします。
5. 同様に[追加]ボタンをクリックし、[環境変数]ダイアログボックスで下記の環境変数も設定します。

LOGNAME=user

TZ=JST-9

- 実行時の動作を設定する

1. [一般]タブを選択します。

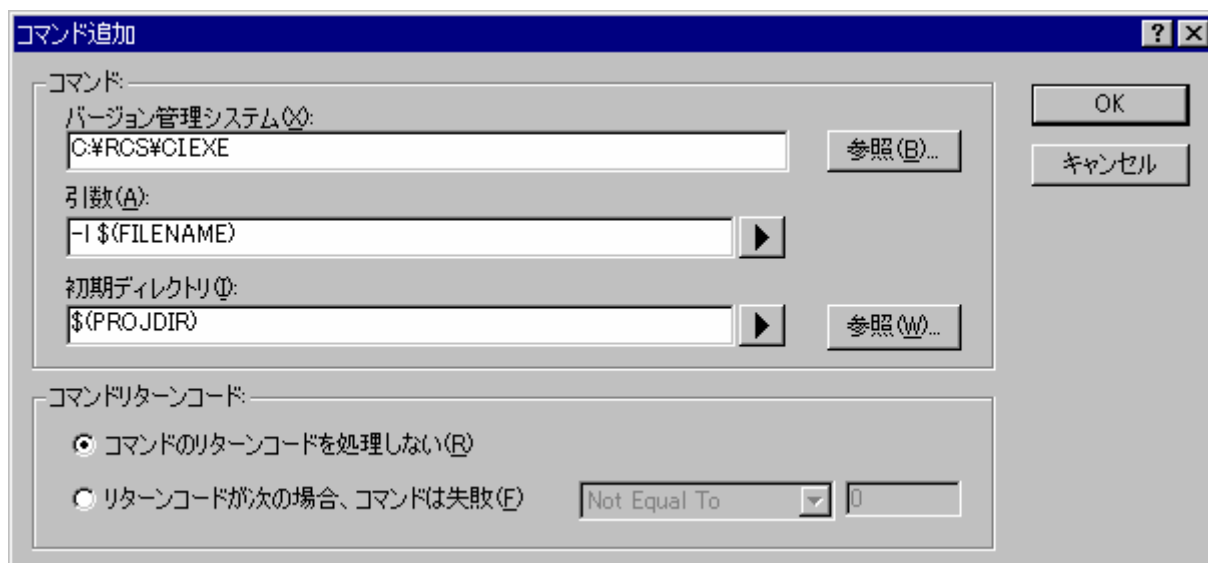


2. [コマンド実行前に確認]チェックボックスをオフにします(コマンド実行前にファイル名を確認したい場合は、オンのままにしてください)。
3. [DOS ウィンドウで実行]チェックボックスをオンにします(RCS のコマンドに MS-DOS コマンドプロンプトから入力する必要があるため)。
4. [コマンド終了後 DOS ウィンドウを閉じる]チェックボックスをオフにします(エラーを MS-DOS コマンドプロンプトで確認するため)。

#### • コマンドを設定する

1. [コマンド]タブを選択します。
2. [ユーザメニューオプション]にある[追加]ボタンをクリックします。[メニューオプションの追加]ダイアログボックスが開きます。
3. [オプション]に"Check in"と入力し、[追加]ボタンをクリックします。[コマンド追加]ダイアログボックスが開きます。





4. [バージョン管理システム]に"C:\RCS\CI.EXE"を設定します。
5. [引数]に"-I \$(FILENAME)"を設定します。
6. [初期ディレクトリ]に"\$(PROJDIR)"を設定します。
7. [OK]ボタンをクリックします。
8. [メニューオプションの追加]ダイアログボックスで[OK]ボタンをクリックします。

#### ポイント：

バージョン管理によく使用するコマンドのみを登録し、あまり使用しないコマンドは、MS-DOS コマンドプロンプトから使用することをお勧めします。

同様に以下の3つのコマンドを設定してください。

コマンド	項目	入力内容
Check out	Version control executable	C:\RCS\CO.EXE
	Arguments	-I -f \$(FILENAME)
	Initial directory	\$(PROJDIR)
Diff	Version control executable	C:\RCS\RCSDIFF.EXE
	Arguments	\$(FILENAME)
	Initial directory	\$(PROJDIR)
Log	Version control executable	C:\RCS\RLOG.EXE
	Arguments	\$(FILENAME)
	Initial directory	\$(PROJDIR)

これで RCS ツールを使う準備ができました。

#### 8.11.1.4 ステップ4：バージョン管理機能を使用する（チェックイン）

1. 後からファイルの変更履歴が分かるように"rcs\_sample.c"ファイルの12行目にキーワードを入力します。  
//\$Id\$
2. ワークスペースウィンドウの[Projects]タブの"rcs\_sample.c"ファイルを選択し、[ツール -> バージョン管理 -> Check in]を選択します。MS-DOS コマンドプロンプトウィンドウが開きます。

3. プロンプト(">>")に続いてチェックイン時のコメントを入力し、Enter キーを押します。
4. ピリオド" ."を 1 文字だけ入力し、Enter キーを押します。

```

rcs_sample.c,v <-- rcs_sample.c
enter description, terminated with single '.' or end of file:
NOTE: This is NOT the log message!
>> .
initial revision: 1.1
done
C:\RCS>

```

5. "exit"を入力し Enter キーを押します。MS-DOS コマンドプロンプトウィンドウが閉じます。
6. 表示される High-performance Embedded Workshop の確認ダイアログボックスで[はい]ボタンをクリックしてください。キーワードがファイル名、リビジョン、日時、ユーザなどに変換されて表示します。

#### ポイント：

MS-DOS コマンドプロンプトウィンドウでコマンドの実行結果を確認しない場合、[バージョン管理設定]ダイアログボックスの[一般]タブの[コマンド終了後 DOS ウィンドウを閉じる]チェックボックスをオンにしてください。

#### 8.11.1.5 ステップ 5：バージョン管理機能を使用する（比較）

1. チェックインしたファイルと比較するため、"rcs\_sample.c"ファイルを編集し上書きします。
2. ワークスペースウィンドウの[Projects]タブの"rcs\_sample.c"ファイルを選択し、[ツール -> バージョン管理 -> Diff]を選択します。MS-DOS コマンドプロンプトウィンドウが開き、比較した結果が表示されます。
3. "exit"を入力し Enter キーを押します。MS-DOS コマンドプロンプトウィンドウが閉じます。

```

RCS file: rcs_sample.c,v
Working file: rcs_sample.c
head: 1.1
branch:
locks: strict
    user: 1.1
access list:
symbolic names:
keyword substitution: kv
total revisions: 1;   selected revisions: 1
description:
-----
revision 1.1   locked by: user;
date: 2006/03/14 07:22:42; author: user; state: Exp;
Initial revision
=====

```

### 8.11.1.6 ステップ 6：バージョン管理機能を使用する（チェックアウト）

ファイルで編集した内容を元に戻す場合は、チェックアウトコマンドを実行します。

1. ワークスペースウィンドウの[Projects]タブの"res\_sample.c"ファイルを選択し、[ツール->バージョン管理->Check out]を選択します。MS-DOS コマンドプロンプトウィンドウが開き、チェックアウトした結果が表示されます。
2. "exit"を入力し Enter キーを押します。MS-DOS コマンドプロンプトウィンドウが閉じます。
3. 表示される High-performance Embedded Workshop の確認ダイアログボックスで[はい]ボタンをクリックしてください。ファイルが元に戻ります。

```
res_sample.c,v <-- res_sample.c
enter description, terminated with single '!' or end of file:
NOTE: This is NOT the log message!
>> .
initial revision: 1.1
done
C:¥RCS>
```

### 8.11.1.7 ステップ 7：バージョン管理機能を使用する（ログの確認）

編集した履歴を確認するため、ログコマンドを実行します。

1. ワークスペースウィンドウの[Projects]タブの"res\_sample.c"ファイルを選択し、[ツール->バージョン管理->Log]を選択します。MS-DOS コマンドプロンプトウィンドウが開き、ログが表示されます。
2. "exit"を入力し Enter キーを押します。MS-DOS コマンドプロンプトウィンドウが閉じます。

```
CS file: res_sample.c,v
Working file: res_sample.c
head: 1.1
branch:
locks: strict
    user: 1.1
access list:
symbolic names:
keyword substitution: kv
total revisions: 1;   selected revisions: 1
description:
-----
revision 1.1   locked by: user;
date: 2006/03/14 07:22:42; author: user; state: Exp;
Initial revision
=====
```

## 8.11.2 High-performance Embedded WorkshopとCVSを接続して使用する

High-performance Embedded Workshop と CVS を接続して使用する例を紹介します。

### • CVS ツールの環境

バージョン管理システムは下記を使用します。

- CVS

この手順では、上記のバージョン管理システムが"C:\cvs-1-11-17"にあると仮定します。

### • ワークスペース

High-performance Embedded Workshop のワークスペースは、以下のフォルダに作成してください。

C:\WorkSpace\sampleCVS

High-performance Embedded Workshop のワークスペース作成後、CVS 用ソースファイルフォルダを作成してください。

C:\src\sampleCVS

### 注：

High-performance Embedded Workshop と CVS を使用したバージョン管理では、管理対象ファイルをソースファイルのみとしています。ソースファイル以外のファイル（ワークスペースファイル、プロジェクトファイルなど）は、管理対象とすることができませんのでご注意願います。

ワークスペースの作成、バージョン管理の設定、バージョン管理機能のリポジトリの作成、モジュールの登録、チェックアウト、ファイルの編集、ステータスの確認、比較、結果の登録、ログの確認の使用手順を説明します。

ステップ 1：ワークスペースを作成する

ステップ 2：カスタムバージョン管理システムを選択する

ステップ 3：バージョン管理を設定する

ステップ 4：バージョン管理機能を使用する（リポジトリの作成）

ステップ 5：バージョン管理機能を使用する（モジュールの登録）

ステップ 6：バージョン管理機能を使用する（チェックアウト）

ステップ 7：バージョン管理機能を使用する（ファイルの編集）

ステップ 8：バージョン管理機能を使用する（ステータスの確認）

ステップ 9：バージョン管理機能を使用する（比較）

ステップ 10：バージョン管理機能を使用する（結果の登録）

ステップ 11：バージョン管理機能を使用する（ログの確認）

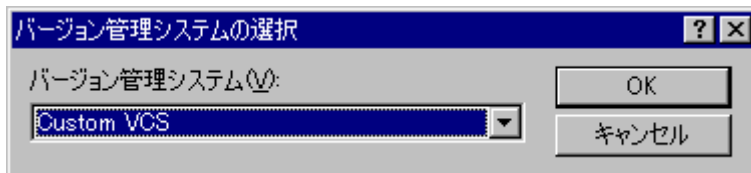
### 8.11.2.1 ステップ 1：ワークスペースを作成する

1. [ファイル->新規ワークスペース]を選択し、[新規プロジェクトワークスペース]ダイアログボックスを開きます。
2. 下記を指定しワークスペースを作成してください。
  - ワークスペース名：sampleCVS

- プロジェクト名 : sampleCVS
  - ディレクトリ : C:¥WorkSpace¥sampleCVS
3. High-performance Embedded Workshop を終了します。
  4. CVS 用ソースファイルフォルダを作成してください。  
C:¥src¥sampleCVS
  5. バージョン管理ツールで管理するファイルを"C:¥WorkSpace¥sampleCVS¥sampleCVS"から"C:¥src¥sampleCVS"へ移動します。
  6. High-performance Embedded Workshop を起動し、作成した"sampleCVS"ワークスペースを開きます。
  7. [プロジェクト -> ファイルの削除]メニューを選択し、[プロジェクトファイルの削除]ダイアログボックスを開きます。
  8. バージョン管理するファイルを選択し、[削除]ボタンをクリックします。
  9. [OK]ボタンをクリックします。
  10. [プロジェクト -> ファイルの追加]メニューを選択し、['sampleCVS'プロジェクトにファイルの追加]ダイアログボックスを開きます。
  11. バージョン管理するファイルを選択し、[追加]ボタンをクリックします。

#### 8.11.2.2 ステップ 2 : カスタムバージョン管理システムを選択する

1. [ツール -> バージョン管理 -> 選択]を選択します。[バージョン管理システムの選択]ダイアログボックスが開きます。



2. [バージョン管理システム]リストから"Custom VCS"を選択します。
3. [OK]ボタンをクリックします。

#### 8.11.2.3 ステップ 3 : バージョン管理を設定する

[ツール -> バージョン管理 -> 構成]を選択し、[バージョン管理設定]ダイアログボックスを開きます。

- **環境変数を設定する**

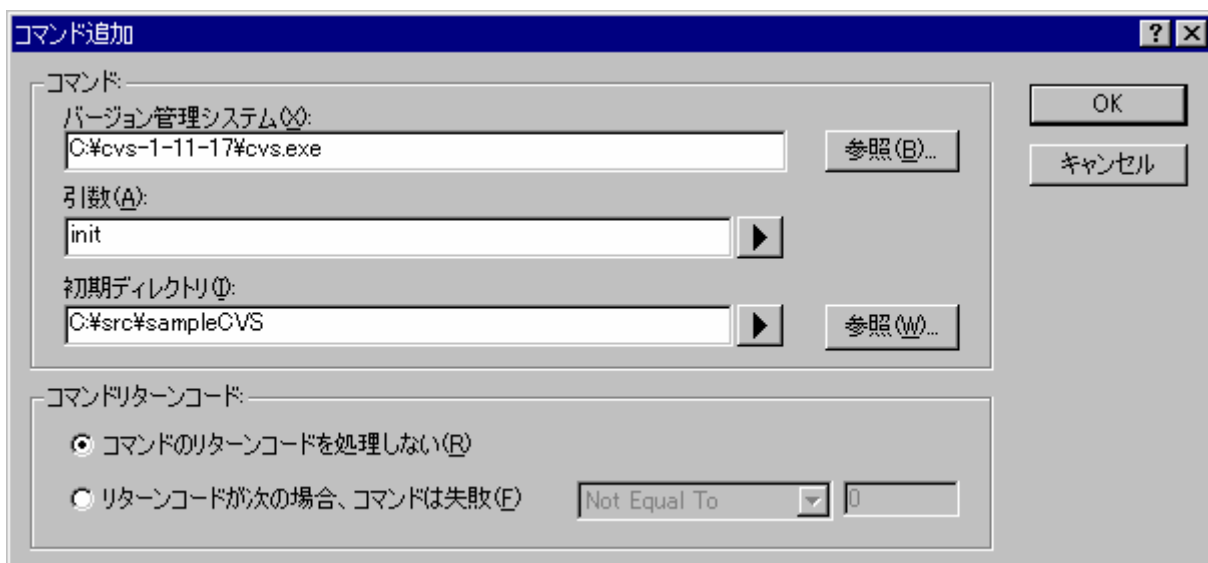
1. [プロジェクト]タブを選択します。
2. [環境]にある[追加]ボタンをクリックします。[環境変数]ダイアログボックスが開きます。
3. [変数]に"CVSROOT"、[値]に"C:¥cvs-1-11-17¥DATABASE"を設定します。



4. [OK]ボタンをクリックします。

#### • コマンドを設定する

1. [コマンド]タブを選択します。
2. [ユーザメニューオプション]にある[追加]ボタンをクリックします。[メニューオプションの追加]ダイアログボックスが開きます。
3. [オプション]に"Initialize"と入力し、[追加]ボタンをクリックします。[コマンド追加]ダイアログボックスが開きます。
4. [バージョン管理システム]に"C:\cvs-1-11-17\cvs.exe"を設定します。
5. [引数]に"init"を設定します。
6. [初期ディレクトリ]に"C:\src\sampleCVS"を設定します。



7. [OK]ボタンをクリックします。
8. [メニューオプションの追加]ダイアログボックスで[OK]ボタンをクリックします。

同様に以下の 6 つのコマンドを設定してください。

コマンド	項目	入力内容
Register	Version control executable	C:\cvs-1-11-17\cvs.exe
	Arguments	import -m "New Source" sampleCVS Renesas rel-20060228
	Initial directory	C:\src\sampleCVS
Checkout	Version control executable	C:\cvs-1-11-17\cvs.exe
	Arguments	checkout sampleCVS
	Initial directory	C:\src
Status	Version control executable	C:\cvs-1-11-17\cvs.exe
	Arguments	status \$(FILENAME)
	Initial directory	C:\src\sampleCVS
Diff	Version control executable	C:\cvs-1-11-17\cvs.exe
	Arguments	diff \$(FILENAME)
	Initial directory	C:\src\sampleCVS
Commit	Version control executable	C:\cvs-1-11-17\cvs.exe
	Arguments	commit -m "modify" \$(FILENAME)
	Initial directory	C:\src\sampleCVS
Log	Version control executable	C:\cvs-1-11-17\cvs.exe
	Arguments	log \$(FILENAME)
	Initial directory	C:\src\sampleCVS

これで CVS ツールを使う準備ができました。

#### 8.11.2.4 ステップ 4：バージョン管理機能を使用する（リポジトリの作成）

1. [ツール -> バージョン管理 -> Initialize]メニューを選択します。
2. [Initialize]ダイアログボックスでファイルを選択し[OK]ボタンをクリックします。アウトプットウィンドウの[Version Control]タブに下記のメッセージが表示されます。

```
Executing C:\cvs-1-11-17\cvs.exe init
Command(s) completed successfully.
```



### 8.11.2.5 ステップ 5：バージョン管理機能を使用する（モジュールの登録）

1. [ツール -> バージョン管理 -> Register]メニューを選択します。
2. [Register]ダイアログボックスでファイルを選択し[OK]ボタンをクリックします。アウトプットウィンドウの[Version Control]タブに下記のメッセージが表示されます。

```
Executing C:\cvs-1-11-17\cvs.exe import -m "New Source" sampleCVS Renesas rel-20060228
N sampleCVS/dbstc.c
(省略)
No conflicts created by this import
Command(s) completed successfully.
```

### 8.11.2.6 ステップ 6：バージョン管理機能を使用する（チェックアウト）

1. [ツール -> バージョン管理 -> Checkout]メニューを選択します。
2. [Checkout]ダイアログボックスでファイルを選択し[OK]ボタンをクリックします。アウトプットウィンドウの[Version Control]タブに下記のメッセージが表示されます。

```
Executing C:\cvs-1-11-17\cvs.exe checkout sampleCVS
U sampleCVS/dbstc.c
(省略)
cvs.exe checkout: Updating sampleCVS
Command(s) completed successfully.
```

### 8.11.2.7 ステップ 7：バージョン管理機能を使用する（ファイルの編集）

High-performance Embedded Workshop のエディタでファイルを編集し上書きします。例として"sampleCVS.c" ファイルを修正します。

### 8.11.2.8 ステップ 8：バージョン管理機能を使用する（ステータスの確認）

1. "sampleCVS.c"ファイルのステータスを確認します。
2. [ツール -> バージョン管理 -> Status]メニューを選択します。
3. [Status]ダイアログボックスでファイルを選択し[OK]ボタンをクリックします。アウトプットウィンドウの[Version Control]タブに下記のメッセージが表示されます。

```
Executing C:\cvs-1-11-17\cvs.exe status sampleCVS.c
```

```
=====
File: sampleCVS.c    Status: Locally Modified
Working revision: 1.1.1.1 Tue Feb 28 02:46:30 2006
Repository revision: 1.1.1.1 C:\cvs-1-11-17\DATABASE/sampleCVS/sampleCVS.c,v
Sticky Tag: (none)
Sticky Date: (none)
Sticky Options: (none)
Command(s) completed successfully.
```

"sampleCVS.c"ファイルのステータスが、"Locally Modified"(修正されている)と確認できます。

### 8.11.2.9 ステップ 9 : バージョン管理機能を使用する (比較)

1. "sampleCVS.c"ファイルの修正内容を確認します。
2. [ツール -> バージョン管理 -> Diff]メニューを選択します。
3. [Diff]ダイアログボックスでファイルを選択し[OK]ボタンをクリックします。アウトプットウィンドウの[Version Control]タブに下記のメッセージが表示されます。

```
Executing C:\cvs-1-11-17\cvs.exe diff sampleCVS.c
```

```
Index: sampleCVS.c
```

```
=====
```

```
RCS file: C:\cvs-1-11-17\DATABASE/sampleCVS/sampleCVS.c,v
```

```
retrieving revision 1.1.1.1
```

```
diff -r1.1.1.1 sampleCVS.c
```

```
28c28,31
```

```
< printf("### Data Input ###\n");
```

```
---
```

```
> //無限ループ追加
```

```
> while(1)
```

```
> {
```

```
> printf("### Data Input ###\n"); //無限ループ開始
```

```
49a53,54
```

```
> //無限ループ終了
```

```
> }
```

```
Command(s) completed successfully.
```

"sampleCVS.c"ファイルの修正内容が、確認できます。

### 8.11.2.10 ステップ 10 : バージョン管理機能を使用する (結果の登録)

1. [ツール -> バージョン管理 -> Commit]メニューを選択します。
2. [Commit]ダイアログボックスでファイルを選択し[OK]ボタンをクリックします。アウトプットウィンドウの[Version Control]タブに下記のメッセージが表示されます。

```
Executing C:\cvs-1-11-17\cvs.exe commit -m "modify" sampleCVS.c
```

```
Checking in sampleCVS.c;
```

```
C:\cvs-1-11-17\DATABASE/sampleCVS/sampleCVS.c,v <-- sampleCVS.c
```

```
new revision: 1.2; previous revision: 1.1
```

```
done
```

```
Command(s) completed successfully.
```

### 8.11.2.11 ステップ 11 : バージョン管理機能を使用する (ログの確認)

1. [ツール -> バージョン管理 -> Log]メニューを選択します。
2. [Log]ダイアログボックスでファイルを選択し[OK]ボタンをクリックします。アウトプットウィンドウの[Version Control]タブに下記のメッセージが表示されます。

```
Executing C:\cvs-1-11-17\cvs.exe log sampleCVS.c
RCS file: C:\cvs-1-11-17\DATABASE/sampleCVS/sampleCVS.c,v
Working file: sampleCVS.c
head: 1.2
branch:
locks: strict
access list:
symbolic names:
rel-20060228: 1.1.1.1
Renasas: 1.1.1
keyword substitution: kv
total revisions: 3; selected revisions: 3
description:
-----
revision 1.2
date: 2006/02/28 03:39:22; author: user; state: Exp; lines: +6 -1
modify
-----
revision 1.1
date: 2006/02/28 02:46:30; author: user; state: Exp;
branches: 1.1.1;
Initial revision
-----
revision 1.1.1.1
date: 2006/02/28 02:46:30; author: user; state: Exp; lines: +0 -0
New Source
=====
Command(s) completed successfully.
```

## 9. Visual SourceSafeバージョン管理システム

High-performance Embedded Workshop は Microsoft® Visual SourceSafe バージョン管理システムをサポートしています。

Visual SourceSafe バージョン管理システムでは、Microsoft® Visual SourceSafe データベースの中のプロジェクトと High-performance Embedded Workshop のワークスペースの中のプロジェクトを関連付けます。[ツール->バージョン管理]サブメニューからメニューオプションを選択するか、[バージョン管理]ツールバーボタンをクリックすることにより、Microsoft® Visual SourceSafe バージョン管理システムで最もよく使用されるコマンドを容易に起動できます。

Microsoft® Visual SourceSafe のインストールおよびオプションの設定は、Microsoft® Visual SourceSafe のヘルプまたはマニュアルを参照してください。Microsoft® Visual SourceSafe のログイン名、パスワードは設定済みとして説明します。

High-performance Embedded Workshop と Microsoft® Visual SourceSafe バージョン管理システムを接続した場合のバージョン管理の簡単な流れを以下に示します。他に利用できる機能について、詳しくは「9.2 Visual SourceSafe コマンド」を参照してください。

### 1. ワークスペースと Microsoft® Visual SourceSafe を関連付ける

High-performance Embedded Workshop のワークスペースの中のプロジェクトと Microsoft® Visual SourceSafe データベースの中のプロジェクトを関連付けます。

- "Microsoft Visual SourceSafe"を High-performance Embedded Workshop と接続するバージョン管理システムとして選択する。
- Microsoft® Visual SourceSafe へログインする。
- Microsoft® Visual SourceSafe に作成するプロジェクト名、および作成する位置を設定する。

### 2. ファイルを追加する

High-performance Embedded Workshop のワークスペースで Microsoft® Visual SourceSafe のプロジェクトに追加するファイルを選択します。

### 3. ファイルをチェックアウトする

High-performance Embedded Workshop のワークスペースで編集するファイルを選択しチェックアウトします。

### 4. ファイルをチェックインする

High-performance Embedded Workshop のワークスペースで編集したファイルを選択しチェックインします。

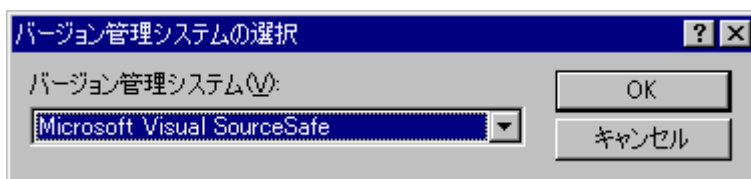
## 9.1 ワークスペースに Visual SourceSafe を関連付ける

Microsoft® Visual SourceSafe と High-performance Embedded Workshop の現在のワークスペースとを関連付ける方法を説明します。

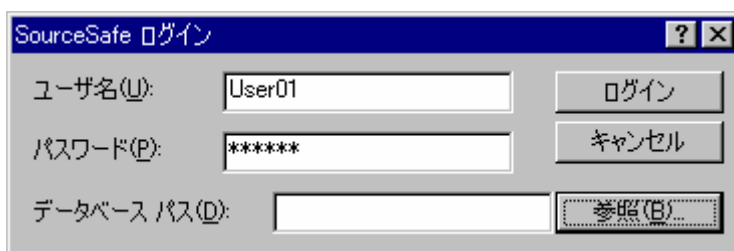
まず、Microsoft® Visual SourceSafe をバージョン管理システムとして選択します。

### ワークスペースに Microsoft® Visual SourceSafe を関連付けるには

1. [ツール->バージョン管理->選択]を選択します。[バージョン管理システムの選択]ダイアログボックスが開きます。
2. [バージョン管理システム]リストから[Microsoft Visual SourceSafe]を選択します。



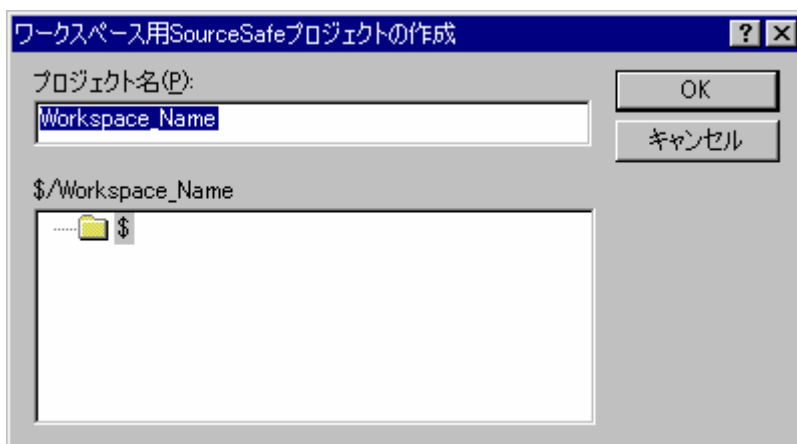
3. [OK] ボタンをクリックします。[SourceSafe ログイン]ダイアログボックスが開きます。



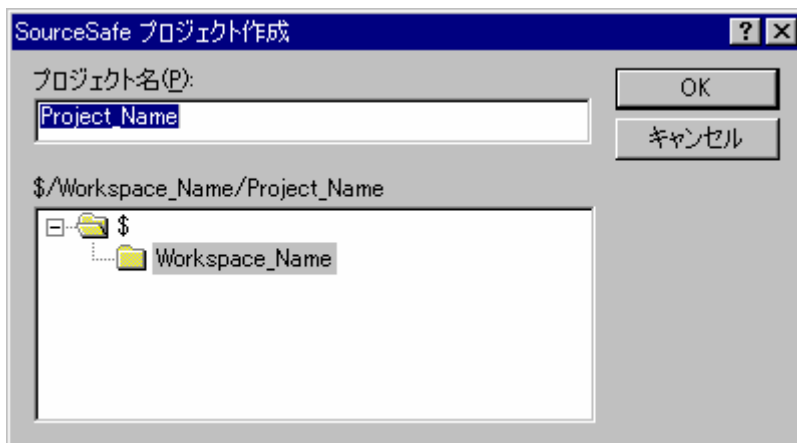
4. [ユーザ名]に Visual SourceSafe のログイン名を、[パスワード]にパスワードを入力してください。
5. [データベースパス]にプロジェクトを追加する Visual SourceSafe データベースファイル (SRCSAFE.INI) を指定します。以下のいずれかの操作を選択してください。
  - [参照]ボタンをクリックしてください。[Visual SourceSafe データベースの選択]ダイアログボックスを開きます。プロジェクトを追加する Visual SourceSafe データベースファイルを選択し[選択]ボタンをクリックしてください。



- プロジェクトを追加する Visual SourceSafe データベースファイルへの絶対パスを[データベースパス]に入力してください。
6. [SourceSafe ログイン]ダイアログボックスの[ログイン]ボタンをクリックしてください。[ワークスペース用 SourceSafe プロジェクトの作成]ダイアログボックスを開きます。



7. [プロジェクト名]に、あらかじめ High-performance Embedded Workshop のワークスペース名が表示されます。データベースに作成するプロジェクト名 (フォルダ名) です。プロジェクト名は変更できません。
8. [プロジェクト名]フィールド下のツリーにはステップ 5 で指定したデータベース構造が表示されます。[プロジェクト名]に指定したフォルダをどのフォルダ内に作成するかを選択してください。
9. [ワークスペース用 SourceSafe プロジェクトの作成]ダイアログボックスの[OK] ボタンをクリックしてください。[SourceSafe プロジェクト作成]ダイアログボックスが開きます。下の図の例では、High-performance Embedded Workshop のワークスペース "Workspace\_Name" がプロジェクト名 "Workspace\_Name" としてデータベースに追加されました。[プロジェクト名]には、次に作成する High-performance Embedded Workshop のプロジェクト "Project\_Name" があらかじめ表示されています。



10. [SourceSafe プロジェクト作成]ダイアログボックスの[OK] ボタンをクリックしてください。
11. 現在のワークスペースにあるプロジェクトの数だけ、ステップ 10 を繰り返してください。

Visual SourceSafe のプロジェクトとサブプロジェクト、High-performance Embedded Workshop のワークスペースとプロジェクト、およびハードディスク上のワークスペースディレクトリの階層イメージを以下の図に示します。

Visual SourceSafe のプロジェクトと サブプロジェクト	HEWワークスペース とプロジェクト	ハードディスク上の ワークスペースディレクトリ (Visual SourceSafe の作業フォルダ)
		

High-performance Embedded Workshop は、Visual SourceSafe 内に必要なプロジェクトを作成し、容易にアクセスできるように、バージョン管理ツールバーやメニューを設定します。

## 9.2 Visual SourceSafe コマンド

バージョン管理ツールバーやメニューを使用することにより以下の 8 つの操作が使用可能です。


ツール -> バージョン管理メニュー	バージョン管理ツールバー	機能
ファイルの追加		Visual SourceSafe にファイルを追加します。
ファイルの削除		Visual SourceSafe からファイルを削除します。
ファイルの取得		Visual SourceSafe からファイルの読み取り専用コピーを取得します。
ファイルのチェックアウト		Visual SourceSafe からファイルの書き込み可能コピーをチェックアウトします。
ファイルのチェックイン		Visual SourceSafe にファイルの書き込み可能コピーをチェックインします。
ファイルの状態		Visual SourceSafe のファイルの状態を表示します。
チェックアウトの取り消し	-	Visual SourceSafe のチェックアウトコマンドを取り消します。
履歴表示	-	Visual SourceSafe のファイル履歴を表示します。

### 9.2.1 Visual SourceSafe にファイルを追加する

前節では、ハードディスク上のワークスペースディレクトリ（作業中ディレクトリ）と Visual SourceSafe のプロジェクトディレクトリ（Visual SourceSafe によって管理されたディレクトリ）との間でマッピングを確立しました。ハードディスクのプロジェクトディレクトリ（サブディレクトリを含む）に複数のソースファイルがあってもマッピング先の Visual SourceSafe のプロジェクトディレクトリにはファイルがありません。

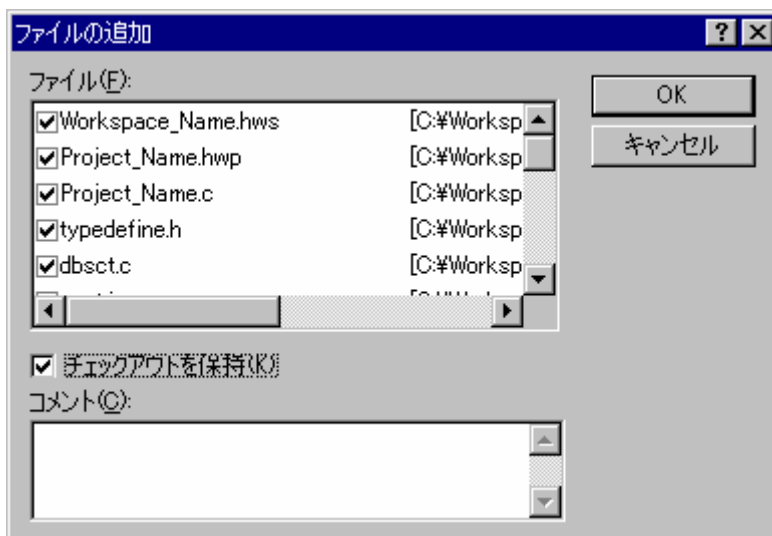
まず、Visual SourceSafe にファイルを追加します。

#### Visual SourceSafe にファイルを追加するには

- ワークスペースウィンドウの[Projects]タブで、Visual SourceSafe に追加するファイル（複数選択可）を選択してください。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはそれらを混在して選択することもできます。プロジェクトフォルダ、またはワークスペースフォルダを選択すると、High-performance Embedded Workshop のシステムファイルをファイルリストに追加します。例えば、プロジェクトフォルダを選択すると、プロジェクトファイル（.HWP）が[ファイル]リストへ追加されます。プロジェクトファイルがチェックアウトされ、バージョンが最後にロードしたときより新しい場合、そのプロジェクトを再読み込みしたいかどうかをユーザに問い合わせます。
- 以下のいずれかの操作を選択してください。[ファイルの追加]ダイアログボックスが開きます。
  - [ファイルの追加]ツールボタン()をクリックする。
  - [ツール -> バージョン管理 -> ファイルの追加]を選択する。
  - 右クリックしポップアップメニューを表示します。[バージョン管理 -> ファイルの追加]を選択する。



- 追加するファイルの左のチェックボックスをオンにし[OK]ボタンをクリックしてください。



Visual SourceSafe にファイルを追加すると、作業中フォルダのローカルファイルは読み取り専用になります。ファイルを追加する際に[ファイルの追加]ダイアログボックスの[チェックアウトを保持]チェックボックスをオンにすると、ファイルを追加した後もファイルをチェックアウトされた状態で書き込み可能にできます。ファイルが追加されたことの確認や、プロジェクト内のすべてのファイル状態を表示するには、以下を行ってください。

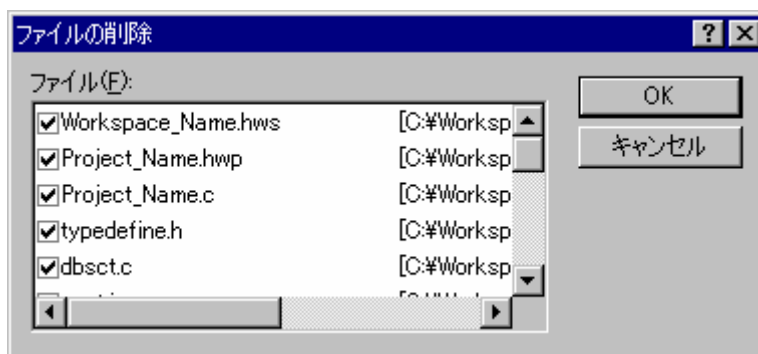
- ワークスペースウィンドウの[Projects]タブで、チェックしたいファイルのプロジェクトフォルダを選択してください。
- 以下のいずれかの操作を選択してください。ファイルの状態の表示先は、アウトプットウィンドウの[Version Control]タブまたは[ファイルの状態]ダイアログボックスです。ファイル状態の表示先の設定は、「9.3 Visual SourceSafe コマンドのオプション」を参照してください。表示された情報により、ファイルがプロジェクトに追加されたかどうか、ファイルがチェックアウトされたかどうか、また、どのユーザによってチェックアウトされたなどの状態も確認できます。
  - ファイルの状態ツールボタン(🔒)をクリックする。
  - [ツール->バージョン管理->ファイルの状態]を選択する。
  - 右クリックしポップアップメニューを表示します。[バージョン管理->ファイルの状態]を選択する。

## 9.2.2 Visual SourceSafeからファイルを削除する

Visual SourceSafe に追加したファイルを Visual SourceSafe による管理から外す場合は、Visual SourceSafe からファイルを削除します。

### Visual SourceSafe からファイルを削除するには

- ワークスペースウィンドウの[Projects]タブで、Visual SourceSafe から削除するファイルを選択してください。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在させて選択することもできます。
- 以下のいずれかの操作を選択してください。[ファイルの削除]ダイアログボックスが開きます。
  - [ファイルの削除]ツールバーボタン(🗑️)をクリックする。
  - [ツール->バージョン管理->ファイルの削除]を選択する。
  - 右クリックしポップアップメニューを表示します。[バージョン管理->ファイルの削除]を選択する。



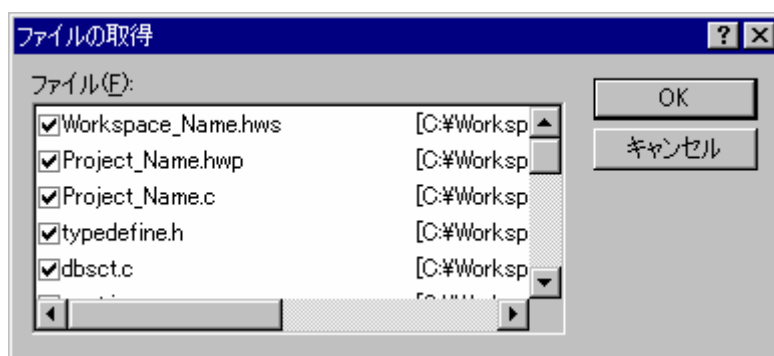
- 削除するファイルの左のチェックボックスをオンにし[OK]ボタンをクリックしてください。

### 9.2.3 Visual SourceSafeからファイルの読み取り専用コピーを取得する

Visual SourceSafe はソースファイルを保護して、管理しているファイルの書き込み可能なコピーを一度に一人のユーザだけが取得できるようにします。しかし、どのユーザもすべてのファイルの読み取り専用コピーを取得できます。

#### Visual SourceSafe からファイルの読み取り専用コピーを取得するには

- ワークスペースウィンドウの[Projects]タブで、Visual SourceSafe から取得するファイルを選択してください。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選択することもできます。
- 以下のいずれかの操作を選択してください。[ファイルの取得]ダイアログボックスが開きます。
  - [ファイルの取得]ツールバーボタン(📁)をクリックする。
  - [ツール -> バージョン管理 -> ファイルの取得]を選択する。
  - 右クリックしポップアップメニューを表示します。[バージョン管理 -> ファイルの取得]を選択する。



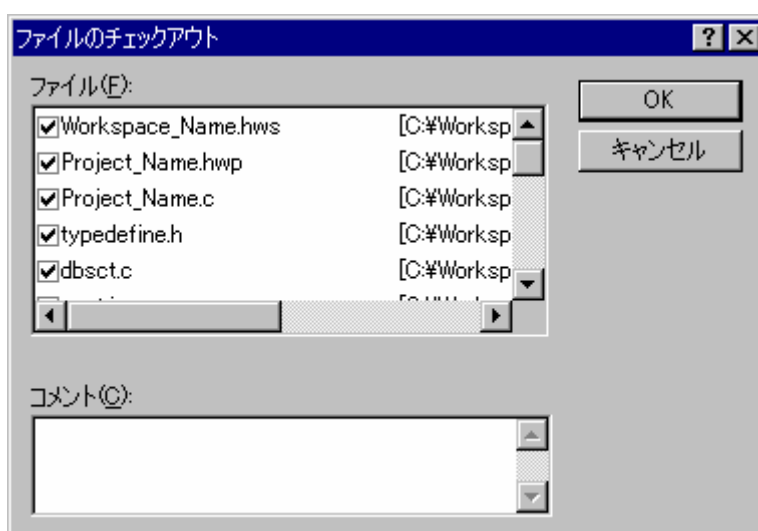
- 取得するファイルの左のチェックボックスをオンにし[OK]ボタンをクリックしてください。

### 9.2.4 Visual SourceSafeからファイルの書き込み可能コピーをチェックアウトする

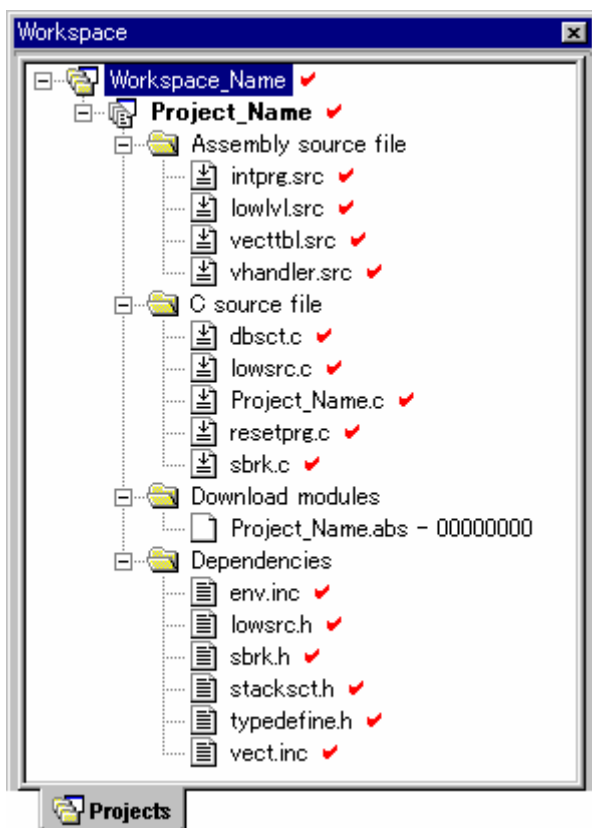
Visual SourceSafe はソースファイルを保護して、管理しているファイルの書き込み可能コピーを一度に一人のユーザだけが取得できるようにします。チェックアウト操作をすると、Visual SourceSafe からファイルの書き込み可能コピーをローカルドライブに取得します。これは、チェックアウトしようとするファイルが既に他のユーザによりチェックアウト（ファイル名の横に青いチェックマーク）されていない場合のみ可能です。

**Visual SourceSafe からファイルの書き込み可能コピーをチェックアウトするには**

1. ワークスペースウィンドウの[Projects]タブで、Visual SourceSafe からチェックアウトしたいファイルを選択してください。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選択することもできます。
2. 以下のいずれかの操作を選択してください。[ファイルのチェックアウト]ダイアログボックスが開きます。
  - [ファイルのチェックアウト]ツールバーボタン(📁)をクリックする。
  - [ツール -> バージョン管理 -> ファイルのチェックアウト]を選択する。
  - 右クリックしポップアップメニューを表示します。[バージョン管理 -> ファイルのチェックアウト]を選択する。



3. チェックアウトするファイルの左のチェックボックスをオンにし[OK]ボタンをクリックしてください。操作が完了すると、ファイル名の横に赤いチェックマークが付き、チェックアウトされたことを示します。

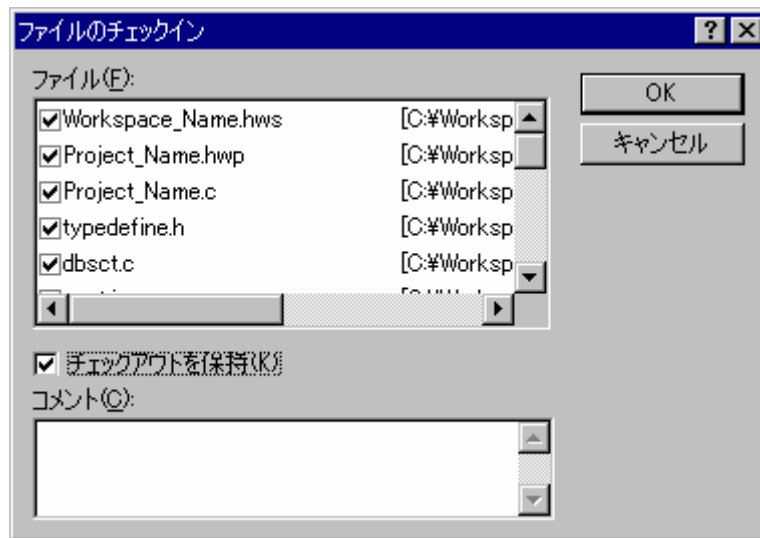


### 9.2.5 Visual SourceSafeにファイルの書き込み可能コピーをチェックインする

Visual SourceSafe はソースファイルを保護して、管理しているファイルの書き込み可能コピーを一度に一人のユーザだけが取得できるようにします。チェックアウト操作をすると、Visual SourceSafe からファイルの書き込み可能コピーをローカルドライブに取得します。チェックアウトしたファイルを編集してチェックインすると、編集結果を他のユーザが見られるようになります。

#### 編集したファイルを Visual SourceSafe にチェックインするには

1. ワークスペースウィンドウの[Projects]タブで、Visual SourceSafe に再びチェックインするファイルを選択してください。複数のファイルを選択できます。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選択することもできます。
2. 以下のいずれかの操作を選択してください。[ファイルのチェックイン]ダイアログボックスが開きます。
  - [ファイルのチェックイン]ツールバーボタン(🔒)をクリックする。
  - [ツール->バージョン管理->ファイルのチェックイン]を選択する。
  - 右クリックしポップアップメニューを表示します。[バージョン管理->ファイルのチェックイン]を選択する。



3. チェックインするファイルの左のチェックボックスをオンにし[OK]ボタンをクリックしてください。
4. ファイル名の横の赤いチェックマークが外れ、チェックインされたことを示します。ファイルをチェックインする際に[ファイルのチェックイン]ダイアログボックスの[チェックアウトを保持]チェックボックスをオンにすると、ファイルをチェックインした後もファイルをチェックアウトされた状態で書き込み可能にできます。

### 9.2.6 Visual SourceSafeのファイルの状態を表示する

High-performance Embedded Workshop プロジェクトにファイルが表示されても（ワークスペースウィンドウの[Projects]タブ）、ファイルが Visual SourceSafe に管理されているとは限りません。Visual SourceSafe に管理されているファイルのうち、チェックインされたり、チェックアウト（編集するため）されたりするものがあります。ファイルの状態コマンドは、現在のファイルの状態を表示します。

ファイルの状態の表示先は、アウトプットウィンドウの[Version Control]タブまたは[ファイルの状態]ダイアログボックスです。ファイル状態の表示先の設定は、「9.3 Visual SourceSafe コマンドのオプション」を参照してください。

#### Visual SourceSafe のファイルの状態を表示するには

1. ワークスペースウィンドウの[Projects]タブで、状態を表示するファイルを選択してください。複数のファイルを選択することもできます。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選択できます。
2. 以下のいずれかの操作を選択してください。アウトプットウィンドウの[Version Control]タブまたは[ファイルの状態]ダイアログボックスにファイルの状態を表示します。
  - [ファイルの状態]ツールバーボタン(🔒)をクリックする。
  - [ツール -> バージョン管理 -> ファイルの状態]を選択する。
  - 右クリックしポップアップメニューを表示します。[バージョン管理 -> ファイルの状態]を選択する。

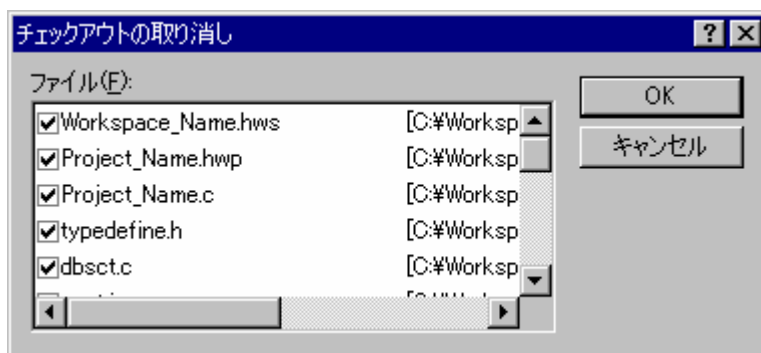
### 9.2.7 Visual SourceSafeのチェックアウトコマンドを取り消す

Visual SourceSafe はソースファイルを保護して、管理しているファイルの書き込み可能コピーを一度に一人のユーザだけが取得できるようにします。チェックアウト操作をすると、Visual SourceSafe からファイルの書き込み可能コピーをローカルドライブに取得します。チェックアウトしたファイルを編集してチェックインす

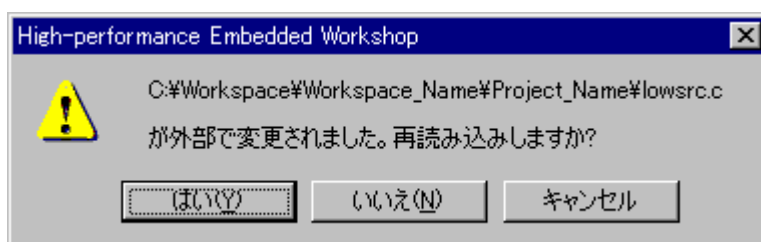
ると、編集結果を他のユーザが見られるようになります。しかし、もしチェックアウト操作を誤って行った場合、または、必要なくなった場合、チェックアウト操作を取り消すことができます。

#### Visual SourceSafe からのファイルのチェックアウト操作を取り消すには

1. ワークスペースウィンドウの[Projects]タブで、以前のチェックアウト操作を取り消したいファイルを選択してください。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選択することもできます。
2. 以下のいずれかの操作を選択してください。[チェックアウトの取り消し]ダイアログボックスが開きます。
  - [ツール->バージョン管理->チェックアウトの取り消し]を選択する。
  - 右クリックしポップアップメニューを表示します。[バージョン管理->チェックアウトの取り消し]を選択する。



3. チェックアウト操作を取り消すファイルの左のチェックボックスをオンにし[OK]ボタンをクリックしてください。
4. ファイルが変更されていると、以下の確認メッセージボックスが表示されます。チェックアウト操作を継続する場合は、[キャンセル]ボタンをクリックしてください。バージョン管理システムから変更前の内容を読み込むには[はい]ボタンをクリックしてください。[いいえ]ボタンをクリックした場合は、ローカルファイルとバージョン管理システムのファイル内容が不一致になります。



### 9.2.8 Visual SourceSafeのファイル履歴を表示する

Visual SourceSafe はプロジェクトのファイルへの編集を管理します。ファイルが最初にプロジェクトに追加されたときからの編集内容の完全なファイル履歴を表示できます。

ファイル履歴の表示先は、アウトプットウィンドウの[Version Control]タブまたは[ファイルの履歴]ダイアログボックスです。ファイル履歴の表示先の設定は、「9.3 Visual SourceSafe コマンドのオプション」を参照してください。

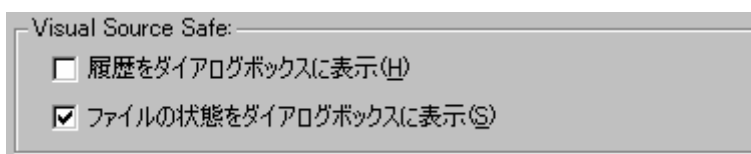
### Visual SourceSafe のファイル履歴を表示するには

1. ワークスペースウィンドウの[Projects]タブで、ファイル履歴を表示するファイルを選択してください。複数のファイルを選択できます。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在してを選択できます。
2. 以下のいずれかの操作を選択してください。アウトプットウィンドウの[Version Control]タブまたは[ファイルの履歴]ダイアログボックスにファイル履歴を表示します。
  - [ツール->バージョン管理->履歴表示]を選択する。
  - 右クリックしポップアップメニューを表示します。[バージョン管理->履歴表示]を選択する。

### 9.3 Visual SourceSafe コマンドのオプション

[ツール->バージョン管理->構成]を選択すると、履歴コマンドや状態コマンドの表示方法を変更できます。

1. [ツール->バージョン管理->構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [一般]タブを選択してください。



3. 履歴コマンドの結果をダイアログボックスで表示するには、[履歴をダイアログボックスに表示]チェックボックスをオンにしてください。履歴コマンドの結果をアウトプットウィンドウの[Version Control]タブに表示するには、[履歴をダイアログボックスに表示]チェックボックスをオフにしてください。
4. 状態コマンドの結果をダイアログボックスで表示するには、[ファイルの状態をダイアログボックスに表示]チェックボックスをオンにしてください。状態コマンドの結果をアウトプットウィンドウの[Version Control]タブに表示するには、[ファイルの状態をダイアログボックスに表示]チェックボックスをオフにしてください。

### 9.4 バージョン管理設定内容

High-performance Embedded Workshop と Visual SourceSafe を接続して使用する際の一般的な表示、設定を行います。

1. [ツール->バージョン管理->構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [一般]タブを選択してください。



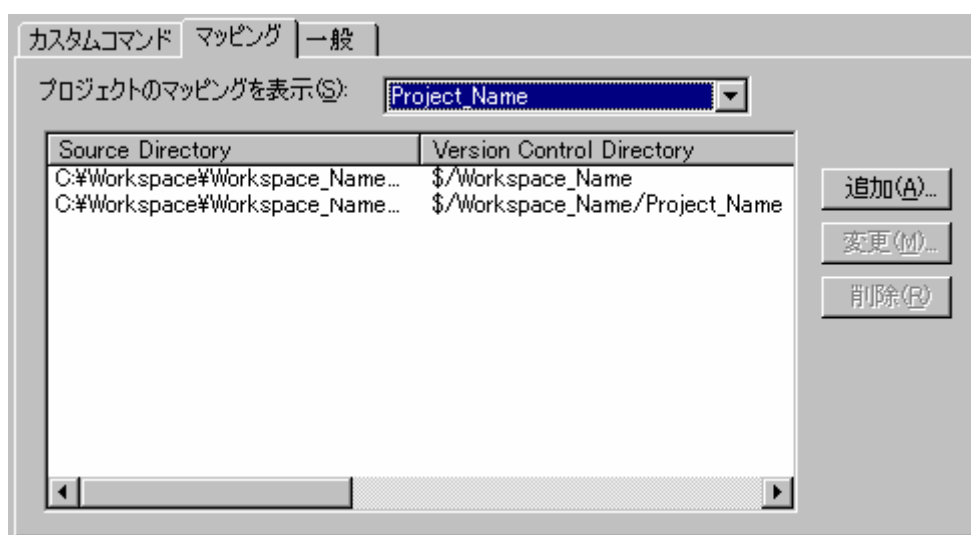
3. [ユーザ詳細]グループボックスに Visual SourceSafe にログインしているユーザ名、パスワード、およびデータベースを表示します。
4. [ログイン]ボタンをクリックすると、ユーザ名、パスワード、およびデータベースを変更できます。
5. [構成ファイル]グループボックスに現在のバージョン管理構成ファイルを表示します。バージョン管理設定の詳細が保存されている構成ファイルです。

## 9.5 ディレクトリをマッピングする

プロジェクトの現在のマッピング情報を表示します。

[プロジェクトのマッピングを表示]ドロップダウンリストでマッピングを表示するプロジェクトを選択します。リストに High-performance Embedded Workshop のワークスペースおよびプロジェクトのディレクトリ、および Visual SourceSafe のプロジェクトのディレクトリを表示します。プロジェクトのマッピングは、追加、変更、または削除できます。

詳細は、「8.6 ディレクトリをマッピングする」を参照してください。



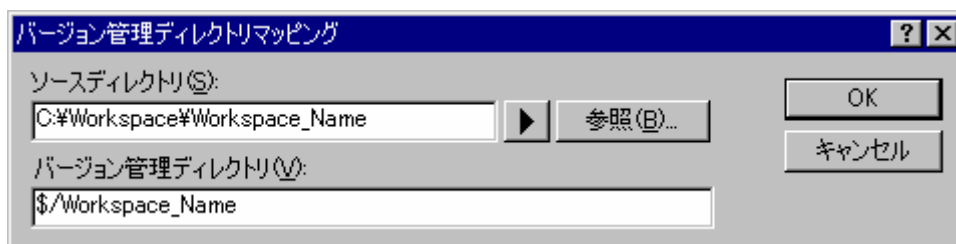
### 新しいマッピングを定義するには

1. [ツール -> バージョン管理 -> 構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [マッピング]タブを選択してください。

3. [追加]ボタンをクリックしてください。[バージョン管理ディレクトリマッピング]ダイアログボックスを開きます。
4. [ソースディレクトリ]に High-performance Embedded Workshop のワークスペースまたはプロジェクトのディレクトリを設定してください。以下のいずれかの操作を選択してください。
  - ディレクトリを直接入力する。
  - プレースホルダボタンをクリックして表示されるメニューから、Project directory、または Workspace directory を選択する。
  - [参照]ボタンをクリックして表示される[ソースディレクトリを参照]ダイアログボックスで、ディレクトリを選び[選択]ボタンをクリックする。
5. [バージョン管理ディレクトリ]にソースディレクトリへマップする Visual SourceSafe のプロジェクトのディレクトリを入力してください。
6. [OK]ボタンをクリックしてください。

#### 既存のマッピングを変更するには

1. [ツール->バージョン管理->構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [マッピング]タブを選択してください。
3. マッピングを変更するリストを選択し、[変更]ボタンをクリックしてください。[バージョン管理ディレクトリマッピング]ダイアログボックスを開きます。



4. [バージョン管理ディレクトリマッピング]ダイアログボックスで新しいマッピングを定義したときと同様の手順で設定し、マッピングを変更してください。
5. [OK]ボタンをクリックしてください。

#### 既存のマッピングを削除するには

1. [ツール->バージョン管理->構成]を選択してください。[バージョン管理設定]ダイアログボックスを開きます。
2. [マッピング]タブを選択してください。
3. マッピングを削除するリストを選択し[削除]ボタンをクリックしてください。

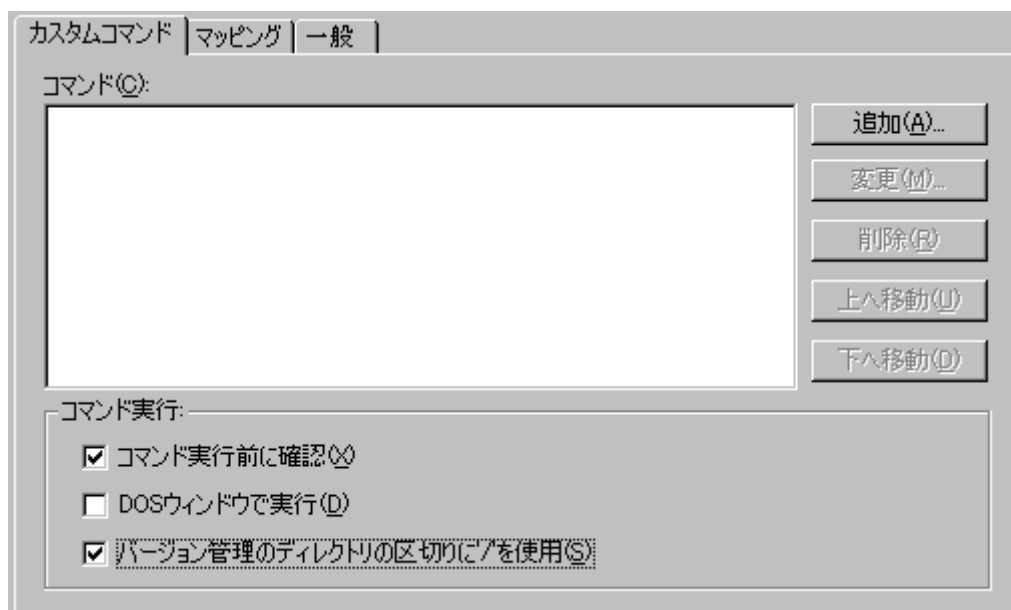
## 9.6 Visual SourceSafe コマンドを追加する

[バージョン管理設定]ダイアログボックスの[カスタムコマンド]タブでバージョン管理サブメニューにない Visual SourceSafe のコマンドを定義したメニューを追加できます。

このメニューを選択すると定義したコマンドが起動できます。このコマンド実行した際の出力情報はアウトプットウィンドウの[Version Control]タブに表示されます。

[ツール->バージョン管理->構成]を選択し、[バージョン管理設定]ダイアログボックスを開きます。

ここでは、Visual SourceSafe の Properties コマンドの定義を例に説明します。



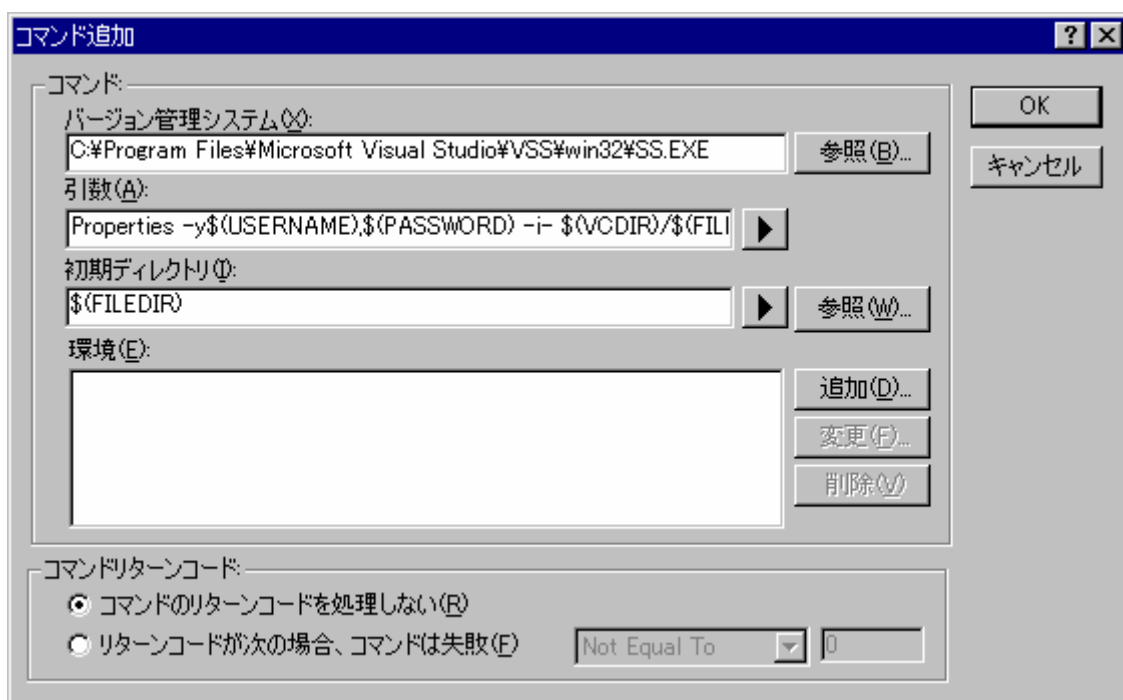
- **バージョン管理コマンド実行時の制御**

1. [コマンド実行前に確認]チェックボックスをオンにします。
2. [DOS ウィンドウで実行]チェックボックスをオフにします。Visual SourceSafeはこのフォーマットを使用します。
3. [バージョン管理のディレクトリの区切りに/を使用]チェックボックスをオンにします。Visual SourceSafeはこのフォーマットを使用します。

詳細は、「8.9 バージョン管理システムの実行を制御する」を参照してください。

- **コマンド追加**

1. [追加]ボタンをクリックします。[メニューオプションの追加]ダイアログボックスが開きます。
2. [オプション]に"Properties"と入力し、[追加]ボタンをクリックします。[コマンド追加]ダイアログボックスが開きます。



3. [バージョン管理システム]にバージョン管理システムの実行可能ファイル(SS.EXE)を入力してください。
4. [引数]に"Properties -y\$(USERNAME),\$(PASSWORD) -i \$(VCDIR)/\$(FILENAME)"を設定します。
5. [初期ディレクトリ]に"\$(FILEDIR)"を設定します。
6. [OK]ボタンをクリックします。
7. [メニューオプションの追加]ダイアログボックスで[OK]ボタンをクリックします。  
詳細は、「8.2 バージョン管理コマンドを定義する」を参照してください。

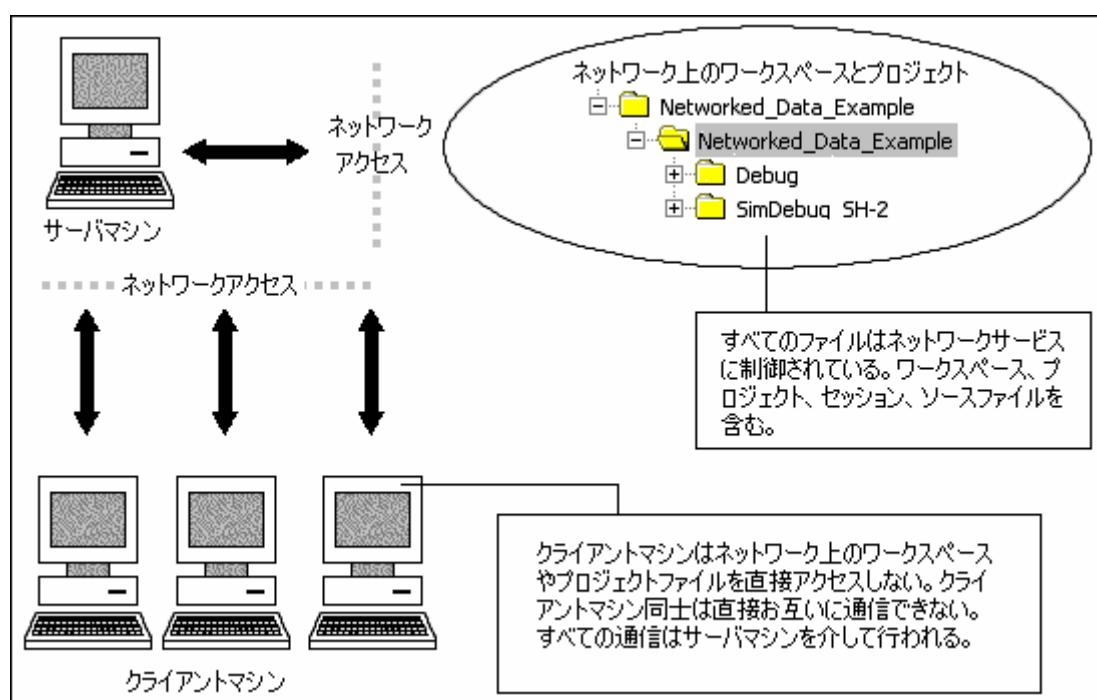
#### • コマンド実行

1. [ツール -> バージョン管理 -> Properties]を選択します。
2. [Properties]ダイアログボックスでファイルを選択し[OK]ボタンをクリックします。
3. アウトプットウィンドウの[Version Control]タブに結果情報が表示されます。

## 10. ネットワークを利用したプロジェクト共有機能

High-performance Embedded Workshop は、ネットワークを介してワークスペースやプロジェクトを共有できます。この機能により、ユーザは共有したプロジェクトを同時に操作し、また、お互いの変更を見ることができます。この機能は、バージョン管理機能とともに使用できます。バージョン管理機能とこの機能の主な違いは、ユーザがワークスペースやプロジェクトを変更、更新したときに、他のユーザがプロジェクトを再読み込みする必要がないことです。

このシステムは、ネットワーク上の1つのコンピュータをサーバマシンとして使うことで実現します。他のすべてのクライアントマシンはサーバマシンが提供するサービスを利用できます。例えば、クライアントマシンの1つが新規ファイルを追加すると、サーバマシンへファイルの追加が通知されます。すると、サーバマシンは他のすべてのクライアントマシンにファイルの追加を通知します。この構成を下図に示します。



ネットワーク機能によってユーザはファイルへのアクセス権を与られます。これによって、プロジェクトアドミニストレータから書き込みを許可されたユーザのみが、プロジェクトやソースファイルを変更できるようになります。アドミニストレータは、各ユーザの書き込み権を、プロジェクトの各自のエリアに制限して、他のエリアを読み出し専用にできます。こうして、各ユーザによるプロジェクトの他の部分への競合や破壊を未然に防ぎます。これらの制限は異なるレベルに設定できます。

### 注：

- ネットワークを利用したプロジェクト共有機能は、Windows Vista® Operating System 上では使用できません。
- ある操作をクライアントマシンで実行している間は、他のクライアントマシンでロックされていることがあります。例えば、あるユーザがツールチェインオプションを変更している間、他のすべてのユーザはこのオプションを変更できません。
- ネットワーク機能を使うと High-performance Embedded Workshop の性能は低下します。小規模なチームで作業する場合は、この機能を使用せずにバージョン管理を使うことを推奨します。

## Windows® XP Service Pack 2 以降を搭載した PC での High-performance Embedded Workshop 起動前に実施していただくこと

次節の「10.1 Windows® XP Service Pack 2 以降を搭載した PC で本機能を使う場合の設定」を参照してください。

### サーバマシンでのプロジェクトファイル設定手順

1. ネットワーク用のプロジェクトファイルを作成します。
2. [新規プロジェクトワークスペース]ダイアログボックスのディレクトリの指定には、サーバマシンとクライアントマシンから認識できるネットワークパス名を指定してください。  
(例) ¥¥WorkGroup¥Sever1¥ShareDirectory

### ワークスペースの共有の仕組み

サーバマシンでのプロジェクトファイル設定手順の 2. で指定したディレクトリは、共有設定をしてください。High-performance Embedded Workshop のネットワーク機能は、サーバマシンにあるソース、オブジェクト、その他のファイルを共有します。これらのファイルは、サーバマシン上のみに存在し、クライアントマシンからはネットワークを介してアクセスします。

### クライアントマシンからワークスペースを開く方法

クライアントマシンからネットワーク上のワークスペースを開くには、[ワークスペースを開く]ダイアログボックスでネットワークパスを検索しワークスペースを選択します。

## 10.1 Windows® XP Service Pack 2 以降を搭載した PC で本機能を使う場合の設定

Windows® XP Service Pack 2 でセキュリティ強化機能が搭載されました。

Windows® XP Service Pack 2 以降を搭載した PC の場合、Windows® operating system の初期状態では High-performance Embedded Workshop のネットワークを利用したプロジェクト共有機能を使用できません。High-performance Embedded Workshop を起動する前に下記のように PC の設定を変更し、サーバマシン、クライアントマシンがネットワークを介して High-performance Embedded Workshop システムへアクセスできるようにする必要があります。サーバマシンおよびクライアントマシンが Windows® XP Service Pack 2 以降を搭載している場合は両方とも設定してください。

### • サーバマシンが Windows® XP Service Pack 2 以降を搭載している場合

#### ステップ 1: Firewall の設定

1. Windows® スタートメニューの[コントロール パネル]を選択します。
2. [コントロール パネル]で[ネットワークとインターネット接続]をクリックします。
3. [ネットワークとインターネット接続]で[Windows ファイアウォール]をクリックします。[Windows ファイアウォール]ダイアログボックスが開きます。
4. [例外]タブをクリックします。
5. [プログラムの追加]ボタンをクリックし、[プログラムの追加]ダイアログボックスを開きます。
6. [参照]ボタンをクリックし、High-performance Embedded Workshop のインストールディレクトリにある

¥System¥Sec¥HewServer¥Hew3Server.exe

を登録します。

7. OK ボタンをクリックし、[プログラムの追加]ダイアログボックスを閉じます。

8. [ポートの追加]ボタンをクリックし、[ポートの追加]ダイアログボックスを開きます。
9. ポート名とポート番号を入力し[TCP]ラジオボタンを選択します。  
名前： DCOM  
ポート番号： 135  
TCP： 選択
10. OK ボタンをクリックし、[ポートの追加]ダイアログボックスを閉じます。
11. [Windows ファイアウォール]ダイアログボックスの OK ボタンをクリックします。
12. [ネットワークとインターネット接続]を閉じます。
13. [コントロール パネル]を閉じます。

## ステップ 2： DCOM の設定

1. Windows® スタートメニューの[ファイル名を指定して実行]を選択します。[ファイル名を指定して実行]ダイアログボックスが開きます。
2. [名前]に"dcomcnfg"を入力し OK ボタンをクリックします。[コンポーネントサービス]ウィンドウが開きます。
3. ツリーを展開します。ツリーの展開時に[Windows セキュリティの重要な警告]ダイアログボックスが表示された場合は、[ブロックする]ボタンをクリックします。  
コンソールルート -> コンポーネントサービス -> コンピュータ -> マイコンピュータ
4. マイコンピュータを右クリックし、表示されるポップアップメニューから[プロパティ]を選択します。[マイコンピュータのプロパティ]ダイアログボックスが開きます。
5. [COM セキュリティ]タブをクリックします。
6. [アクセス許可]の[制限の編集]ボタンをクリックし、[アクセス許可]ダイアログボックスを開きます。
7. [グループ名またはユーザ名]リストで"ANONYMOUS LOGON"を選択します。
8. [ANONYMOUS LOGON のアクセス許可]リストで[リモートアクセス]の[許可]チェックボックスをオンにします。
9. OK ボタンをクリックし、[アクセス許可]ダイアログボックスを閉じます。
10. [起動とアクティブ化のアクセス許可]の[制限の編集]ボタンをクリックし、[起動許可]ダイアログボックスを開きます。
11. [グループ名またはユーザ名]リストで"Everyone"を選択します。
12. [Everyone のアクセス許可]リストで、[ローカルからの起動]および[リモートからのアクティブ化]の[許可]チェックボックスをオンにします。
13. OK ボタンをクリックし、[起動許可]ダイアログボックスを閉じます。
14. [マイコンピュータのプロパティ]ダイアログボックスの OK ボタンをクリックします。
15. [コンポーネントサービス]ウィンドウを閉じます。

### • クライアントマシンが Windows® XP Service Pack 2 以降を搭載している場合

#### Firewall の設定

1. Windows® スタートメニューの[コントロール パネル]を選択します。
2. [コントロール パネル]で[ネットワークとインターネット接続]をクリックします。
3. [ネットワークとインターネット接続]で[Windows ファイアウォール]をクリックします。[Windows ファイアウォール]ダイアログボックスが開きます。
4. [例外]タブをクリックします。

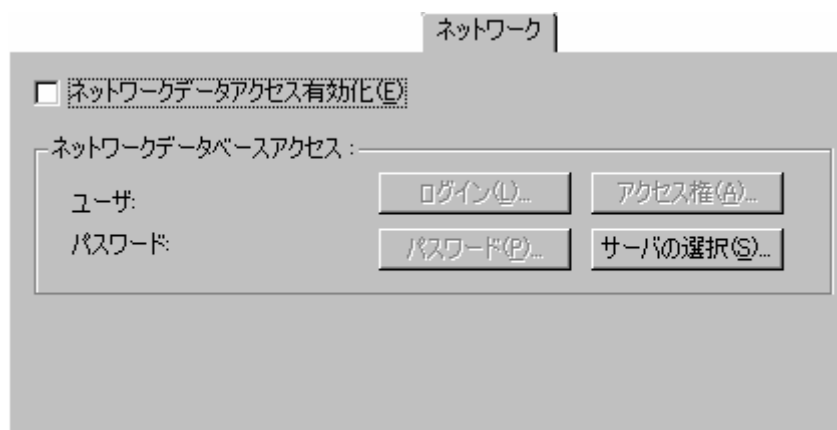


5. [プログラムの追加]ボタンをクリックし、[プログラムの追加]ダイアログボックスを開きます。
6. [参照]ボタンをクリックし、High-performance Embedded Workshop インストールディレクトリにある  
HEW2.exe  
を登録します。
7. OK ボタンをクリックし、[プログラムの追加]ダイアログボックスを閉じます。
8. [ポートの追加]ボタンをクリックし、[ポートの追加]ダイアログボックスを開きます。
9. ポート名とポート番号を入力し[TCP]ラジオボタンを選択します。  
名前： DCOM  
ポート番号： 135  
TCP： 選択
10. OK ボタンをクリックし、[ポートの追加]ダイアログボックスを閉じます。
11. [Windows ファイアウォール]ダイアログボックスの OK ボタンをクリックします。
12. [ネットワークとインターネット接続]を閉じます。
13. [コントロール パネル]を閉じます。

## 10.2 ネットワークを利用したプロジェクト共有機能を使う

### ネットワークを利用したプロジェクト共有機能を使うには

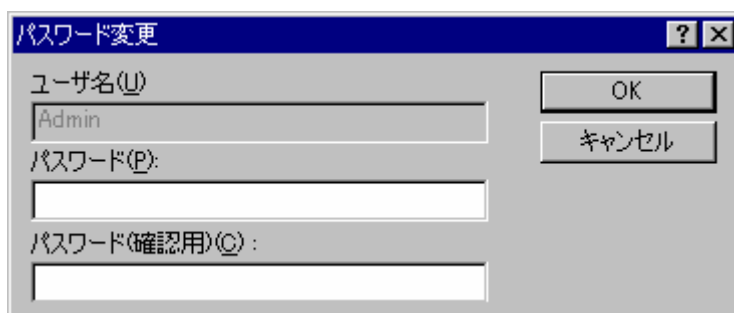
1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ネットワーク]タブを選択してください。
3. [ネットワークデータアクセス有効化]チェックボックスをオンにしてください。パスワードなしでアドミニストレータをシステムに追加します。アドミニストレータのみが、ユーザを追加したり、ユーザアクセス権を変更したりできます。
4. [ネットワーク]ダイアログボックスで、アドミニストレータはパスワードを設定します。「10.3 アドミニストレータのパスワードを設定する」で説明します。
5. [オプション]ダイアログボックスを閉じます。
6. ダイアログボックスを閉じると、ワークスペースを保存して再び開くかどうか確認されます。これは、共有アクセスモードでワークスペースを再び開く必要があるためです。ワークスペースの変更を保存しないと、変更は無効になります。ネットワークを利用したプロジェクト共有機能を使う場合、[OK]ボタンをクリックしてください。
7. ワークスペースを再び開きログインすると、現在のあなたのアクセス権を示すダイアログボックスを開きます。例えば、アドミニストレータユーザの場合、レベルはアドミニストレータになります。このダイアログボックスを閉じると、High-performance Embedded Workshop サーバウィンドウが開き、ネットワークを利用したプロジェクト共有機能が使えるようになります。



### 10.3 アドミニストレータのパスワードを設定する

#### パスワードを設定するには

1. 「10.2 ネットワークを利用したプロジェクト共有機能を使う」の手順の4から続けて操作してください。
2. [パスワード]ボタンをクリックしてください。ネットワークデータベースアクセスを有効にしたとき、このボタンも有効になります。[パスワード変更]ダイアログボックスを開きます。



3. [ユーザ名]は読み出し専用です。この場合、Adminになります。
4. 両方のフィールドに新規パスワードを入力して、[OK]ボタンをクリックしてください。[オプション]ダイアログボックスの[ネットワーク]タブにユーザ名とパスワードを設定します。

### 10.4 新規ユーザを追加する

ネットワークデータベースの初期設定では、アドミニストレータユーザとゲストユーザをシステムに追加します。High-performance Embedded Workshop システムでは、以下のアクセス権のレベルを設定できます。

#### アドミニストレータ

High-performance Embedded Workshop のすべての機能にフルアクセスできます。プロジェクトからユーザを削除したり、アクセス権レベルを変更できます。ワークスペース、プロジェクトファイル、ソースファイルを変更できます。

#### フルリード/ライトアクセス

ワークスペース、プロジェクトファイル、ソースファイルを変更できます。ユーザのアクセス権レベルは変更できません。

#### ファイルのみリード/ライトアクセス

ソースファイルのみ変更できます。すべてのプロジェクト設定を表示できますが、変更できません。

#### リードアクセス

すべてのソースファイルとプロジェクトファイルが表示できます。変更はできません。

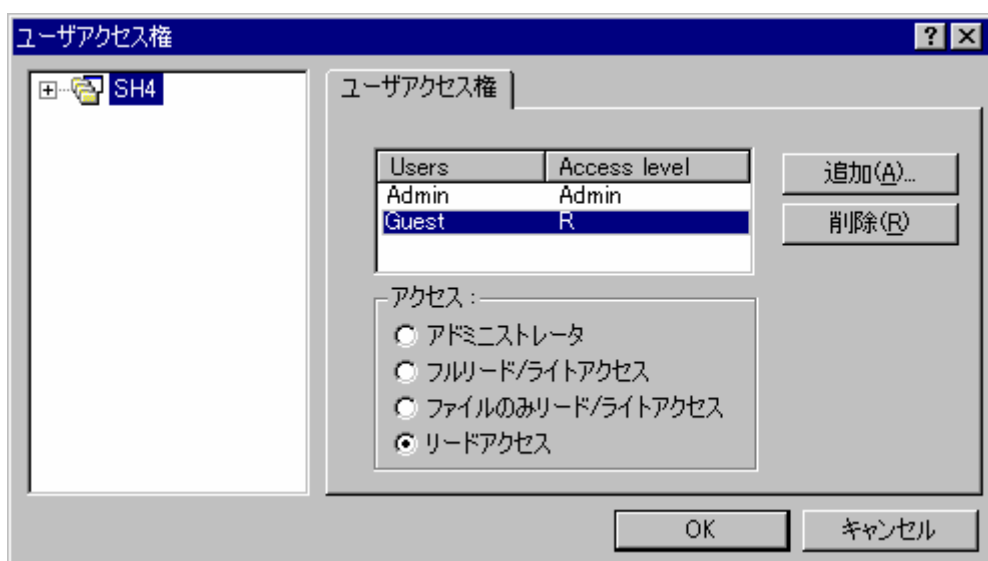
どのユーザも、ネットワーク上で使用できるプロジェクトを開くときにはユーザ名とパスワードを入力します。入力しないと、アクセス権は与えられません。入力すると、ユーザには上記のアクセス権レベルのうちのいずれかのレベルを与えられます。

#### 新規ユーザを追加するには

1. アドミニストレータのアクセス権をもつユーザがログインします。この詳細は、上記を参照してください。
2. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
3. [ネットワーク]タブを選択してください。
4. [アクセス権]ボタンをクリックしてください。[ユーザアクセス権]ダイアログボックスを開きます。
5. [追加]ボタンをクリックしてください。[新規ユーザ追加]ダイアログボックスを開きます。あなたをアドミニストレータユーザとして新規のログイン名とパスワードを追加できます。通常、パスワードはデフォルトのテキストか空白のまま設定します。次に、[OK]ボタンをクリックしてください。
6. [OK]ボタンをクリックすると、ユーザをリードアクセスのアクセス権レベルで追加します。アクセス権レベルを変更するには、ユーザ名を選んで必要なラジオボタンをクリックしてください。[OK]ボタンをクリックすると、アクセス権レベルの変更を保存します。

#### 既存ユーザを削除するには

1. アドミニストレータのアクセス権をもつユーザがログインします。この詳細は、上記を参照してください。
2. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
3. [ネットワーク]タブを選択してください。
4. [アクセス権]ボタンをクリックしてください。[ユーザアクセス権]ダイアログボックスを開きます。
5. 削除するユーザ名を選択してください。
6. [削除]ボタンをクリックしてください。
7. [OK]ボタンをクリックするとアクセス権の変更を保存します。



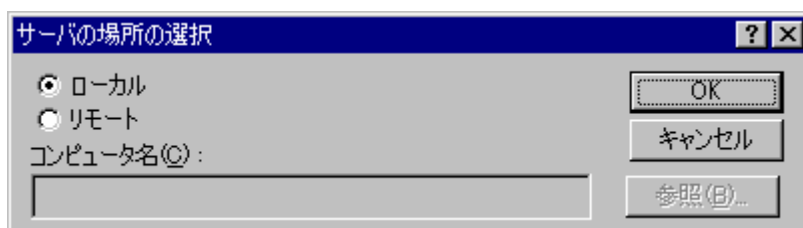
## 10.5 パスワードを変更する

### パスワードを変更するには

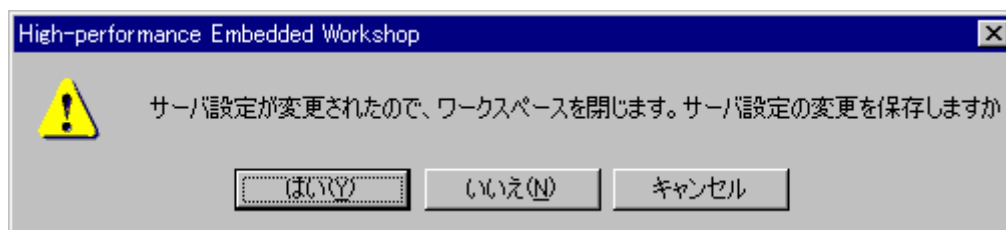
1. パスワードを変更する High-performance Embedded Workshop ネットワークデータベースにログインして、[基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [ネットワーク]タブを選択してください。
3. [パスワード]ボタンをクリックしてください。
4. 旧パスワードを入力してください。
5. 新しいパスワードを入力して3番目のエディットボックスで同じパスワードを入力してください。
6. [OK]ボタンをクリックしてください。
7. 次に、[OK]ボタンをクリックするとパスワードの変更を保存します。

## 10.6 ネットワークを利用したプロジェクト共有機能を使う場合の留意事項

最初にネットワーク上のプロジェクトを共有するとき、High-performance Embedded Workshop は自動的にサーバマシンを探して接続します。サーバマシンはマシン名で定義します。ワークスペースが、マシン名で見つからないときは、以下に示すダイアログボックスを開きます。サーバマシンの場所を入力するか参照して、[OK]ボタンをクリックしてください。設定するマシンをサーバマシンとして使用する場合は、ラジオボタンをデフォルト設定(ローカル)のままにしてください。



ワークスペースのサーバマシンとして使用していたマシンを他のマシンに接続しようとする時、以下のメッセージを表示します。[はい]ボタンをクリックすると新しいマシンに接続します。



### 注：


ネットワークを利用したプロジェクト共有機能を有効にした場合、ユーザは複数のワークスペースの中から1つのワークスペースにしかアクセスできません。ただし、同一マシン(サーバマシンとクライアントマシンが同じ)の場合、複数のワークスペースにアクセスできます。

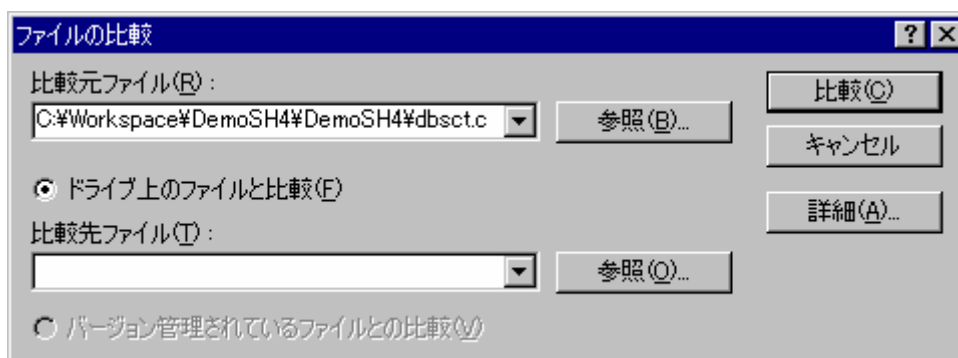
## 11. ファイルの比較

High-performance Embedded Workshop には差分ウィンドウがあります。ドライブの2つのローカルファイル、またはローカルファイルと Microsoft® Visual SourceSafe システムのファイルと比較して相違点を表示できます。

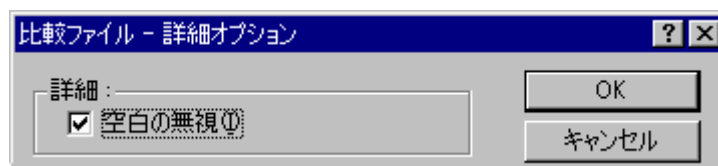
### 11.1 差分ウィンドウを開く

#### ローカルドライブの2つのファイルと比較するには

- 以下のいずれかの操作を選択し[ファイルの比較]ダイアログボックスを開きます。
  - ワークスペースウィンドウの[Projects]タブで比較したいファイルを選択してください。選択したファイル上で右クリックしポップアップメニューを表示してください。[ファイルの比較]を選択してください。
  - [差分]ツールバーボタン(  )をクリックしてください。
  - [表示 -> 差分]を選択してください。ウィンドウ内で右クリックしポップアップメニューを表示してください。[比較]を選択してください。




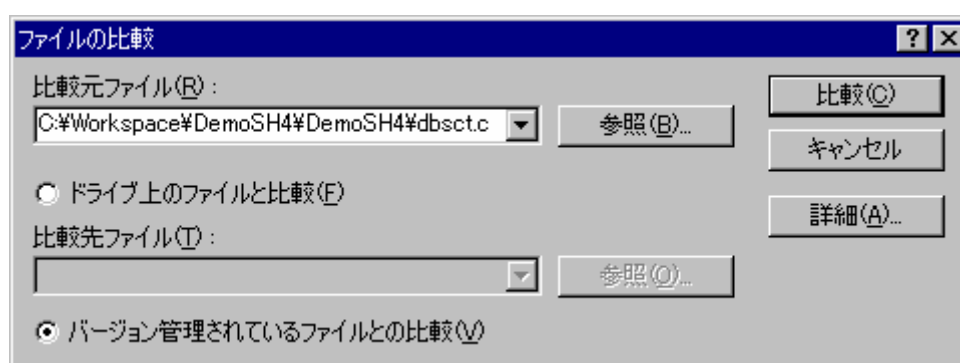
- [ドライブ上のファイルと比較]ラジオボタンが選択されていることを確認してください。
- 比較する2つのファイルを入力してください。「1.」でワークスペースウィンドウからダイアログボックスを表示した場合は、[比較元ファイル]にファイルが入力されています。ドロップダウンリストボックスから前回指定したファイルを選択するか、[参照]ボタンをクリックしファイルを参照できます。
- [詳細]ボタンをクリックすると、[比較ファイル-詳細オプション]ダイアログボックスを開きます。空白を考慮せずに相違点の比較ができます。[比較ファイル-詳細オプション]ダイアログボックスを終了するには[OK]ボタンをクリックしてください。



- [比較]ボタンをクリックしてください。

### ローカルファイルと Visual SourceSafe のファイルを比較するには

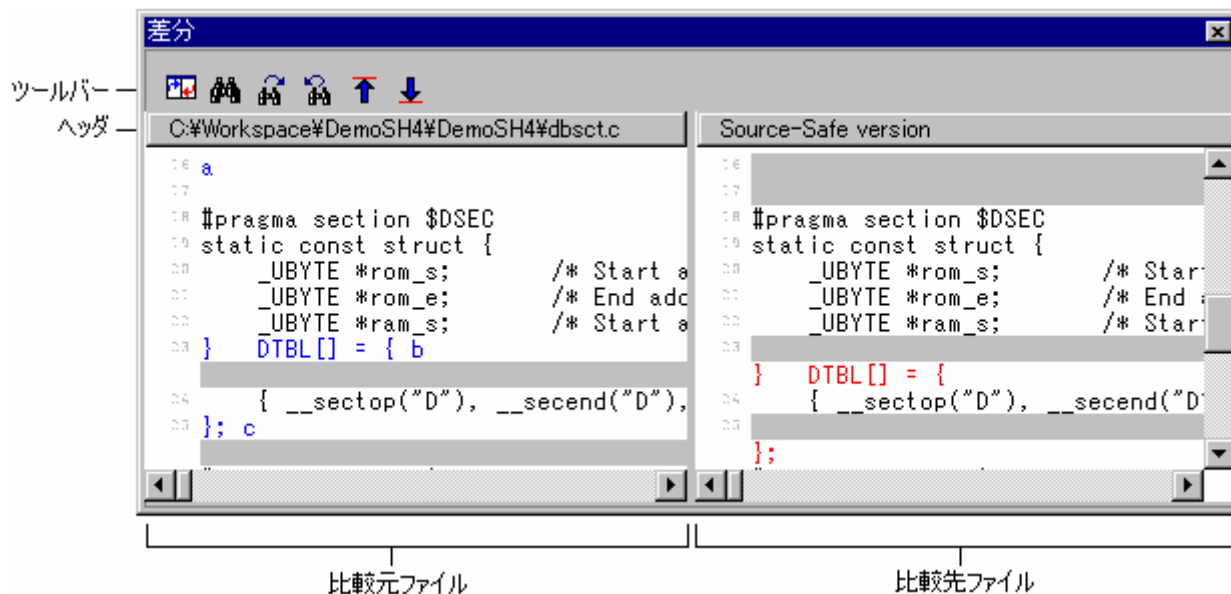
1. Visual SourceSafe システムと High-performance Embedded Workshop が接続されている必要があります。また、比較するファイルはあらかじめ Visual SourceSafe システムに追加されている必要があります。
2. 以下のいずれかの操作を選択し[ファイルの比較]ダイアログボックスを開きます。
  - ワークスペースウィンドウの[Projects]タブで比較したいファイルを選択してください。選択したファイル上で右クリックしポップアップメニューを表示してください。[ファイルの比較]を選択してください。
  - [差分]ツールバーボタン()をクリックしてください。
  - [表示 -> 差分]を選択してください。ウィンドウ内で右クリックしポップアップメニューを表示してください。[比較]を選択してください。



3. [バージョン管理されているファイルとの比較]ラジオボタンをクリックしてください。Visual SourceSafe システムと High-performance Embedded Workshop が接続されている場合は、[バージョン管理されているファイルとの比較]ラジオボタンは選択可能になっています。
4. [比較元ファイル]に比較するファイルを入力してください。2.でワークスペースウィンドウからダイアログボックスを表示した場合は、[比較元ファイル]にファイルが入力されています。ドロップダウンリストボックスから前回指定したファイルを選択するか、[参照]ボタンをクリックしファイルを参照できます。
5. [詳細]ボタンをクリックすると[比較ファイル-詳細オプション]ダイアログボックスを開きます。空白を考慮せずに相違点の比較ができます。[比較ファイル-詳細オプション]ダイアログボックスを終了するには[OK]ボタンをクリックしてください。
6. [比較]ボタンをクリックしてください。

[ファイルの比較]ダイアログボックスが閉じ、差分ウィンドウが開きます。

## ウィンドウの構成



- 差分ウィンドウの左ペインに比較元ファイルを表示し、右ペインに比較先ファイルを表示します。
- ファイル名が各ペインのヘッダに表示されます。
- ペインの境界上にマウスイカーソルを置くと左右の矢印カーソルに変わります。クリックし希望する位置までマウスをドラッグするとペインの横幅を調節できます。

## ウィンドウのオプション

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

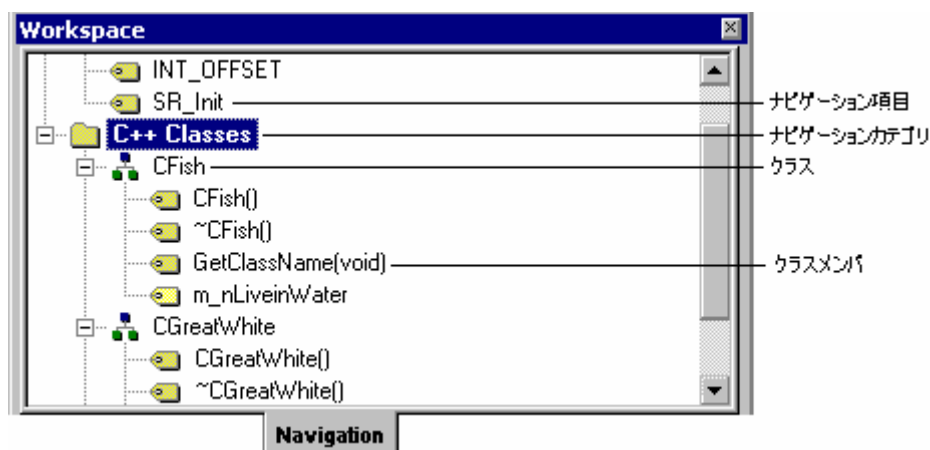
[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	マクロの記録	機能
比較		-	[ファイルの比較]ダイアログボックスを開きます。新たにファイルと比較し相違点を表示できます。
結果のファイル出力	-	-	ダイアログボックスが開き、現在の相違点の結果をテキストフォーマットで保存できます。
空白の無視		-	空白文字を無視します。このメニューオプションは[詳細オプション]ダイアログボックスの[空白の無視]オプションから選択することもできます。
検索		-	標準の[検索]ダイアログボックスを開きます。High-performance Embedded Workshop エディタと同じダイアログボックスです。
前を検索		-	検索項目に当てはまる次の前方の文字列を検索します。
次を検索		-	検索項目に当てはまる次の後方の文字列を検索します。
前の差分		-	次の後方の相違点に自動的にジャンプします。
次の差分		-	次の前方の相違点に自動的にジャンプします。
再比較	-	-	表示をリフレッシュして再び手動で相違点の比較を実行します。どちらかのファイルと比較後、変更したときに便利です。
ツールバー表示	-	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	-	ツールバーボタンをカスタマイズします。



## 12. ナビゲーション機能

High-performance Embedded Workshop にはナビゲーション機能があります。サポートされているすべてのナビゲーションの種類をワークスペースウィンドウの[Navigation]タブに表示します。



以下のナビゲーション機能を標準でサポートしています。

ナビゲーションの種類(カテゴリ)	機能
C Defines	C ソースファイルのすべての#define の表示
C Functions	C ソースファイルのすべての ANSI C 標準関数の表示
C++ Classes	C++ソースファイルのすべてのクラス、関数、メンバの表示

デフォルトでナビゲーション項目をカテゴリ単位で表示できます。各カテゴリの下にアクティブプロジェクトのナビゲーション項目をアルファベット順に表示します。

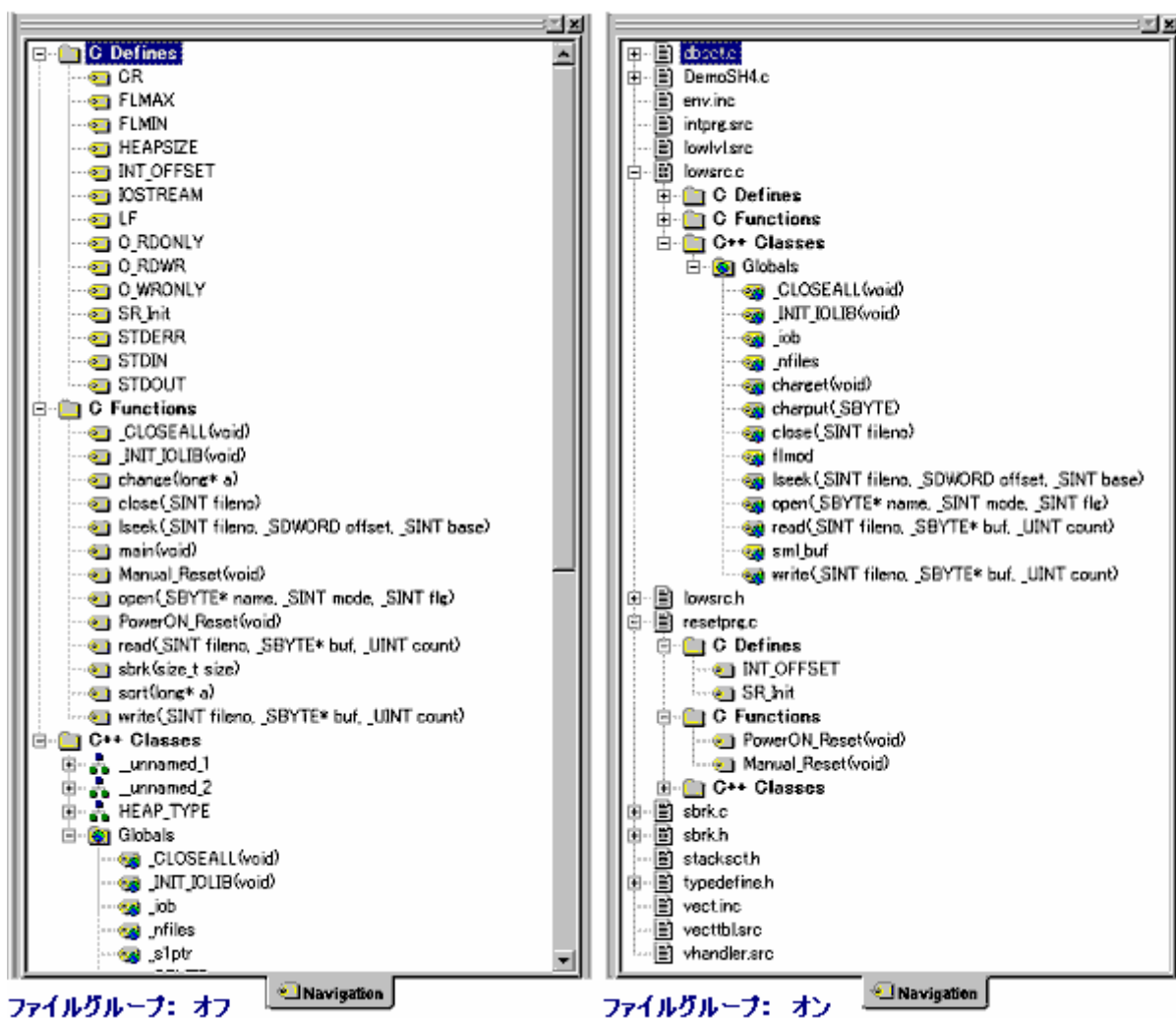
### カテゴリ単位でナビゲーション項目を表示するには

1. [Navigation]タブ内の任意の位置で右クリックしポップアップメニューを表示してください。
2. [ファイルグループ]のチェックをオフにしてください。デフォルトはチェックがオフです。

ナビゲーション項目をファイル単位で表示できます。アクティブプロジェクト内の各ファイルをツリーで表示し、その下に各ファイルのナビゲーション項目をアルファベット順に表示します。

### ファイル単位でナビゲーション項目を表示するには

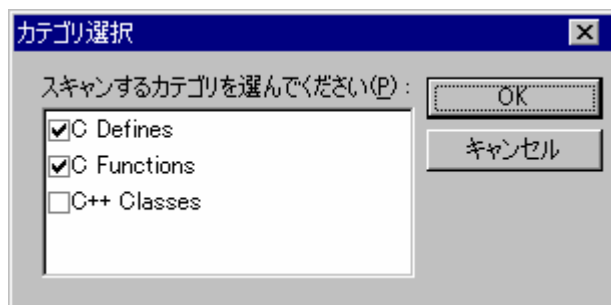
1. [Navigation]タブ内の任意の位置で右クリックしポップアップメニューを表示してください。
2. [ファイルグループ]のチェックをオンにしてください。



特定のナビゲーションカテゴリの情報を必要としない場合はスキャンを無効にできます。

### ナビゲーションカテゴリを無効にするには

1. [Navigation]タブ内の任意の位置で右クリックしポップアップメニューを表示してください。
2. [カテゴリの選択]を選択してください。[カテゴリ選択]ダイアログボックスを開きます。



3. 定義を参照する必要のないカテゴリのチェックボックスをオフにしてください。
4. [OK]ボタンをクリックしてください。

### ナビゲーション表示を更新するには

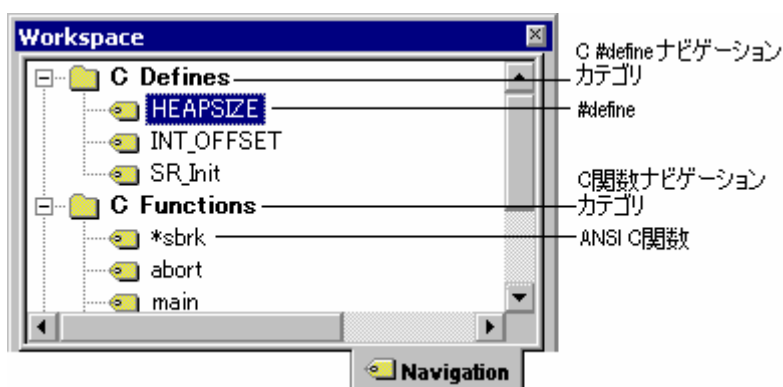
1. [Navigation]タブ内の任意の位置で右クリックしポップアップメニューを表示してください。
2. [リフレッシュ]を選択してください。

#### 注：

- ファイルのスキャン状況に応じて、ナビゲーション項目を表示します。ファイルが多数あると、ナビゲーション表示の更新を終了するのに時間がかかります。
- ファイルを保存する際に再びスキャンします。ファイルを保存するまで、新しいクラスや関数のナビゲーション情報を表示しません。
- [ファイルグループ]と[アクセスグループ]は、同時にオンにできません。一方をオンにするともう一方はオフになります。

## 12.1 C関数と#defineナビゲーションの構成

この機能では、ナビゲーション表示に関数と#define 定義を追加します。



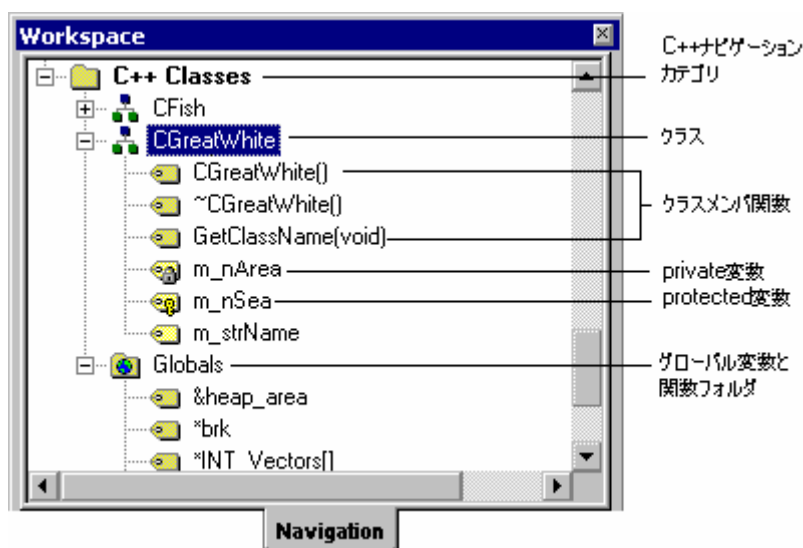
### 定義へジャンプするには

以下のいずれかの操作を選択してください。

- [Navigation]タブの関数または#define 定義項目上でダブルクリックしてください。
- [Navigation]タブの関数または#define 定義項目上で右クリックしポップアップメニューを表示してください。[定義へジャンプ]を選択してください。

## 12.2 C++ナビゲーションの構成

C++ナビゲーション機能はC++ソースファイルの表示で以下の構成をサポートします。情報の基本構成を下図に示します。



C++ナビゲーション表示では、アイコンの示す関数や変数の種類を示すためにいくつかのアイコンを用います。これらを下図に示します。

アイコン	説明
	public関数
	protected関数
	private関数
	public変数
	protected変数
	private変数

デフォルトで、C++ナビゲーション項目をダブルクリックすると、関連するC++ナビゲーション項目の宣言へジャンプします。このデフォルト動作は、ポップアップメニューの[ダブルクリックで定義へジャンプ]の選択で変更できます。デフォルトはチェックがオフです。チェックマークを付けるとナビゲーション項目をダブルクリックしたときに関連するナビゲーション項目の定義へジャンプします。

### 定義へジャンプするには

以下のいずれかの操作を選択してください。

- [Navigation]タブのC++ナビゲーション項目上で右クリックしポップアップメニューを表示してください。[ダブルクリックで定義へジャンプ]のチェックマークが付いていることを確認してください。  
[Navigation]タブのC++ナビゲーション項目上でダブルクリックしてください。
- [Navigation]タブのC++ナビゲーション項目上で右クリックしポップアップメニューを表示してください。[定義へジャンプ]を選択してください。

### 宣言ヘジャンプするには

以下のいずれかの操作を選択してください。

- [Navigation]タブの C++ナビゲーション項目上で右クリックしポップアップメニューを表示してください。[ダブルクリックで定義ヘジャンプ]のチェックマークが外れていることを確認してください。[Navigation]タブの C++ナビゲーション項目上でダブルクリックしてください。
- [Navigation]タブの C++ナビゲーション項目上で右クリックしポップアップメニューを表示してください。[宣言ヘジャンプ]を選択してください。

### メンバ変数と関数を各々アルファベット順で表示するには

1. [Navigation]タブの C++ナビゲーション項目上で右クリックしポップアップメニューを表示してください。
2. [アクセスグループ]のチェックマークをオフにしてください。デフォルトはチェックがオフです。

### public、private、protected メンバ変数と関数をグループ化して表示するには

1. [Navigation]タブの C++ナビゲーション項目上で右クリックしポップアップメニューを表示してください。
2. [アクセスグループ]のチェックマークをオンにしてください。デフォルトはチェックがオフです。

[ファイルグループ]と[アクセスグループ]は、同時にオンにできません。一方をオンにするともう一方はオフになります。

他の機能としては、特定の範囲の基底クラス、または派生クラスを表示します。

### 基底クラス、派生クラスを表示するには

1. [Navigation]タブのクラス上で右クリックしポップアップメニューを表示してください。
2. [派生クラスの表示]を選択すると、選択した範囲の派生クラスを表示します。[基底クラスの表示]を選択すると、選択した範囲の基底クラスを表示します。
3. 選択により、拡張ツリー形式で選択されたクラス構造をダイアログボックスに表示します。
4. [閉じる]ボタンをクリックしてください。

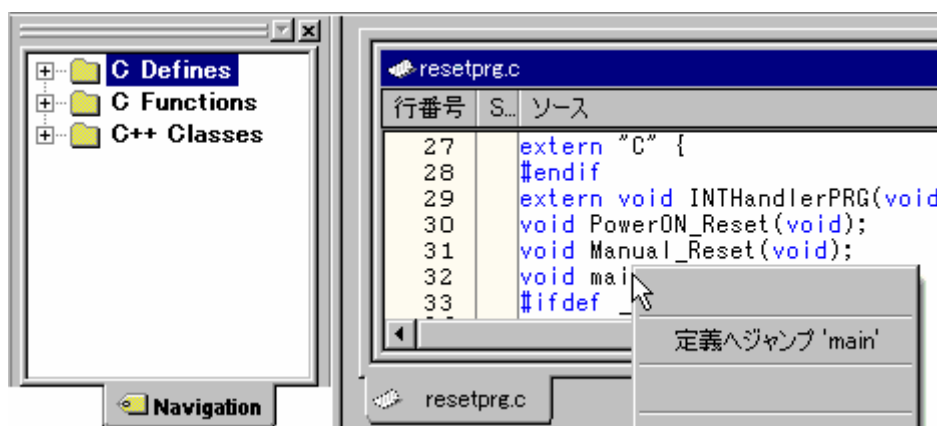
## 12.3 エディタ内から定義ヘジャンプする

エディタウィンドウに表示されているソースコードから#define、C関数、またはC++クラスなどのナビゲーション項目名を選択し、それが定義されている位置をエディタウィンドウに表示できます。

定義位置を表示できるのは、ワークスペースウィンドウの[Navigation]タブで有効なカテゴリのナビゲーション項目です。

### 定義ヘジャンプするには

1. エディタウィンドウの[ソース]領域で定義位置を表示したいナビゲーション項目名上で右クリックしてください。
2. 表示されるポップアップメニューから[定義ヘジャンプ 'ナビゲーション項目名']を選択してください。



3. 該当するナビゲーション項目が含まれる複数存在する場合は、[ナビゲーションアイテムの選択]ダイアログボックスが開きます。ファイル名および行番号がリストで表示されます。リスト項目をダブルクリックするか、リスト項目を選択して[OK]ボタンをクリックしてください。

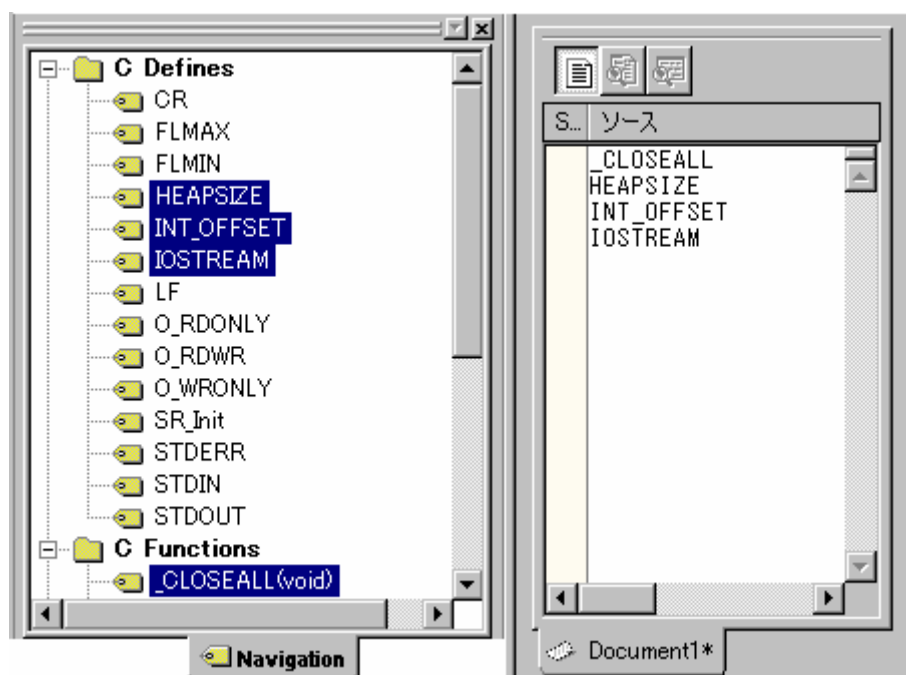


## 12.4 ナビゲーション項目をドラッグアンドドロップする

ワークスペースウィンドウの[Navigation]タブに表示されている#define、C 関数、または C++クラスなどのナビゲーション項目名をドラッグアンドドロップできます。カテゴリ (C Defines, C Functions, C++ Classes)、Globals フォルダ、ファイル項目はドラッグできません。

### エディタで開いているファイルにナビゲーション項目をドラッグアンドドロップする

ファイルに関連した項目をエディタウィンドウ内にドラッグできるので、それらをタイプする必要がなくコードを書くことを容易にします。



- 複数のアイテムを同時にドラッグできます（別々の行に表示されます）。
- ナビゲーション項目名はすべてアルファベット順に挿入されます。
- 関数をドラッグする場合、関数名だけがドラッグされます。

## 12.5 スマートエディタ

High-performance Embedded Workshop エディタの機能の 1 つにスマートエディタ機能があります。この機能はデフォルトではすべての C および C++ ソースファイルで有効です。C 関数、#define、C++ クラスやメンバ関数を使っているとき、High-performance Embedded Workshop エディタは、C 関数、#define、および C++ ナビゲーション情報にアクセスしスマートエディタリストをポップアップウィンドウで表示します。

### スマートエディタ機能を使用するには

1. [基本設定 -> オプション] を選択してください。[オプション] ダイアログボックスを開きます。
2. [エディタ] タブを選択してください。
3. [Smart-editor 機能] チェックボックスをオンにしてください。
4. [OK] ボタンをクリックしてください。

smart-editor機能(R)

このオプションをオンにすると、操作している C および C++ ファイルでスマートエディタ機能が有効になります。

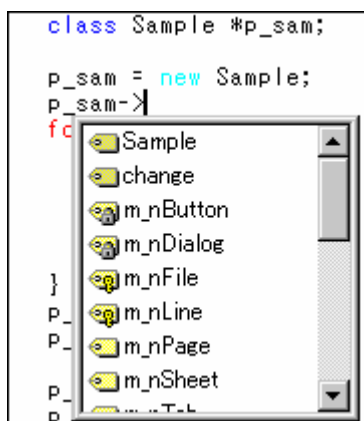
### 注：

ワークスペースウィンドウの[Navigation]タブで、"C Defines"、"C Functions"、または"C++ Classes" ナビゲーションカテゴリがオフのとき、High-performance Embedded Workshop のスマートエディタ機能は無効です。

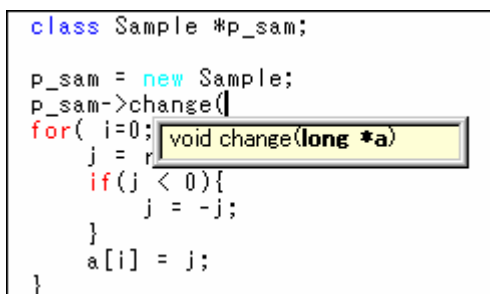


High-performance Embedded Workshop エディタを編集するとき、以下のエディタ操作によりスマートエディタリストをポップアップウィンドウで表示します。

- オブジェクトを使用し「.」や「->」でメンバをアクセスしているときは、ポップアップウィンドウが表示され、入力するより効率良くメンバを選択できます。入力中、ポップアップウィンドウ内は上下矢印キーでメンバを選択できます。Enter キーを押すと、選択したメンバを追加します。「::」を使っているときもこのポップアップメニューを表示します。メンバを追加するまでは、Ctrl+Space キーを押すとポップアップウィンドウを表示できます。



- C および C++関数を使用しているときは、最初に関数名を入力すると、ポップアップウィンドウを表示します。このポップアップウィンドウにより、現在のオブジェクトでどの関数を使用できるかわかります。関数を選択すると自動的に残りのパラメータを入力します。閉じ括弧を入力するまでは、Ctrl+Shift+Space キーを押すとポップアップウィンドウを表示できます。



- High-performance Embedded Workshop エディタウィンドウで、C 関数、#define、C++クラス、メンバ関数などを追加したい位置にポップアップウィンドウから選択した項目を追加できます。以下のいずれかの操作を選択してください。
  - High-performance Embedded Workshop エディタウィンドウ内で右クリックしポップアップメニューを表示します。[ナビゲーションリスト]を選択し、表示されるポップアップウィンドウで追加したい項目を選択し Enter キーを押す。
  - Ctrl+Space キーを押し、表示されるポップアップウィンドウで追加したい項目を選択し Enter キーを押す。

## 13. マップ

リンケージエディタのセクション設定、およびリンケージエディタが出力するリンケージリストファイルの情報からセクション情報とシンボル情報を表示します。下記の機能を備えています。

### セクション情報

- アドレス - セクショングループ - セクションの順にツリービュー表示
- ドラッグアンドドロップによるセクションの配置の変更が可能
- アドレス、セクショングループ、セクションの追加、変更、削除が可能
- 選択行のアドレスに対応したソースファイルをエディタウィンドウで表示可能
- 表示内容の印刷が可能

### シンボル情報

- すべてまたはセクション毎のシンボル情報のリスト表示
- 各シンボル情報の検索、フィルタリング、およびソートが可能
- 選択行のアドレスに対応したソースファイルをエディタウィンドウで表示可能
- 表示内容の印刷が可能

#### • 対応ツールチェーン

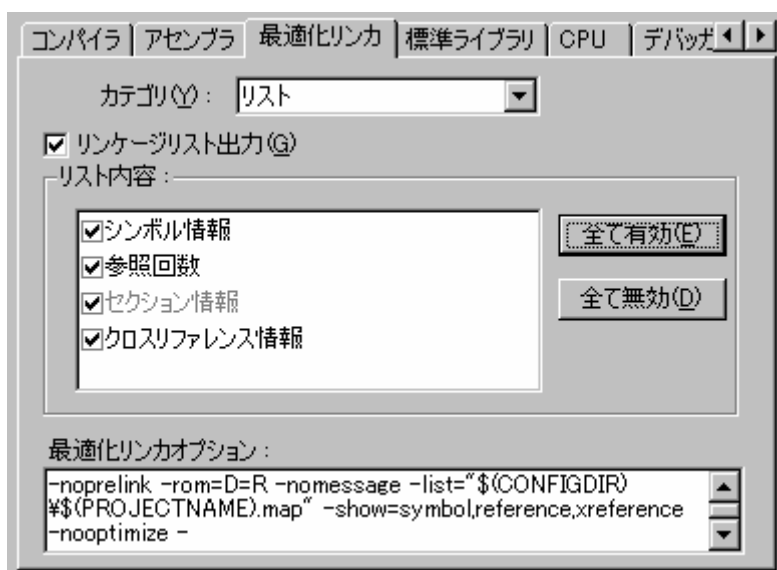
以下に示すコンパイラパッケージに含まれるツールチェーンを使用している場合、マップ機能に対応しています。

- SuperH ファミリ用 C/C++コンパイラパッケージ V.7.1.03 およびそれ以降
- H8SX, H8S, H8 ファミリ用 C/C++コンパイラパッケージ V.5.0.05 およびそれ以降
- M16C シリーズ用 C コンパイラパッケージ V.5.42 Release 00 およびそれ以降
- M32C シリーズ用 C コンパイラパッケージ V.5.41 Release 00 およびそれ以降
- R32C シリーズ用 C コンパイラパッケージ V.1.01 Release 00 およびそれ以降
- RX ファミリ用 C/C++コンパイラパッケージ V.1.00 Release 00 およびそれ以降

#### • セクション情報およびシンボル情報を表示するには

セクション情報およびシンボル情報を表示するには、あらかじめ下記を行う必要があります。

1. [ビルド -> xxxxxx Standard Toolchain]を選択し、ツールチェーンのビルドオプション設定ダイアログボックスを開きます。
2. [最適化リンク]タブの[カテゴリ]で"リスト"を選択します。
3. [リンケージリスト出力]チェックボックスをオンにします。これによりセクション情報が出力されます。
4. [全て有効]ボタンをクリックします。これによりシンボル情報が出力されます。
5. [ビルド -> ビルド]を選択します。

**注：**

下記のようにサブコマンドファイルを指定した場合は、セクション情報ウィンドウにリンケージエディタのセクション設定情報は表示されません。




### 13.1 セクション設定を管理する

Map Section Information ウィンドウを使用すると、セクション情報の追加、編集、削除が GUI 上で簡単に実行できます。

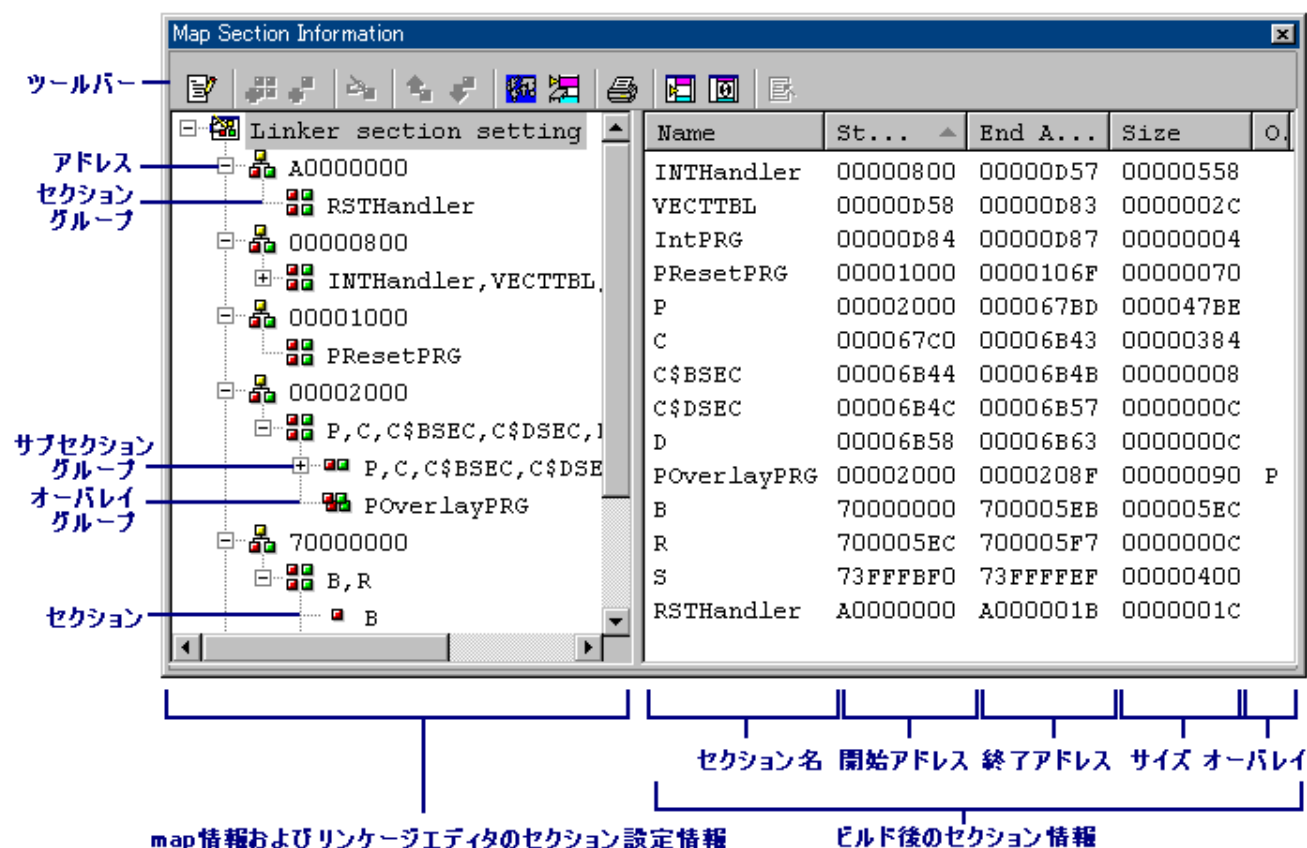
セクション編集後の確認も容易です。

### 13.1.1 Map Section Informationウィンドウを開く





#### Map Section Information ウィンドウを開くには








1. 以下のいずれかの操作を選択し[マップ種別の選択]ダイアログボックスを開きます。
  - [表示 -> マップ]を選択する。
  - [マップ]ツールバーボタン(  )をクリックする。
2. [マップ種別]ドロップダウンリストから"Map Section Information"を選択します。
3. [OK]ボタンをクリックします。

#### ウィンドウの構成



- map 情報およびリンケージエディタのセクション設定情報を左側のペインに表示します。
  - リンケージエディタの[セクション]オプションの設定情報が表示されます。
  - 編集モード時、サブセクショングループ、オーバーレイグループ、およびセクションは、ドラッグアンドドロップが可能です。

操作	ドラッグ操作中のマウスポインタの状態	ドロップ結果
Ctrl キーを押した状態で、サブセクショングループ (  )、またはセクション (  ) を選択してドラッグ	 ITEM コピー可能なドロップ対象項目上では、通常の選択時のマウスポインタになります。マウスポインタにドラッグ項目名と'+'マークが付きます。	ドラッグした項目をドロップ対象項目の次にコピーします。
Ctrl キーを押した状態で、オーバーレイグループ (  ) を選択してドラッグ		ドラッグした項目をドロップ対象項目の最後にコピーします。

操作	ドラッグ操作中のマウスポインタの状態	ドロップ結果
Shift キーを押した状態で、サブセクショングループ (  )、またはセクション (  ) を選択してドラッグ	 ITEM 移動可能なドロップ対象項目上では、通常の実際のマウスポインタになります。マウスポインタにドラッグ項目名と上矢印が付きます。	ドラッグした項目をドロップ対象項目の前に移動します。
サブセクショングループ (  )、またはセクション (  ) を選択してドラッグ	 ITEM 移動可能なドロップ対象項目上では、通常の実際のマウスポインタになります。マウスポインタにドラッグ項目名が付きます。	ドラッグした項目をドロップ対象項目の次に移動します。
オーバーレイグループ (  ) を選択してドラッグ		ドラッグした項目をドロップ対象項目の最後の項目に移動します。

- ビルド後のセクション情報を右側のペインに表示します。
  - リンケージエディタが出力するリンケージリストファイル(.map)がある場合に表示されます。
  - カラムヘッダ上のマウスのドラッグで各カラムの幅を調節できます。また、カラムヘッダ上で右クリックするとポップアップメニューを表示します。カラムが利用できる場合は、エントリの横にはチェックマークがあります。エントリをクリックすると、各カラムの表示、非表示を切り替えます。
  - リストのカラムヘッダをクリックすると、リスト項目はそのカラムによってソートされます。ウィンドウ開くと、[Start Address]の昇順でソートされています。
  - [Overlay]カラムにはオーバーレイグループのプライマリセクション名が表示されます。

## ウィンドウのオプション

- map 情報およびリンケージエディタのセクション設定情報 (左側のペイン)

左側のペイン内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
セクション情報編集モード		編集モードを切り替えます。
セクショングループの追加		セクショングループを追加します。
セクションの追加		セクションを追加します。
オーバーレイグループの追加 *1		オーバーレイグループを追加します。
セクションの自動登録		未登録セクションを自動登録します。
選択項目の編集		選択項目を編集します。
切り取り		セクション情報を切り取ります。
コピー		セクション情報をコピーします。
貼り付け		セクション情報を貼り付けます。
削除		セクション情報を削除します。
1つ上へ		セクション情報を1つ上へ移動します。
1つ下へ		セクション情報を1つ下へ移動します。
プライマリセクションに設定 *1		プライマリセクションに設定します。
メモリマップ設定 *2		メモリマップを設定するダイアログボックスを開きます。
メモリリソース自動確保 *2		メモリリソースを自動的に確保します。
印刷		セクション設定ツリーを印刷します。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。

**注：**

\*1. リンケージエディタがオーバーレイをサポートしていない場合は使用できません。

\*2. 以下に示すデバuggがサポートしています。

- SuperH ファミリ用シミュレータデバugg
- H8SX, H8S, H8 ファミリ用シミュレータデバugg
- RX ファミリ用シミュレータデバugg

- **ビルド後のセクション設定情報（右側のペイン）**

右側のペイン内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
未割り当て領域の表示		未割り当て領域を表示します。
サイズ0のセクション表示		サイズ0のセクションを表示します。
ソースファイル表示		選択行のアドレスに対応したソースコードを表示します。
印刷		セクションリストを印刷します。

### 13.1.2 セクション情報編集モードの開始と終了

左側のペインに表示されるセクション情報を編集するには、編集モードに切り替える必要があります。デフォルトは編集モードではありません。

#### 編集モードに移行するには

1. 左側のペイン内で右クリックしポップアップメニューを表示します。
2. [セクション情報編集モード]を選択します。これにより編集モードに移行し、[セクション情報編集モード]にチェックマークが付きます。

編集モードに移行すると、下記のオプションが使用可能になります。

左側のペインの各項目上で右クリックすると下記の"✓"を記載したオプションが使用可能です。

ポップアップメニューオプション	ツールバーボタン	項目					
		ルート	アドレス	セクショングループ	サブセクショングループ	オーバーレイグループ	セクション
セクショングループの追加		✓	✓	✓	✓	✓	✓
セクションの追加			✓	✓	✓	✓	✓
オーバーレイグループの追加 *1				✓	✓		✓ *2
セクションの自動登録		✓	✓	✓	✓	✓	✓
選択項目の編集			✓	✓	✓	✓	✓
切り取り					✓	✓	✓
コピー					✓	✓	✓
貼り付け			✓	✓	✓	✓	✓
削除			✓	✓	✓	✓	✓

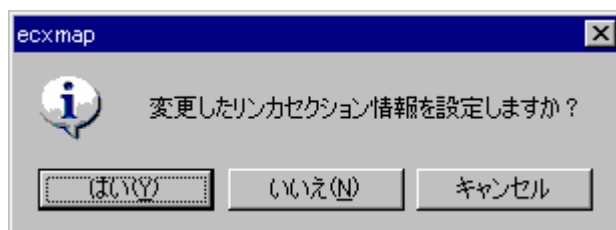
	ツールバー ボタン	項目					
		ルート	アドレス	セクション グループ	サブセクション グループ	オーバーレイ グループ	セクション
ポップアップメニューオプション							
1つ上へ							<input checked="" type="checkbox"/>
1つ下へ							<input checked="" type="checkbox"/>
プライマリセクションに設定 *1						<input checked="" type="checkbox"/>	

**注：**

- \*1. リンケージエディタがオーバーレイをサポートしていない場合は使用できません。
- \*2. 以下に示すコンパイラパッケージに含まれるツールチェーンを使用している場合にサポートしています。
  - SuperH ファミリ用 C/C++コンパイラパッケージ V.9.00 Release 04 以降
  - H8SX, H8S, H8 ファミリ用 C/C++コンパイラパッケージ V.6.01 Release 02 以降
  - RX ファミリ用 C/C++コンパイラパッケージ V.1.00 Release 00 以降

**編集モードを終了するには**

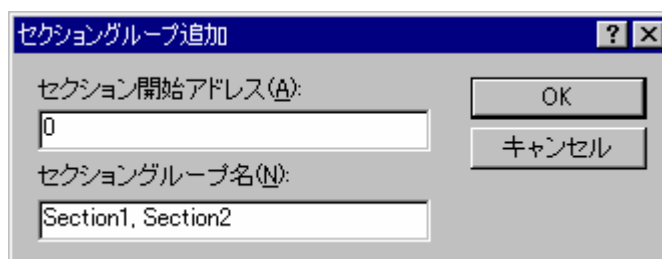
1. 左側のペイン内で右クリックしポップアップメニューを表示します。
2. [セクション情報編集モード]を選択します。
3. 下記の確認ダイアログボックスが開きます。編集した結果をリンケージエディタのセクション設定に反映するには「はい」を選択してください。これにより編集モードが終了します。[セクション情報編集モード]のチェックマークが外れます。

**13.1.3 セクショングループを追加する**

編集モードでは、セクショングループの追加が可能です。

**セクショングループを追加するには**

1. 左側のペイン内で右クリックしポップアップメニューを表示します。
2. [セクショングループの追加]を選択します。[セクショングループ追加]ダイアログボックスが開きます。





3. [セクション開始アドレス]にセクション開始アドレスを入力します。
4. [セクショングループ名]にセクショングループ名を入力します。セクショングループ名をコンマで区切って指定すると、各々がセクションとして追加されます。上の図の例を設定すると、セクション "Section1"、およびセクション "Section2"が追加されます。

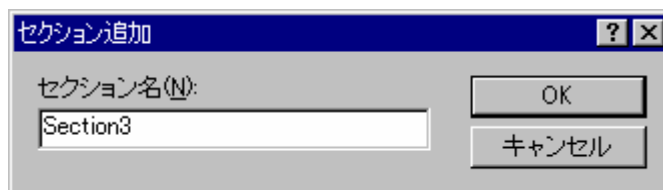
左側のペインのセクション設定ツリーで、アドレス下にセクショングループが追加されます。

### 13.1.4 セクションを追加する

編集モードでは、セクショングループ、サブセクショングループ、またはオーバーレイグループ下にセクションの追加が可能です。

#### セクションを追加するには

1. 左側のペイン内のセクショングループ、サブセクショングループ、またはオーバーレイグループ上で右クリックしポップアップメニューを表示します。
2. [セクションの追加]を選択します。[セクション追加]ダイアログボックスが開きます。



3. [セクション名]にセクション名を入力します。

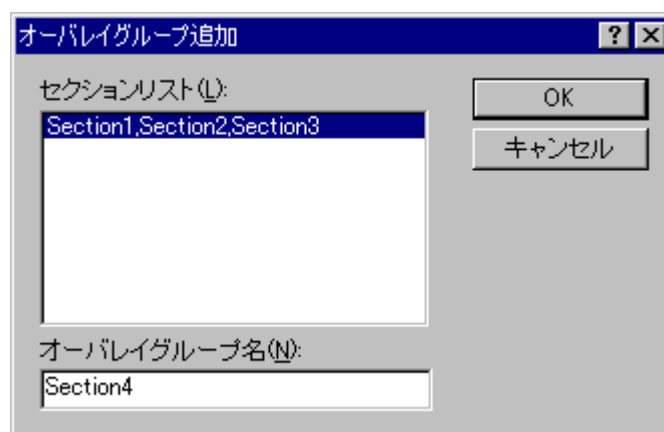
左側のペインのセクション設定ツリーで、セクショングループ、サブセクショングループ、またはオーバーレイグループ下にセクションが追加されます。

### 13.1.5 オーバレイグループを追加する

編集モードでは、セクショングループ、サブセクショングループ、またはセクション (\*1) 下にオーバーレイ (\*2) グループの追加が可能です。

#### オーバーレイグループを追加するには

1. 左側のペイン内のセクショングループ、サブセクショングループ、またはセクション (\*1) 上で右クリックしポップアップメニューを表示します。
2. [オーバーレイグループの追加]を選択します。[オーバーレイグループ追加]ダイアログボックスが開きます。



3. [オーバーレイグループ名]にオーバーレイグループ名を入力します。

左側のペインのセクション設定ツリーで、セクショングループ、サブセクショングループ、またはセクション (\*1) 下にオーバーレイグループが追加されます。

**注：**

\*1. 以下に示すコンパイラパッケージに含まれるツールチェインを使用している場合にサポートしています。

- SuperH ファミリ用 C/C++コンパイラパッケージ V.9.00 Release 04 以降
- H8SX, H8S, H8 ファミリ用 C/C++コンパイラパッケージ V.6.01 Release 02 以降
- RX ファミリ用 C/C++コンパイラパッケージ V.1.00 Release 00 以降

\*2. リンケージエディタがオーバーレイをサポートしていない場合は使用できません。

### 13.1.6 未登録セクションを自動登録する

編集モードでは、右側のペインに表示しているセクションリストの中で、未登録セクションを左側のペインのセクションツリーへ自動登録できます。

#### 未登録セクションを自動登録するには

1. 左側のペイン内で右クリックしポップアップメニューを表示します。
  2. [セクションの自動登録]を選択します。
- 未登録セクションの中で一番小さいアドレスより小さいアドレスのセクショングループがセクションツリーに存在する場合、そのセクショングループの最後にすべての未登録セクションをアドレス指定なしで追加します。
  - 未登録セクションの中で一番小さいアドレスより小さいアドレスのセクショングループがセクションツリーに存在しない場合、未登録セクションの中で一番小さなアドレスにすべての未登録セクションをセクショングループとして追加します。

**注：**

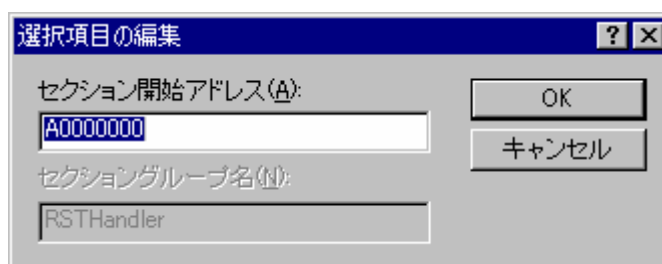
未登録セクションがオーバーレイセクションだった場合は、オーバーレイセクションとしては登録せず、通常のセクションとして登録します。

### 13.1.7 選択項目を編集する

編集モードでは、アドレス値、およびセクショングループ、サブセクショングループ、オーバーレイグループ、セクションの名前の編集が可能です。

#### 選択項目を編集するには

1. 以下のいずれかの操作を選択し[選択項目の編集]ダイアログボックスを開きます。
  - 左側のペイン内の項目上で右クリックしポップアップメニューを表示する。[選択項目の編集]を選択する。
  - 左側のペイン内の項目をダブルクリックする。
2. 編集フィールドにデータを入力してください。



### 13.1.8 プライマリセクションに設定する

編集モードでは、選択したオーバーレイグループをプライマリセクションに設定(\*)できます。

#### プライマリセクションに設定するには

1. 左側のペイン内のオーバーレイグループ上で右クリックしポップアップメニューを表示します。
2. [プライマリセクションに設定]を選択します。

左側のペインのセクション設定ツリーで、選択したオーバーレイグループがプライマリセクションに設定されます。

#### 注:

\*. リンケージエディタがオーバーレイをサポートしていない場合は使用できません。

### 13.1.9 メモリマップを設定する

デバッガと接続している場合にメモリマップを設定できます。

#### メモリマップを設定するには

1. 左側のペイン内で右クリックしポップアップメニューを表示します。
2. [メモリマップ設定]を選択します。メモリマップを設定するダイアログボックスが開きます。
3. アドレス空間の現在のマッピング状況がリストで表示されているため、必要に応じてメモリマップおよびメモリリソースを設定してください。
4. [OK]ボタンをクリックします。

**注：**

以下に示すデバッガがサポートしています。

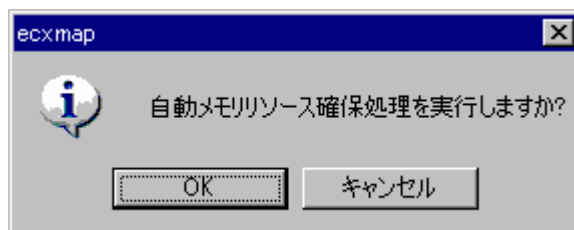
- SuperH ファミリ用シミュレータデバッガ
- H8SX, H8S, H8 ファミリ用シミュレータデバッガ
- RX ファミリ用シミュレータデバッガ

### 13.1.10 メモリリソースを自動的に確保する

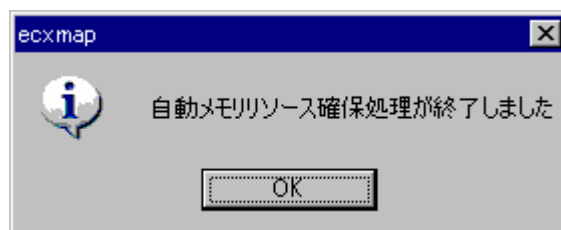
リンケージエディタが出力するリンケージリストファイル (.map) がある場合、メモリマップおよびリンケージマップ情報に基づきメモリリソースを自動的に確保できます。デバッガと接続している場合に使用できます。

**メモリリソースを自動的に確保するには**

1. 左側のペイン内で右クリックしポップアップメニューを表示します。
2. [メモリリソース自動確保]を選択します。下記のダイアログボックスが開きます。



3. 処理を継続する場合は[OK]をクリックしてください。
4. 処理が完了すると下記のダイアログボックスが開きます。

**注：**

以下に示すデバッガがサポートしています。

- SuperH ファミリ用シミュレータデバッガ
- H8SX, H8S, H8 ファミリ用シミュレータデバッガ
- RX ファミリ用シミュレータデバッガ

### 13.1.11 セクション設定ツリーを印刷する

左側のペインのセクション設定ツリーを印刷できます。

**セクション設定ツリーを印刷するには**

1. 左側のペイン内で右クリックしポップアップメニューを表示します。

2. [印刷]を選択します。
3. 標準のプリントフォーマットを選択するダイアログボックスが開きます。ここで、使用するプリンタとページ設定のオプションを選択できます。[印刷]ボタンをクリックしてください。

### 13.1.12 未割り当て領域を表示する

セクションが割り当てられていない領域の表示が可能です。デフォルトでは表示されません。

#### 未割り当て領域を表示するには

1. 右側のペイン内で右クリックしポップアップメニューを表示します。
2. [未割り当て領域の表示]を選択します。[未割り当て領域の表示]にチェックマークが付きます。これにより右側のペインで、セクションが割り当てられていない領域がすべて表示されます。Name カラムに"Unallocated Area"と表示しています。

### 13.1.13 サイズ0のセクションを表示する

サイズ0のセクションの表示が可能です。デフォルトでは表示されません。

#### サイズ0のセクションを表示するには

1. 右側のペイン内で右クリックしポップアップメニューを表示します。
2. [サイズ0のセクション表示]を選択します。右側のペインで、サイズ0のセクションがすべて表示されます。このとき"End Address"カラムに終了アドレスは表示されません。

### 13.1.14 アドレスに対応するソースコードを表示する

選択している行の該当アドレスに対応したソースファイルをエディタウィンドウで開きます。

#### アドレスに対応するソースコードを表示するには

以下のいずれかの操作を選択してください。

- 右側のペイン内のソースを表示したいセクション情報行の上で右クリックし、ポップアップメニューを表示します。[ソースファイル表示]を選択します。
- 右側のペイン内のソースを表示したいセクション情報行をダブルクリックします。

### 13.1.15 セクションリストを印刷する

右側のペインのセクションリストを印刷できます。

#### セクションリストを印刷するには


1. 右側のペイン内で右クリックしポップアップメニューを表示します。
2. [印刷]を選択します。
3. 標準のプリントフォーマットを選択するダイアログボックスが開きます。ここで、使用するプリンタとページ設定のオプションを選択できます。[印刷]ボタンをクリックしてください。

## 13.2 シンボル情報を見る

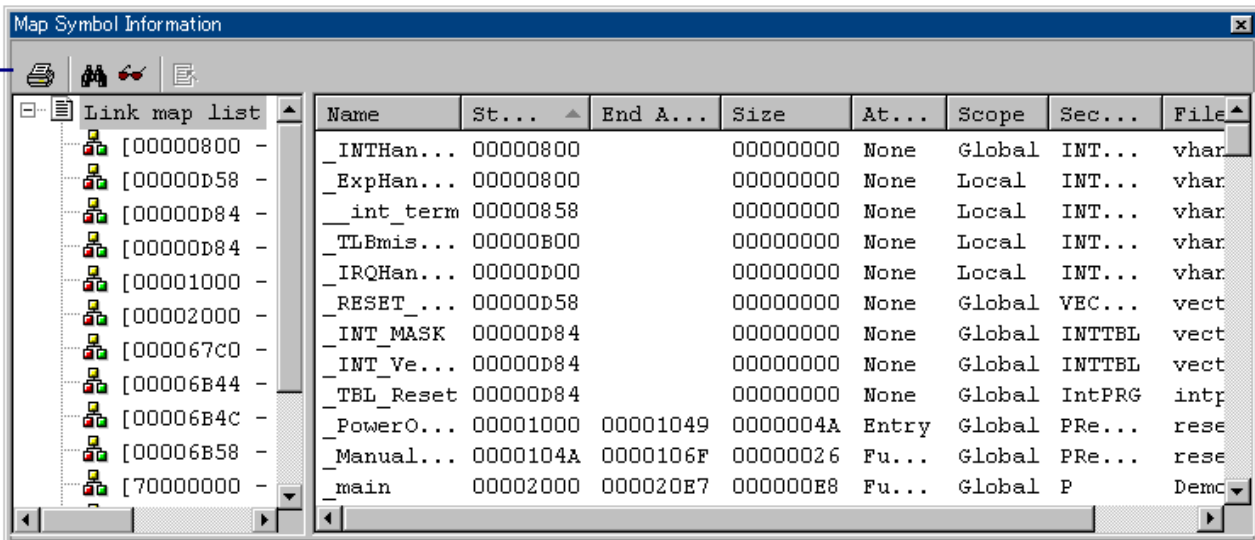
Map Symbol Information ウィンドウを使用すると、セクションごとにシンボルを表示できます。シンボル数が多い場合も、検索機能、フィルタ機能で必要な情報だけを表示できます。

### 13.2.1 Map Symbol Information ウィンドウを開く

#### Map Symbol Information ウィンドウを開くには

- 以下のいずれかの操作を選択し[マップ種別の選択]ダイアログボックスを開きます。
  - [表示 -> マップ]を選択する。
  - [マップ]ツールバーボタン(  )をクリックする。
- [マップ種別]ドロップダウンリストから"Map Symbol Information"を選択します。
- [OK]ボタンをクリックします。

#### ウィンドウの構成



The screenshot shows the 'Map Symbol Information' window. On the left is a 'Link map list' tree view. On the right is a table of symbol information. Below the window, labels identify the components: 'ツールバー' (Toolbar) points to the icons at the top; 'リンケージエディタのマップリスト' (Linker Editor Map List) points to the left pane; 'リンケージエディタのシンボル情報' (Linker Editor Symbol Information) points to the right pane. The table columns are labeled: 'シンボル名' (Symbol Name), '開始アドレス' (Start Address), '終了アドレス' (End Address), 'シンボルサイズ' (Symbol Size), 'シンボル属性' (Symbol Property), 'シンボル存在セクション' (Symbol Existence Section), and 'シンボル存在ファイル' (Symbol Existence File).

Name	St...	End A...	Size	At...	Scope	Sec...	File
_INTHan...	00000800		00000000	None	Global	INT...	vhar
_ExpHan...	00000800		00000000	None	Local	INT...	vhar
_int_term	00000858		00000000	None	Local	INT...	vhar
_TLBmis...	00000B00		00000000	None	Local	INT...	vhar
_IRQHan...	00000D00		00000000	None	Local	INT...	vhar
_RESET_...	00000D58		00000000	None	Global	VEC...	vect
_INT_MASK	00000D84		00000000	None	Global	INTTBL	vect
_INT_Ve...	00000D84		00000000	None	Global	INTTBL	vect
_TBL_Reset	00000D84		00000000	None	Global	IntPRG	intp
_PowerO...	00001000	00001049	0000004A	Entry	Global	PRE...	rese
_Manual...	0000104A	0000106F	00000026	Fu...	Global	PRE...	rese
_main	00002000	000020E7	000000E8	Fu...	Global	P	Demc

- リンケージエディタのマップリストを左側のペインに表示します。
  - リンケージエディタが出力するリンケージリストファイル(.map)がある場合に表示されます。
  - マップリストのルート(Linker map list)を選択すると、すべてのシンボル情報を右側のペインに表示します。
  - マップリストのセクションを選択すると、選択したセクションに属するシンボル情報を右側のペインに表示します。
- リンケージエディタのシンボル情報を右側のペインに表示します。
  - リンケージエディタが出力するリンケージリストファイル(.map)にシンボルリストがある場合に表示します。


- カラムヘッダ上のマウスのドラッグで各カラムの幅を調節できます。また、カラムヘッダ上で右クリックするとポップアップメニューを表示します。カラムが利用できる場合は、エントリの横にはチェックマークがあります。エントリをクリックすると、各カラムの表示、非表示を切り替えます。
- リストのカラムヘッダをクリックすると、リスト項目はそのカラムによってソートされます。ウィンドウ開くと、[Start Address]の昇順でソートされています。
- [Attribute]カラムにはシンボル属性を表示します。
  - Entry : エントリ関数
  - Function : 関数名
  - Data : 変数名
  - None : 未設定(ラベル、アセンブラシンボル)
- [Scope]カラムにはスコープ属性を表示します。
  - Global : グローバルシンボル
  - Local : ローカルシンボル

## ウィンドウのオプション

### • リンケージエディタのマップリスト (左側のペイン)

左側のペイン内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
印刷		リンケージエディタのマップリストを印刷します。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。

### • リンケージエディタのシンボル情報 (右側のペイン)

右側のペイン内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
検索		シンボルを検索します。
次を検索		次のシンボルを検索します。
フィルタ		シンボル情報をフィルタ表示します。
ソースファイル表示		選択行のアドレスに対応したソースコードを表示します。
印刷		リンケージエディタのシンボル情報を印刷します。

## 13.2.2 マップリストを印刷する

左側のペインのリンケージエディタのマップリストを印刷できます。



### マップリストを印刷するには

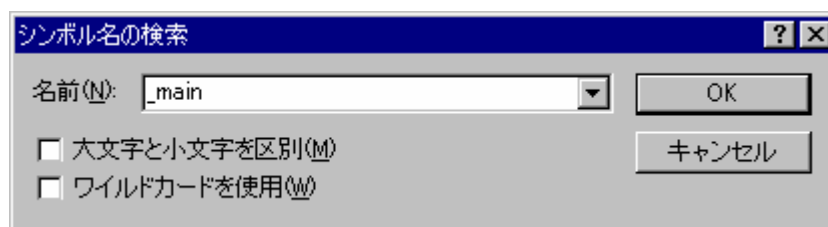
1. 左側のペイン内で右クリックしポップアップメニューを表示します。
2. [印刷]を選択します。
3. 標準のプリントフォーマットを選択するダイアログボックスが開きます。ここで、使用するプリンタとページ設定のオプションを選択できます。[印刷]ボタンをクリックしてください。

### 13.2.3 シンボルを検索する

リンケージエディタのシンボルの検索が可能です。

#### シンボルを検索するには

1. 右側のペイン内で右クリックしポップアップメニューを表示します。
2. [検索]を選択します。[シンボル名の検索]ダイアログボックスが開きます。



3. [名前]に検索するシンボルを入力、またはドロップダウンリストボックスから以前検索したシンボルを選択してください。
4. 大文字と小文字を識別する場合は[大文字と小文字を区別]チェックボックスをオンにしてください。
5. ワイルドカード ("?"または"\*)")を使用する場合は[ワイルドカードを使用]チェックボックスをオンにしてください。

? : 任意の1文字

\* : 任意の文字列

6. [OK]ボタンをクリックします。

右側のペインのリンケージエディタのシンボル情報で該当行が強調表示されます。

検索文字列の次の該当箇所を検索する場合は、[次を検索]を選択してください。

### 13.2.4 シンボル情報をフィルタ表示する

リンケージエディタのシンボル情報のフィルタ表示が可能です。

#### シンボル情報をフィルタ表示するには

1. 右側のペイン内で右クリックしポップアップメニューを表示します。
2. [フィルタ]を選択します。[シンボルのフィルタ設定]ダイアログボックスが開きます。



3. [フィルタを有効にする]チェックボックスをオンにします。チェックボックスのデフォルトはオフです。
4. [フィルタ対象カラム]でフィルタリングするカラムを選択します。
5. [フィルタ項目]でカラムに表示する項目を選択します。

フィルタ対象カラム	フィルタ項目	機能
Attribute (シンボル属性)	Function	関数でフィルタリングを行う
	Data	データシンボルでフィルタリングを行う
	Entry	実行開始位置でフィルタリングを行う
	None	上記以外(ラベル等)のシンボルでフィルタリングを行う
Scope (スコープ属性)	Global	グローバルシンボルでフィルタリングを行う
	Local	ローカルシンボルでフィルタリングを行う
Section (シンボル存在セクション)	セクション名	セクション名でフィルタリングを行う
File (シンボル存在ファイル名)	ファイル名	ファイル名でフィルタリングを行う

6. [すべて設定]ボタンをクリックするとフィルタ項目のすべてのチェックボックスがオンになります。
7. [すべて解除]ボタンをクリックするとフィルタ項目のすべてのチェックボックスがオフになります。
8. [OK]ボタンをクリックします。

右側のペインのリンケージエディタのシンボル情報が、選択した項目でフィルタ表示されます。

複数条件でのフィルタ表示を行う場合は、さらに別のフィルタ対象カラムを選択し、そのカラム内のフィルタ項目を選択してください。

フィルタを無効にするには、[シンボルのフィルタ設定]ダイアログボックスで[フィルタを有効にする]チェックボックスをオフにしてください。

### 13.2.5 アドレスに対応するソースコードを表示する

選択している行の該当アドレスに対応したソースファイルをエディタウィンドウで開きます。

### アドレスに対応するソースコードを表示するには

以下のいずれかの操作を選択します。

- 右側のペイン内のソースを表示したいシンボル情報行の上で右クリックし、ポップアップメニューを表示します。[ソースファイル表示]を選択します。
- 右側のペイン内のソースを表示したいシンボル情報行をダブルクリックします。

### 13.2.6 シンボル情報を印刷する

右側のペインのリンクージエディタのシンボル情報を印刷できます。

#### シンボル情報を印刷するには

1. 右側のペイン内で右クリックしポップアップメニューを表示します。
2. [印刷]を選択します。
3. 標準のプリントフォーマットを選択するダイアログボックスが開きます。ここで、使用するプリンタとページ設定のオプションを選択できます。[印刷]ボタンをクリックしてください。

## 14. コマンドライン


High-performance Embedded Workshop コマンドラインインタプリタにより、ウィンドウメニュー/コマンドの代わりにテキストベースのコマンドで、ターゲットプラットフォームを制御できます。これはあらかじめ決まったコマンド列をバッチファイルから呼び出して、ターゲットプラットフォームに送る場合に特に便利です。さらに出力をログファイルに記録することもできます。

### 注：

コマンドラインでファイルを指定する場合、プレースホルダを使用してください（TCL コマンドは除く）。プレースホルダに含まれていないディレクトリを指定したいときは、絶対パスを指定してください。絶対パスで指定すると、他のマシンやパスの内容が異なる環境に移動する場合、正しくファルを参照できなくなりますので、ファイル指定をやり直してください。

(例) FILE\_LOAD ELF/DWARF2 \$(CONFIGDIR)¥¥demo.abs

### 14.1 コマンドラインウィンドウを開く

コマンドラインウィンドウを開くには、[表示 -> コマンドライン]を選択するか、[コマンドライン]ツールボタンをクリックします。ウィンドウタイトルとしてバッチファイル名とログファイル名をコロンで区切って表示します。

High-performance Embedded Workshop コマンドおよび TCL コマンドを実行できます。使用できるコマンドについては、「リファレンス 3. コマンド」を参照してください。

コマンドラインウィンドウには、2つのタブ（[Console]および[Debugger]）があります。

### ウィンドウの構成

#### • [Console]タブ

テキストベースのコマンドを入力し、ターゲットプラットフォームを制御できるウィンドウです。コマンドの実行結果は、[Debugger]タブの下側ペインの表示内容と連動しています。

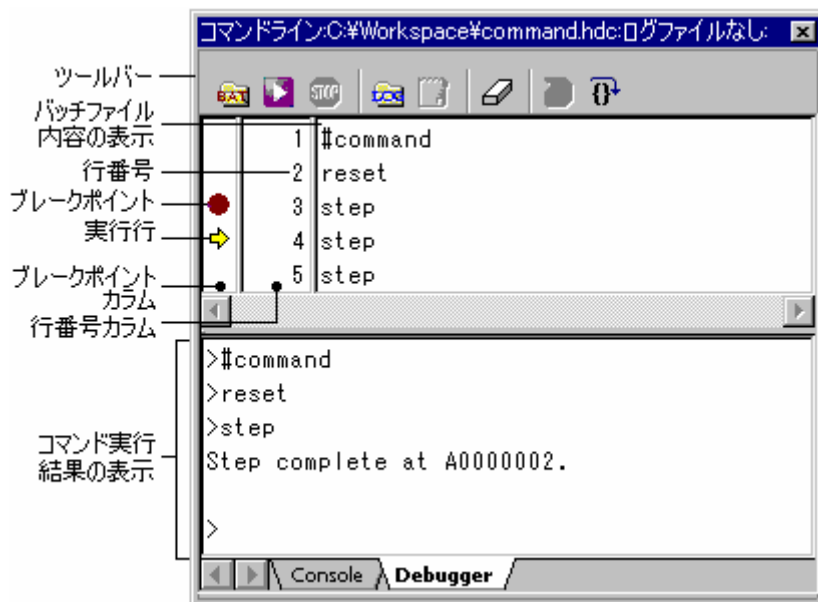


- コマンドラインウィンドウ最終行のコマンドプロンプト(">")に続けて入力後、"Enter"キーを押すとコマンドを実行します。
- 最終行で"Ctrl+↑"または"Ctrl+↓"キーを押すと直前に入力したコマンド行を表示できます。
- バッチファイルからコマンドを呼び出して実行、および実行結果をファイルに出力できます。設定したバッチファイルの内容は、[Debugger]タブの上側ペインに表示します。
- ウィンドウの表示内容をクリアできます。[Debugger]タブの下側ペインの表示内容もクリアします。

### • [Debugger]タブ

設定したバッチファイルの内容を1行単位で実行、またはブレークポイントを設定して実行し、ターゲットプラットフォームを制御できるウィンドウです。

バッチファイルを読み込んでいる場合、ペインの境界上にマウスカーソルを置くと上下の矢印カーソルに変わります。クリックし希望する位置までマウスをドラッグするとペインの幅を調節できます。



#### ○ 上側のペイン (バッチファイル内容の表示)

設定したバッチファイルの内容を表示し、バッチファイルをデバッグできます。このウィンドウ上で表示内容を編集できません。

- 設定したバッチファイルの内容を表示します。
- バッチファイルを1行単位で実行できます。
- バッチファイルの先頭からバッチファイルの実行を開始できます。
- コマンド行に対応するブレークポイントカラムでブレークポイント(●)を設定できます。
- ブレークポイントカラム上の黄色の矢印(➡)は、バッチファイルの実行行 (開始位置) を指します。
- バッチファイルの先頭に実行開始位置 (➡) を戻すことができます。
- コマンド実行結果をファイルに出力できます。

#### ○ 下側のペイン (コマンド実行結果の表示)

コマンドの実行結果を表示します。コマンドの実行結果は、[Console]タブの表示内容と連動しています。このウィンドウ上でコマンドを入力できません。

- ウィンドウの表示内容をクリアできます。[Console]タブの表示内容もクリアします。

### ウィンドウのオプション

#### • [Console]タブ

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	マクロの記録	機能
バッチファイル指定		-	バッチファイルを設定します。
バッチファイル実行		-	バッチファイルの先頭からバッチファイルの実行を開始します。
バッチファイル停止		-	バッチファイルの実行を停止します。
ログファイル指定		-	ログファイルを設定します。
ロギング開始/停止		-	ログファイルへの出力を開始/停止します。
参照		-	ファイルの絶対パスを入力します。
プレースホルダ	Configuration Directory	-	プレースホルダ\$(CONFIGDIR)を挿入します。
コンフィグレーションディレクトリ		-	プレースホルダ\$(CONFIGNAME)を挿入します。
コンフィグレーション名		-	プレースホルダ\$(PROJDIR)を挿入します。
プロジェクトディレクトリ		-	プレースホルダ\$(PROJECTNAME)を挿入します。
プロジェクト名		-	プレースホルダ\$(WORKSPDIR)を挿入します。
ワークスペースディレクトリ		-	プレースホルダ\$(WORKSPNAME)を挿入します。
ワークスペース名		-	プレースホルダ\$(HEWDIR)を挿入します。
HEW インストールディレクトリ		-	
すべて選択		-	アクティブウィンドウのすべての内容を選択します。
コピー		-	選択部分をコピーし Windows® クリップボードに貼り付けます。
カット		-	選択部分を切り取り Windows® クリップボードに貼り付けます。
貼り付け		-	Windows® クリップボードの内容をコピーしてアクティブウィンドウのカーソル位置に貼り付けます。
クリア		-	ウィンドウの表示内容をクリアします。
元に戻す		-	直前の入力を元に戻します。
-		-	開き括弧（{または[）が入力され閉じ括弧（}または]）が入力されていない場合ボタンが点灯します（ボタン選択不可）。
ツールバー表示	-	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	-	ツールバーボタンをカスタマイズします。

## • [Debugger]タブ



### ○ 上側のペイン（バッチファイル内容の表示）

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	マクロの記録	機能
バッチファイル指定		-	バッチファイルを設定します。
バッチファイル実行		-	バッチファイルの先頭からバッチファイルの実行を開始します。
バッチファイル停止		-	バッチファイルの実行を停止します。
リセット	-	-	バッチファイルの実行状態をリセットします。
ステップ		-	バッチファイルを1行単位で実行します。
ブレークポイントの挿入/削除	-	-	ブレークポイントを挿入/削除します。


ポップアップメニューオプション	ツールバーボタン	マクロの記録	機能
ブレークポイントの有効化/無効化	-	-	現在設定しているブレークポイントの有効/無効を切り替えます。
ブレークポイントをすべて削除	-	-	すべてのブレークポイントを削除します。
ログファイル指定		-	ログファイルを設定します。
ロギング開始/停止		-	ログファイルへの出力を開始/停止します。
ツールバー表示	-	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	-	ツールバーボタンをカスタマイズします。

○ 下側のペイン（コマンド実行結果の表示）

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	マクロの記録	機能
クリア		-	ウィンドウの表示内容をクリアします。

## 14.2 バッチファイルを設定する

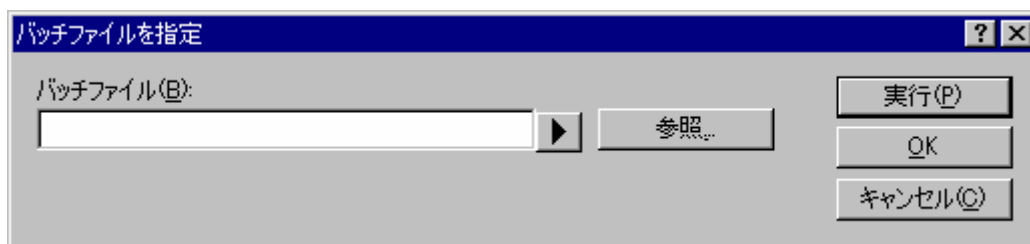
自動テストなどのあらかじめ定義した一連のコマンド行を実行する場合は、バッチファイルを利用すると便利です。

バッチファイルはテキストエディタで作成し、実行するコマンドラインを記述しておきます。バッチファイルの拡張子のデフォルトは".hdc"です。

バッチファイルを設定するには、[バッチファイルを指定]ダイアログボックスを使用します。

コマンドラインウィンドウのポップアップメニューから[バッチファイル指定]を選択すると、[バッチファイル指定]ダイアログボックスを開きます。バッチファイル名(\*.hdc)を入力できます。

[実行]ボタンをクリックすると、設定したバッチファイルを実行します。[OK]ボタンをクリックすると、設定したバッチファイル名を、ウィンドウタイトルに表示します。[キャンセル]ボタンをクリックすると、設定を変更しないでダイアログボックスを閉じます。



バッチファイルを設定すると、コマンドラインウィンドウの[Debugger]タブの上側のタブにバッチファイルの内容を表示します。

セッションの保存時に設定したバッチファイル名も保存します。

注：

- 以下の GUI 機能およびコマンドを実行するとバッチファイルを閉じるため、コマンドラインウィンドウの[Debugger]タブの上側のタブの表示はクリアされます。
  - ワークスペースを閉じる（CLOSE\_WORKSPACE コマンド）



- 新規ワークスペースを作成する
- プロジェクトを挿入する
- セッションを追加する
- バッチファイル設定後にバッチファイルの内容を変更すると、コマンドラインウィンドウの[Debugger]タブの上側のタブに表示しているバッチファイルの内容を更新します。そのため、バッチファイルの実行状態をリセットし、設定しているブレークポイントをすべて削除します。

バッチファイル設定後にバッチファイルの内容を変更すると、再びバッチファイルを読み込むことを通知するメッセージを表示します。

確認要求ダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。

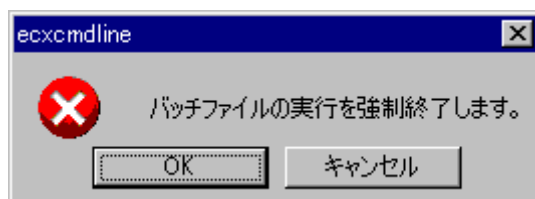
#### 確認要求ダイアログボックスを再び表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスが開きます。
2. [確認]タブを選択してください。
3. [バッチファイルの更新]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. OK をクリックしてください。

### 14.3 バッチファイルを実行する

バッチファイルを実行するには、[バッチファイルを指定]ダイアログボックスで[実行]ボタンをクリックするか、コマンドラインウィンドウのポップアップメニューから[バッチファイル実行]を選択します。[バッチファイル実行]メニューはバッチファイル実行中にグレー表示となり、バッチファイル実行が停止してユーザに制御が戻ったときに有効表示になります。

バッチファイルを実行中にコマンドラインウィンドウを閉じようとする、下記のメッセージボックスを表示します。



バッチファイルの実行を強制終了し、コマンドラインウィンドウを閉じる場合は、[OK]ボタンをクリックしてください。

バッチファイルの実行を継続する場合は、[キャンセル]ボタンをクリックしてください。この時コマンドラインウィンドウは閉じません。

### 14.4 コマンド実行を中断する

コマンド実行を中断する場合は、コマンドラインウィンドウのポップアップメニューから[バッチファイル停止]を選択します。

[バッチファイル停止]メニューはコマンド実行中に有効表示になります。

#### 注：

TCL コマンドの場合、TCL コマンドが完了しないとコマンド実行が中断されないことがあります。

### 14.5 ログファイルを設定する

コマンド実行結果を保存するログファイルは、[ログファイルを開く]ダイアログボックスで設定します。

[ログファイルを開く]ダイアログボックスを開くには、コマンドラインウィンドウのポップアップメニューから[ログファイル指定]を選択します。



### 14.6 ログファイルへの出力を開始/停止する

ファイルへのロギング処理を実行するか、停止するか切り替えるは、コマンドラインウィンドウのポップアップメニューの[ロギング開始/停止]で行います。ログファイルの内容は、ロギングが終了するか、チェックボックスをクリアしてロギングを一時的に停止しなければ表示できないことにご注意ください。ロギングを再び開始すると、ログファイルに追加します。

### 14.7 ファイルの絶対パスを入力する

コマンドラインウィンドウのポップアップメニューで[参照]を選択すると、参照ダイアログボックスを開きます。ここでファイルを選択し[開く]をクリックすると、絶対パスファイル名をカーソル位置に貼り付けます。カーソルが最終行にある場合のみ使用できます。

### 14.8 プレースホルダを挿入する

コマンドラインウィンドウのポップアップメニュー[プレースホルダ]のサブメニューで選択したプレースホルダをカーソル位置に貼り付けます。

カーソルが最終行にある場合のみ使用できます。

[プレースホルダ]のサブメニュー	カーソル位置に貼り付けるプレースホルダ
コンフィグレーションディレクトリ	\$(CONFIGDIR)
コンフィグレーション名	\$(CONFIGNAME)
プロジェクトディレクトリ	\$(PROJDIR)
プロジェクト名	\$(PROJECTNAME)
ワークスペースディレクトリ	\$(WORKSPDIR)
ワークスペース名	\$(WORKSPNAME)
HEW インストールディレクトリ	\$(HEWDIR)

### 14.9 ウィンドウの表示内容をすべて選択する

コマンドラインウィンドウのポップアップメニューから[すべて選択]を選択すると、コマンドラインウィンドウの出力をすべて選択します。

### 14.10 選択部分をクリップボードにコピーする

コマンドラインウィンドウのポップアップメニューから[コピー]を選択すると、選択したテキストブロックをWindows® クリップボードにコピーします。テキストブロックを選択している場合のみ使用できます。

### 14.11 選択部分をクリップボードに切り取る

コマンドラインウィンドウのポップアップメニューから[切り取り]を選択すると、選択したテキストブロックを削除し Windows® クリップボードに貼り付けます。現在入力中のカーソル位置のテキストブロックを選択している場合にのみ使用できます。

### 14.12 クリップボードの内容を貼り付ける

コマンドラインウィンドウのポップアップメニューから[貼り付け]を選択すると、Windows® クリップボードの内容を現在入力中のカーソル位置に貼り付けます。カーソルが最終行にある場合のみ使用できます。



### 14.13 ウィンドウの表示内容をクリアする













コマンドラインウィンドウのポップアップメニューから[クリア]を選択すると、[Console]タブ、および [Debugger]タブの下側のペインに表示している内容をすべてクリアします。

### 14.14 直前の操作を元に戻す

コマンドラインウィンドウのポップアップメニューから[元に戻す]を選択すると、現在のカーソル位置で、直前の操作を元に戻します。

### 14.15 括弧の呼応状態を確認する

コマンドラインウィンドウで[開き括弧インジケータ]ツールバーボタン()が表示されている場合、コマンドラインウィンドウに入力中の式に含まれる括弧の呼応状態を確認できます。[開き括弧インジケータ]ツールバーボタンの点灯状態により括弧の呼応状態を確認します。ボタンは呼応状態の表示のみで操作はできません。括弧の開始である "[" または "{" が入力されネスト部分が入力されると、括弧の終了である "]" または "}" が入力されるまでは、[開き括弧インジケータ]ツールバーボタンが点灯()します。

コマンドラインウィンドウへの TCL 関数の入力例	括弧の入力状況	ツールバーボタンの点灯状態
>set bit 1	なし	
1	なし	
>set value 1	なし	
1	なし	
>set r 1	なし	
1	なし	
>if {\$bit & \$value} {	開き括弧が入力された	
> set r 0\$r	なし	
>} else {	閉じ括弧および開き括弧が入力された	
> set r 1\$r	なし	
>}	閉じ括弧が入力された	
>01	なし	

### 14.16 バッチファイルの実行状態をリセットする

コマンドラインウィンドウの [Debugger] タブでバッチファイルの実行状態をリセットします。バッチファイルを設定していない場合は使用できません。

以下をリセットします。

- バッチファイルの先頭に実行開始位置を戻します。
- コマンドの対話モードの場合、対話モードを終了します。
- 括弧を開いている場合、括弧を閉じます。

### バッチファイルの実行状態をリセットするには

1. コマンドラインウィンドウの[Debugger]タブを開きます。
2. 上側のペインで、右クリックしてポップアップメニューを表示します。
3. [リセット]を選択します。

#### 注：


- 設定したバッチファイル名、バッチファイルのブレークポイントなどの情報は各セッションに保存されています。コマンドラインウィンドウの[Debugger]タブを開き、セッションを切り替える操作を行うと、そのセッションで設定したバッチファイルの表示内容に切り替わります。バッチファイルの実行状態も自動的にリセットされます。バッチファイルにこれらのコマンドが含まれている場合も同様です。
  - アクティブプロジェクトを切り替える（CHANGE\_PROJECT コマンド）
  - デバッグセッションを切り替える（CHANGE\_SESSION コマンド）
- コンフィグレーションを切り替えると、バッチファイルの実行状態も自動的にリセットされます。バッチファイルにこのコマンドが含まれている場合も同様です。
  - ビルドのコンフィグレーションを切り替える（CHANGE\_CONFIGURATION コマンド）


### 14.17 バッチファイルを 1 行単位で実行する

コマンドラインウィンドウの[Debugger]タブで、バッチファイルの開始位置からバッチファイルのコマンド行を 1 行単位で実行します。バッチファイルを設定していない場合は使用できません。



### バッチファイルを 1 行単位で実行するには

1. コマンドラインウィンドウの[Debugger]タブを開きます。
2. 上側のペインで、以下の操作のいずれかを選択します。
  - ウィンドウのツールバーの[ステップ]ツールバーボタン()をクリックする。
  - ウィンドウ内を右クリックしてポップアップメニューを表示し、[ステップ]を選択する。
  - ウィンドウ内にフォーカスがある場合に、"Ctrl+F10"キーを押す。

バッチファイルの開始位置()は次のコマンド行へ移動します。コマンド実行で発生したエラーによっては、バッチファイルの開始位置は次のコマンド行へ移動しません。

#### 注：


以下のコマンドがバッチファイルに含まれている場合、コマンドラインウィンドウの[Debugger]タブで設定したバッチファイルのデバッグを継続できません。コマンドラインウィンドウの[Console]タブでバッチファイルを実行してください。


- CLOSE\_WORKSPACE コマンド  
コマンドを実行すると、バッチファイルを閉じます。
- CHANGE\_PROJECT コマンド、および CHANGE\_SESSION コマンド  
コマンドを実行すると、切り替えたセッションで設定したバッチファイルの表示内容に切り替わります。
- CHANGE\_CONFIGURATION コマンド  
コマンドを実行すると、バッチファイルの実行状態も自動的にリセットされます。

### 14.18 バッチファイルにブレークポイントを設定する

コマンドラインウィンドウの[Debugger]タブで、バッチファイルのコマンド行にブレークポイントを設定します。バッチファイルを設定していない場合は使用できません。

#### バッチファイルにブレークポイントを設定するには

1. コマンドラインウィンドウの[Debugger]タブを開きます。
2. 上側のペインで、以下の操作のいずれかを選択します。
  - バッチファイルを停止したい行のブレークポイントカラムをダブルクリックします。
  - バッチファイルを停止したい行にカーソルを置きます。右クリックしてポップアップメニューを表示し、[ブレークポイントの挿入/削除]を選択します。  
ブレークポイントカラム上にブレークポイントマーク()を表示します。これは、ブレークポイントを設定したことを示します。
3. 設定したブレークポイントの有効、無効を切り替えるには、右クリックしてポップアップメニューを表示し、[ブレークポイントの有効化/無効化]を選択します。
4. 設定したブレークポイントを削除するには、右クリックしてポップアップメニューを表示し[ブレークポイントの挿入/削除]を選択するか、カラムをダブルクリックします。
5. 設定したブレークポイントをすべて削除するには、右クリックしてポップアップメニューを表示し、[ブレークポイントをすべて削除]を選択します。

コマンドラインウィンドウの[Debugger]タブを開いた状態で、バッチファイルを実行します。ブレーク発生時には、ブレークポイントを設定したコマンド行を実行する直前で停止します。ブレークポイントで停止した後に、ステップまたはバッチファイル実行を選択した場合、黄色い矢印アイコン()を表示した行から実行し

ます。ブレークポイントは最大 256 箇所設定できます。セッション保存時に設定したブレークポイントも保存します。

**注：**

- コマンドラインウィンドウの[Console]タブを開いた状態で、バッチファイルを実行した場合はブレークしません。
- バッチファイル設定後にバッチファイルの内容を変更すると、コマンドラインウィンドウの[Debugger]タブの上側のペインに表示しているバッチファイルの内容を更新します。そのため、バッチファイルの実行状態をリセットし、設定しているブレークポイントをすべて削除します。

## 15. マクロ生成支援機能

マクロ生成支援機能は、High-performance Embedded Workshop システムのアプリケーション関連 (\*1)、ビルド関連 (\*2)、およびデバッグ関連 (\*3) などの一部の操作を High-performance Embedded Workshop コマンドラインのコマンドとして記録、または記録したコマンドを実行する機能です。

記録されるファイル (High-performance Embedded Workshop マクロファイル) は、拡張子が hdc のコマンドバッチファイルであり編集可能なテキストファイルです。このファイルは、各ユーザプロファイルのアプリケーションデータフォルダにある High-performance Embedded Workshop 管理情報フォルダ下の "Macros" フォルダに作成されます。デフォルトのファイル名は Default.hdc です。

High-performance Embedded Workshop コマンドラインのコマンドに対応する High-performance Embedded Workshop システムのすべての操作が記録できるわけではありません。各機能のメニューオプション一覧には "マクロの記録" 欄にマクロの記録マーク (●) を表示し、High-performance Embedded Workshop マクロファイルへ記録可能かどうかを示しています。

### 注:

- \*1. プロジェクトの変更、セッションの変更、コンフィグレーションの変更など
- \*2. コンパイル、ビルドなど
- \*3. モジュールのダウンロード、メモリ値の変更、レジスタ値の変更、ソフトウェアブレークの設定 / 解除、プログラム実行など

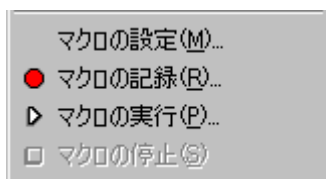
マクロ生成支援機能は [ツール] メニューと [マクロ] ツールバーで利用できます。

[マクロの記録] を選択してから [マクロの停止] を選択するまでが 1 つの "マクロ" で、1 つの High-performance Embedded Workshop マクロファイルには複数のマクロを記録できます。マクロには、複数の High-performance Embedded Workshop コマンドラインのコマンドが含まれます。

### 15.1 マクロメニューとツールバー

マクロ生成支援機能はメニューとツールバーで利用できます。

マクロのメニューは [ツール] メニューにあります。標準で以下のメニューオプションがあります。



[マクロの設定] から [マクロの設定] ダイアログボックスを開くことができます。ここで、使用中のマクロファイルやマクロを管理します。この他のメニューオプションは [マクロ] ツールバー上の項目と同じです。

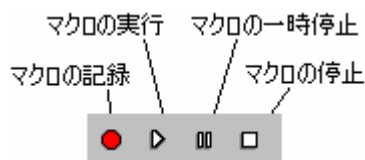
マクロをメニューに割り当てることもできます。これを行うとマクロの標準のメニューオプションの下にメニューオプションが追加されます。以下にその例を示します。





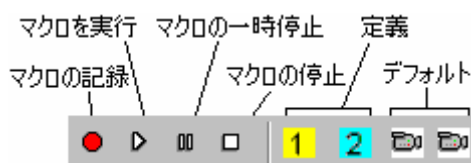
マクロをメニューおよびツールバーに割り当てた場合は、メニューオプションの横にアイコンが表示されません。

[マクロ]ツールバーの標準ボタンを以下に示します。



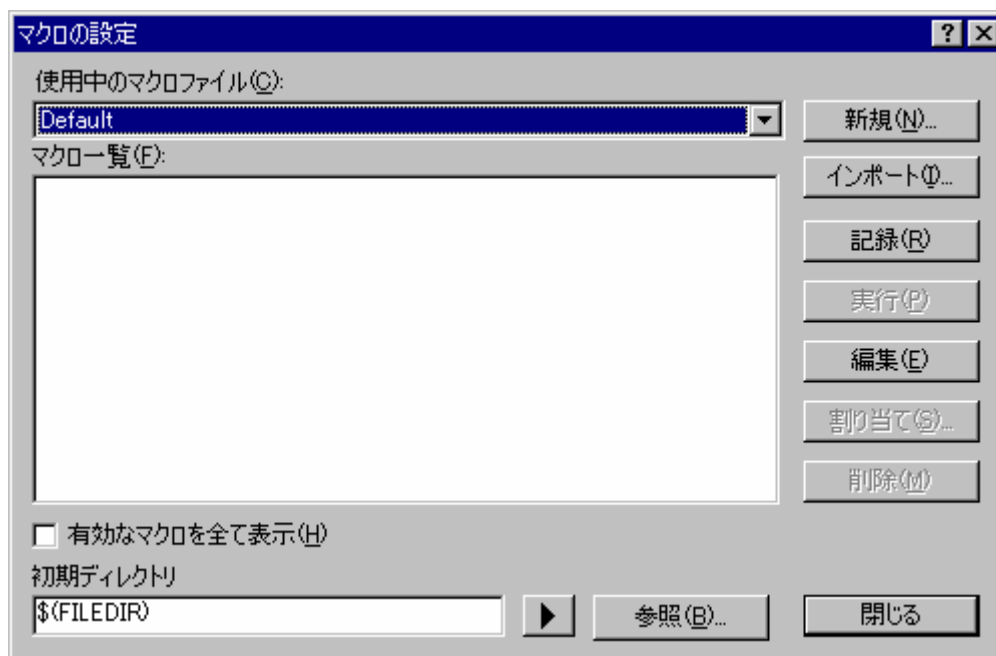
- [マクロの記録]ボタンでマクロの記録を開始します。このボタンをクリックすると、デバッグや High-performance Embedded Workshop の管理操作が記録されます。これは[ツール]メニューの[マクロの記録]メニューと同じです。
- [マクロの実行]ボタンでマクロ機能の再生を開始します。マクロを定義すると[マクロの選択]ダイアログボックスが開きます。再生する特定のマクロを指定してください。
- [マクロの一時停止]ボタンはマクロを記録または実行しているときのみ有効です。このボタンは現在の操作を一時停止して後で再開できるようにします。
- [マクロの停止]ボタンはマクロを記録または実行しているときのみ有効です。このボタンは実行中のマクロ操作を停止します。

マクロのメニューへの割り当てができると同様に、マクロのツールバーへの割り当てもできます。マクロを割り当てたツールバーのアイコンには、定義したアイコンか、デフォルトのアイコンを使用できます。マクロをツールバーに割り当てると、割り当てたマクロが標準のマクロボタンの右側に表示されます。ボタンをクリックすると、割り当てたマクロが起動します。



## 15.2 マクロの設定ダイアログボックスを使用する

[マクロの設定]ダイアログボックスから High-performance Embedded Workshop システムの現在使用中のマクロファイルをアクセスしたり管理したりできます。[マクロの設定]ダイアログボックスを以下に示します。



[使用中のマクロファイル]ドロップダウンリストには High-performance Embedded Workshop システムで現在使用中のマクロファイルをすべて表示します。これらのファイルは、各ユーザプロファイルのアプリケーションデータフォルダにある High-performance Embedded Workshop 管理情報フォルダ下の "Macros" フォルダに格納しています。"Macros" フォルダにコピーしたファイルはすべて自動的に [使用中のマクロファイル] ドロップダウンリストに追加します。

[使用中のマクロファイル]ドロップダウンリストで選択したファイルに含まれるマクロは、[マクロ一覧]リストボックスに表示します。[使用中のマクロファイル]の選択を変更すると自動的にリストを更新します。[有効なマクロを全て表示]チェックボックスをオンにすると、現在定義されているすべてのマクロを表示します。チェックしてこのオプションを指定することにより、[使用中のマクロファイル]リストボックスの値は無視されます。

[初期ディレクトリ]は相対パスを持つマクロファイルを実行するために初期ディレクトリを設定します。ディレクトリを変更するコマンドと同様にディレクトリを変更後、初期ディレクトリからマクロファイルを実行します。

### 新規のマクロファイルを作成するには

1. [ツール -> マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. [新規]ボタンをクリックしてください。[マクロファイルの新規追加]ダイアログボックスが開きます。
3. 新規のマクロファイル名を入力してください。マクロファイル名は 64 文字以内で、半角英数字、半角下線のみ使用してください。
4. OK をクリックすると、[使用中のマクロファイル]リストに新規マクロファイル名を追加します。

### 既存のマクロファイルをインポートするには

1. [ツール->マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. [インポート]ボタンをクリックしてください。[インポートファイルの選択]ダイアログボックスが開きます。
3. 既存のマクロファイルを選択してください。
4. [選択]をクリックしてください。[使用中のマクロファイル]リストに新しくマクロファイル名を追加します。
5. リスト内のマクロファイル名を選択すると、インポートできるマクロを表示します。

マクロファイルはテキストベースの High-performance Embedded Workshop コマンドバッチファイルのため編集可能です。

### 既存のマクロファイルを編集するには

1. [ツール->マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. [使用中のマクロファイル]にあるマクロファイルを選択してください。
3. [マクロ一覧]から編集したいマクロを選択してください。
4. [編集]ボタンをクリックしてください。
5. [マクロの設定]ダイアログボックスが閉じファイルをエディタ内に表示します。

通常のテキストファイルと同様に High-performance Embedded Workshop ウィンドウ上にドラッグアンドドロップしてマクロファイル開くことも可能です。

### 既存のマクロファイルからマクロを削除するには

1. [ツール->マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. [使用中のマクロファイル]にあるマクロファイルを選択してください。
3. [マクロ一覧]から削除したいマクロを選択してください。
4. [削除]ボタンをクリックしてください。

[閉じる]ボタンをクリックすると[マクロの設定]ダイアログボックスを閉じます。このダイアログボックスでの操作はやり直しできません。

## 15.3 既存のマクロファイルをインポートする

既に複数のマクロがある他のマシンから既存の High-performance Embedded Workshop マクロファイルをインポートするには、以下の操作を実行してください。

一つの方法は、[マクロの設定]ダイアログボックスを使用して、マクロファイルをインポートすることです。以下に操作方法を示します。

### 既存のマクロファイルを High-performance Embedded Workshop にインポートするには

1. [ツール->マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. [インポート]ボタンをクリックしてください。標準のファイルブラウザが開き、インポートしたい既存のマクロファイルを選択できます。

3. ファイルを選択し[選択]ボタンをクリックするとファイルをインポートします。インポートしたマクロは、[使用中のマクロファイル]ドロップダウンリストで選択状態になります。また、インポートしたマクロファイルは、各ユーザプロファイルのアプリケーションデータフォルダにある High-performance Embedded Workshop 管理情報フォルダ下の"Macros"フォルダにコピーされます。
4. [マクロの設定]ダイアログボックスで[閉じる]をクリックすると、インポートしたマクロファイルが使用可能になります。

もう一つの方法は、追加するマクロファイルを各ユーザプロファイルのアプリケーションデータフォルダにある High-performance Embedded Workshop 管理情報フォルダ下の"Macros"フォルダにコピーすることです。High-performance Embedded Workshop は、この"Macros"フォルダにあるマクロファイルを自動的に検出して[マクロの設定]ダイアログボックスにその情報を追加します。追加したファイルはデフォルトでは使用できません。デフォルトのマクロファイルを追加したファイルに切り替えるには、[マクロの設定]ダイアログボックスを使用してください。

## 15.4 マクロを記録する

マクロファイルを記録するには下記に示すいくつかの方法があります。

### ツールバーまたはメニューからマクロを記録するには

1. [マクロ]ツールバーの[マクロの記録]ボタン(●)、または[ツール]メニューの[マクロの記録]を選択してください。マウスポインタにマクロの記録アイコンが付き(●)、記録されていることを示します。マクロが記録されているとき、[マクロの一時停止]ボタン(⏸)および[マクロの停止]ボタン(⏹)やメニューが有効になります。記録の一時停止中はマウスポインタに[マクロの一時停止]アイコンが付きます(⏸)。
2. マクロ機能をサポートしないデバッガを使用している場合、"WARNING: ターゲットがマクロ機能をサポートしていないため、記録内容は制限されます。続けますか?"という確認メッセージを表示します。「はい」をクリックすると記録を続けます。確認のダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。確認のダイアログボックスを再び表示するには、[基本設定 -> オプション]を選択し[オプション]ダイアログボックスを開きます。[確認]タブの[マクロ機能をサポートしていないターゲットでの記録]チェックボックスをオンにしてください。デフォルトはチェックボックスがオンです。
3. [マクロの停止]をクリックするまで記録を続行します。マクロファイルに記録する操作については、次節の「15.5 マクロファイルにマクロを記録できる機能」を参照してください。
4. [マクロの停止]をクリックすると[マクロの新規追加]ダイアログボックスが表示されるのでマクロ名を入力してください。
5. OK をクリックするとマクロ名でマクロを保存し、現在のマクロファイルに追加します。キャンセルをクリックするとダイアログボックスを閉じマクロの記録を停止します。

[マクロの設定]ダイアログボックスでも記録できます。

### [マクロの設定]ダイアログボックスからマクロを記録するには

1. [ツール -> マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. [記録]ボタンをクリックしてください。ダイアログボックスが閉じ記録を開始します。

## 15.5 マクロファイルにマクロを記録できる機能

High-performance Embedded Workshop システムのすべての操作がスクリプトとしてマクロファイルに記録できるわけではありません。記録できる操作については、各機能のメニューオプション一覧にも"マクロの記録"として記録機能のサポートの有無を記載していますので参照してください。






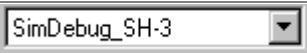
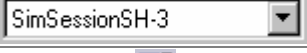






- マクロを記録できる機能(High-performance Embedded Workshop 共通)
- マクロを記録できる機能(デバッガ依存)

### 15.5.1 マクロを記録できる機能 (High-performance Embedded Workshop共通)

連携ツールの製品パッケージにバンドルされている High-performance Embedded Workshop で共通の機能の中で、マクロを記録できる機能について示します。

- メニューオプション、ショートカットキー、およびツールバーボタン**

マクロの記録中、以下に示すメニューオプション、ショートカットキー、およびツールバーボタンの操作、または表示されるダイアログボックスへの設定を記録できます。

メニュー	メニューオプション	ショートカットキー	ツールバーボタン
ファイル	ワークスペースを開く	-	-
	ワークスペースの保存	-	-
	ワークスペースを閉じる	-	-
	新規セッション	-	-
	セッションのインポート	-	-
	セッションの保存	-	-
	セッションのリフレッシュ	-	-
	ダウンロードモジュールの追加 *2	-	-
	最近使ったワークスペース	-	-
	最近ダウンロードしたモジュール	-	-
編集	ブレークポイントの挿入/削除	F9	
	ブレークポイントの有効化/無効化	Ctrl+F9	
プロジェクト	アクティブプロジェクトに設定	-	-
	プロジェクトの挿入	-	-
	構成の編集 *3	-	-
ビルド *1	コンパイル	Ctrl+F7	
	ビルド	F7	
	すべてをビルド	-	
	複数ビルド	-	-
	クリーン アクティブプロジェクト	-	-
	すべてをクリーン	-	-
	ビルドの構成 *4	-	
デバッグ	デバッグセッション *5	-	
	CPUのリセット	-	
	実行	F5	
	リセット後実行	Shift+F5	
	フリー実行 *3	-	
	カーソル位置まで実行	-	
	カーソル位置を PC 値に設定	-	

メニュー	メニューオプション	ショートカットキー	ツールバーボタン
	条件を指定して実行	-	-
	ステップイン	F11	
	ステップオーバ	F10	
	ステップアウト	Shift+F11	
	ステップ	-	-
	ステップモード		
	自動	-	-
	アセンブリ	-	-
	ソース	-	-
	プログラムの停止	-	
	初期化	-	-
	接続 *3	-	
	接続解除 *3	-	
	メモリの保存	-	-
	メモリのベリファイ *3	-	-
	ダウンロード		
	ダウンロードモジュール名	-	-
	All Download Modules	-	-
	アンロード		
	ダウンロードモジュール名	-	-
	All Downloaded Modules	-	-
基本設定	基数		
	16進数	-	
	10進数	-	
	8進数	-	
	2進数	-	

注：

- \*1. High-performance Embedded Workshop V.4.01 以降で作成したデバッグ専用プロジェクトを使用する場合、このメニューは表示されません。
- \*2. [オフセット]、[ファイルフォーマット]、[ファイル名]、[アクセスサイズ]、および[ダウンロード時のメモリベリファイ]オプションを記録します。[デバッグ情報のみのダウンロード]および[ターゲット接続時にダウンロード]オプションは記録しません。
- \*3. 機能のサポートはデバッガに依存します。
- \*4. [現在のコンフィグレーション]ドロップダウンリストの切り替えを記録します。
- \*5. [現在のセッション]ドロップダウンリストの切り替えを記録します。






• ウィンドウ

マクロの記録中、以下に示すウィンドウへの操作を記録します。記録する操作については、各ウィンドウに関する説明を参照してください。

ウィンドウ名	ウィンドウ表示メニュー
ワークスペースウィンドウの [Projects] タブ	[表示 -> ワークスペース]
エディタ	ワークスペースウィンドウの [Projects] タブ上にツリー表示されるソースファイルのダブルクリック
逆アセンブリ	[表示 -> 逆アセンブリ]
レジスタ	[表示 -> CPU -> レジスタ]
メモリ	[表示 -> CPU -> メモリ]
IO	[表示 -> CPU -> IO]

### 15.5.1.1 ワークスペースウィンドウの Projects タブ

マクロの記録中、以下に示す操作を記録できます。

	操作対象	操作内容	機能
 ワークスペース	[クリーン 全プロジェクト]ポップアップメニューオプション *1	クリック	すべてのプロジェクトのビルドの中間ファイルおよび出力ファイルの削除
 プロジェクト	[ビルド]ポップアップメニューオプション *1		ビルド
	[すべてをビルド]ポップアップメニューオプション *1		すべてをビルド
	[クリーン アクティブプロジェクト]ポップアップメニューオプション *1		現在のコンフィグレーションのビルドの中間ファイルおよび出力ファイルの削除
 プロジェクトファイル	[アクティブプロジェクトに設定]ポップアップメニューオプション		アクティブプロジェクトに設定
	[コンパイル <ファイル名>]ポップアップメニューオプション *1		ファイルのコンパイル
 Download modules フォルダ	[全てダウンロード]ポップアップメニューオプション		すべてのオブジェクト(プログラム)ファイルのロード
	[ダウンロードモジュールの追加]ポップアップメニューオプション	クリックにより表示されるダイアログボックスで設定 *2	オブジェクト(プログラム)ファイルのロード
 ダウンロードモジュール名	-	ダブルクリック	
	[ダウンロード]ポップアップメニューオプション	クリック	
	[ダウンロード (debug 情報のみ)]ポップアップメニューオプション		オブジェクト(プログラム)ファイルのアンロード
	[アンロード]ポップアップメニューオプション		
	[ダウンロードモジュールの追加]ポップアップメニューオプション	クリックにより表示されるダイアログボックスで設定 *2	オブジェクト(プログラム)ファイルのロード

注：

\*1. ツールチェーンが登録されていない場合は、使用できません。

\*2. [オフセット]、[ファイルフォーマット]、[ファイル名]、[アクセスサイズ]、および[ダウンロード時のメモリバリエーション]オプションを記録します。[デバッグ情報のみのダウンロード]および[ターゲット接続時にダウンロード]オプションは記録しません。

### 15.5.1.2 エディタウィンドウ

マクロの記録中、以下に示す操作を記録できます。

表示モード	操作対象	操作内容	機能
ソースモード	[コンパイル '<ファイル名>']ポップアップメニューオプション	クリック	ファイルのコンパイル
	[ブレークポイントの挿入/削除]ポップアップメニューオプション		ブレークポイントの挿入/削除
	[ブレークポイントの有効化/無効化]ポップアップメニューオプション		ブレークポイントの有効化/無効化
	[カーソル位置まで実行]ポップアップメニューオプション		カーソル位置まで実行
	[カーソル位置に PC 値を設定]ポップアップメニューオプション		カーソル位置に PC 値を設定
	S/W ブレークポイントカラム	ダブルクリック	ブレークポイントの挿入/削除
混合モード/ 逆アセンブリ モード	[カーソル位置まで実行]ポップアップメニューオプション	クリック	カーソル位置まで実行
	[カーソル位置に PC 値を設定]ポップアップメニューオプション		カーソル位置に PC 値を設定
	[ブレークポイントの挿入/削除]ポップアップメニューオプション		ブレークポイントの挿入/削除
	[ブレークポイントの有効化/無効化]ポップアップメニューオプション		ブレークポイントの有効化/無効化
	[S/W ブレークポイント]カラム	ダブルクリック	ブレークポイントの挿入/削除



### 15.5.1.3 逆アセンブリウィンドウ

マクロの記録中、以下に示す操作を記録できます。

表示モード	操作対象	操作内容	機能
ソースモード	[ブレークポイントの挿入/削除]ポップアップメニューオプション	クリック	ブレークポイントの挿入/削除
	[ブレークポイントの有効化/無効化]ポップアップメニューオプション		ブレークポイントの有効化/無効化
	[カーソル位置まで実行]ポップアップメニューオプション		カーソル位置まで実行
	[カーソル位置に PC 値を設定]ポップアップメニューオプション		カーソル位置に PC 値を設定
	[S/W ブレークポイント]カラム	ダブルクリック	ブレークポイントの挿入/削除
混合モード/ 逆アセンブリ モード	[カーソル位置まで実行]ポップアップメニューオプション	クリック	カーソル位置まで実行
	[カーソル位置に PC 値を設定]ポップアップメニューオプション		カーソル位置に PC 値を設定
	[ブレークポイントの挿入/削除]ポップアップメニューオプション		ブレークポイントの挿入/削除
	[ブレークポイントの有効化/無効化]ポップアップメニューオプション		ブレークポイントの有効化/無効化
	[S/W ブレークポイント - ASM]カラム	ダブルクリック	ブレークポイントの挿入/削除

### 15.5.1.4 レジスタウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[編集]ポップアップメニューオプション	クリックにより表示されるダイアログボックスで設定	レジスタ値の設定
フラグレジスタ *	クリック	
値	インプレース編集	
	ダブルクリックにより表示されるダイアログボックスで設定	

注：\*. 機能のサポートはデバッガに依存します。

### 15.5.1.5 メモリウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[設定]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	メモリ内容の設定
[フィル]ポップアップメニュー項/ツールバーボタン		指定データによるメモリ内容の一括変更
[コピー]ポップアップメニュー項/ツールバーボタン		メモリ内容のコピー
[比較]ポップアップメニュー項/ツールバーボタン *		メモリ内容の比較
[保存]ポップアップメニュー項/ツールバーボタン		メモリ内容の保存
[読み込み]ポップアップメニュー項/ツールバーボタン		メモリ内容の読み込み
値	インプレース編集	メモリ内容の設定
	ダブルクリックにより表示されるダイアログボックスで設定	

注：\*. 機能のサポートはデバッガに依存します。

### 15.5.1.6 IO ウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
値	インプレース編集	I/O レジスタの値の編集
	ダブルクリックにより表示されるダイアログボックスで設定	

### 15.5.2 マクロを記録できる機能（デバッガ依存）

連携ツールの製品パッケージにバンドルされているデバッガに依存する機能の中で、マクロを記録できる機能について示します。

- H8S/Tiny E8a エミュレータデバッガ
- H8 Tiny/Super Low Power E8a エミュレータデバッガ
- H8 Tiny/Super Low Power E8 エミュレータデバッガ
- E10A-USB エミュレータソフトウェア（H8SX デバイスグループ）

すべてのデバッグ機能についてマクロを記録できます。

- R8C/Tiny E8 エミュレータデバッガ
- M16C/Tiny, M16C/62P E8 エミュレータデバッガ
- M32C/80 E8 エミュレータデバッガ

マクロの記録中、以下に示すウィンドウへの操作を記録できます。記録する操作については、各ウィンドウに関する説明を参照してください。

ウィンドウ	ウィンドウ表示メニュー
エディタ	ワークスペースウィンドウの[Projects]タブ上にツリー表示されるソースファイルのダブルクリック
逆アセンブリ	[表示 -> 逆アセンブリ]
ラベル	[表示 -> シンボル -> ラベル]
ウォッチ	[表示 -> シンボル -> ウォッチ]
ローカル	[表示 -> シンボル -> ローカル]
イベントポイント	[表示 -> コード -> イベントポイント]





- E10A-USB エミュレータソフトウェア（H8SX デバイスグループを除く）
- E6000 エミュレータソフトウェア
- E6000H エミュレータソフトウェア

マクロの記録中、以下に示すウィンドウへの操作を記録できます。記録する操作については、各ウィンドウに関する説明を参照してください。

ウィンドウ	ウィンドウ表示メニュー
エディタ	ワークスペースウィンドウの[Projects]タブ上にツリー表示されるソースファイルのダブルクリック
逆アセンブリ	[表示 -> 逆アセンブリ]
ラベル	[表示 -> シンボル -> ラベル]
ウォッチ	[表示 -> シンボル -> ウォッチ]
ローカル	[表示 -> シンボル -> ローカル]

- E100 エミュレータソフトウェア

マクロの記録中、以下に示すメニューオプション、ショートカットキー、およびツールバーボタンの操作を記録できます。

メニュー	メニューオプション	ショートカットキー	ツールバーボタン	
デバッグ	RTOS デバッグ	カーソル位置まで実行	-	
		ステップイン	Alt+F11	
		ステップオーバ	Alt+F10	
		ステップアウト	Shift+Alt+F11	

マクロの記録中、以下に示すウィンドウまたはダイアログボックスへの操作を記録できます。記録する操作については、各ウィンドウまたはダイアログボックスに関する説明を参照してください。

ウィンドウ/ダイアログボックス	ウィンドウ/ダイアログボックス表示メニュー
エディタ	ワークスペースウィンドウの[Projects]タブ上にツリー表示されるソースファイルのダブルクリック
逆アセンブリ	[表示 -> 逆アセンブリ]
ラベル	[表示 -> シンボル -> ラベル]
ウォッチ	[表示 -> シンボル -> ウォッチ]
ローカル	[表示 -> シンボル -> ローカル]
コードカバレッジ	[表示 -> コード -> コードカバレッジ]
データカバレッジ	[表示 -> コード -> データカバレッジ]
ハードウェアブレイク条件設定	[表示 -> イベント -> ハードウェアブレイク設定]
トレース条件設定	[表示 -> イベント -> トレース条件設定]
パフォーマンス条件設定	[表示 -> イベント -> パフォーマンス設定]
トリガ出力条件設定	[表示 -> イベント -> トリガ出力条件設定]
リアルタイムプロファイル	[表示 -> パフォーマンス -> リアルタイムプロファイル]
コンフィグレーションプロパティ	[基本設定 -> エミュレータ -> システム]
デバイスセッティング	[基本設定 -> エミュレータ -> デバイスセッティング]
Start/Stop ファンクション設定	[基本設定 -> エミュレータ -> Start/Stop ファンクション設定]

- M32C PC7501 エミュレータデバッグ
- M16C R8C PC7501 エミュレータデバッグ
- M32C PC4701 エミュレータデバッグ
- M16C PC4701 エミュレータデバッグ
- M32C コンパクトエミュレータデバッグ
- M16C R8C コンパクトエミュレータデバッグ
- M32C FoUSB/UART デバッグ
- M16C R8C FoUSB/UART デバッグ

マクロの記録中、以下に示すウィンドウへの操作を記録できます。記録する操作については、各ウィンドウに関する説明を参照してください。

ウィンドウ	ウィンドウ表示メニュー
ASM ウォッチ	[表示 -> シンボル -> ASM ウォッチ]
C ウォッチ	[表示 -> シンボル -> C ウォッチ]
S/W ブレイクポイント	[表示 -> ブレイク -> S/W ブレイクポイント]
アドレス一致ブレイクポイント	[表示 -> ブレイク -> アドレス一致ブレイクポイント]

- E200F エミュレータソフトウェア

マクロの記録中、以下に示すウィンドウまたはダイアログボックスへの操作を記録できます。記録する操作については、各ウィンドウまたはダイアログボックスに関する説明を参照してください。

ウィンドウ/ダイアログボックス	ウィンドウ/ダイアログボックス表示メニュー
エディタ	ワークスペースウィンドウの[Projects]タブ上にツリー表示されるソースファイルのダブルクリック
逆アセンブリ	[表示 -> 逆アセンブリ]
ラベル	[表示 -> シンボル -> ラベル]
ウォッチ	[表示 -> シンボル -> ウォッチ]
ローカル	[表示 -> シンボル -> ローカル]
トレース	[表示 -> コード -> トレース]
イベントポイント	[表示 -> コード -> イベントポイント]
リアルタイムプロファイル	[表示 -> パフォーマンス -> リアルタイムプロファイル]
コンフィグレーション	[基本設定 -> エミュレータ -> システム]

- M32C シミュレータデバッグ
- M16C R8C シミュレータデバッグ

マクロの記録中、以下に示すウィンドウへの操作を記録できます。記録する操作については、各ウィンドウに関する説明を参照してください。

ウィンドウ	ウィンドウ表示メニュー
ASM ウォッチ	[表示 -> シンボル -> ASM ウォッチ]
C ウォッチ	[表示 -> シンボル -> C ウォッチ]
S/W ブレークポイント	[表示 -> ブレーク -> S/W ブレークポイント]

- SuperH ファミリ用シミュレータデバッガ
- H8SX, H8S, H8 ファミリ用シミュレータデバッガ
- RX ファミリ用シミュレータデバッガ

マクロの記録中、以下に示すウィンドウへの操作を記録できます。記録する操作については、各ウィンドウに関する説明を参照してください。

ウィンドウ	ウィンドウ表示メニュー
エディタ	ワークスペースウィンドウの[Projects]タブ上にツリー表示されるソースファイルのダブルクリック
逆アセンブリ	[表示 -> 逆アセンブリ]
ラベル	[表示 -> シンボル -> ラベル]
ウォッチ	[表示 -> シンボル -> ウォッチ]
ローカル	[表示 -> シンボル -> ローカル]
カバレッジ	[表示 -> コード -> カバレッジ]
トレース	[表示 -> コード -> トレース]
イベントポイント	[表示 -> コード -> イベントポイント]
I/O シミュレーション	[表示 -> CPU -> I/O シミュレーション]

機能のサポートは、デバッガに依存するため、エミュレータデバッガまたはシミュレータデバッガのユーザーズマニュアル、ヘルプ、およびリリースノートを参照してください。

### 15.5.2.1 エディタウィンドウ

マクロの記録中、以下に示す操作を記録できます。

表示モード	操作対象	操作内容	機能
ソースモード	[インスタントウォッチ]ポップアップメニューオプション	クリックにより表示されるダイアログボックスで設定	シンボルの登録

- E8 エミュレータソフトウェア

操作対象	操作内容	機能
[EVENT ポイント]カラム	ダブルクリック	ハードウェアブレークポイントの挿入/削除

### 15.5.2.2 逆アセンブリウィンドウ

マクロの記録中、以下に示す操作を記録できます。

表示モード	操作対象	操作内容	機能
ソースモード	[インスタントウォッチ]ポップアップメニューオプション	クリックにより表示されるダイアログボックスで設定	シンボルの登録

- E8 エミュレータソフトウェア

操作対象	操作内容	機能
[EVENT ポイント]カラム	ダブルクリック	ハードウェアブレークポイントの挿入/削除

### 15.5.2.3 ラベルウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[追加]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	シンボルの追加
[削除]ポップアップメニューオプション/ツールバーボタン	クリック	シンボルの削除
[すべてを削除]ポップアップメニューオプション/ツールバーボタン		全シンボルの削除
[ロード]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	シンボルの追加
[BP]カラム	ダブルクリック	ソフトウェアブレークポイントの挿入/削除

### 15.5.2.4 ウォッチウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[自動更新有効化]ポップアップメニューオプション/ツールバーボタン	クリック	シンボルの自動更新有効化
[全シンボル自動更新有効化]ポップアップメニューオプション/ツールバーボタン		全シンボルの自動更新有効化
[自動更新無効化]ポップアップメニューオプション/ツールバーボタン		シンボルの自動更新無効化
[全シンボル自動更新無効化]ポップアップメニューオプション/ツールバーボタン		全シンボルの自動更新無効化
[値更新記録->記録開始]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	値更新の記録の開始
[値更新記録->記録終了]ポップアップメニューオプション/ツールバーボタン	クリック	値更新の記録の終了
[シンボル登録]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	シンボルの登録
[シンボル名の編集]ポップアップメニューオプション/ツールバーボタン		シンボルの削除および登録
[値の編集]ポップアップメニューオプション/ツールバーボタン		シンボルの値の編集
[削除]ポップアップメニューオプション/ツールバーボタン	クリック	シンボルの削除
[全シンボル削除]ポップアップメニューオプション/ツールバーボタン		全シンボルの削除
[基数->16進数表示]ポップアップメニューオプション/ツールバーボタン		シンボルの16進数表示
[基数->10進数表示]ポップアップメニューオプション/ツールバーボタン		シンボルの10進数表示
[基数->8進数表示]ポップアップメニューオプション/ツールバーボタン		シンボルの8進数表示
[基数->2進数表示]ポップアップメニューオプション/ツールバーボタン		シンボルの2進数表示
'+'マークおよび'-'マーク		シンボルの展開/縮小
変数名	ダブルクリックにより表示されるダイアログボックスで設定	シンボルの削除および登録
値	インプレース編集	シンボルの値の編集
	ダブルクリックにより表示されるダイアログボックスで設定	

### 15.5.2.5 ローカルウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[編集]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	シンボルの値の編集
値	ダブルクリックにより表示されるダイアログボックスで設定	
	インプレース編集	

### 15.5.2.6 イベントポイントウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[追加]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	ブレイクポイントまたはブレイクコンディションの設定
[編集]ポップアップメニューオプション/ツールバーボタン		ブレイクポイントまたはブレイクコンディションの編集
[有効]ポップアップメニューオプション/ツールバーボタン	クリック	ブレイクポイントまたはブレイクコンディションの有効
[無効]ポップアップメニューオプション/ツールバーボタン		ブレイクポイントまたはブレイクコンディションの無効
[削除]ポップアップメニューオプション/ツールバーボタン		ブレイクポイントまたはブレイクコンディションの削除
[全て削除]ポップアップメニューオプション/ツールバーボタン		すべてのブレイクポイントまたはブレイクコンディションの削除

### 15.5.2.7 コードカバレッジウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[ハードウェア設定]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	<ul style="list-style-type: none"> <li>コードカバレッジメモリの割当て設定</li> <li>コードカバレッジメモリの割当て解除</li> </ul>

### 15.5.2.8 データカバレッジウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[ハードウェア設定]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	<ul style="list-style-type: none"> <li>データカバレッジメモリの割当て設定</li> <li>データカバレッジメモリの割当て解除</li> </ul>

### 15.5.2.9 ハードウェアブレイク条件設定ダイアログボックス

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[適用]ボタン	クリック	ハードウェアブレイク条件の設定
[アクセスプロテクト違反]アイテムの[詳細]ボタン	クリックにより表示されるダイアログボックスで設定	違反検出領域の設定
[初期化抜け]アイテムの[詳細]ボタン		
[スタックアクセス違反]アイテムの[詳細]ボタン		
[タスクスタックアクセス違反]アイテムの[詳細]ボタン		

### 15.5.2.10 トレース条件設定ダイアログボックス

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[適用]ボタン	クリック	トレース条件の設定
[アクセスプロテクト違反]アイテムの[詳細]ボタン	クリックにより表示されるダイアログボックスで設定	違反検出領域の設定
[初期化抜け]アイテムの[詳細]ボタン		
[スタックアクセス違反]アイテムの[詳細]ボタン		
[タスクスタックアクセス違反]アイテムの[詳細]ボタン		

### 15.5.2.11 パフォーマンス条件設定ダイアログボックス

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[適用]ボタン	クリック	パフォーマンス条件の設定

### 15.5.2.12 トリガ出力条件設定ダイアログボックス

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[適用]ボタン	クリック	トリガ出力条件の設定

### 15.5.2.13 リアルタイムプロファイルウィンドウ

マクロの記録中、以下に示す操作を記録できます。

#### • E200F エミュレータ ソフトウェア

操作対象	操作内容	機能
[すべてのデータをクリア]ポップアップメニューオプション/ツールバーボタン	クリック	測定結果をすべてクリア
[ファイルに保存]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	測定結果をファイルに保存

#### • E100 エミュレータ ソフトウェア

操作対象	操作内容	機能
[範囲の設定]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	<ul style="list-style-type: none"> <li>プロファイルメモリの割当て設定</li> <li>プロファイルメモリの割当て解除</li> <li>リアルタイムプロファイルの測定モード設定</li> <li>各タスク ID の測定有効/無効</li> </ul>
[プロパティ]ポップアップメニューオプション/ツールバーボタン		計測単位の設定
[ファイルに保存]ポップアップメニューオプション/ツールバーボタン		測定結果をファイルに保存
[すべてのデータをクリア]ポップアップメニューオプション/ツールバーボタン	クリック	測定結果をすべてクリア

### 15.5.2.14 コンフィグレーションプロパティダイアログボックス

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[OK]ボタン	クリック	<ul style="list-style-type: none"> <li>動作クロック選択</li> <li>プラットフォームの設定</li> <li>排他機能の選択</li> <li>エミュレーションメモリの設定</li> <li>フラッシュメモリ上書き設定</li> <li>例外警告の設定</li> </ul>



### 15.5.2.15 デバイスセッティングダイアログボックス

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[OK]ボタン	クリック	デバイスの選択および動作モードの設定

### 15.5.2.16 Start/Stop ファンクション設定ダイアログボックス

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[OK]ボタン	クリック	ワークエリアの設定/実行関数の設定

### 15.5.2.17 ASM ウォッチウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[値の編集]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	ウォッチポイントの値の編集
値	ダブルクリックにより表示されるダイアログボックスで設定 インプレース編集	

### 15.5.2.18 C ウォッチウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[値の編集]ポップアップメニューオプション/ツールバーボタン	クリックにより表示されるダイアログボックスで設定	シンボルの値の編集
値	ダブルクリックにより表示されるダイアログボックスで設定 インプレース編集	

### 15.5.2.19 S/W ブレークポイントウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作方法	機能
[追加]ボタン	クリック	ソフトウェアブレークポイントの設定
[削除]ボタン		ソフトウェアブレークポイントの解除
[全て削除]ボタン		全ソフトウェアブレークポイントの解除
[有効]ボタン		ソフトウェアブレークポイントの有効化
[全て有効]ボタン		全ソフトウェアブレークポイントの有効化
[無効]ボタン		ソフトウェアブレークポイントの無効化
[全て無効]ボタン		全ソフトウェアブレークポイントの無効化
[S/W ブレークポイント]リストの選択項目	ダブルクリック	ソフトウェアブレークポイントの有効化または無効化

### 15.5.2.20 アドレス一致ブレークポイントウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作方法	機能
[追加]ボタン	クリック	アドレス一致ブレークポイントの設定
[削除]ボタン		アドレス一致ブレークポイントの解除
[全て削除]ボタン		全アドレス一致ブレークポイントの解除
[有効]ボタン		アドレス一致ブレークポイントの有効化
[全て有効]ボタン		全アドレス一致ブレークポイントの有効化
[無効]ボタン		アドレス一致ブレークポイントの無効化
[全て無効]ボタン		全アドレス一致ブレークポイントの無効化
[アドレス一致ブレークポイント]リストの選択項目	ダブルクリック	アドレス一致ブレークポイントの有効化または無効化

### 15.5.2.21 トレースウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[設定]ポップアップメニュー／ツールバーボタン	クリックにより表示されるダイアログボックスで設定	トレース情報取得の設定

### 15.5.2.22 コンフィグレーションダイアログボックス

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[OK]ボタン	クリック	プラットフォームの設定

### 15.5.2.23 カバレッジウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[すべて有効]ポップアップメニューオプション／ツールバーボタン	クリック	カバレッジ測定をすべて有効化
[すべてクリア]ポップアップメニューオプション／ツールバーボタン		カバレッジ測定をすべてクリア
[測定範囲追加]ポップアップメニューオプション／ツールバーボタン	クリックにより表示されるダイアログボックスで設定	カバレッジ測定範囲を追加
[測定範囲編集]ポップアップメニューオプション／ツールバーボタン		カバレッジ測定範囲を編集
[有効]ポップアップメニューオプション／ツールバーボタン	クリック	カバレッジ測定の有効化
[クリア]ポップアップメニューオプション／ツールバーボタン		カバレッジ測定をクリア
[保存]ポップアップメニューオプション／ツールバーボタン	クリックにより表示されるダイアログボックスで設定	カバレッジ情報の保存
[ロード]ポップアップメニューオプション／ツールバーボタン		カバレッジ情報のロード

### 15.5.2.24 I/O シミュレーションウィンドウ

マクロの記録中、以下に示す操作を記録できます。

操作対象	操作内容	機能
[削除]ポップアップメニューオプション／ツールバーボタン	クリック	I/O シミュレーションウィンドウの内容のクリア

## 15.6 マクロを実行する

マクロファイルを実行するには下記に示すいくつかの方法があります。

### ツールバーまたはメニューからマクロを実行するには

1. [マクロの実行]ボタン(▶)またはマクロツールバーや[ツール]メニューの[マクロの実行]を選択してください。
2. 定義されたマクロが1つの場合は自動的にそれを実行します。現在のデフォルトのマクロファイルに複数のマクロがある場合、ダイアログボックスが表示され、実行するマクロを選択できます。実行中はマウスポインタに[マクロの実行]アイコンが付きます(▶)。
3. OK をクリックするとマクロを実行します。マクロ実行中、[マクロの一時停止]ボタン(⏸)および[マクロの停止]ボタン(⏹)やメニューが有効になります。実行の一時停止中はマウスポインタに[マクロの一時停止]アイコンが付きます(⏸)。

[マクロの設定]ダイアログボックスでマクロの割り当てを設定すると、マクロツールバー、メニュー、またはキーボードのショートカットから簡単にマクロを実行することもできます。割り当ての詳細は「15.8 マクロを割り当てる」を参照してください。

[マクロの設定]ダイアログボックスでマクロを実行することもできます。

### [マクロの設定]ダイアログボックスからマクロを実行するには

1. [ツール -> マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. 実行したい機能を含むマクロファイルを選択してください。
3. マクロ一覧から実行したマクロ機能を選択してください。
4. [実行]ボタンをクリックしてください。
5. [マクロの設定]ダイアログボックスが閉じマクロを実行します。

## 15.7 マクロを編集する

マクロファイルを編集するには下記に示すいくつかの方法があります。

### マクロを編集するには


1. [ツール -> マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. 編集したい機能があるマクロファイルを選択してください。
3. 編集するマクロ機能をマクロ一覧から選択してください。
4. [編集]ボタンをクリックしてください。
5. [マクロの設定]ダイアログボックスが閉じ High-performance Embedded Workshop のエディタウィンドウでファイルが開きます。

マクロファイルは、各ユーザプロファイルのアプリケーションデータフォルダにある High-performance Embedded Workshop 管理情報フォルダ下の "Macros" フォルダに格納されています。これらのファイルはテキストベースの High-performance Embedded Workshop コマンドラインのコマンドファイルなので、High-performance Embedded Workshop のエディタウィンドウで手動により編集することもできます。編集結果は次回マクロを使用するときに反映されます。

## 15.8 マクロを割り当てる

カスタムのメニューオプション、ツールバー、またはキーボードのショートカットにマクロの割り当てが可能です。これにより、頻繁に使用するマクロにすばやくアクセスできます。[ツール]メニュー、[マクロ]ツールバー、キーボードのショートカットすべてに同じマクロの割り当てが可能です。カスタムのメニューオプションおよびツールバーの表示順序は、アルファベット順です。

### ツールバーボタンにマクロを割り当てるには

1. [ツール->マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. 割り当てたい機能があるマクロファイルを選択してください。
3. 割り当てるマクロ機能を[マクロ一覧]から選択してください。
4. [割り当て]ボタンをクリックし、[割り当ての詳細]ダイアログボックスを開きます。
5. [ツールバーの割り当て]チェックボックスをオンにしてください。
6. [説明]でマクロの記述を変更できます。マクロをツールバーボタンに割り当てたときにツールバーでのツールチップとして使用されます。
7. デフォルトのマクロツールバーボタン(  )を使用するか、既存のビットマップファイル(\*.bmp)から独自のツールバーボタンの画像を指定するかを選択できます。
8. 独自の画像を使用する場合、[参照]ボタンをクリックして標準のファイルブラウザを開いてください。これによりマシン上のファイルを手動で検索できます。
9. OK をクリックすると[マクロ]ツールバーに追加します。

### メニューにマクロを割り当てるには

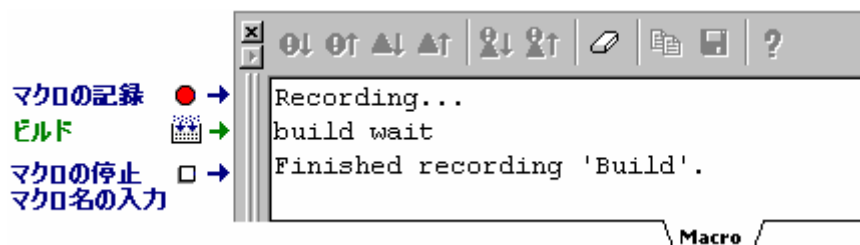
1. [ツール->マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. 割り当てたい機能があるマクロファイルを選択してください。
3. 割り当てるマクロ機能を[マクロ一覧]から選択してください。
4. [割り当て]ボタンをクリックし、[割り当ての詳細]ダイアログボックスを開きます。
5. [メニュー名の割り当て]チェックボックスをオンにしてください。
6. [説明]でマクロの記述を変更できます。これはマクロをメニューに割り当てたときにメニューのツールチップとして使用されます。
7. [ツール]メニューで表示したいメニュー名を入力してください。
8. OK をクリックすると[ツール]メニューに追加します。

### キーボードのショートカットにマクロを割り当てるには

1. [ツール->マクロの設定]を選択し、[マクロの設定]ダイアログボックスを開きます。
2. 割り当てたい機能があるマクロファイルを選択してください。
3. 割り当てるマクロ機能を[マクロ一覧]から選択してください。
4. [割り当て]ボタンをクリックし、[割り当ての詳細]ダイアログボックスを開きます。
5. [キーボードショートカットの割り当て]チェックボックスをオンにしてください。
6. マクロに割り当てたいキーボードのショートカットをドロップダウンリストで選択してください。
7. OK をクリックするとショートカットが使用可能になります。

## 15.9 アウトプットウィンドウのMacroタブの構成




アウトプットウィンドウの[Macro]タブは、現在のマクロの記録状況を出力します。[ツール->マクロの記録]による記録の開始から[ツール->マクロの停止]による記録の終了までに High-performance Embedded Workshop マクロファイルに記録した High-performance Embedded Workshop コマンドラインのコマンドなどの情報を出力します。マクロを記録中に記録内容を確認できます。



ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
クリア		ウィンドウ内容をクリアします。
保存		ウィンドウ内容をテキストファイルに保存します。
コピー		ウィンドウ内容の選択部分を Windows® クリップボードにコピーします。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。

## 16. テスト支援機能

High-performance Embedded Workshop にはユーザアプリケーションの回帰テストを行うためのテスト支援機能があります。回帰テストとはプログラムの修正後に前回成功したテストを再び実行し、その結果を検証するために実行されるものです。

この機能を使用することで、生成したマクロおよびバッチファイルを実行し、High-performance Embedded Workshop システムと元の取得データと比較し一致しているか否かを確認できます。様々なテストやスクリプトを作成し、テスト手順を自動化できます。テストが完了すると結果がテストブラウザウィンドウに表示されます。テスト手順の自動化により複数の回帰テスト実行での作業を軽減できます。

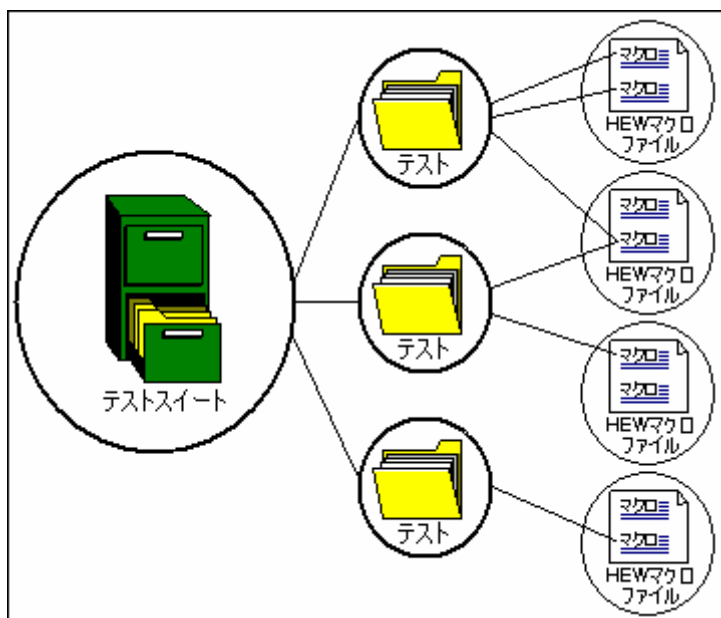
テスト支援機能は、High-performance Embedded Workshop の機能を実行する手段としてコマンドラインバッチファイルを使用しています。マクロ生成支援機能を使用するか、コマンドラインバッチファイルを編集することで作成できます。テスト実行を自動化するために、テストスイートを開く / 閉じる、テストデータを実行 / 比較するといったテスト支援機能用のコマンドをサポートしています。

最初のステップはテストスイートの作成です。これは[テスト]メニューで作成します。次にそのテストスイートを編集し、実際のテストを作成します。[テストを実行]を使用して作成したテストを実行できます。

High-performance Embedded Workshop システムにあるすべての機能のテストイメージデータを保存できるわけではありません。テストイメージデータを保存できる機能については、「16.6 テストイメージファイルにテストイメージデータとして保存できる機能」を参照してください。

### 16.1 テストスイートを作成する

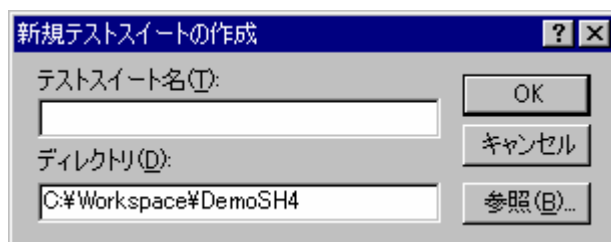
テストスイートはテストの集まりです。テストスイートの概念はワークスペースと似ています。実行する複数のマクロを含む複数のテストをテストスイートに含めることができます。



テストスイートは、現在の High-performance Embedded Workshop ワークスペースに依存しません。このため、テストスイートを複数のワークスペースおよびプロジェクトで動作させ、1つのテストスイートで様々な状況のテストを行うことが可能です。テストスイート機能で使用できる"open\_workspace"および"change\_project"コマンドは現在使用しているワークスペースを変更できます。

### テストスイートを作成するには

1. [テスト -> テストスイートの作成]を選択してください。[新規テストスイートの作成]ダイアログボックスを開きます。



2. [テストスイート名]にテストスイート名を入力してください。
3. [ディレクトリ] はあらかじめワークスペースディレクトリを表示しています。必要に応じて編集してください。
4. [OK]ボタンをクリックしてください。テストスイートが作成されます。[テスト]メニューにある他のオプションが使用可能になります。

テストスイートを作成すると、ワークスペースウィンドウの[Test]タブにテストスイートを追加します。このタブにより、テストスイートの操作およびテストへ容易にアクセスできます。

テストスイートファイルは、保存先の場所に拡張子".HTS" (High-performance Embedded Workshop テストスイート) の付いたファイルとして保存します。

## 16.2 テストスイートを開く／閉じる

テストスイートを作成すると、拡張子".HTS" (High-performance Embedded Workshop テストスイート) の付いたファイルとして保存します。以前に作成したテストスイートファイルを開きたい場合は、以下の操作を実行してください。

### テストスイートを開くには

1. [テスト -> テストスイートを開く]を選択してください。[テストスイートを開く]ダイアログボックスを開きます。
2. テストスイートファイルを選択し、[選択]ボタンをクリックしてください。テストスイートを読み込みます。[テスト]メニューにある他の項目が使用可能になり、ワークスペースウィンドウの[Test]タブに、テストスイートを表示します。

最近開いたファイルを[ファイル -> 最近使ったテストスイート]のサブメニューに表示します。最近使ったテストスイートを再び開きたいときに使用してください。

### テストスイートを閉じるには

[テスト -> テストスイートを閉じる]を選択してください。現在のテストスイートが閉じ、ワークスペースウィンドウの[Test]タブからすべての項目を削除します。

これらの操作は、コマンドラインウィンドウから High-performance Embedded Workshop コマンドラインのコマンドを使用することによっても可能です。対応するコマンドは"open\_test\_suite"と"close\_test\_suite"です。



ワークスペースウィンドウの[Test]タブのポップアップメニューからも、テストスイートを閉じることができます。

### 16.3 テストスイートを編集する

テストスイートを作成したら、次のステップはテストの追加です。この操作により、[テストスイートの編集]ダイアログボックスにアクセスし、実行したいテストを追加できます。

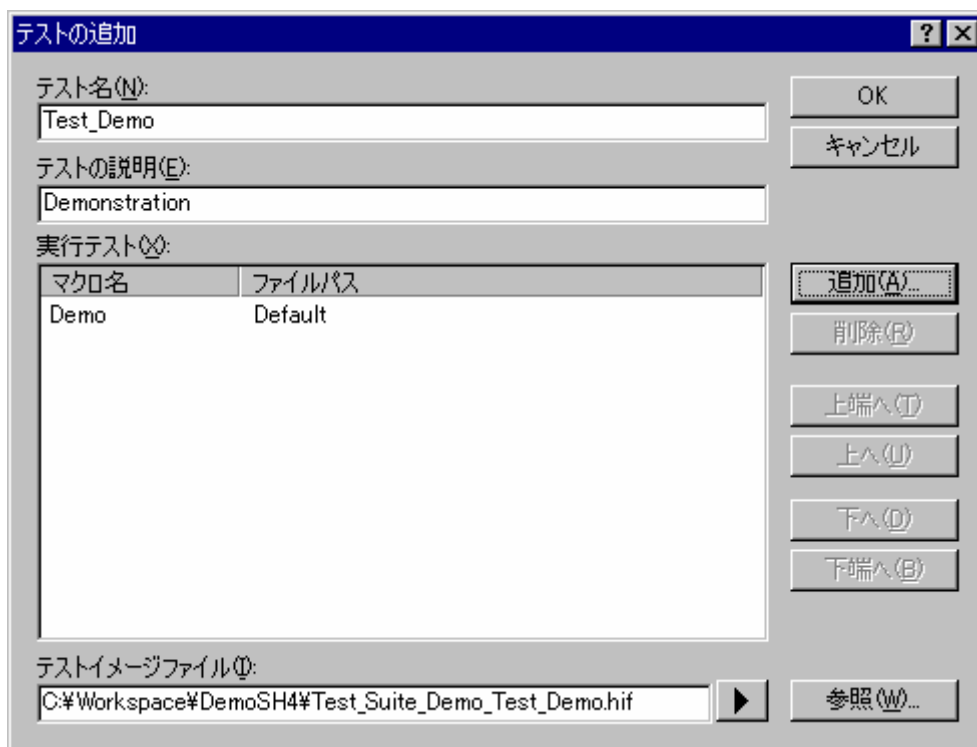
#### テストスイートを編集するには

1. [テスト -> テストスイートの編集]を選択してください。[テストスイートの編集]ダイアログボックスを開きます。



[テストスイートの編集]ダイアログボックスは、現在定義しているテストケースを表示します。各テストケースには名前があり、テストの目的を説明するため、詳細な説明を付けることができます。

このダイアログボックスで[追加]ボタンをクリックすると、[テストの追加]ダイアログボックスが開きます。



テストを選択し[変更]ボタンをクリックすると、選択したテストについて現在定義している詳細を変更できます。これらの詳細は、[テストの追加]で使用したダイアログボックスにも表示します。[削除]ボタンをクリックすると、テストを削除します。

[テストの追加]ダイアログボックスでは、後で実行するためのテストを設定できます。詳細は「テストスイートにテストを追加する」で説明します。

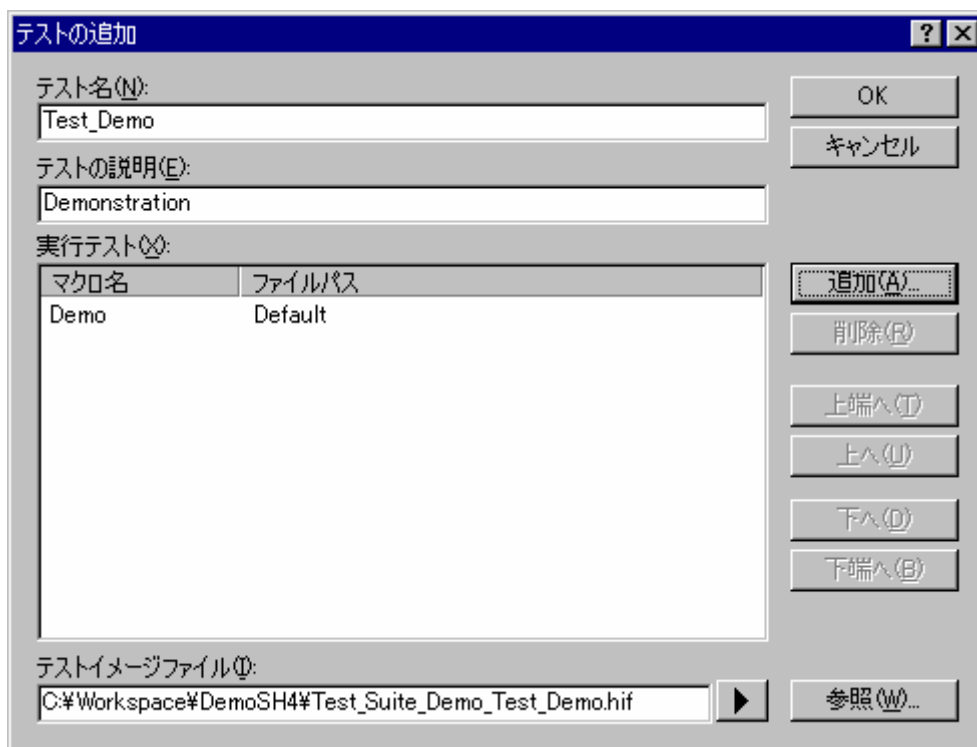
ワークスペースウィンドウの[Test]タブのポップアップメニューからも、テストスイートを編集できます。

#### 16.4 テストスイートにテストを追加する

テストスイートを作成したら、次のステップはテストの追加です。この操作により、[テストスイートの編集]ダイアログボックスにアクセスし、実行したいテストを追加できます。

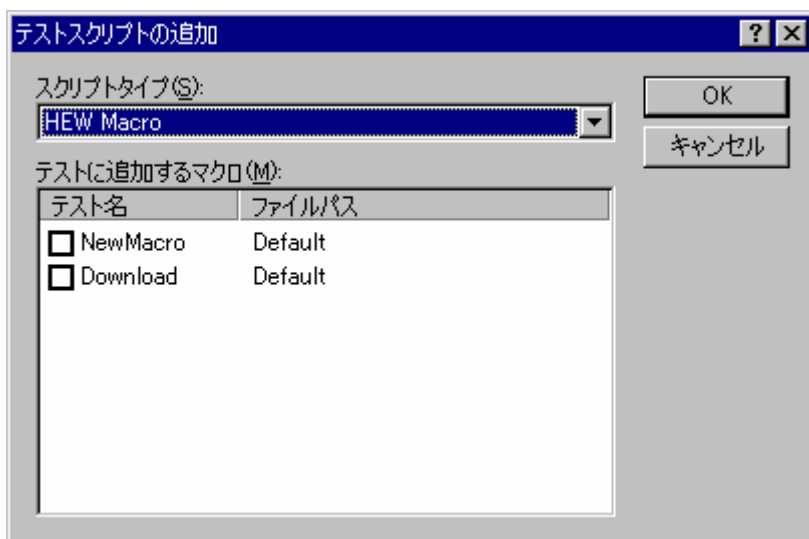
##### テストスイートにテストを追加するには

1. [テスト->テストスイートの編集]を選択してください。[テストスイートの編集]ダイアログボックスを開きます。
2. [追加]ボタンをクリックしてください。[テストの追加]ダイアログボックスを開きます。



#### 新しいテストを追加するには以下のデータの設定が必要です:

1. テスト名を入力してください。名前にはスペースを含まないようにしてください。
2. テストの説明を入力してください。後から分かりやすいように詳細な説明を入力してください。
3. [追加]ボタンをクリックし、実行するテストを設定してください。[テストスクリプトの追加]ダイアログボックス(下記参照)が開きます。
4. [スクリプトタイプ]ドロップダウンリストで、High-performance Embedded Workshop マクロ ("HEW Macro") または High-performance Embedded Workshop コマンドラインバッチファイル ("TCL command line batch file") が選択できます。
5. [スクリプトタイプ]ドロップダウンリストで"HEW Macro"を選択すると、すべての登録済みのマクロを[テストに追加するマクロ]リストに表示します。
6. 各テストに対し、実行したいマクロファイルを複数選択できます。使用したいマクロ名の横にあるチェックボックスをオンにしてください。
7. [スクリプトタイプ]ドロップダウンリストで"TCL command line batch file"を選択すると、[テストスクリプトの追加]ダイアログボックスが変更され、テスト実行時に実行するファイルを定義できます。
8. OK をクリックすると、結果を保存し[テストの追加]ダイアログボックスに追加します。
9. 最後に、[テストイメージファイル]を指定できます。このファイルには、テスト実行後に比較するシステム用の比較データを保存します。"Test\_Demo"テストアイコン上で右クリックしポップアップメニューを表示します。[テストイメージファイルの編集]を選択し[テストイメージファイルの編集]ダイアログボックスを開きます。[テスト -> テストイメージファイルの作成]を使用することで、ファイルを作成できます。
10. OK をクリックすると[テストスイートの編集]ダイアログボックスにテストを追加します。



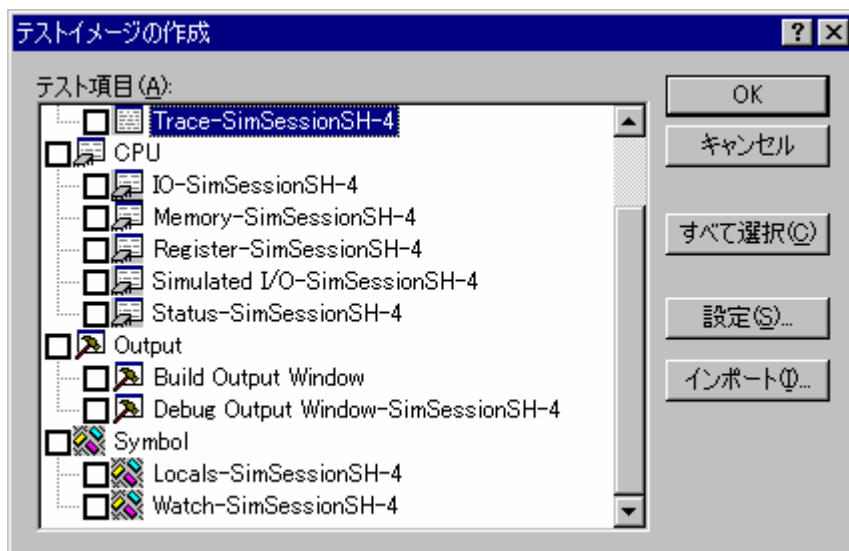
### 16.5 テストイメージファイルを作成する

テストイメージデータは、テストシステムでの比較に使用します。比較できるのは、テストイメージファイルに保存した項目のみです。

例えば、デバイス全体のメモリでなく、特定の範囲のメモリにあるメモリデータのみを比較したいとします。各テストで比較する項目数が多いほど比較処理は遅くなり複数のテストを実行すると差が生じます。

テストイメージファイルに保存するテストイメージデータを作成するには

1. [テスト -> テストイメージファイルの作成]を選択してください。[テストイメージの作成]ダイアログボックスを開きます。



2. [すべて選択]ボタンをクリックするとすべてのチェックボックスがオンになります。[すべて解除]ボタンをクリックするとすべてのチェックボックスがオフになります。
3. [インポート]ボタンをクリックすると、[テストイメージファイルのインポート]ダイアログボックスが開きます。インポートしたいHIFファイルをブラウズしてください。既存のテストイメージファイルの設定をインポートします。

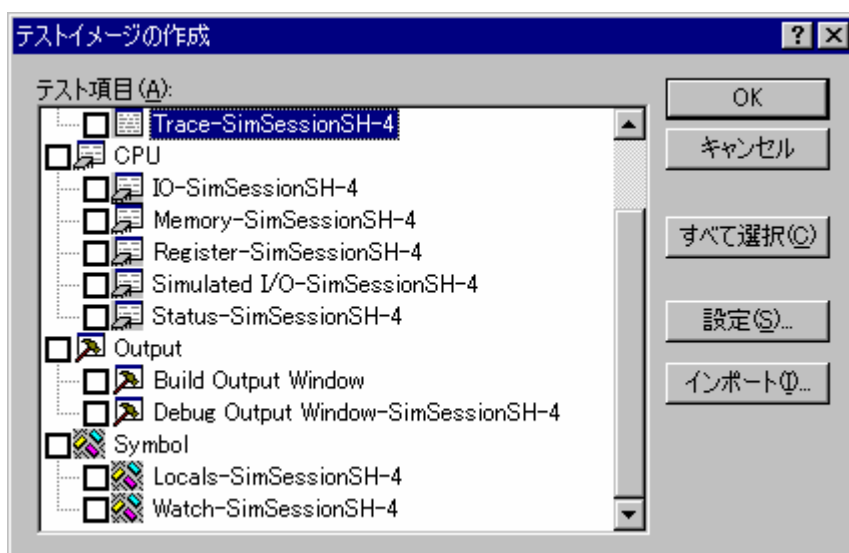
4. ダイアログボックスに各テスト項目を表示します。テスト項目名の横にあるチェックボックスをオンにすると、そのテスト項目のテストイメージデータを保存します。



5. ダイアログボックスでテスト項目を選択し、[設定]ボタンをクリックしてください。選択されたテスト項目用の詳細設定ダイアログボックスが開きます。テストイメージファイルに保存するテストイメージデータをカスタマイズできます。詳細は次節の「16.6 テストイメージファイルにテストイメージデータとして保存できる機能」を参照してください。
6. OK ボタンをクリックします。[テストイメージファイルの保存]ダイアログボックスが開きます。
7. [テストの追加]ダイアログボックスで指定した High-performance Embedded Workshop テストイメージファイルを選択し保存するか、または新規 High-performance Embedded Workshop テストイメージファイルに保存してください。チェックボックスがオンのテスト項目のテストイメージデータのみをテストイメージファイルに保存します。

ワークスペースウィンドウの[Test]タブのポップアップメニューからも、テストイメージファイルを編集できます。

## 16.6 テストイメージファイルにテストイメージデータとして保存できる機能



High-performance Embedded Workshop システムにあるすべての機能のテストイメージデータをテストイメージファイルに保存できるわけではありません。

テストイメージデータとしてテストイメージファイルに保存できる機能については、下記を参照してください。また、テスト実行後またはテストイメージファイルの比較の際に、各テスト項目のテスト結果が不一致 (FAIL) の場合のテスト結果の詳細情報についても記載しています。

- テストイメージファイルに保存できる機能(High-performance Embedded Workshop 共通)
- テストイメージファイルに保存できる機能(デバッガ依存)

テストイメージデータを保存する際にテスト項目の詳細設定が必要な場合があります。その場合は、[テストイメージの作成]ダイアログボックスで各テスト項目をダブルクリックすると、テスト項目の詳細設定ダイアログボックスが開きます。

テスト項目の詳細設定があるものは、テストイメージファイルに保存するテストイメージデータをカスタマイズできます。

テスト項目の詳細設定がないものは、テスト項目の左横のチェックボックスがオンの場合にはそのテスト項目のすべてのテストイメージデータをテストイメージファイルに保存します。

テストイメージデータをテストイメージファイルに保存できる機能については、[表示]メニュー一覧にも"テストイメージデータの保存"としてファイルに保存マーク(📁)を記載していますので参照してください。

### 16.6.1 テストイメージファイルに保存できる機能 (High-performance Embedded Workshop共通)

連携ツールの製品パッケージにバンドルされている High-performance Embedded Workshop で共通の機能については、以下のテスト項目のテストイメージファイルに保存するテストイメージデータが選択できます。

[テストイメージの作成]ダイアログボックスのテスト項目と対応するウィンドウ名を以下に示します。

テストイメージファイルに保存するテストイメージデータとその詳細設定、およびテスト結果の詳細情報については、各ウィンドウ用のテスト項目に関する説明を参照してください。

テストイメージの作成ダイアログボックス		ウィンドウ名	ウィンドウ表示メニュー
テストグループ名	テスト項目名		
Output	Build Output Window	アウトプットウィンドウの[Build]タブ	[表示 -> アウトプット]
	Debug Output Window	アウトプットウィンドウの[Debug]タブ	
CPU	Register *	レジスタ	[表示 -> CPU -> レジスタ]
	IO *	IO	[表示 -> CPU -> IO]
	Status *	ステータス	[表示 -> CPU -> ステータス]
	Memory *	メモリ	[表示 -> CPU -> メモリ]
Code	StackTrace *	スタックトレース	[表示 -> コード -> スタックトレース]

注：

\*. ターゲットに接続していない場合はテスト項目を指定できません。

#### 16.6.1.1 Output-Build/Debug (アウトプットウィンドウ用)

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果 (不一致の場合) の詳細情報について以下に示します。

テストグループ名	Output	
テスト項目名	Build Output Window	
テストイメージファイルに保存するテストイメージデータの内容	アウトプットウィンドウの[Build]タブのすべての表示内容です。	
テスト結果の詳細情報	Original	テストイメージファイルの内容
	New	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	<b>Original:</b> <b>New:</b> Building - New - Debug

テストグループ名	Output	
テスト項目名	Debug Output Window	
テストイメージファイルに保存するテストイメージデータの内容	アウトプットウィンドウの[Debug]タブのすべての表示内容です。	
テスト結果の詳細情報	Original	テストイメージファイルの内容
	New	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	<b>Original:</b> Connected <b>New:</b> Step Normal End

ウィンドウに対するテスト項目の詳細設定はありません。

### 16.6.1.2 CPU-Register（レジスタウィンドウ用）

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果（不一致の場合）の詳細情報について以下に示します。

テストグループ名	CPU	
テスト項目名	Register	
テストイメージファイルに保存するテストイメージデータの内容	レジスタウィンドウのすべての表示内容です。	
テスト結果の詳細情報	Failed at register	不一致のあったレジスタ名
	Src	テストイメージファイルの内容
	Dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	<b>Failed at register R11, Src = 0x00000000, Dest = 0x000000ff</b>

ウィンドウに対するテスト項目の詳細設定はありません。

### 16.6.1.3 CPU-IO（IO ウィンドウ用）

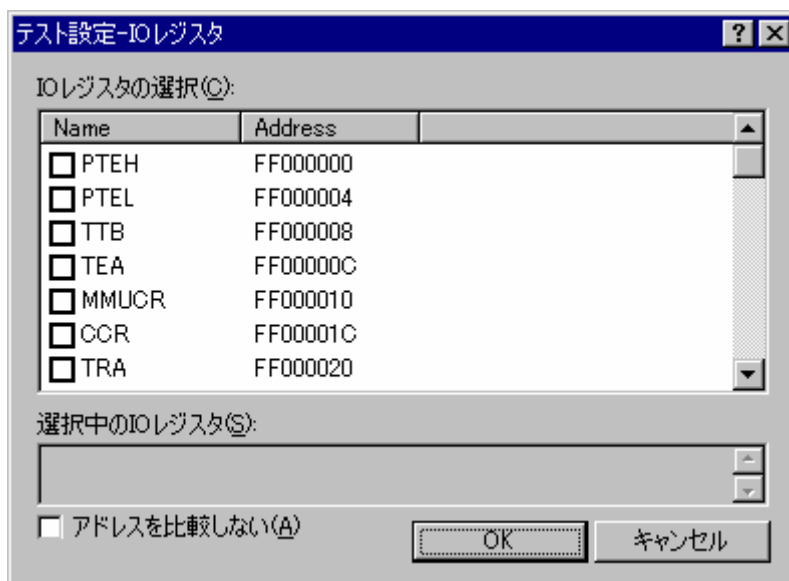
テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果（不一致の場合）の詳細情報について以下に示します。

テストグループ名	CPU	
テスト項目名	IO	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびテスト項目の詳細設定値で取得した[IO]ウィンドウの内容です。デフォルトで I/O レジスタは指定されていません。	
テスト結果の詳細情報	Failed at	不一致のあった I/O レジスタ名
	Src	テストイメージファイルの内容
	Dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	<b>Failed at IPRC, register value is different: Src = 0000, Dest = FFFF</b>

#### 詳細設定するには

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。





1. テストイメージファイルに保存する I/O レジスタを指定するには、[IO レジスタの選択]リストの I/O レジスタ名のチェックボックスをオンにします。チェックボックスのデフォルトはオフです。選択した I/O レジスタは[選択中の IO レジスタ]に表示します。
2. 選択した I/O レジスタのアドレスを比較しない場合は、[アドレスを比較しない]チェックボックスをオンにします。チェックボックスのデフォルトはオフです。
3. OK ボタンをクリックします。

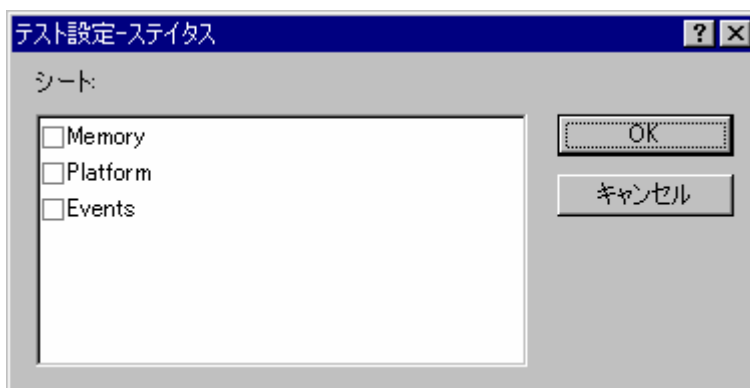
#### 16.6.1.4 CPU-Status（ステータスウィンドウ用）

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果（不一致の場合）の詳細情報について以下に示します。

テストグループ名	CPU	
テスト項目名	Status	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびステータスウィンドウのすべての内容です。デフォルトでシートは指定されていません。	
テスト結果の詳細情報	Sheet	不一致のあったシート名
	Line	不一致のあった行番号
	Src	テストイメージファイルの内容
	Dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	Status differs. Sheet = Platform Line = 6 Src = Execute From, Pipeline Reset Dest = Execute From, EX Stage

#### 詳細設定するには

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。



1. テストイメージファイルで比較するシートを指定するには、[Memory]、[Platform]、または[Events]チェックボックスをオンにします。チェックボックスのデフォルトはオフです。
2. OK ボタンをクリックします。

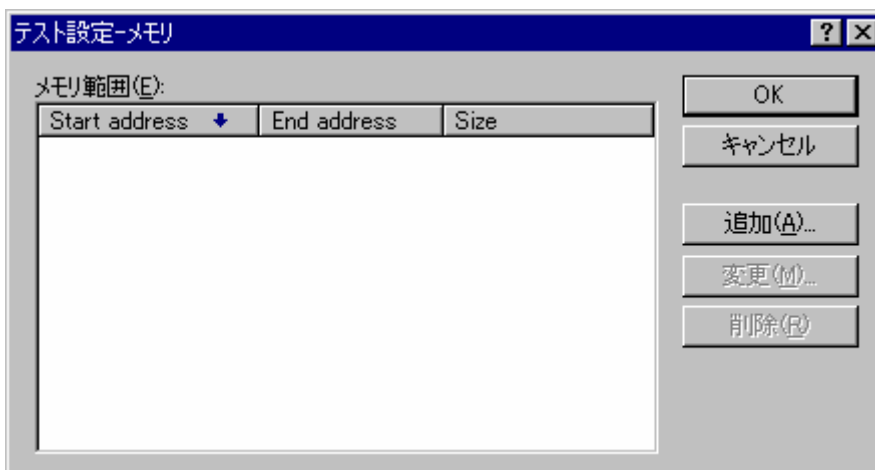
### 16.6.1.5 CPU-Memory (メモリウィンドウ用)

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果 (不一致の場合) の詳細情報について以下に示します。

テストグループ名	CPU	
テスト項目名	Memory	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびテスト項目の詳細設定値で取得したメモリウィンドウの内容です。デフォルトでメモリ範囲は指定されていません。	
テスト結果の詳細情報	Failed at address	不一致のあったアドレス
	Src	テストイメージファイルの内容
	Dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	<b>Failed at address 0x70000014. Src = 0x00002f5a, Dest = 0x00002704</b>

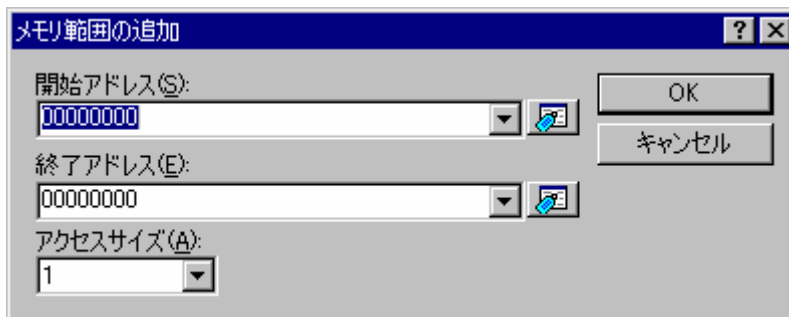
#### 詳細設定するには

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。



### • メモリ範囲を追加する

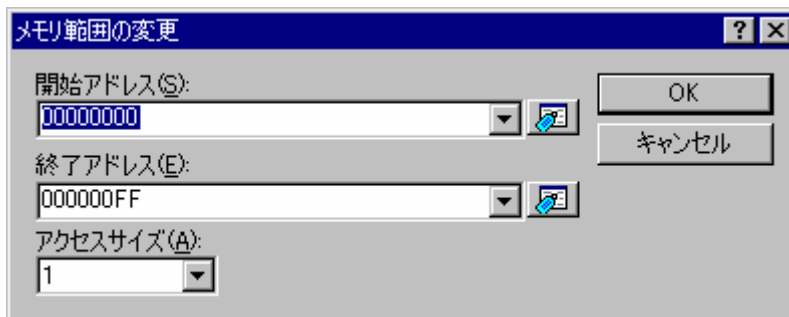
1. [テスト設定-メモリ]ダイアログボックスの[追加]ボタンをクリックし[メモリ範囲の追加]ダイアログボックスを開きます。



2. テストイメージファイルに保存するメモリ範囲は、[開始アドレス]、[終了アドレス]、および[アクセスサイズ]で指定します。
3. OK ボタンをクリックします。

### • メモリ範囲を変更する

1. [テスト設定-メモリ]ダイアログボックスの[変更]ボタンをクリックします。[メモリ範囲]リストで選択したメモリ範囲を変更できます。リストは単一選択のみ可能です。[メモリ範囲の変更]ダイアログボックスを開きます。



2. テストイメージファイルに保存するメモリ範囲は、[開始アドレス]、[終了アドレス]、および[アクセスサイズ]で指定します。
3. OK ボタンをクリックします。

### • メモリ範囲を削除する

1. [テスト設定-メモリ]ダイアログボックスの[削除]ボタンをクリックします。[メモリ範囲]リストで選択したメモリ範囲を削除できます。リストは単一選択のみ可能です。

OK ボタンをクリックします。

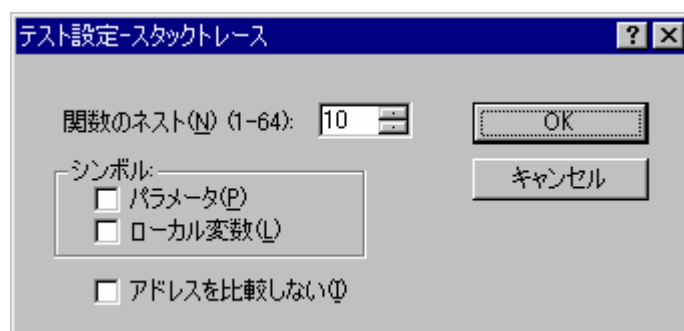
### 16.6.1.6 Code-Stack Trace（スタックトレースウィンドウ用）

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果（不一致の場合）の詳細情報について以下に示します。

テストグループ名	Code	
テスト項目名	StackTrace	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびスタックトレースウィンドウの関数コールのネスト数分の内容です。デフォルトで[関数のネスト]に 10 が指定されています。	
テスト結果の詳細情報	Src	テストイメージファイルの内容
	Dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	Value differs. Src = F, PowerON_Reset_PC(), { 0000080E } Dest = F, PowerON_Reset_PC(), { 0000081C }

#### 詳細設定するには

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。



1. テストイメージファイルに保存するスタックトレース範囲は、[関数のネスト]で指定した関数コールネスト数分の表示内容です。
2. パラメータまたはローカル変数を比較する場合は、[パラメータ]、[ローカル変数]チェックボックスをオンにします。チェックボックスのデフォルトはオフです。
3. アドレスを比較しない場合は、[アドレスを比較しない]チェックボックスをオンにします。チェックボックスのデフォルトはオフです。
4. OK ボタンをクリックします。

### 16.6.2 テストイメージファイルに保存できる機能（デバッガ依存）

連携ツールの製品パッケージにバンドルされているデバッガに依存する機能について、下記のテスト項目のテストイメージファイルに保存するテストイメージデータが選択できます。

以下に示すデバッガがサポートしている[テストイメージの作成]ダイアログボックスのテスト項目と対応するウィンドウ名を以下に示します。

テストイメージファイルに保存するテストイメージデータとその詳細設定、およびテスト結果の詳細情報については、各ウィンドウ用のテスト項目に関する説明を参照してください。

テストイメージの作成ダイアログボックス		ウィンドウ名	ウィンドウ表示メニュー	対応デバッグ
テストグループ名	テスト項目名			
Symbol	Watch	ウォッチ	[表示 -> シンボル -> ウォッチ]	<ul style="list-style-type: none"> <li>• SuperH ファミリ用シミュレータデバッグ</li> <li>• H8SX, H8S, H8 ファミリ用シミュレータデバッグ</li> <li>• RX ファミリ用シミュレータデバッグ</li> <li>• SuperH ファミリ用エミュレータデバッグ</li> <li>• H8SX, H8S, H8 ファミリ用エミュレータデバッグ ただし、以下を除く。 - H8/300H Tiny シリーズコンパクトエミュレータデバッグ</li> <li>• E100 エミュレータソフトウェア</li> <li>• E8 エミュレータソフトウェア</li> <li>• E8a エミュレータソフトウェア</li> </ul>
	Locals	ローカル	[表示 -> シンボル -> ローカル]	
	ASM Watch	ASM ウォッチ	ASM ウォッチ	[表示 -> シンボル -> ASM ウォッチ]
C Watch	C ウォッチ	C ウォッチ	[表示 -> シンボル -> C ウォッチ]	
CPU	Simulated I/O	I/O シミュレーション	[表示 -> CPU -> I/O シミュレーション]	<ul style="list-style-type: none"> <li>• SuperH ファミリ用シミュレータデバッグ</li> <li>• H8SX, H8S, H8 ファミリ用シミュレータデバッグ</li> <li>• RX ファミリ用シミュレータデバッグ</li> </ul>
Code	Coverage	カバレッジ	[表示 -> CPU -> カバレッジ]	<ul style="list-style-type: none"> <li>• SuperH ファミリ用シミュレータデバッグ</li> <li>• H8SX, H8S, H8 ファミリ用シミュレータデバッグ</li> <li>• RX ファミリ用シミュレータデバッグ</li> <li>• SuperH ファミリ用エミュレータデバッグ</li> <li>• H8SX, H8S, H8 ファミリ用エミュレータデバッグ</li> </ul>
	Trace	トレース	[表示 -> コード -> トレース]	

ターゲットに接続していない場合はテスト項目を指定できません。

機能のサポートは、デバッグに依存するため、エミュレータデバッグまたはシミュレータデバッグのユーザーズマニュアル、ヘルプ、およびリリースノートを参照してください。

### 16.6.2.1 Symbol-Watch (ウォッチウィンドウ用)

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果 (不一致の場合) の詳細情報について以下に示します。

テストグループ名	Symbol	
テスト項目名	Watch	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびテスト項目の詳細設定値で取得したウォッチウィンドウの内容です。デフォルトで各シンボルのチェックボックスはオフです。	
テスト結果の詳細情報	Failed at symbol	不一致のあったシンボル名
	Src	テストイメージファイルの内容
	Dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	<b>Failed at symbol a[6], Src = (long)H'00002704{70000018}, Dest = (long)H'00000daa{70000018}</b>

### 詳細設定するには

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。

テスト対象は、スコープを[Current Scope]に設定した変数です。そのため、スコープを[Current Scope]以外に設定している場合、[シンボル]リストに表示されません。



1. [ウォッチシート]ドロップダウンリストには、[Watch1]、[Watch2]、[Watch3]、および[Watch4]を表示します。[ウォッチシート]ドロップダウンリストで選択した項目によって、[シンボル]リストに表示するシンボルが変わります。
2. [シンボル]リストの各項目にはチェックボックスを表示します。チェックボックスはデフォルトでオフです。テストイメージファイルに保存するシンボルを指定するには、[ウォッチシート]を選択しシンボルのチェックボックスをオンにします。選択したシンボルは[選択中のシンボル]に表示します。
3. [アドレスを比較しない]チェックボックスをオンにします。チェックボックスのデフォルトはオフです。



4. OK ボタンをクリックします。

### 16.6.2.2 Symbol-Locals (ローカルウィンドウ用)

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果 (不一致の場合) の詳細情報について以下に示します。

テストグループ名	Symbol	
テスト項目名	Locals	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびテスト項目の詳細設定値で取得したローカルウィンドウの内容です。デフォルトで各シンボルのチェックボックスはオフです。	
テスト結果の詳細情報	Failed at symbol	不一致のあったシンボル名
	Src	テストイメージファイルの内容
	Dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	Failed at symbol i, Src = (int)H'00000001{R14}, Dest = (int)H'00000000{R14}

#### 詳細設定するには

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。



1. テストイメージファイルに保存するシンボルを指定するには、[シンボル]のチェックボックスをオンにします。チェックボックスのデフォルトはオフです。選択したシンボルは[選択中のシンボル]に表示されます。



2. OK ボタンをクリックします。



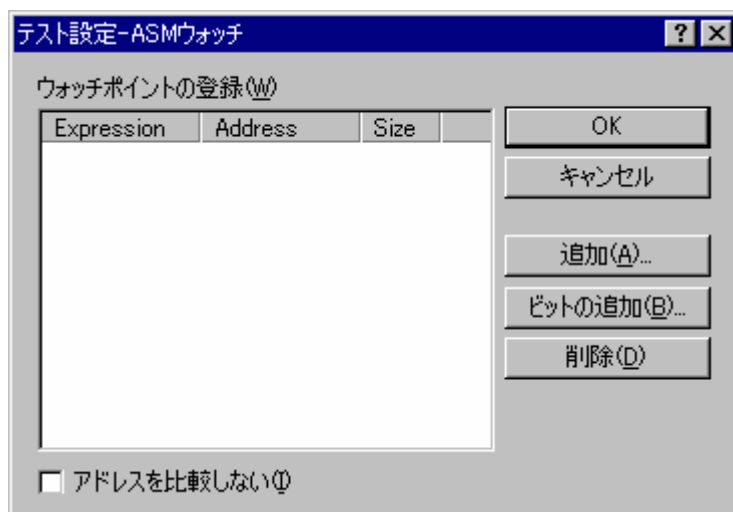
### 16.6.2.3 Symbol-ASM Watch (ASM ウォッチウィンドウ用)

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果（不一致の場合）の詳細情報について以下に示します。

テストグループ名	Symbol	
テスト項目名	ASM Watch	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびテスト項目の詳細設定値で取得した ASM ウォッチウィンドウの内容です。デフォルトでウォッチポイントは指定されていません。	
テスト結果の詳細情報	Failed at xxx, data value is different:	不一致のあったウォッチポイント
	Src	テストイメージファイルの内容
	Dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	Failed at 414, data value is different: Src = 0000, Dest = 0001

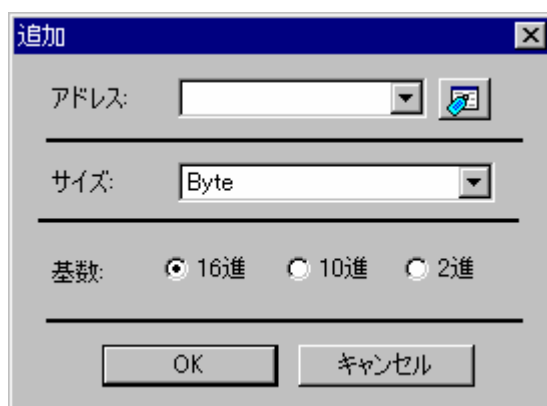
#### 詳細設定するには

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。



- ウォッチポイントを追加する

1. [テスト設定-ASM ウォッチ]ダイアログボックスの[追加]ボタンをクリックします。[追加]ダイアログボックスを開きます。



2. テストイメージファイルに保存するウォッチポイントは、[アドレス]、[サイズ]、および[基数]で指定します。
3. OK ボタンをクリックします。

- **ビット形式のウォッチポイントを追加する**

1. [テスト設定-ASM ウォッチ]ダイアログボックスの[ビットの追加]ボタンをクリックします。[ビットの追加]ダイアログボックスを開きます。



2. テストイメージファイルに保存するビット形式のウォッチポイントは、[ビットシンボル]、[アドレス]、[サイズ]、および[ビット]で指定します。
3. OK ボタンをクリックします。

- **ウォッチポイントを削除する**

1. [テスト設定-ASM ウォッチ]ダイアログボックスの[削除]ボタンをクリックします。[ウォッチポイントの登録]リストで選択したウォッチポイントを削除できます。リストは単一選択のみ可能です。

- **ウォッチポイントのアドレスを比較しない**

1. [テスト設定-ASM ウォッチ]ダイアログボックスの[アドレスを比較しない]チェックボックスをオンにします。チェックボックスのデフォルトはオフです。



OK ボタンをクリックします。

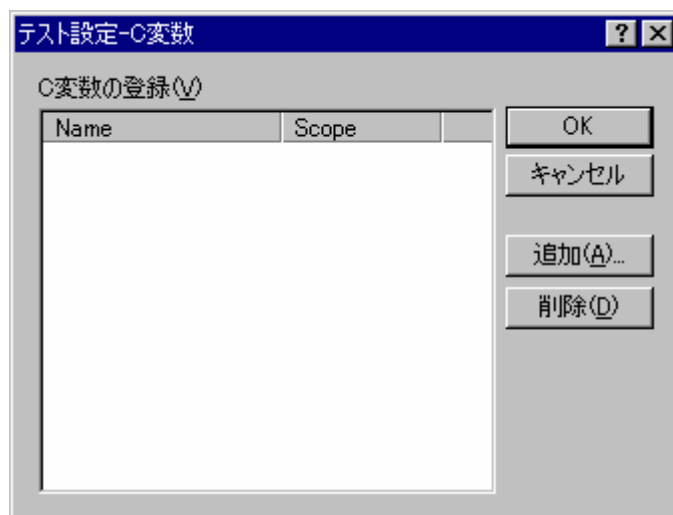
### 16.6.2.4 Symbol-C Watch (C ウォッチウィンドウ用)

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果（不一致の場合）の詳細情報について以下に示します。

テストグループ名	Symbol	
テスト項目名	C Watch	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびテスト項目の詳細設定値で取得した C ウォッチウィンドウの内容です。デフォルトで各変数のチェックボックスはオフです。	
テスト結果の詳細情報	Failed at xxx , data value is different:	不一致のあった変数名
	Src	テストイメージファイルの内容
	Dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	<b>Failed at (a)[0] ([Global]), data value is different: Src = 1783, Dest = 0</b>

#### 詳細設定するには

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。



- 変数を追加する

- [テスト設定-C変数]ダイアログボックスの[追加]ボタンをクリックします。[C変数の追加]ダイアログボックスを開きます。変数をスコープ毎に追加できます。



2. [スコープ]ドロップダウンリストには、[Global]、[Local]、およびファイル名の一覧を表示します。  
[スコープ]ドロップダウンリストで選択した項目によって、[Name]リストに表示する変数が変わります。  
[スコープ]ドロップダウンリストで[Global]を選択すると、[Name]リストにグローバル変数の一覧を表示します。また、[Local]を選択すると、現在のスコープ（プログラムカウンタ位置）で参照可能なローカル変数の一覧を表示します。また、ファイル名を選択すると、そのファイルが属するコンパイルユニットに含まれるファイルローカル変数の一覧を表示します。
3. [Name]リストの各項目にはチェックボックスを表示します。チェックボックスはデフォルトでオフです。  
[Name]リストで '+' が表示されている変数は、ダブルクリックによって展開できます。ただし、 '-' が表示されている変数をダブルクリックしても折りたたむことはできません。また、[Name]リストで '+' が表示されている変数のチェックボックスをオンにしても、その要素のチェックボックスはオフです。
4. スコープを選択し変数のチェックボックスをオンにします。
5. OK ボタンをクリックします。

#### • 変数を削除する

1. [テスト設定-C変数]ダイアログボックスの[削除]ボタンをクリックします。  
[C変数の登録]リストで選択した変数を削除できます。リストは単一選択のみ可能です。



OK ボタンをクリックします。

### 16.6.2.5 CPU-Simulated I/O (I/O シミュレーションウィンドウ用)

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果（不一致の場合）の詳細情報について以下に示します。

テストグループ名	CPU	
テスト項目名	Simulated I/O	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびテスト項目の詳細設定値で取得した I/O シミュレーションウィンドウの内容です。デフォルトで保存する I/O シミュレーション範囲は指定されていません。	
テスト結果の詳細情報	Failed at Line	不一致のあった行番号
	src	テストイメージファイルの内容
	dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	Failed at Line 1: src = ### Data Input ### dest = a[1]=21468

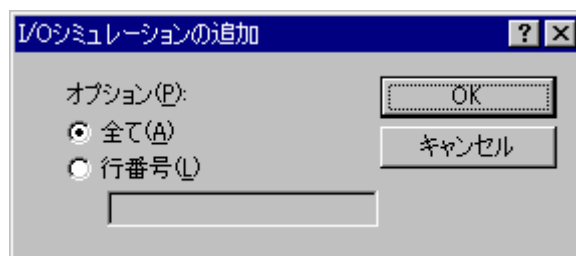
#### 詳細設定するには

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。



- I/O シミュレーション範囲を追加する

1. [テスト設定-I/O シミュレーション]ダイアログボックスの[追加]ボタンをクリックします。[I/O シミュレーションの追加]ダイアログボックスを開きます。



2. テストイメージファイルに保存する I/O シミュレーション範囲は、[全て]または[行番号]で指定します。
3. OK ボタンをクリックします。

### • I/O シミュレーション範囲を変更する

1. [テスト設定-I/Oシミュレーション]ダイアログボックスの[変更]ボタンをクリックします。[I/Oシミュレーション]リストで選択したI/Oシミュレーション範囲を変更できます。リストは単一選択のみ可能です。[I/Oシミュレーションの変更]ダイアログボックスを開きます。



2. テストイメージファイルに保存する I/O シミュレーション範囲は、[全て]または[行番号]で指定します。
3. OK ボタンをクリックします。

### • I/O シミュレーション範囲を削除する

1. [テスト設定-I/Oシミュレーション]ダイアログボックスの[削除]ボタンをクリックします。[I/Oシミュレーション]リストで選択したI/Oシミュレーション範囲を削除できます。リストは単一選択のみ可能です。

OK ボタンをクリックします。

#### 16.6.2.6 Code-Coverage (カバレッジウィンドウ用)

テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果(不一致の場合)の詳細情報について以下に示します。

テストグループ名	Code	
テスト項目名	Coverage	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびテスト項目の詳細設定値で取得したカバレッジウィンドウの内容です。デフォルトでテスト範囲は指定されていません。	
テスト結果の詳細情報	Range	カバレッジ測定範囲
	Instruction Execution mismatch at	不一致のあったテストオプション
	Image	不一致のあったアドレス
	System	テストイメージファイルの内容
	表示例	現在のシステムまたは比較で使用するテストイメージファイルの内容
	<b>Range DemoSH4.c:</b> <b>Instruction Execution Time mismatch at 0x20E4 -- Image: 1 System: 0</b>	

#### 詳細設定するには

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。



1. [有効な範囲]には設定されているカバレッジ測定範囲が表示されます。
2. テストイメージファイルに保存するカバレッジ測定範囲は、[>>]ボタンをクリックし[テストの範囲]に移動します。
3. テストイメージファイルにカバレッジ測定範囲を保存する場合、アドレスは常に保存するため、テストオプションの[アドレス]チェックボックスは常にオンです（チェックボックスは選択できません）。テストオプションの[実行パス]、[実行回数]、または[分岐条件]チェックボックスをオンにします。[実行パス]チェックボックスのデフォルトはオンです。[実行回数]、または[分岐条件]チェックボックスのデフォルトはオフです。
4. OK ボタンをクリックします。

### 16.6.2.7 Code-Trace（トレースウィンドウ用）

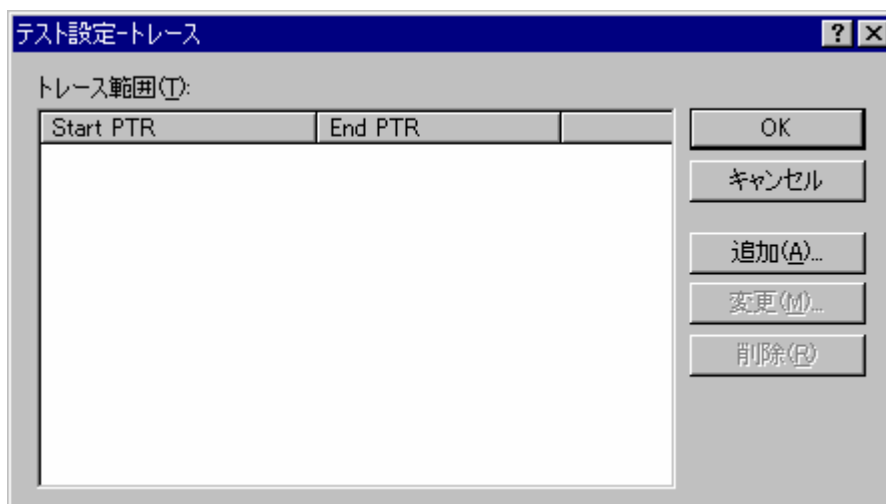
テストイメージファイルに保存するテストイメージデータの内容、およびテスト結果（不一致の場合）の詳細情報について以下に示します。

テストグループ名	Code	
テスト項目名	Trace	
テストイメージファイルに保存するテストイメージデータの内容	テスト項目の詳細設定値、およびテスト項目の詳細設定値で取得したトレースウィンドウの内容です。デフォルトでトレース範囲は指定されていません。	
テスト結果の詳細情報	Comparing PTR	比較したトレース範囲
	Trace type	トレースタイプ
	Trace data is not matching. PTR	不一致のあったトレース範囲
	Src	テストイメージファイルの内容
	Dest	現在のシステムまたは比較で使用するテストイメージファイルの内容
	表示例	<b>Comparing PTR</b> -3 to -1. <b>Trace type</b> : トレース。 <b>Trace data is not matching. PTR</b> :-3 to PTR:-1 Trace data at the beginning of difference. PTR:-3 <b>Src</b> : 0000001879 0000107C F->DEMMW MOV.L @R15+, R1 R1<-00000010 <b>Dest</b> : 0000001887 00000818 FD<E JSR @R5 PC<-00001000

#### 詳細設定するには（トレース種別が単一の場合）

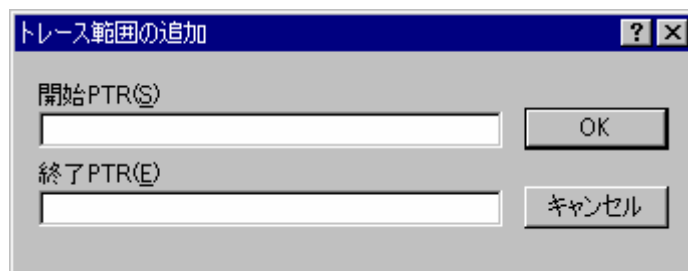
[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。





- **トレース範囲を追加する**

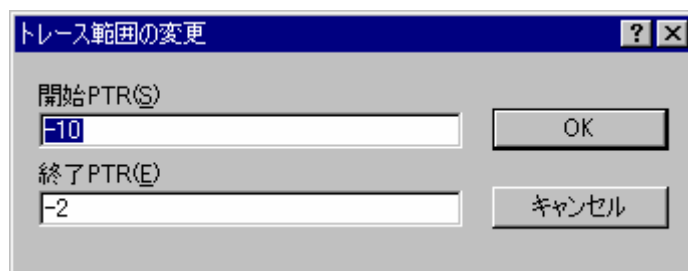
1. [テスト設定-トレース]ダイアログボックスの[追加]ボタンをクリックします。[トレース範囲の追加]ダイアログボックスを開きます。



2. テストイメージファイルに保存するトレース範囲は、[開始 PTR]および[終了 PTR]で指定します。
3. OK ボタンをクリックします。

- **トレース範囲を変更する**

1. [テスト設定-トレース]ダイアログボックスの[変更]ボタンをクリックします。[トレース範囲]リストで選択したトレース範囲を変更できます。リストは単一選択のみ可能です。[トレース範囲の変更]ダイアログボックスを開きます。



2. テストイメージファイルに保存するトレース範囲は、[開始 PTR]および[終了 PTR]で指定します。
3. OK ボタンをクリックします。

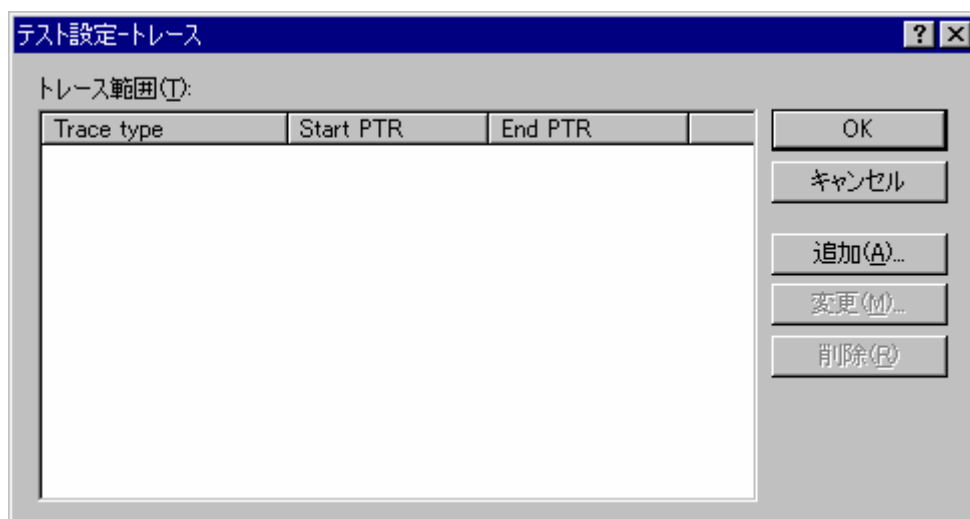
- **トレース範囲を削除する**

1. [テスト設定-トレース]ダイアログボックスの[削除]ボタンをクリックします。[トレース範囲]リストで選択したトレース範囲を削除できます。リストは単一選択のみ可能です。

OK ボタンをクリックします。

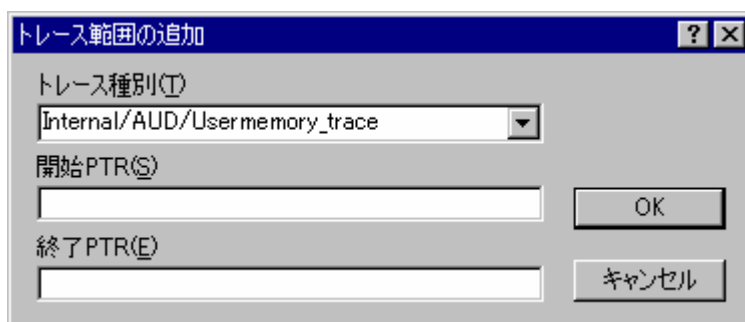
**詳細設定するには (トレース種別が複数の場合)**

[テストイメージの作成]ダイアログボックスでテスト項目をダブルクリックすると、ウィンドウに対するテスト項目の詳細設定用のダイアログボックスを開きます。



- **トレース範囲を追加する**

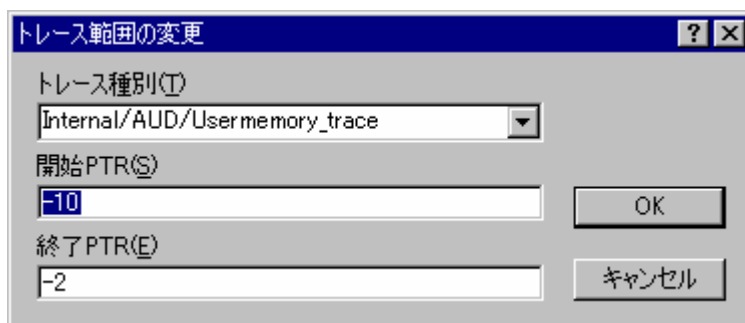
1. [テスト設定-トレース]ダイアログボックスの[追加]ボタンをクリックします。[トレース範囲の追加]ダイアログボックスを開きます。



2. テストイメージファイルに保存するトレース範囲は、[トレース種別]、[開始 PTR]、および[終了 PTR]で指定します。
3. OK ボタンをクリックします。

- **トレース範囲を変更する**

1. [テスト設定-トレース]ダイアログボックスの[変更]ボタンをクリックします。[トレース範囲]リストで選択したトレース範囲を変更できます。リストは単一選択のみ可能です。[トレース範囲の変更]ダイアログボックスを開きます。



2. テストイメージファイルに保存するトレース範囲は、[トレース種別]、[開始 PTR]、および[終了 PTR]で指定します。
3. OK ボタンをクリックします。

- **トレース範囲を削除する**

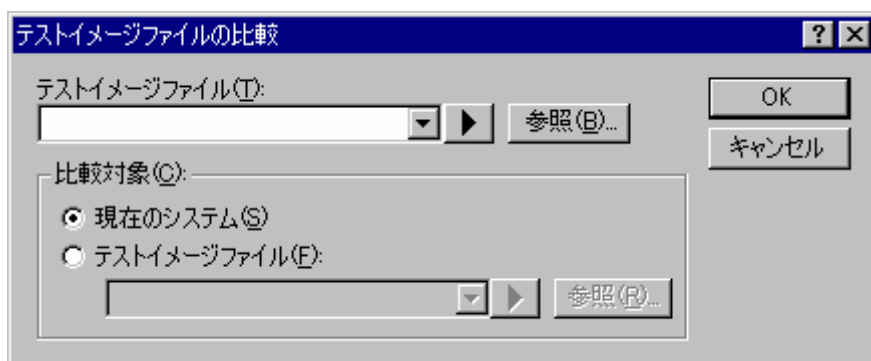
1. [テスト設定-トレース]ダイアログボックスの[削除]ボタンをクリックします。[トレース範囲]リストで選択したトレース範囲を削除できます。リストは単一選択のみ可能です。

OK ボタンをクリックします。

## 16.7 テストイメージファイルを比較する

### テストイメージファイルを比較するには

1. [テスト -> テストイメージファイルの比較]を選択してください。[テストイメージファイルの比較]ダイアログボックスを開きます。



2. [テストイメージファイル]に、比較したいテストイメージファイルのある場所を入力してください。このファイルは"\*.HIF"ファイル(High-performance Embedded Workshop テストイメージファイル)で、中にはテストイメージデータの詳細が保存されています。
3. 次に選択したファイルと比較する対象を選択してください。現在のシステム、または以前に保存された別のテストイメージファイルを選択できます。スクリプトを手動で実行し、現在のテストイメージデータを過去に保存したテストイメージデータと比較して確認したい場合は、[現在のシステム]オプションが便利です。
4. OK ボタンをクリックしてください。
5. テストブラウザに結果を表示します。

テスト実行時でなくとも、テストイメージファイルの比較が可能です。実際にテストを実行した後に、現在のシステムまたは2つのテストイメージファイルを手動で比較できます。

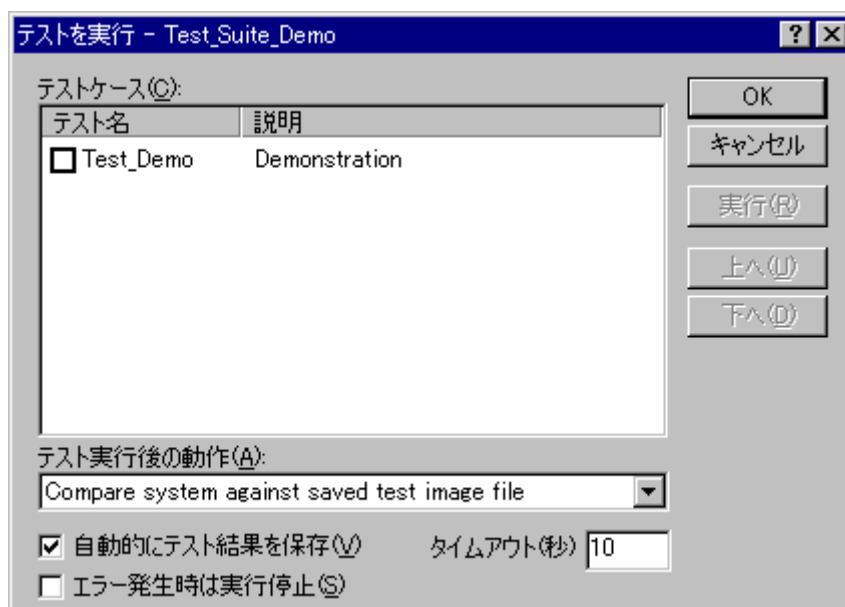
ワークスペースウィンドウの[Test]タブのポップアップメニューからも、テストイメージファイルを比較できます。

## 16.8 テストを実行する

テストを選択し、その実行を自動化できます。

### テストを実行するには

1. [テスト -> テストを実行]を選択してください。[テストを実行]ダイアログボックスを開きます。



2. High-performance Embedded Workshop テストシステムで現在定義されているすべてのテストを[Testケース]リストに表示します。
3. チェックボックスをクリックすると、今回実行するテストを選択します。
4. テストを選択して[上へ]および[下へ]ボタンをクリックすることで、テストの実行される順序を変更することもできます。
5. 1つ以上のテストを選択していれば、[実行]ボタンが使用可能です。[実行]ボタンをクリックすると、テストの実行を開始します。テスト実行中、アウトプットウィンドウの[Test]タブに情報を表示します。
6. テストが完了するとテストブラウザが開き、実行したすべてのテストの結果を表示します。

[テスト実行後の動作]ドロップダウンリストには2つのオプションがあります。

- "Compare system against saved test image file"オプションは、通常の動作で、関連するテストケースに添付したテストイメージファイル(\*.HIF)と現在の High-performance Embedded Workshop システムを比較します。これらの結果をテストブラウザに追加し、テストの成功/失敗、失敗した理由が情報として提供されます。

- "Refresh test image file"オプションは、テストケースを実行後に関連するテストケースに添付したテストイメージファイル(\*.HIF)ファイルを更新します。

[自動的にテスト結果を保存]チェックボックスをオンにすると、各テストの実行結果を自動的にテキストファイルに保存できます。このファイルの場所は、テストスイートと同じディレクトリになります。ファイル名は、現在のテストスイート名とテスト実行時の日付が付加されています。

[エラー発生時は実行停止]チェックボックスをオンにすると、最初にエラーが発生した時点でテストの実行を停止します。最初のテストが原因で他のテストも失敗してしまう場合、何度もテストを実行せずに済みます。

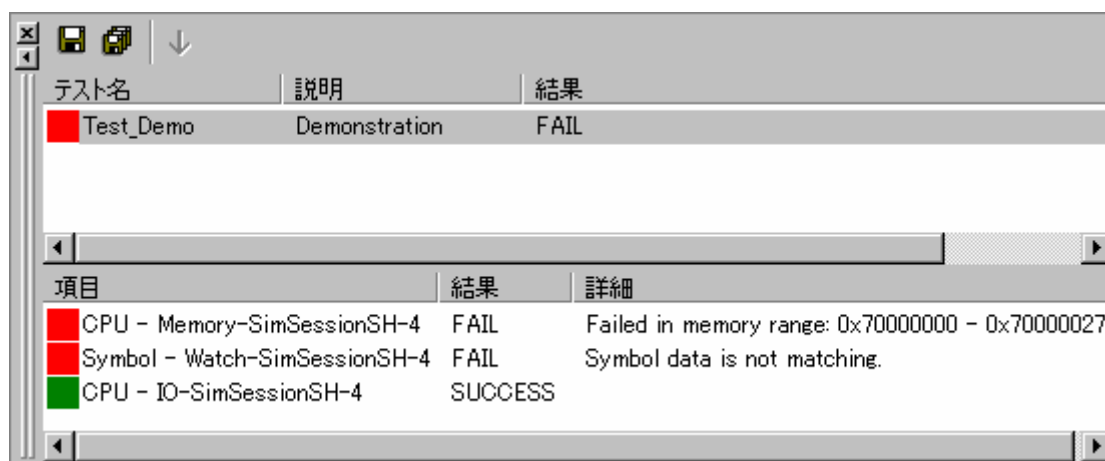
[タイムアウト]はユーザコードを実行する場合に使用します。場合によっては、バグがあるコードを実行しテスト実行中に終了しない可能性があります。テストがこのボックスに入力された秒数よりも長くかかる場合、実行を中止しテストが失敗したとみなします。

ワークスペースウィンドウの[Test]タブのポップアップメニューからも、テストを実行できます。

## 16.9 テストブラウザを使用する

テストブラウザは、前回実行したテストの結果を表示します。

[テスト -> テスト結果ブラウザ]を選択すると、テストブラウザが開きます。



テストブラウザ上側のペインには、前回実行されたすべてのテストを表示します。テスト名の左にある緑のアイコンは、テストが成功したことを示します。赤のアイコンは、テストが失敗したことを示します。

上側のペインでテストを選択すると、その下側のペインにテストの詳細を表示します。テストブラウザの下側のペインには、チェックしたすべてのテスト項目を表示しています。テスト項目名の左にある緑のアイコンは、テストが成功したことを示します。赤のアイコンは、テストが失敗したことを示します。テストが失敗した場合、このウィンドウに詳細を表示します。

テスト項目をダブルクリックすると、テストイメージファイルの比較が失敗した原因について詳細な情報を表示します。テスト項目を選択し、ツールバーの[詳細]ボタンをクリックした場合もこの情報を参照できます。



テストの結果は、ファイルにエクスポートできます。[エクスポート]と[すべてエクスポート]の2つのオプションがあります。[エクスポート]を選択すると、現在選択しているテストの結果だけを保存します。[すべてエクスポート]を選択すると、実行したすべてのテストの結果を保存します。

結果は、他の解析ツールにインポートするためのテキスト形式またはコンマ区切りのファイル形式で保存します。

ワークスペースウィンドウの[Test]タブのポップアップメニューからも、[テストブラウザ]を表示できます。


- テスト（上側のペイン）

上側のペイン内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	ツールバーボタン	機能
エクスポート		結果をエクスポートします。
すべてエクスポート		結果をすべてエクスポートします。
クリア	-	比較結果表示をクリアします。

- テスト項目（下側のペイン）

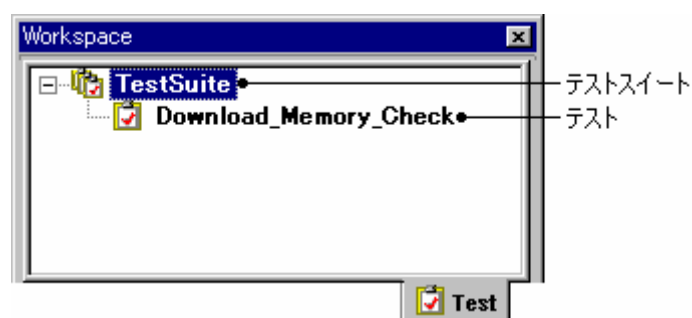
下側のペイン内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	ツールバーボタン	機能
詳細		現在選択したテスト結果の詳細をダイアログボックスに表示します。


## 16.10 ワークスペースウィンドウのTestタブの構成

ワークスペースウィンドウの[Test]タブは、テストスイート内のテストに容易にアクセスできます。


テストスイートを開くと、このテストスイートはワークスペースウィンドウの[Test]タブに追加され、その下にテストスイート内のテストが追加されます。これを以下に示します。



テストスイート名またはテストを右クリックすると、テスト機能へ容易にアクセスできます。

テストスイートのアイコン()を右クリックすると、次のオプションが使用できます。

ポップアップメニューオプション	機能
テストスイートの編集	現在のテストスイートを編集します。High-performance Embedded Workshop テストシステムへのテストの追加および削除ができます。
テストスイートを閉じる	現在開いているテストスイートを閉じます。
テストイメージファイルの比較	テストイメージファイルを、現在の High-performance Embedded Workshop システム、または以前に保存した別のテストイメージファイルと比較します。結果はテストブラウザに表示します。
テストを実行	テストスイートで定義した複数のテストを実行し、比較した結果をテストブラウザで見ることができます。テストの実行方法を指定する様々なオプションがあります。また、テストイメージファイルに小さな修正を加え更新する必要がある場合、このオプションを使用してテストイメージファイルを自動的にリフレッシュすることもできます。
テスト結果ブラウザ	前回実行したテストの結果を表示します。テストの成功/失敗と、失敗した要因の詳細を見ることができます。
プロパティ	[テストスイートのプロパティ]ダイアログボックスを開きます。テストスイートの名前、説明、場所、前回修正した日付を表示します。

テストのアイコン()を右クリックすると、次のオプションが使用できます。

ポップアップメニューオプション	機能
テストケースの実行	選択したテストケースを実行します。
テストケースの編集	選択したテストケースを編集します。
テストケースの削除	選択したテストケースを削除します。
テストイメージファイルの作成	選択したテストケースのテストイメージファイルを作成します。
テストイメージファイルの比較	選択したテストケースに設定したテストイメージファイルを現在のシステムと比較します。
テストイメージファイルの編集	選択したテストケースに設定したテストイメージファイルを編集します。
テストイメージファイルの保存	現在の High-performance Embedded Workshop システムの状態を選択したテストケースに設定したテストイメージファイルに保存します。
名前を付けて保存	現在の High-performance Embedded Workshop システムの状態を、テストイメージファイル作成時のテスト項目への指定を元に、別のテストイメージファイルに保存します。
プロパティ	[テストのプロパティ]ダイアログボックスを開きます。テストの名前、説明、場所、前回修正した日付を表示します。

## 16.11 アウトプットウィンドウのTestタブの構成

アウトプットウィンドウの[Test]タブは、現在のテスト実行の結果および進行状況を出力します。テスト実行の進行状況は、現在実行中のテストと、残りのテストの数で構成しています。エラーが発生した場合、このウィンドウに出力します。

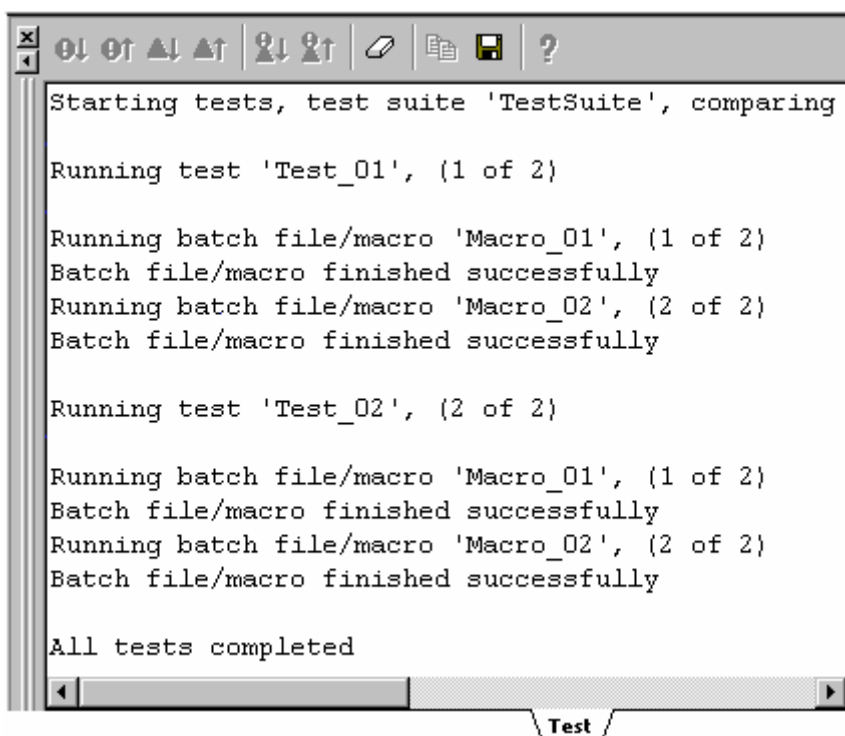
**Starting tests, test suite** <テストスイート名>, <比較条件>, <保存条件>, <エラー条件>, <タイムアウト>

**Running test,** <テスト名>, (X of Y)

**Running batch file/macro,** <マクロ名>, (X of Y)




分類	項目	テストを実行ダイアログボックス		表示例(下図参照)
<b>Starting tests, test suite</b>	<テストスイート名>	-	-	'TestSuite'
	<比較条件>	[テスト実行後の動作]ドロップダウンリストボックス	Compare system against saved test image file (デフォルト) Refresh test image file	comparing results refreshing results
	<保存条件>	[自動的にテスト結果を保存]チェックボックス	<input checked="" type="checkbox"/> (デフォルト) <input type="checkbox"/>	saving results not saving results
	<エラー条件>	[エラー発生時は実行停止]チェックボックス	<input checked="" type="checkbox"/> <input type="checkbox"/> (デフォルト)	stop on failure continue on failure
	<タイムアウト>	[タイムアウト(秒)]エディットボックス	任意 (デフォルト 10)	timeout=10
	<b>Running test</b>	<テスト名>	-	-
(X of Y) Xは現在のテスト番号。 Yはテストの総数		-	-	(1 of 2) (2 of 2)
<b>Running batch file/macro</b>	<マクロ名>	-	-	'Macro_01' 'Macro_02'
	(X of Y) Xは現在のマクロ番号。 Yはマクロの総数	-	-	(1 of 2) (2 of 2)





ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。  
 基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	機能
クリア		ウィンドウ内容をクリアします。
保存		ウィンドウ内容をテキストファイルに保存します。
コピー		ウィンドウ内容の選択部分を Windows® クリップボードにコピーします。
ツールバー表示	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	ツールバーボタンをカスタマイズします。

## 17. デバッグ機能

この章では、主に連携ツールの製品パッケージにバンドルされている High-performance Embedded Workshop で共通のデバッグ機能について説明します。

デバッグの使用に必要な High-performance Embedded Workshop の基本概念は「1.概要」を参照してください。

ご使用のデバッガで利用可能な機能の詳細説明は、エミュレータまたはシミュレータのユーザーズマニュアルおよびヘルプを参照してください。

### 17.1 デバッグの準備をする

この節では、プログラムのデバッグを開始するための準備作業について説明します。デバッグを行うためのデバッガの選択および設定方法、デバッグ対象ファイルのロード、およびデバッグセッションについて説明します。

#### 17.1.1 デバッグの前にビルドを行う

プログラムを C/C++ソースレベルでデバッグするには、C/C++プログラムをビルドして、[Debug]オプションが有効な状態でリンクする必要があります。このオプションが有効の場合、コンパイラは、C/C++コードのデバッグに必要な情報をすべてアブソリュートファイルまたはマネージメントファイルに入れます。これらのファイルはその後「デバッグオブジェクトファイル」と呼ばれます。プロジェクトを作成すると、初期セットアップで通常のデバッグを設定します。

注：

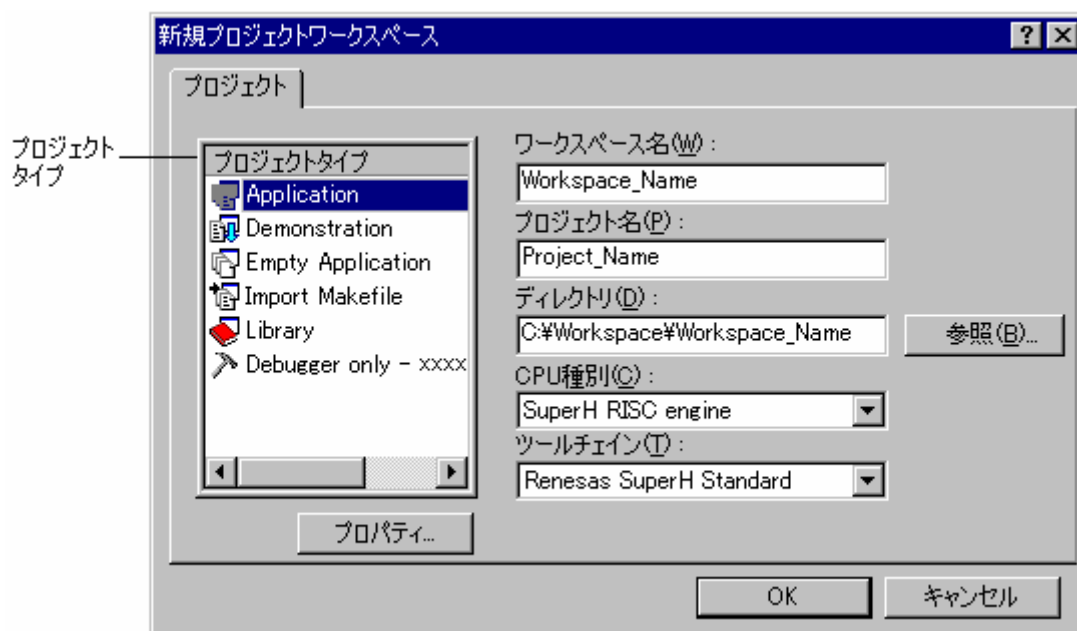
- デバッグ用のオブジェクトファイルを生成する際はコンパイラおよびリンカの[Debug]オプションを必ず有効にしてください。
- デバッグオブジェクトファイルにデバッグ情報(例えばSレコードフォーマットなど)がない場合でもデバッガにロードできますが、アセンブリ言語レベルでのデバッグとなります。

#### 17.1.2 デバッガを選択する

デバッガの選択は High-performance Embedded Workshop のインストール方法に大きく依存します。High-performance Embedded Workshop にツールチェーンをインストールしている場合、アプリケーションプロジェクトジェネレータはツールチェーンおよびデバッグのターゲットを同時にセットアップできます。これによってターゲットおよびツールチェーンオプションを一致させ、矛盾が起こらないようにします。ツールチェーンをインストールしていない場合には、デバッグ専用プロジェクトのみ選択できます。High-performance Embedded Workshop はデフォルトの設定で、[新規プロジェクトワークスペース]ダイアログボックスの中に、生成するそれぞれの CPU ファミリのデバッグ専用プロジェクト生成タイプを表示します。

注：

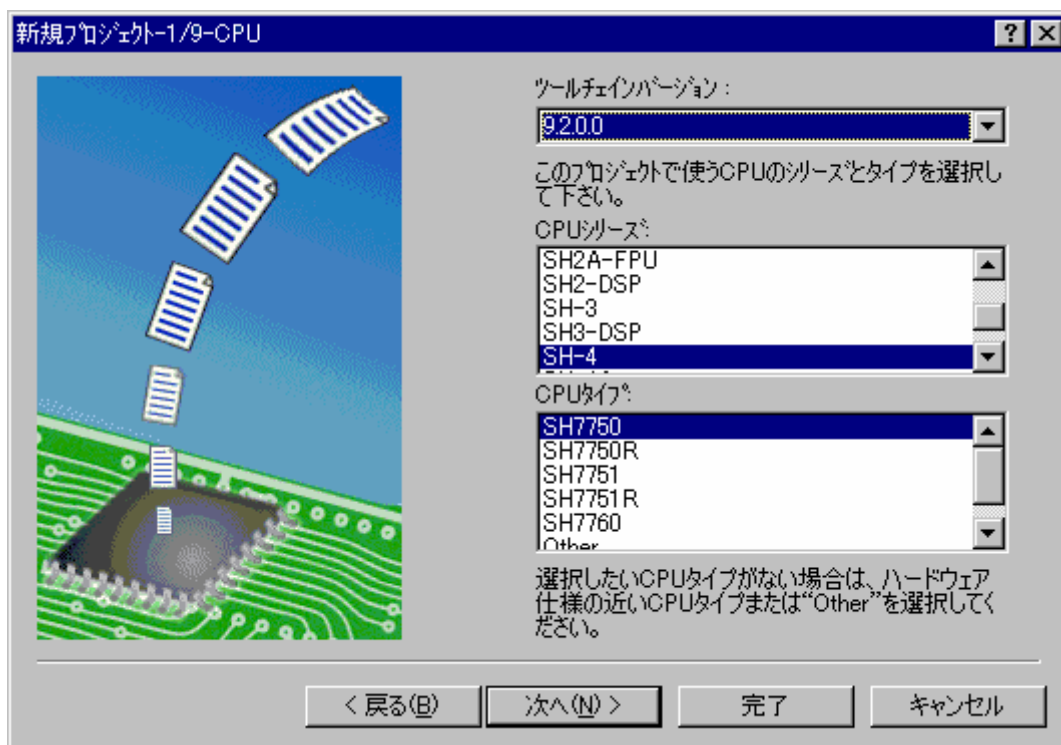
High-performance Embedded Workshop のプロジェクトジェネレータは、マイコンに対応したサンプルのソースファイルを自動生成します(選択するプロジェクトタイプによっては、生成ファイルはありません)。生成されたソースファイルのコードは一例であり、これを使用して作成したすべてのプログラムの動作を保証するものではありません。該当するマイコンのハードウェアマニュアルをご参照の上、必要に応じてコードを変更しコードの適用可否をご判断ください。



[新規プロジェクトワークスペース]ダイアログボックスを使用して、ターゲット CPU に合ったプロジェクト生成タイプを選択できます。

プロジェクトタイプ	説明
[Application]	C/C++言語で記述された初期ルーチンファイルを含む実行プログラムを生成するためのプロジェクト
[Assembly Application]	アセンブリ言語で記述された初期ルーチンファイルを含む実行プログラムを生成するためのプロジェクト
[Demonstration]	C言語で記述されたデモンストレーションプログラムを作成するためのプロジェクト
[C source startup Application]	C言語で記述されたスタートアッププログラムを作成するためのプロジェクト
[Empty Application]	ツールチェーン環境の設定のみのプロジェクト(生成ファイルなし)
[Import Makefile]	既存のメイクファイルを取り込み実行プログラムを作成するためのプロジェクト
[Library]	ライブラリファイルを作成するためのプロジェクト(生成ファイルなし)
[Debugger only - xxxxxx]	デバッグ専用プロジェクト(生成ファイルなし)

ここでは SH-4 シミュレータデバッガを例に説明します。ご使用のデバッガで利用可能な機能の説明は、デバッガのユーザーズマニュアルを参照してください。

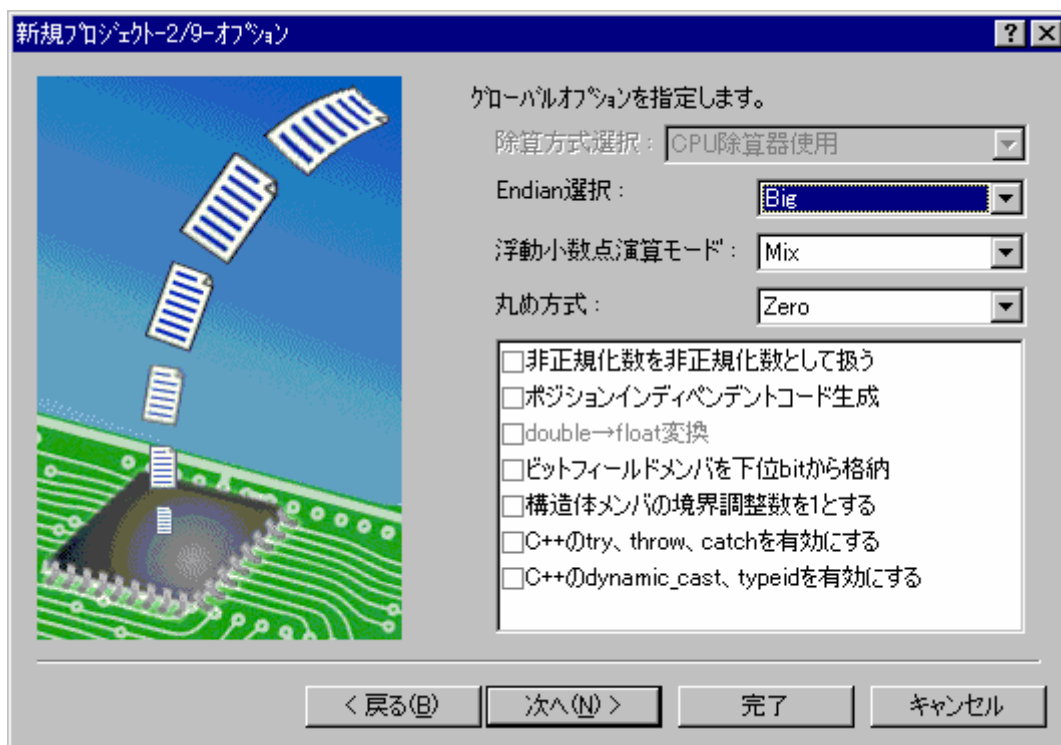


- ステップ1ではツールチェーンバージョンとCPUを選択します。使用するCPUの種類([CPUタイプ])はCPUのシリーズ([CPUシリーズ])ごとに分類しています。ツールチェーンバージョン、[CPUシリーズ]、および[CPUタイプ]の選択により生成するファイルが異なりますので、開発するプログラムに対応した設定を選択してください。対応するCPUがない場合は、ハードウェア仕様の近いCPUまたは[Other]を選択してください。

ウィザード下部のボタンは[新規プロジェクト]ウィザード共通です。

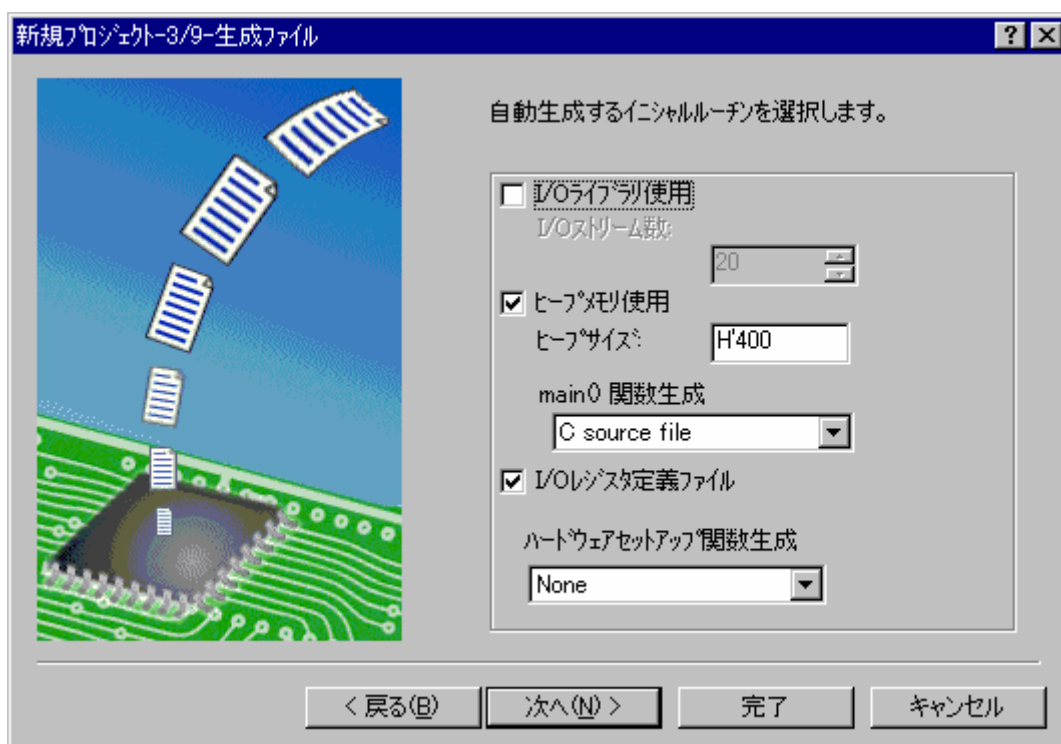
- [次へ>] 次の画面に移ります。  
 [<戻る] 前の画面に戻ります。  
 [完了] 以降の選択はデフォルトとなり、[概要]ダイアログボックスを開きます。  
 [キャンセル] [新規プロジェクトワークスペース]ダイアログボックスに戻ります。

ステップ1画面で[次へ>]をクリックすると、ステップ2に進みます。



2. ステップ2では全プロジェクトファイルで共通のオプションを設定します。  
設定できる項目はステップ1で選択したCPUに依存します。詳細はコンパイラユーザズマニュアルで確認してください。

ステップ2画面で[次へ>]をクリックすると、ステップ3に進みます。



## 3. ステップ 3 では生成するファイルを設定します。

[I/O ライブラリを使用]	チェックすると標準入出力ライブラリを活用できます。
[I/O ストリーム数]	同時に使用する入出力ストリームの本数を設定します。
[ヒープメモリを使用]	チェックするとヒープ領域管理用の関数 <code>sbrk()</code> を使用できます。
[ヒープサイズ]	管理するヒープ領域サイズ単位を指定します。
[main()関数生成]	main 関数の雛型生成を選択できます。
	メイン関数ファイル[(プロジェクト名).c/cpp)]を生成します。
[I/O レジスタ定義ファイル]	チェックすると C 言語で記述した I/O レジスタ定義ファイル ("iodefine.h") を生成します。
[ハードウェアセットアップ 関数生成]	I/O レジスタ初期設定プログラムの雛型生成を選択します。 ハードウェア設定用のファイル("hwsetup.c/cpp") または"hwsetup.src")を生成します。

## 注：

既に作成した main 関数を組み込む場合は、[main()関数生成]で[None]を選択し、プロジェクトを作成後に main 関数を含んだファイルをプロジェクトに追加してください。なお、組み込む関数名が異なる場合は、`resetprg.c` の関数呼び出し部分を修正する必要があります。また、プロジェクトジェネレータが生成するベクタテーブル定義および I/O レジスタ定義等のサンプルファイル内容は、使用する CPU のハードウェアマニュアルで確認してください。

ステップ 3 画面で[次へ>]をクリックすると、ステップ 4 に進みます。



## 4. ステップ 4 では C/C++ コンパイラで使用する標準ライブラリの構成を設定します。

チェックした項目で宣言されている関数と runtime 関数を組み込みます。

[全て有効]	すべての標準ライブラリ関数を選択します。
[全て無効]	すべての標準ライブラリ関数を選択しません。 ただし、最低限必要な runtime、new は選択します。

ステップ 4 画面で[次へ>]をクリックすると、ステップ 5 に進みます。



- ステップ 5 ではスタック領域を設定します。スタックポインタの初期値とスタックサイズを設定します。スタック領域の初期値はステップ 1 で選択した CPU に依存します。

**注：**

スタック領域は High-performance Embedded Workshop が生成する"stacksct.h"で定義しています。"stacksct.h"をエディタで編集した場合は High-performance Embedded Workshop の[プロジェクト->構成の編集]では変更できません。

ステップ 5 画面で[次へ>]をクリックすると、ステップ 6 に進みます。





6. ステップ 6 ではベクタを設定します。

[ベクタテーブル定義] チェックするとベクタ定義ファイルとベクタテーブル設定関数の定義ファイルを生成します。

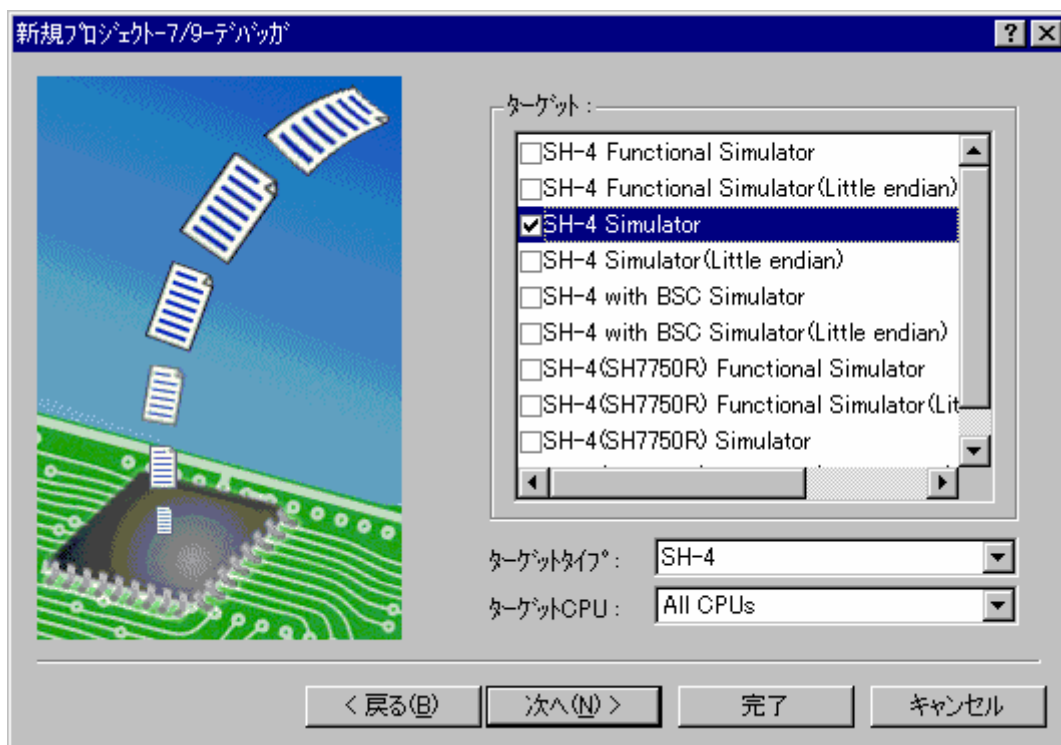
[ベクタハンドラ] [ハンドラ] リセットベクタのハンドラプログラム名を表示します。ハンドラプログラムを変更したい場合は、ハンドラプログラム名を選択してクリック後入力してください。なお、ハンドラプログラムを変更すると、リセットプログラム("resetprg.c")は生成しません。

[ベクタ] ベクタの説明を表示します。

**注：**

生成されたリセットプログラム、割込み関数、リセットベクタハンドラ、および割込み要因レジスタ定義はサンプルであるため、使用される前に CPU のハードウェアマニュアルで確認してください。

ステップ 6 画面で[次へ>]をクリックすると、ステップ 7 に進みます。



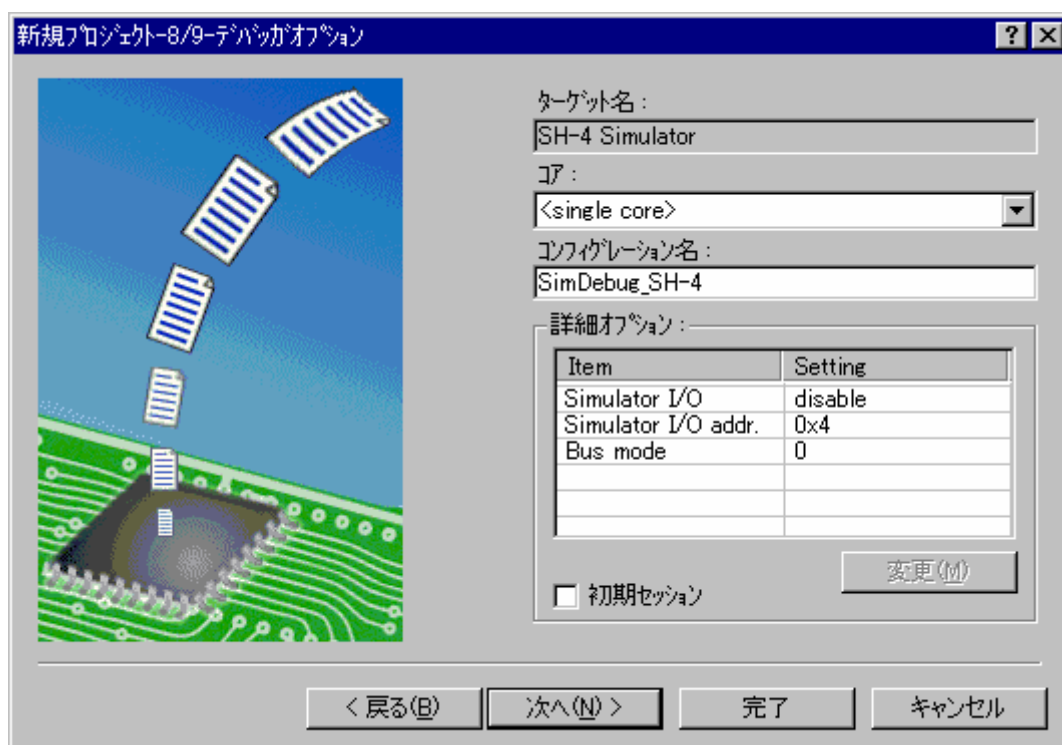
7. ステップ7ではデバッガターゲットを設定します。

- |             |  |
|-------------|--|
| [ターゲット]     | デバッガターゲットを設定します。デバッガターゲットを選択(チェック)してください。デバッガターゲットは未選択でも複数選択してもかまいません。 |
| [ターゲットタイプ]  | [ターゲット]に表示するターゲットの種類を指定します。  |
| [ターゲット CPU] | [ターゲット]に表示するターゲットの CPU を指定します。   |

**注：**

ステップ2で設定したエンディアンはコンパイラの設定に反映します。  
 ステップ7で選択したデバッガターゲットのエンディアンとは独立しています。

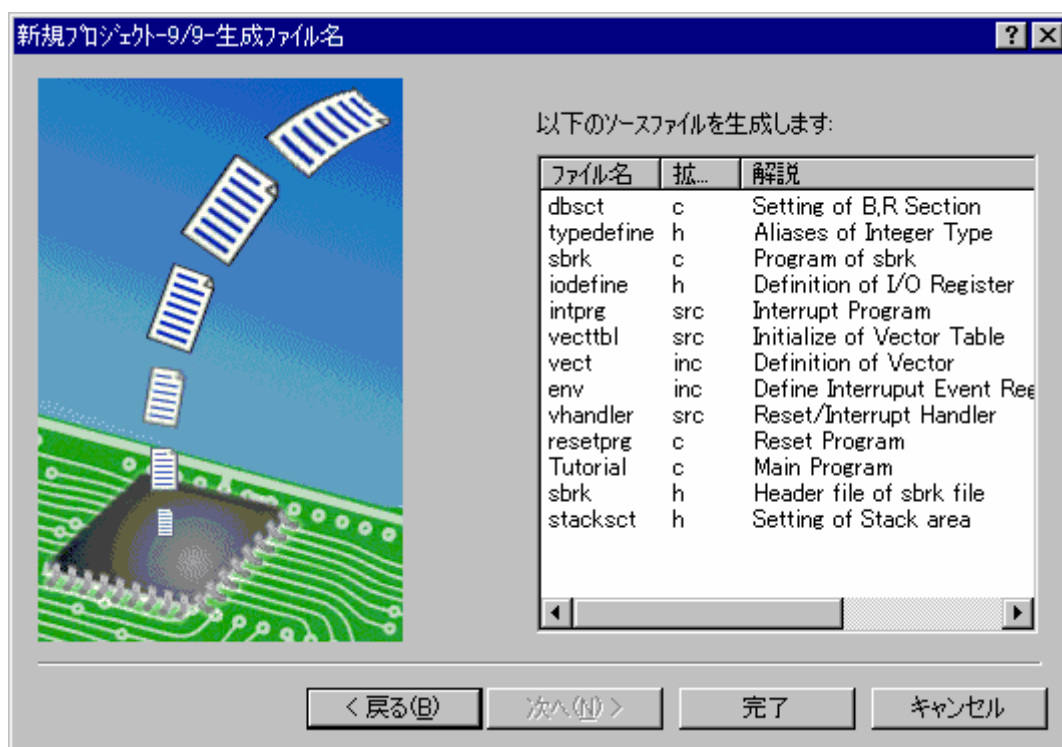
ステップ7画面で[次へ>]をクリックすると、ステップ8に進みます。



8. ステップ 8 では選択したデバッガターゲットのオプションを設定します。

[コア]	ターゲットのコアを指定します。
[コンフィギュレーション名]	High-performance Embedded Workshop はデフォルトで[Release]と[Debug]の2つのコンフィギュレーションを作成しますが、デバッガターゲットを選択すると選択したターゲット用のコンフィギュレーションも作成します(ターゲット名を含んだ略称となります)。このコンフィギュレーション名は[コンフィギュレーション名]で変更できます。
[詳細オプション]	デバッガターゲットのオプションを設定します。変更する場合は、[Item]を選択して[変更]をクリックしてください。なお、変更できない項目の場合、[Item]を選択しても[変更]はグレーのままです。
[Simulator I/O]	ユーザプログラムから標準入出力またはファイル入出力を行うシステムコールは有効([Enable])または無効([Disable])
[Simulator I/O addr.]	上記システムコールアドレス
[Bus mode]	シミュレータデバッガでは現状未使用
[初期セッション]	チェックを付けると本セッションを初期セッションとします。

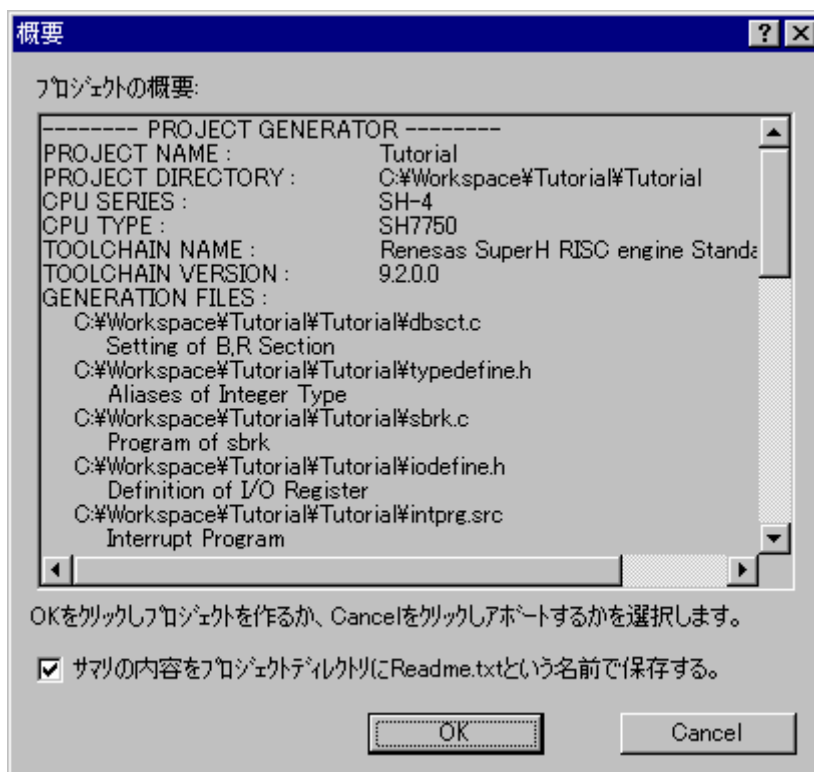
ステップ 8 画面で[次へ>]をクリックすると、ステップ 9 に進みます。



9. ステップ9ではこれまでの設定により High-performance Embedded Workshop が生成するファイルをリスト表示します。

[ファイル名]    ファイル名。変更したい場合は、ファイル名を選択してクリック後入力してください。  
 [拡張子]        拡張子  
 [解説]          ファイルの説明

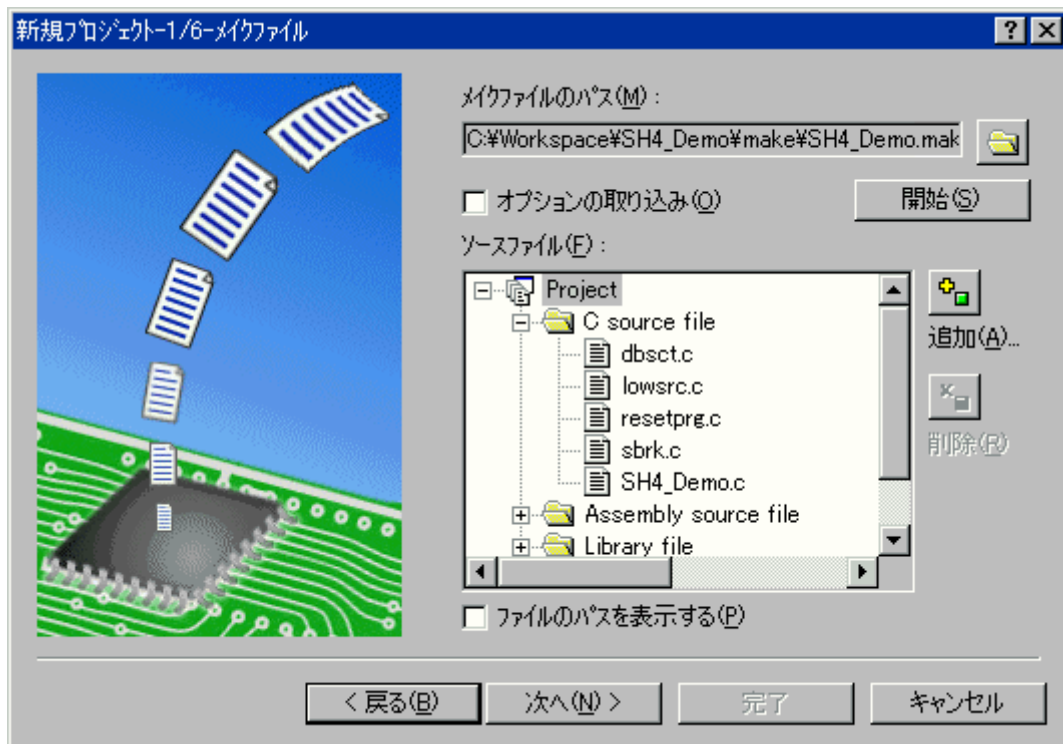
ステップ9画面で[完了]をクリックすると、[概要]ダイアログボックスを開きます。



10. プロジェクトジェネレータは、生成するプロジェクトに関する情報を[概要]ダイアログボックスに表示するので、確認後[OK]ボタンをクリックしてください。[Cancel]ボタンをクリックすると[新規プロジェクト]ウィザードに戻ります。  
 なお、[サマリの内容をプロジェクトディレクトリに Readme.txt という名前で保存する]をチェックすると、[概要]ダイアログボックスで表示したプロジェクトの情報を"Readme.txt"という名称のテキストファイルでプロジェクトディレクトリに保存します。

### 17.1.2.1 メイクファイルから情報を取り込みプロジェクトを作成する

High-performance Embedded Workshop は、GNU make フォーマットまたは、Hmake フォーマット (High-performance Embedded Workshop が出力) のメイクファイルを解析し、作成するワークスペースにファイル情報を取り込むことができます。[新規プロジェクトワークスペース]ダイアログボックスを開きプロジェクトタイプとして[Import Makefile]を選択します。ワークスペース名等の必要な情報を入力後、[OK]ボタンをクリックすると、[新規プロジェクト - メイクファイル]画面に進みます。



[メイクファイルのパス]でメイクファイルを選択すると[ソースファイル]にメイクファイルに登録されているソースファイルを表示します。メイクファイルに登録されているソースファイルを再び表示するには[開始]ボタンをクリックします。

コンパイラ等のツールオプションを取り込むには、[オプションの取り込み]チェックボックスをオンにします。不要なファイルがある場合は、そのファイルを選択し[削除]ボタンで削除できます。また、さらに必要なファイルがある場合は、[追加]ボタンで追加できます。[ファイルのパスを表示する]チェックボックスをオンにするとファイルの絶対パスを表示します。

### 17.1.3 プロジェクトの構成を編集する

SuperH ファミリー用、または H8SX, H8S, H8 ファミリー用のツールチェーンを使用する場合、プロジェクトジェネレータを使用してシミュレータを再度設定できます。ただし、デモンストレーションプロジェクトでは無効です。

1. [プロジェクト -> 構成の編集]を選択してください。[構成の編集]ダイアログボックスが開きます。
2. [ターゲット]タブをクリックしてください。
3. 使用したいターゲットを選択し、[OK]をクリックしてください。

### 17.1.4 デバッガを設定する

デバッガにプログラムをロードする前に、アプリケーションシステムに合うようにデバッガを設定します。必要な設定内容は、デバイスの種類、動作モード、クロックスピード、およびメモリマップなどです。その中でもプログラムをロードするためにデバッガにメモリを設定する必要があるため、特にメモリマップの設定が大切です。High-performance Embedded Workshop では、プロジェクトの生成処理でこの作業のほとんどが完了します。しかし、標準とは異なるボードのコンフィグレーションを使用する場合には、カスタマイズが必要になります。

#### 17.1.4.1 セットアップ

デバッグをセットアップするには、[基本設定 -> シミュレータ]または[基本設定 -> エミュレータ]を選択します。このサブメニューの下に、デバッグの設定に使用できるメニューがあります。

SuperH ファミリ用のシミュレータの場合、[システム]と[メモリリソース]のメニューがあります。どちらのオプションでも、必要に応じてシミュレータをカスタマイズし、セットアップできます。[デバッグ -> デバッグの設定]ダイアログボックスから選択してデバッグ依存のセットアップダイアログボックスを表示できます。

ご使用のデバッグで利用可能な機能の詳細説明は、エミュレータまたはシミュレータのユーザーズマニュアルを参照してください。

#### 17.1.4.2 メモリマップ

デバッグは、デバイスのアドレス空間が ROM、RAM、オンチップレジスタ、またはメモリのない領域なのかを知る必要があります。そのため、メモリマップを設定する必要があります。

プロジェクトジェネレータでデバイスの種類およびモードを選択すると、High-performance Embedded Workshop は自動的にそのデバイスのマップおよびそのプロセッサが動作するモードを設定します。例えば、内部 ROM および RAM を持つデバイスにおいて、これらのメモリがデバイスメモリマップ内で位置する領域をデフォルトで設定します。

内部メモリを持たないデバイス、またはその代わりあるいはその追加として外部メモリを持つデバイスを使用する場合には、そういったメモリが存在することをデバッグに知らせる必要があります。

##### 補足説明:

エミュレータでデバッグを行い、かつ必要なメモリがオンチップまたは外部に存在しない場合は、アプリケーションのアドレス空間用にエミュレータのエミュレーションメモリをマッピングできます。

詳細は新規プロジェクトで選択したデバッグに依存します。

メモリマップに関する追加情報はステータスウィンドウの[Memory]タブにも表示します。"Target Device Configuration"領域にはデバイスのアドレス空間内のメモリを表示します。

##### 注:

いくつかのエミュレータではページ長の制限から、入力アドレスと設定された領域アドレスが一致しない場合があります。

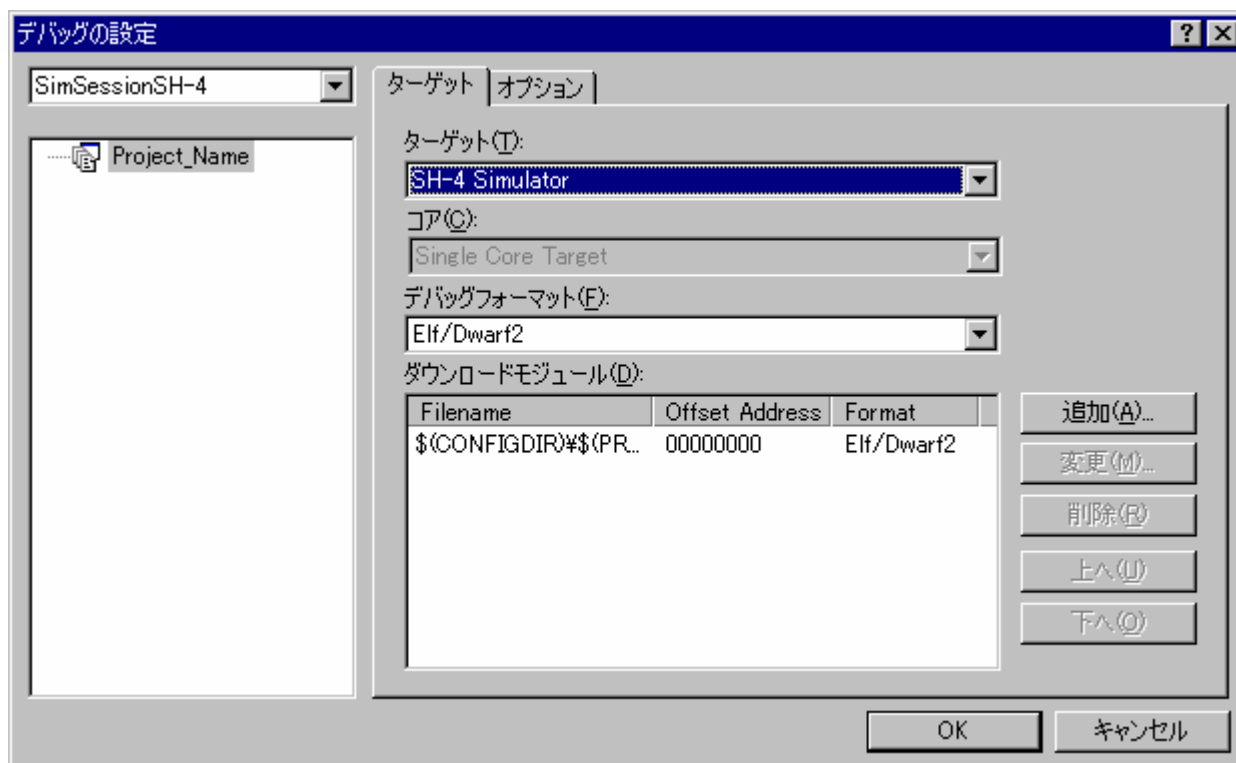
#### 17.1.4.3 デバッグの設定を変更する

High-performance Embedded Workshop デバッグの通常動作では、ユーザのターゲットおよびダウンロードモジュールはプロジェクト生成過程で自動的に設定します。ただし時にはデバッグセッションを手動で設定する必要があります。古いツールチェーンや、最新の High-performance Embedded Workshop インタフェースをサポートしていないプロジェクトジェネレータを使用している場合などが考えられます。

デバッグセッションの設定を確認するには、[デバッグの設定]ダイアログボックスを開きます。以下のいずれかの操作を選択してください。

- [デバッグ -> デバッグの設定]を選択する。
- ワークスペースウィンドウの[Projects]タブの[Download modules]フォルダまたはダウンロードモジュール上で右クリックする。表示されるポップアップメニューから[デバッグの設定]を選択する。





[デバッグの設定]ダイアログボックスの[ターゲット]タブでは、[ターゲット]、[コア]、[デバッグフォーマット]、および[ダウンロードモジュール]を選択できます。ただし、[コア]は、同期デバッグ機能を使用している場合に選択できます。

ターゲットを変更するには、次の操作が必要です。

1. ダイアログボックス左側にあるツリーから、変更したいプロジェクトを選択してください。現在のプロジェクトのデフォルトとして設定します。
2. ツリーの上にあるドロップダウンリストから、変更したいセッションを選択してください。
3. ターゲットのリストを操作して、ターゲットを変更してください。これにより、過去に設定したターゲット固有の設定オプションを削除します。

さらに[デバッグの設定]ダイアログボックスの[オプション]タブでは、以下のオプションを選択できます。

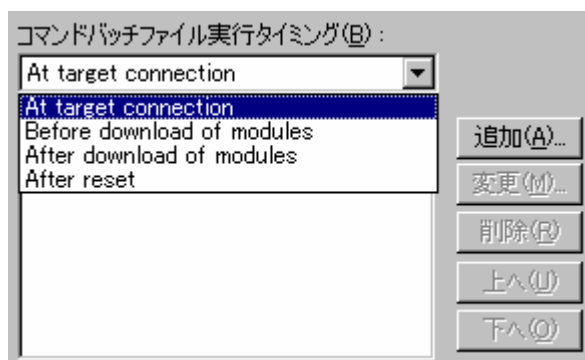
- コマンドバッチファイルを自動実行する
- デバッグ情報のみのダウンロード時にバッチファイルを実行しない
- ビルド後にダウンロードする
- ダウンロード時にブレークポイントを削除する
- ターゲット接続時のバッチファイル実行が終わるまでメモリアクセスを無視する
- 逆アセンブリ表示時のメモリアクセス範囲を最小限に抑える
- 自動的にターゲットを接続しない
- ダウンロード後に CPU リセットする
- ターゲット実行中の GUI 操作によるメモリアクセスを無視する

### (1) コマンドバッチファイルを自動実行する

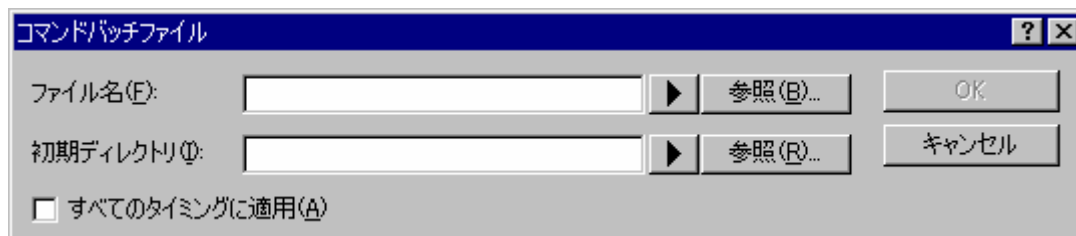
High-performance Embedded Workshop のデバッグは TCL コマンドライン機能と統合されています。特定のタイミングで自動的に実行できる、High-performance Embedded Workshop のデバッグ用のコマンドバッチファイルを記述できます。[コマンドバッチファイル実行順序]リストは、指定されたタイミングで実行するコマンドバッチファイルのリストです。このリストに対してコマンドバッチファイルを追加、変更、削除、上へ移動、下へ移動が可能です。

#### コマンドバッチファイルを自動実行するには

1. [デバッグ -> デバッグの設定]を選択します。[デバッグの設定]ダイアログボックスが開きます。
2. [オプション]タブを選択します。
3. [コマンドバッチファイル実行タイミング]ドロップダウンリストでコマンドバッチファイル実行タイミングを選択してください。[At target connection]、[Before download of modules]、[After download of modules]、および[After reset]から選択できます。



4. [追加]ボタンをクリックします。[コマンドバッチファイル]ダイアログボックスを開きます。



5. [ファイル名]にコマンドバッチファイルのファイル名を入力します。[ファイル名]にプレースホルダを挿入するには、プレースホルダボタン(▶)をクリックし、ポップアップメニューからプレースホルダを選択します。ファイルを参照するには[参照]ボタンをクリックします。
6. [初期ディレクトリ]にコマンドバッチファイルを実行するディレクトリを入力します。[初期ディレクトリ]の設定により相対パスがあるコマンドバッチファイルの使用が可能になります。[初期ディレクトリ]にプレースホルダを挿入するには、プレースホルダボタン(▶)をクリックし、ポップアップメニューからプレースホルダを選択します。ディレクトリを参照するには[参照]ボタンをクリックします。
7. [すべてのタイミングに適用]チェックボックスをオンにすると、すべてのタイミング ([At target connection]、[Before download of modules]、[After download of modules]、および[After reset]) にコマンドバッチファイルを適用します。
8. [OK]ボタンをクリックすると、コマンドバッチファイルを追加します。
9. 追加後は、[上へ]ボタンと[下へ]ボタンを使って順番を移動できます。複数のコマンドバッチファイルを追加している場合のみ有効です。

10. [OK]ボタンをクリックします。

**注：**

- FILE\_LOAD または FILE\_LOAD\_ALL コマンドでダウンロードした場合、[Before download of modules]および[After download of modules]タイミングのコマンドバッチファイルは実行されません。
- RESET コマンドでリセットした場合、[After reset]タイミングのコマンドバッチファイルに記述された RESET コマンドは実行されません。

**(2) デバッグ情報のみのダウンロード時にバッチファイルを実行しない**

このオプションの設定により、[コマンドバッチファイル実行タイミング]で"Before download of modules"または"After download of modules"を選択し、バッチファイルを登録していても、ワークスペースウィンドウ上にあるモジュールを右クリックし、[ダウンロード(debug 情報のみ)]を選択した場合は、バッチファイルを実行しません。

**デバッグ情報のみのダウンロード時にバッチファイルを実行しないようにするには**

1. [デバッグ -> デバッグの設定]を選択します。[デバッグの設定]ダイアログボックスが開きます。
2. [オプション]タブを選択します。
3. [デバッグ情報のみダウンロード時にバッチファイルを実行しない]チェックボックスをオンにします。このチェックボックスのデフォルトはオンです。
4. [OK]ボタンをクリックします。

**(3) ビルド後にダウンロードする**

このオプションの設定により、ビルド後にユーザプログラムを自動的にダウンロードします。

**ビルド後にダウンロードするには**

1. [デバッグ -> デバッグの設定]を選択します。[デバッグの設定]ダイアログボックスが開きます。
2. [オプション]タブを選択します。
3. [ビルド後のダウンロード]チェックボックスをオンにします。このチェックボックスのデフォルトはオンです。
4. [OK]ボタンをクリックします。

**(4) ダウンロード時にブレークポイントを削除する**

このオプションの設定により、ユーザプログラムのダウンロード時自動的にブレークポイントを解除します。

**ダウンロード時にブレークポイントを削除するには**

1. [デバッグ -> デバッグの設定]を選択します。[デバッグの設定]ダイアログボックスが開きます。
2. [オプション]タブを選択します。
3. [ダウンロード時にブレークポイントを削除]チェックボックスをオンにします。このチェックボックスのデフォルトはターゲットに依存します。
4. [OK]ボタンをクリックします。

#### (5) ターゲット接続時のバッチファイル実行が終わるまでメモリアクセスを無視する

このオプションの設定により、ターゲット接続時に自動実行するコマンドバッチファイルを指定している場合、そのコマンドバッチファイルの実行が終了するまで、ターゲットに対してメモリアクセスを行いません。ターゲット接続時にメモリの初期設定が必要な場合に有効です。

#### ターゲット接続時のバッチファイル実行が終わるまでメモリアクセスを無視するには

1. [デバッグ -> デバッグの設定]を選択します。[デバッグの設定]ダイアログボックスが開きます。
2. [オプション]タブを選択します。
3. [ターゲット接続時のバッチファイル実行が終わるまでメモリアクセスを無視]チェックボックスをオンにします。このチェックボックスのデフォルトはオフです。
4. [OK]ボタンをクリックします。

#### (6) 逆アセンブリ表示時のメモリアクセス範囲を最小限に抑える

このオプションの設定により、逆アセンブリ表示で表示範囲以外のメモリからデータのリードを無効にします。

#### 逆アセンブリ表示時のメモリアクセス範囲を最小限に抑えるには

1. [デバッグ -> デバッグの設定]を選択します。[デバッグの設定]ダイアログボックスが開きます。
2. [オプション]タブを選択します。
3. [逆アセンブリ表示時のメモリアクセス範囲を最小限に抑える]チェックボックスをオンにします。このチェックボックスのデフォルトはターゲットに依存します。
4. [OK]ボタンをクリックします。

#### (7) 自動的にターゲットを接続しない

このオプションの設定により、[デバッグ -> 接続]を選択するまでターゲットに接続しません。機能のサポートはデバッグに依存します。

#### 自動的にターゲットを接続しないようにするには

1. [デバッグ -> デバッグの設定]を選択します。[デバッグの設定]ダイアログボックスが開きます。
2. [オプション]タブを選択します。
3. [自動的にターゲットを接続しない]チェックボックスをオンにします。選択したターゲットでこの機能をサポートしている場合、このチェックボックスは有効です。新しいターゲットを選択した場合、ターゲットのデフォルト設定でこのオプションをリセットします。このチェックボックスのデフォルトはターゲットに依存します。
4. [OK]ボタンをクリックします。

#### (8) ダウンロード後に CPU リセットする

このオプションの設定により、ユーザプログラムのダウンロード完了後に自動的にターゲットをリセットします。機能のサポートはデバッグに依存します。

### ダウンロード後に CPU リセットするには

1. [デバッグ -> デバッグの設定]を選択します。[デバッグの設定]ダイアログボックスが開きます。
2. [オプション]タブを選択します。
3. [ダウンロード後に CPU リセット]チェックボックスをオンにします。選択したターゲットでこの機能をサポートしている場合、このチェックボックスは有効です。新しいターゲットを選択した場合、ターゲットのデフォルト設定でこのオプションをリセットします。このチェックボックスのデフォルトはターゲットに依存します。
4. [OK]ボタンをクリックします。

### (9) ターゲット実行中の GUI 操作によるメモリアクセスを無視する

このオプションの設定により、プログラム実行中に High-performance Embedded Workshop からのメモリアクセスを制限できます。これは、メモリの読み書きや実行低下によりターゲットに負荷がかかることを抑止します。機能のサポートはデバッガに依存します。

### ターゲット実行中の GUI 操作によるメモリアクセスを無視するには

1. [デバッグ -> デバッグの設定]を選択します。[デバッグの設定]ダイアログボックスが開きます。
2. [オプション]タブを選択します。
3. [ターゲット実行中の GUI 操作によるメモリアクセスを無視]チェックボックスをオンにします。選択したターゲットでこの機能をサポートしている場合、このチェックボックスは有効です。新しいターゲットを選択した場合、ターゲットのデフォルト設定でこのオプションをリセットします。このチェックボックスのデフォルトはターゲットに依存します。
4. [OK]ボタンをクリックします。

この機能は、メモリアクセスを完全に抑止するわけではありません。

- メモリアクセスを抑止する機能

以下の機能では、メモリ関連の操作が無効になるか、操作してもメモリを読み書きしません。表示する値はすべてダミーの"FF"です。

- エディタウィンドウまたは逆アセンブリウィンドウ（ソースモード）のツールチップウォッチの表示
- エディタウィンドウまたは逆アセンブリウィンドウ（混合モード/逆アセンブリモード）の逆アセンブリコード表示または編集
- メモリウィンドウ、[IO]ウィンドウ、または MR ウィンドウのメモリ内容の表示、または編集\*

- メモリアクセスを抑止しない機能

以下の機能ではメモリアクセスを抑止しません。

- コマンドファイルまたはコマンドラインウィンドウのコマンド実行
- リアルタイム更新時におけるウォッチウィンドウのメモリ内容の表示、または編集\*
- モニタウィンドウ、画像ウィンドウ、または波形ウィンドウのリアルタイム更新時におけるメモリ内容の表示\*
- C ウォッチウィンドウまたは ASM ウォッチウィンドウのリアルタイム RAM モニタ機能を使用したメモリ内容の表示または編集\*
- RAM モニタウィンドウまたは MR ウィンドウのリアルタイム RAM モニタ機能を使用したメモリ内容の表示\*

- C ウォッチウィンドウまたは ASM ウォッチウィンドウのメモリ内容の表示、または編集 \*
- GUI/O ウィンドウ、画像ウィンドウ、または波形ウィンドウのメモリ内容の表示 \*

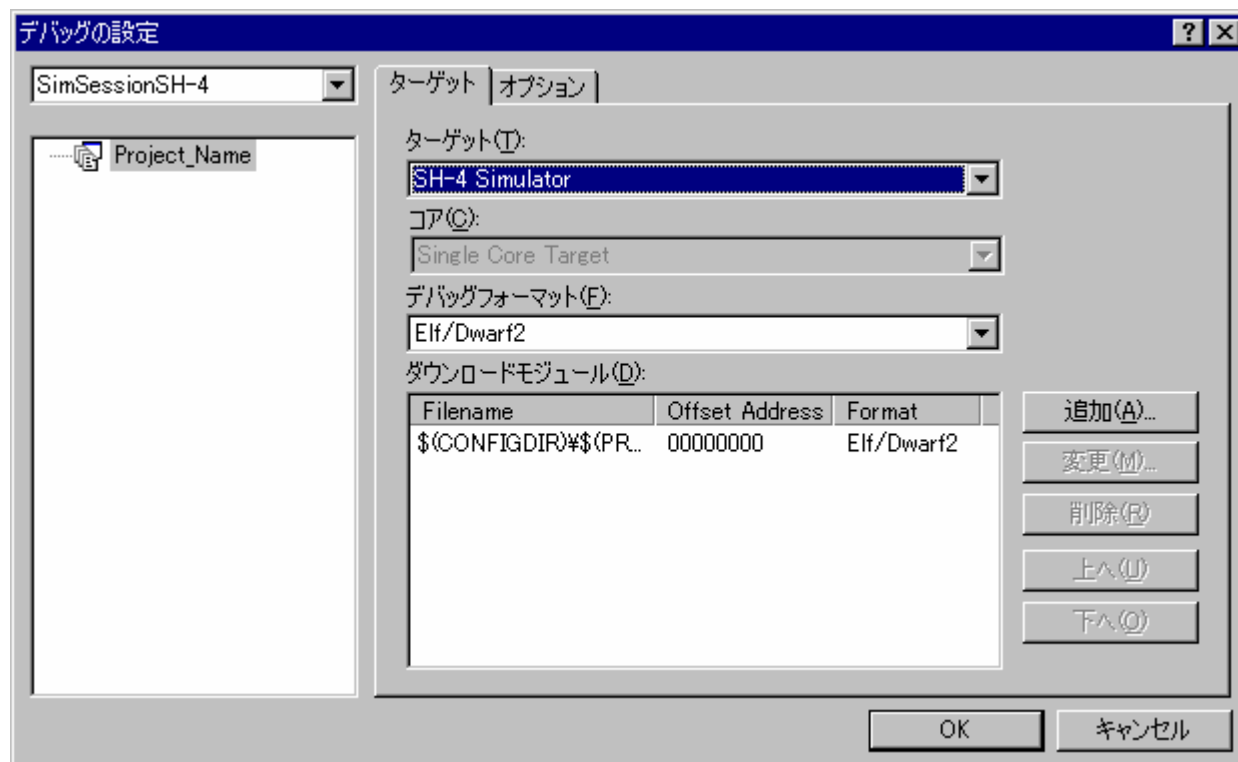
注：

\*. 機能のサポートはデバッガに依存します。

## 17.1.5 ダウンロードモジュール

### 17.1.5.1 ダウンロードモジュールを指定する

プログラムをダウンロードするためのメモリがシステムに十分ある場合、デバッグするプログラムをダウンロードできます。[Application]ジェネレータは、リンカからの出力のため、ダウンロードモジュールの初期選択を自動的に行います。デバッグ専用プロジェクトジェネレータは、ダウンロードモジュールを初期選択しませんが参照は可能です。概要については、「17.1.2 デバッガを選択する」を参照してください。



プロジェクトの作成後もダウンロードモジュールを手動で選択できます。これは、[デバッグの設定]ダイアログボックスを使用して行うことができます。このダイアログボックスでは、ワークスペース全体のデバッグの設定を制御できます。ダイアログボックスの左側にあるツリーには、現在のプロジェクトをすべて含んでいます。このツリーでプロジェクトを選択すると、そのプロジェクトの設定の表示、およびセッションドロップダウンリストには、セッションの選択を表示します。このリストボックスでは、複数のセッションの選択、またはすべてのセッションの選択もできます。複数のセッションを選択した場合には、1つまたは複数のセッションの設定を同時に修正できます。[デバッグの設定]ダイアログボックスは、以下のデバッグオプションを表示します。

- 現在のプロジェクトのための現在のデバッグターゲットおよびセッションの選択
- 現在のプロジェクトのためのダウンロードモジュールおよびセッションの選択



[デバッグフォーマット]リストでデバッグフォーマットを選択します。ここで指定したフォーマットと一致するモジュールのみデバッグできます。[ダウンロードモジュール]で複数のフォーマットが指定されている場合も、このリストで選択したフォーマットを使用します。

[ダウンロードモジュール]リストは、ターゲットにダウンロードするファイルを順番に表示します。このリストでは、モジュールの追加、変更、削除、上へ移動、下へ移動が可能です。

### 新規のダウンロードモジュールを追加するには

- 以下のいずれかの操作を選択し[デバッグの設定]ダイアログボックスを開きます。
  - [デバッグ -> デバッグの設定]を選択する。
  - ワークスペースウィンドウの[Projects]タブの[Download modules]フォルダまたはダウンロードモジュール上で右クリックする。表示されるポップアップメニューから[デバッグの設定]を選択する。
- プロジェクトのツリーで、ダウンロードモジュールを追加するプロジェクトおよびセッションを選択します。
- [追加]ボタンをクリックしてください。[ダウンロードモジュール]ダイアログボックスを開きます。すべてのフィールドは、ダウンロードモジュールが正確に構成されるようにセットアップしてください。



- [オフセット]には、モジュールがロードされるメモリアドレスのオフセットを指定してください。デフォルトは0です。モジュールの追加後、[デバッグ -> ビルド]のサブメニュー、ワークスペースウィンドウの[Projects]タブの[Download modules]フォルダ内、および[ファイル -> 最近ダウンロードしたモジュール]のサブメニューで、モジュール名の右横にオフセット値が表示されます。
- [ファイルフォーマット]には、サポートしているオブジェクトフォーマットのリストを表示します。ここではダウンロードするモジュールのファイルフォーマットを選択します。したがって、[デバッグの設定]ダイアログボックスの[デバッグフォーマット]と必ずしも一致させる必要はありません。ただし、スタックトレースウィンドウなどは[デバッグフォーマット]で指定したフォーマットに依存して提供されます。
- [ファイル名]は、プレースホルダ、または絶対パス形式として指定できます。プレースホルダを使用することをお勧めします。
- [アクセスサイズ]は、メモリアクセス時のアクセスサイズを指定します。特定のアクセスサイズでメモリにアクセスする必要がある場合、このオプションを使用してアクセスサイズを指定してください。
- [デバッグ情報のみのダウンロード]チェックボックスは、デバッグ情報だけをダウンロードする場合に使用します。
- [ダウンロード時のメモリベリファイ]チェックボックスは、モジュールをダウンロードしたとき、ターゲットデバイスへ正確にダウンロードされたことを検証するために使用します。



10. [ターゲット接続時にダウンロード]チェックボックスは、ターゲットに接続したときに自動的にダウンロードを行う場合に使用します。
11. [OK]ボタンをクリックすると、[ダウンロードモジュール]リストの最後にモジュールが追加されます。

ダウンロードモジュールの追加と同時にターゲットにダウンロードするには、「17.1.5.2 モジュールをダウンロードする」を参照してください。

#### ダウンロードモジュールの設定を変更するには

1. 以下のいずれかの操作を選択し[ダウンロードモジュール]ダイアログボックスを開きます。
  - [デバッグの設定]ダイアログボックスを表示し、[ダウンロードモジュール]リストから設定を変更したいモジュールを選択し[変更]ボタンをクリックする。
  - ワークスペースウィンドウの[Projects]タブの[Download modules]フォルダ内のダウンロードモジュール上で右クリックする。表示されるポップアップメニューから[プロパティ]を選択する。
2. 設定を変更後、[OK]ボタンをクリックします。

#### ダウンロードモジュールを削除するには

1. [デバッグの設定]ダイアログボックスを開きます。
2. [ダウンロードモジュール]リストから削除したいモジュールを選択し、[削除]ボタンをクリックします。

#### ワークスペースウィンドウの[Projects]タブで選択したダウンロードモジュールを削除するには

1. ワークスペースウィンドウの[Projects]タブで削除したいダウンロードモジュールを選択してください。複数のダウンロードモジュールを選択するときは"Shift"キーまたは"Ctrl"キーを押してください。
2. 以下のいずれかの操作を選択してください。
  - 右クリックしポップアップメニューから[削除]を選択する。
  - "Delete"キーを押す。
3. ダウンロードモジュールをプロジェクトから削除するかを確認するダイアログボックスが開きます。ダウンロードモジュールをプロジェクトから削除する場合は「はい」を選択してください。削除しない場合は「いいえ」または「キャンセル」を選択してください。

確認要求のダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。確認要求のダイアログボックスを再び表示するには、[基本設定 -> オプション]を選択し[オプション]ダイアログボックスを開きます。[確認]タブの[プロジェクトからダウンロードモジュールを削除]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。

#### モジュールをターゲットにダウンロードする順番を変更するには

1. [デバッグの設定]ダイアログボックスを開きます。
2. [ダウンロードモジュール]リストで、ダウンロードする順番を変更したいモジュールを選択し、[上へ]または[下へ]ボタンをクリックします。

[デバッグの設定]ダイアログボックスで行った変更はすべて、[OK]ボタンをクリックするまで更新しません。

### 17.1.5.2 モジュールをダウンロードする

デバッグするためにオブジェクトプログラムをダウンロードします。

#### モジュールをダウンロードするには

以下のいずれかの操作を選択してください。

- [デバッグ -> ダウンロード]のサブメニューからダウンロードしたいモジュールを選択する。
- ワークスペースウィンドウの[Projects]タブの[Download modules]フォルダ内でダウンロードしたいモジュールをダブルクリックする。
- ワークスペースウィンドウの[Projects]タブの[Download modules]フォルダ内のダウンロードしたいモジュール上で右クリックする。表示されるポップアップメニューから[ダウンロード]または[ダウンロード(debug 情報のみ)]を選択する。この場合モジュールは複数選択できます。

High-performance Embedded Workshop では、最近ダウンロードしたモジュールを[ファイル -> 最近ダウンロードしたモジュール]のサブメニューに表示します。最近使ったモジュールを再びダウンロードしたいときに使用してください。

#### すべてのモジュールをダウンロードするには

以下のいずれかの操作を選択してください。

- [デバッグ -> ダウンロード -> All Download Modules]を選択してください。
- ワークスペースウィンドウの[Projects]タブの[Download modules]フォルダ上で右クリックする。表示されるポップアップメニューから[全てダウンロード]を選択する。

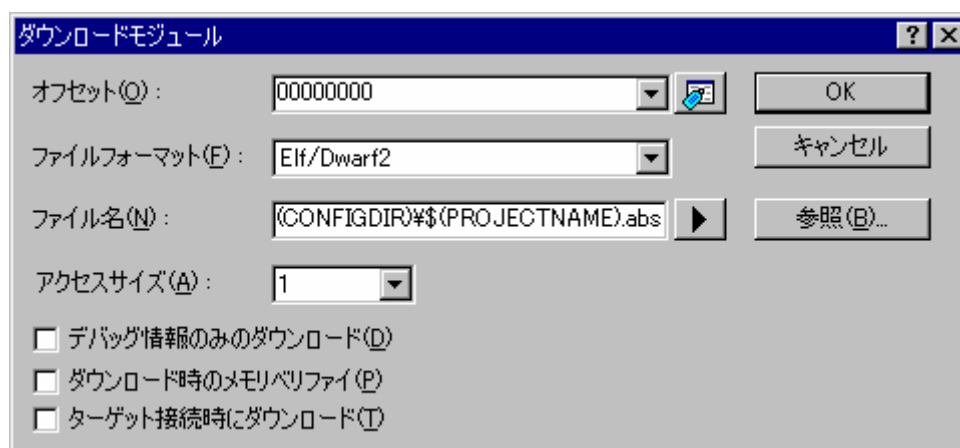
[デバッグ -> デバッグの設定]を選択し、[デバッグの設定]ダイアログボックスを開きます。[ターゲット]タブの[ダウンロードモジュール]で指定した順番でターゲットにダウンロードされます。

モジュールは存在してもダウンロードモジュールリストに追加されていない場合、ダウンロードモジュールリストへの追加と同時にターゲットにダウンロードできます。

#### ダウンロードモジュールリストへの追加と同時にターゲットにダウンロードするには

1. 以下のいずれかの操作を選択し[ダウンロードモジュール]ダイアログボックスを開きます。
  - [ファイル -> ダウンロードモジュールの追加]を選択する。
  - ワークスペースウィンドウの[Projects]タブの[Download modules]フォルダまたはダウンロードモジュール上で右クリックする。表示されるポップアップメニューから[ダウンロードモジュールの追加]を選択する。

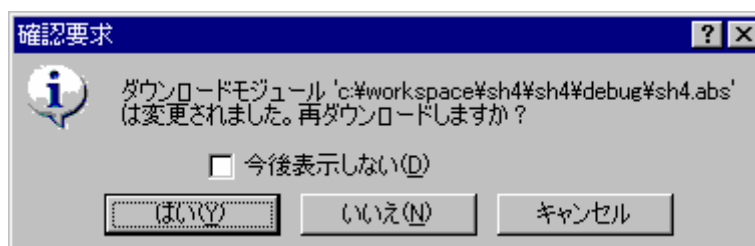
すべてのフィールドは、ダウンロードモジュールが正確に構成されるようにセットアップしてください。



2. [オフセット]には、モジュールがロードされるメモリアドレスのオフセットを指定してください。デフォルトは0です。モジュールの追加後、[デバッグ -> ビルド]のサブメニュー、ワークスペースウィンドウの[Projects]タブの[Download modules]フォルダ内、および[ファイル -> 最近ダウンロードしたモジュール]のサブメニューで、モジュール名の右横にオフセット値が表示されます。
3. [ファイルフォーマット]には、サポートしているオブジェクトフォーマットのリストを表示します。ここではダウンロードするモジュールのファイルフォーマットを選択します。したがって、[デバッグの設定]ダイアログボックスの[デバッグフォーマット]と必ずしも一致させる必要はありません。ただし、スタックトレースウィンドウなどは[デバッグフォーマット]で指定したフォーマットに依存して提供されます。
4. [ファイル名]は、プレースホルダ、または絶対パス形式として指定できます。プレースホルダを使用することをお勧めします。
5. [アクセスサイズ]は、メモリアクセス時のアクセスサイズを指定します。特定のアクセスサイズでメモリにアクセスする必要がある場合、このオプションを使用してアクセスサイズを指定してください。
6. [デバッグ情報のみのダウンロード]チェックボックスは、デバッグ情報のみをダウンロードする場合に使用します。
7. [ダウンロード時のメモリベリファイ]チェックボックスは、モジュールをダウンロードしたとき、ターゲットデバイスへ正確にダウンロードされたことを検証するために使用します。
8. [ターゲット接続時にダウンロード]チェックボックスは、ターゲットに接続したときに自動的にダウンロードを行う場合に使用します。
9. [OK]ボタンをクリックします。

[デバッグ -> デバッグの設定]を選択し、[デバッグの設定]ダイアログボックスを開きます。[ターゲット]タブの[ダウンロードモジュール]リストの最後にモジュールが追加されています。

また、既にダウンロードしたモジュールが High-performance Embedded Workshop の外部で変更された場合、モジュールを再びダウンロードするかを確認するダイアログボックスが開きます。モジュールを再びダウンロードする場合は「はい」を選択してください。ダウンロードしない場合は「いいえ」または「キャンセル」を選択してください。



確認要求ダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。

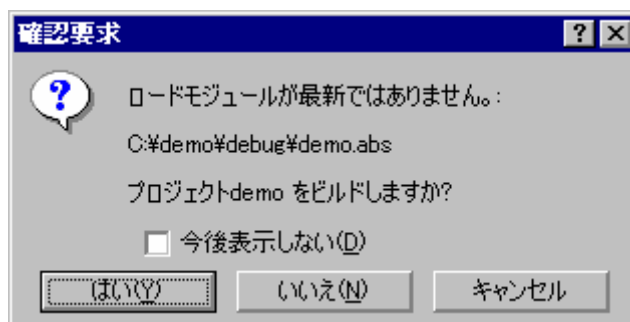
#### 確認要求ダイアログボックスを再び表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスが開きます。
2. [確認]タブを選択してください。
3. [古いダウンロードモジュールの再ダウンロード]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. OK をクリックしてください。

#### 17.1.5.3 ダウンロード前に変更したソースファイルを確認する

High-performance Embedded Workshop は、現在のプロジェクトのダウンロードモジュールをダウンロードする前にソースファイルが変更されているかどうかを確認します。

ファイルが変更されていた場合、確認のダイアログボックスが起動され、ダウンロード実行前にコードを再度ビルドするかどうかを確認するダイアログボックスが開きます。ダウンロード実行前にコードを再度ビルドする場合は「はい」を選択してください。ダウンロード実行前にコードを再度ビルドしない場合は「いいえ」を選択してください。



確認要求ダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。

#### 確認要求ダイアログボックスを再び表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスが開きます。
2. [確認]タブを選択してください。
3. [ダウンロードモジュールの更新]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. OK をクリックしてください。

#### 17.1.5.4 ダウンロード後にソースツリーを表示する（デバッグ専用プロジェクト）

デバッグ専用プロジェクト（[Debugger only - xxxxxx]）を使用している場合、ダウンロードするとワークスペースウィンドウの[Projects]タブ内にダウンロードモジュールを構成するソースファイルをツリーで表示します。

High-performance Embedded Workshop はダウンロード時、ダウンロードモジュールを構成するソースファイルの情報を取得します。

相対パス名、またはファイル名の場合は、ダウンロードモジュールのディレクトリを基準にファイルを検索します。

この初期検索で見つからなかったファイルは[ファイルの検索]ダイアログボックス内に表示します。ダウンロード時、[ファイルの検索]ダイアログボックスを表示しないようにできます。

#### ダウンロード時、[ファイルの検索]ダイアログボックスを表示しないようにするには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [確認]タブを選択してください。
3. [ディレクトリ参照ダイアログボックスをダウンロード時に表示しない]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオフです。
4. [OK]をクリックしてください。

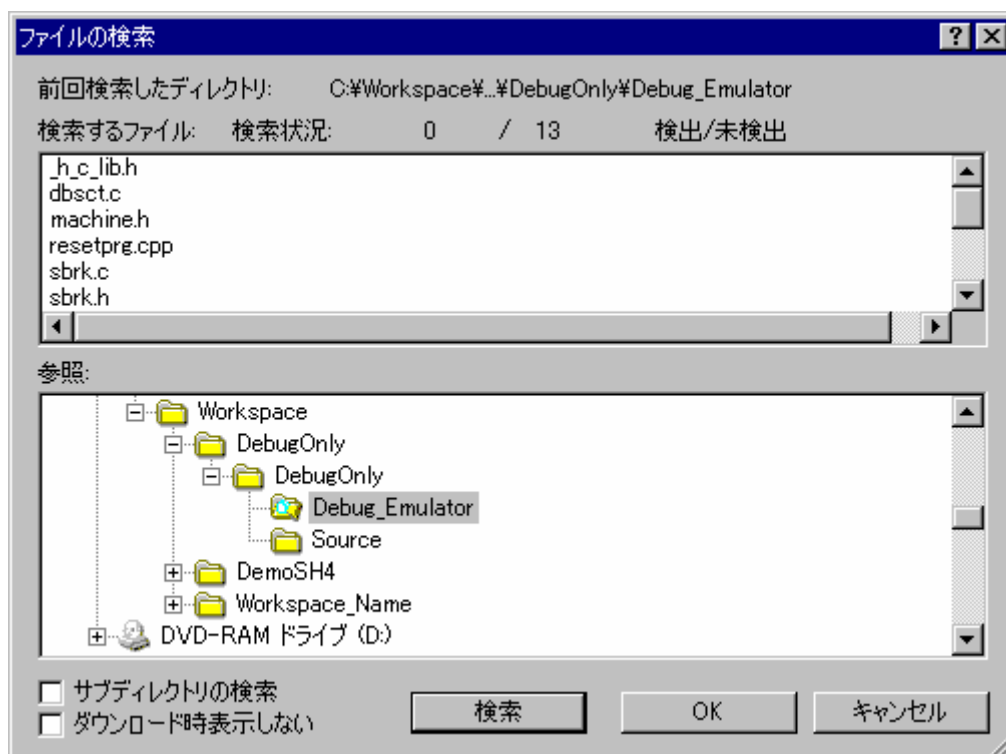
#### • 対応デバッガ

デバッグ専用プロジェクトを作成できるデバッガがダウンロード後にソースツリーを表示する機能にサポートしています。デバッガによりサポートする条件が異なります。


デバッガ	サポート条件
SuperH ファミリー用エミュレータデバッガ	High-performance Embedded Workshop のアップデートによって、High-performance Embedded Workshop のバージョンが V.4.02 以降になった。
H8SX, H8S, H8 ファミリー用エミュレータデバッガ	
M16C ファミリー用エミュレータデバッガ	製品パッケージに同梱される High-performance Embedded Workshop のバージョンが V.4.02 以降である。

#### (1) ファイルを検索する

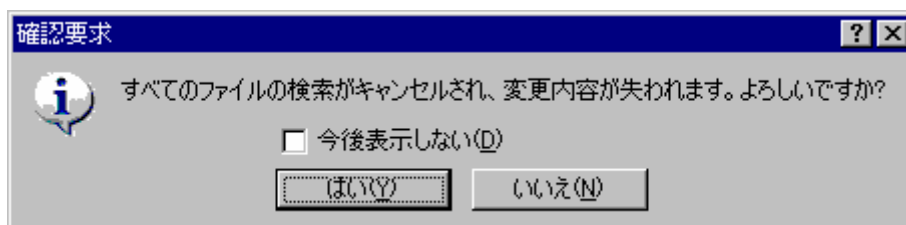
初期検索で見つからなかったソースファイルは、[ファイルの検索]ダイアログボックス内 ([検索するファイル]リスト) に表示します。



## ファイルを検索するには

1. [前回検索したディレクトリ]は、前回ソースファイルを検索したディレクトリを表示します。
2. [検索状況]は、見つからなかったファイル数（[未検出]）に対して見つかったファイル数（[検出]）を表示します。ダイアログボックスに表示されているファイルの数で、前回検索した時の数は含みません。
3. [検索するファイル]リストは、見つからなかったすべてのファイルのリストを表示します。リストにはファイル名のみを表示し、このファイル名を元に検索します。
4. [参照]は、ファイルを検索するためのディレクトリを選択します。
5. [サブディレクトリの検索]チェックボックスをオンにすると、選択されたディレクトリのすべてのサブディレクトリでファイルを検索します。このチェックボックスのデフォルトはオフです。
6. [ダウンロード時表示しない]チェックボックスをオンにすると、ダウンロード時に見つからなかったファイルがある場合もこのダイアログボックスを開きません。このチェックボックスのデフォルトはオフです。このチェックボックスは、[オプション]ダイアログボックスの[確認]タブにもあります。
7. [検索]ボタンをクリックすると、[参照]で選択したディレクトリにあるファイルを検索し、見つかったファイルは[検索するファイル]リストから消去されます。  
[参照]で検索済みのディレクトリは、フォルダと虫眼鏡のアイコン()で表示されます。  
[検索]ボタンをクリックすると、検索を取り消せる[中止]ボタンに変わります。[中止]ボタンをクリックすると、検索を取り消す前に見つかったファイルは、場所が特定されない状態に戻ります。
8. [OK]をクリックすると、検索して見つかったファイルをそのディレクトリに特定します。すべてのファイルの検索が保存され、ダイアログボックスを閉じます。
9. [キャンセル]をクリックすると、すべてのファイルの検索が取り消され、ダイアログボックスを閉じます。

ファイルを検索した後で[キャンセル]を選択すると、下記の確認要求ダイアログボックスが開きます。



「はい」を選択すると、[ファイルの検索]ダイアログボックスを閉じます。「いいえ」を選択すると、[ファイルの検索]ダイアログボックスは閉じません。

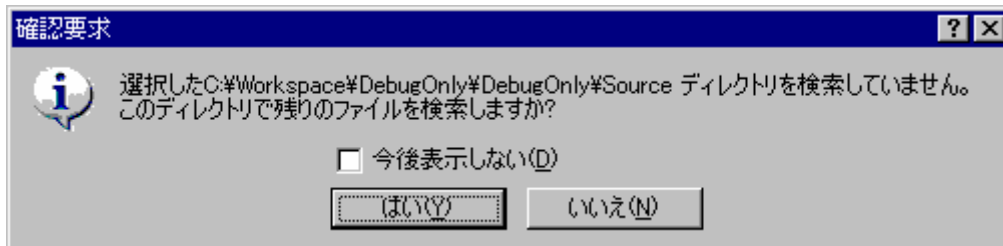
確認要求ダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。[今後表示しない]を選択すると、「はい」を選択したときの動作がデフォルトです。

## 確認要求ダイアログボックスを再び表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスが開きます。
2. [確認]タブを選択してください。
3. [ディレクトリ参照ダイアログでキャンセルした時の確認]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. [OK]をクリックしてください。



ディレクトリを選択した後で[検索]を選択せずに[OK]を選択すると、下記の確認要求ダイアログボックスが開きます。



「はい」を選択すると、ファイルを検索し、[ファイルの検索]ダイアログボックスは閉じません。必要に応じて検索は中止できます。

「いいえ」を選択すると、メッセージを表示し、[ファイルの検索]ダイアログボックスを閉じます。

確認要求ダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。[今後表示しない]を選択すると、「いいえ」を選択したときの動作がデフォルトです。

#### 確認要求ダイアログボックスを再び表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスが開きます。
2. [確認]タブを選択してください。
3. [ディレクトリ参照ダイアログで OK した時の検索の確認]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. [OK]をクリックしてください。

#### (2) ソースツリーを表示する

ダウンロードモジュールから取得したソースファイル名は、ワークスペースウィンドウのダウンロードモジュール下に表示します。

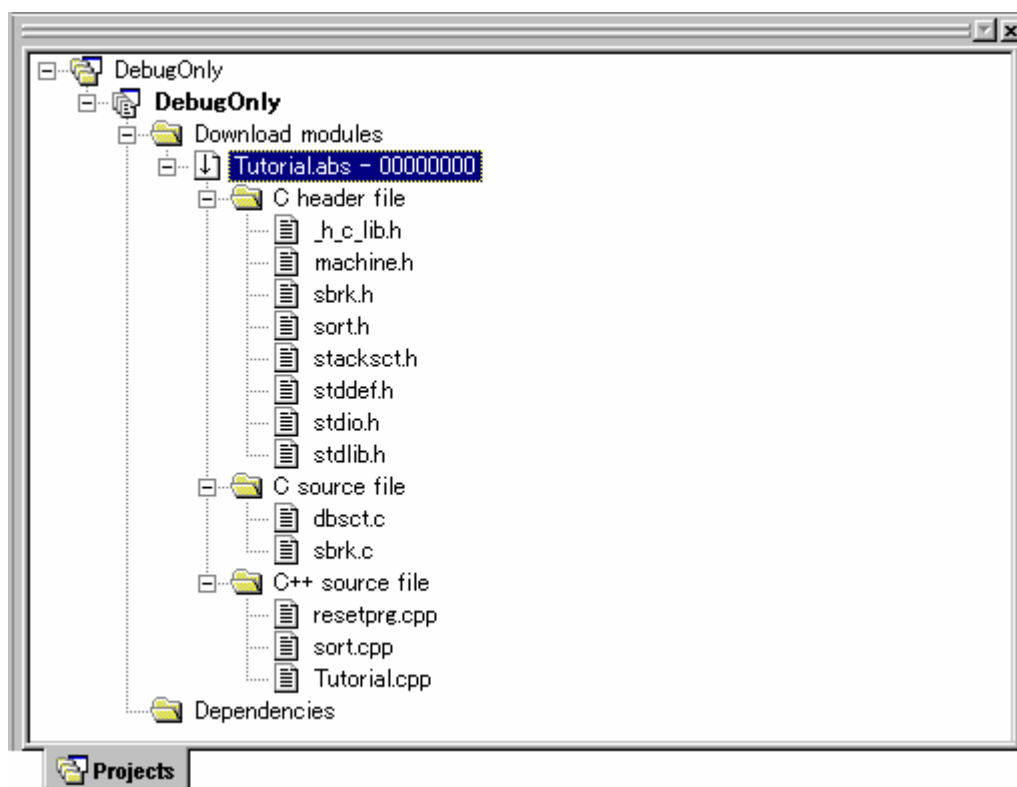
ソースファイルは、ファイルの種類でグループ分けし関連したフォルダ下に表示します。


ファイルの種類が定義されていない場合、ファイルの拡張子フォルダ下にグループ分けします。

この場合、ファイルをダブルクリックすると外部エディタでファイルを開きます。エディタウィンドウで開くには、「2.5 ファイル拡張子とファイルグループ」を参照してください。

見つかったソースファイルは、ダウンロードモジュール下に通常のファイルアイコン(目)で表示します。

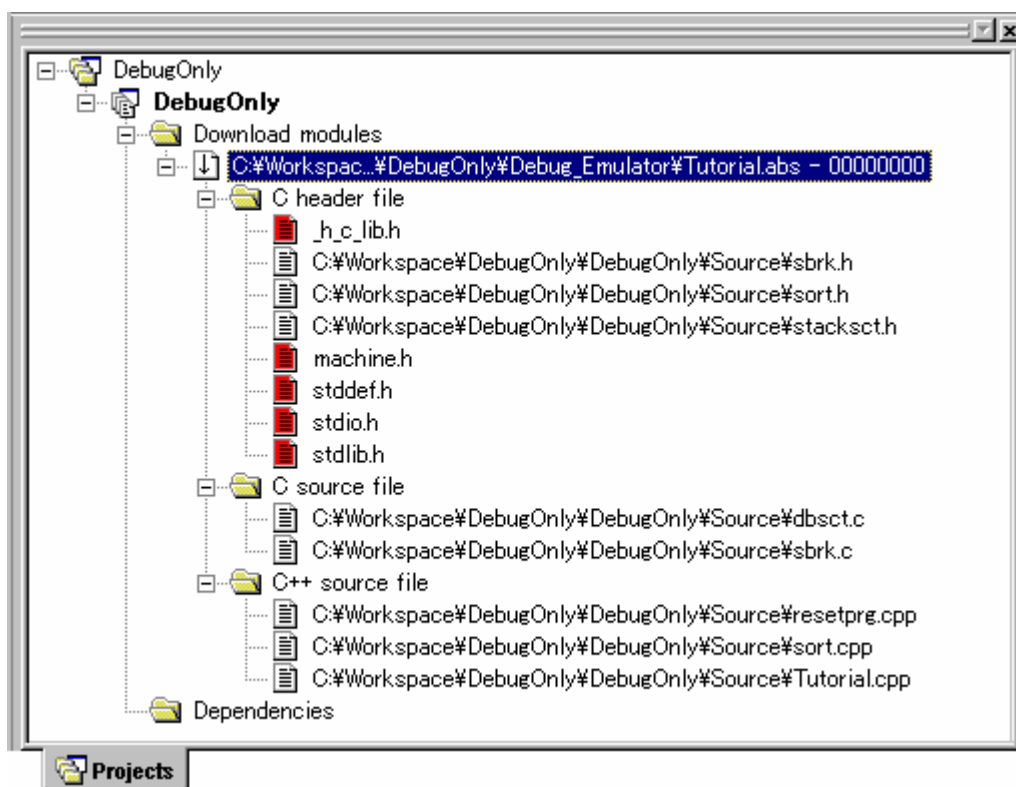




見つからなかったソースファイルは、ダウンロードモジュール下に赤いファイルアイコン(  )で表示します。これらのファイルを区別するためにファイルを絶対パスで表示できます。

#### すべてのファイルを絶対パスで表示するには

1. ワークスペースウィンドウの[Projects]タブ内で右クリックします。
2. [表示の構成]を選択してください。[表示の構成]ダイアログボックスを開きます。
3. [ファイルパスの表示]チェックボックスをオンにしてください。
4. [OK]をクリックしてください。



ポップアップメニューからファイルを別のディレクトリで再検索することができます。

### (3) ファイルを別のディレクトリで再検索する

ダウンロードモジュール内のすべてのソースファイルを別のディレクトリで再検索するには、ダウンロードモジュールのポップアップメニューから行ってください。

#### ダウンロードモジュール内のすべてのソースファイルを別のディレクトリで再検索するには

1. ワークスペースウィンドウの[Projects]タブ内のダウンロードモジュール上で右クリックします。
2. [モジュールの検索]を選択します。Windows® の標準的なディレクトリを開くダイアログボックスを開きます。



3. 検索するディレクトリを選択し、[開く]ボタンをクリックします。

選択したディレクトリで見つからなかったソースファイルは、[ファイルの検索]ダイアログボックスを表示し、これらのファイルを検索できます。

また、ソースファイルを別のディレクトリで再検索するには、ファイルのポップアップメニューから行ってください。

#### ファイルを別のディレクトリで再検索するには

1. ワークスペースウィンドウの[Projects]タブ内のファイル上で右クリックします。
2. [ファイルの検索]を選択します。Windows® の標準的なファイルを開くダイアログボックスを開きます。



3. 同じ名前のファイルだけを選択できます。
4. ファイルを選択し、[開く]ボタンをクリックします。

複数のファイルを選択することもできます。その場合は、[ファイルの検索]ダイアログボックスを開きます。

#### 17.1.5.5 ダウンロード後に main 関数を表示する

モジュールのダウンロード後に main 関数を含むソースファイルを自動的に High-performance Embedded Workshop エディタのソースモードで表示する機能があります。

#### ダウンロード後に main 関数を表示するには

1. [基本設定 -> オプション]を選択し、[オプション]ダイアログボックスを開きます。
2. [デバッグ]タブを選択します。
3. [ダウンロード後に main 関数を表示]チェックボックスをオンにします。デフォルトはオフです。
4. OK をクリックします。
5. モジュールをダウンロードします。ダウンロードについては、「17.1.5.2 モジュールをダウンロードする」を参照してください。

### 17.1.5.6 モジュールをアンロードする

ダウンロードしたモジュールを手動でアンロードします。

モジュールをアンロードすると、そのモジュールのシンボルは、High-performance Embedded Workshop デバッグシステムからなくなりますが、ターゲットのメモリ内容は変更しません。一度アンロードしたモジュールは、再度ロードするまでデバッグすることはできません。

#### モジュールをアンロードするには

以下のいずれかの操作を選択してください。

- [デバッグ -> アンロード]のサブメニューからアンロードしたいモジュールを選択する。
- ワークスペースウィンドウの[Projects]タブの[Download modules]フォルダ内のアンロードしたいモジュールを選択し、右クリックしポップアップメニューを表示してください。ポップアップメニューから[アンロード]を選択する。この場合モジュールは複数選択できます。

#### すべてのモジュールをアンロードするには

- [デバッグ -> アンロード -> All Download Modules]を選択する。

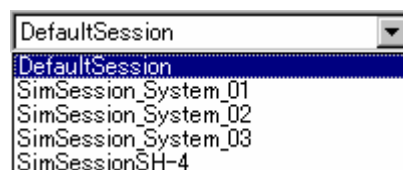
[デバッグ -> デバッグの設定]を選択し、[デバッグの設定]ダイアログボックスを開きます。[ターゲット]タブの[ダウンロードモジュール]リストで指定した順番でターゲットからアンロードされます。

### 17.1.6 デバッグセッション

High-performance Embedded Workshop は、ビルドオプションをコンフィグレーションへ保存できます。同様に、High-performance Embedded Workshop は、デバッグオプションをセッションに保存することもできます。

セッションには、デバッガ、ダウンロードするプログラム、各デバッガのオプションを保存できます。これは、各セッションが異なるデバッガをターゲットにできることを意味します。

この機能は異なるデバッグオプションを持つ多くの異なるセッションを許します。例えば、各セッションは同じターゲットで少し違うオプション設定にできます。これはレジスタ値やクロックスピードなどのターゲット設定を、セッションで簡単に変更したり切り替えたりできることを意味します。下図はこの典型例です。5つのセッションがターゲットを共有しています。しかしセッションでは異なるオプション定義が可能です。



セッションは、コンフィグレーションとは直接関連がありません。これは、複数のセッションが同じダウンロードモジュールを共有し、プログラムの不要なリビルドを避けられることを意味します。

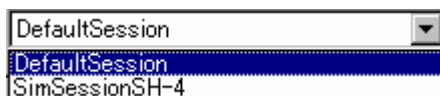
各セッションのデータは、別々のファイルで High-performance Embedded Workshop プロジェクトに保存します。プロジェクト内でデータを共有したり、必要に応じて変更したりできます。

#### 17.1.6.1 セッションを選択する

##### セッションを選択するには

1. [デバッグ -> デバッグセッション]を選択してください。[デバッグセッション]ダイアログボックスを開きます。

2. [現在のセッション]ドロップダウンリストから使用するセッションを選択してください。
3. [OK]ボタンをクリックしてください。



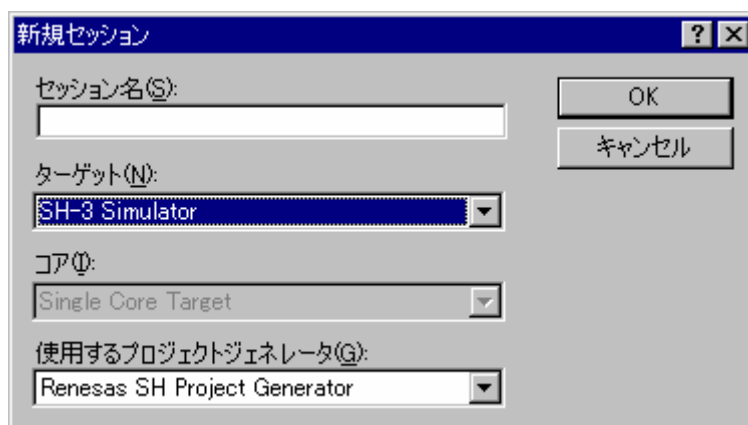
また、[標準]ツールバーのドロップダウンリストから異なるセッションを選択することもできます。

### 17.1.6.2 セッションを追加する

ターゲットを取り付けてセットアップし、セッションを作成できます。セッションには名前を付け、ターゲットを選択できます。

#### ターゲットを取り付けてセットアップし、新しいセッションを作成するには

1. [ファイル -> 新規セッション]を選択してください。[新規セッション]ダイアログボックスが開きます。



2. [セッション]に新しいセッションの名前を入力してください。セッション名は、32文字以内で、半角英数字、半角下線のみ使用してください。特に、日本語文字、空白、または半角マイナス記号は使用しないでください。
3. [ターゲット]で使用するターゲットを選択してください。
4. [コア]で使用するコアを選択してください。同期デバッグ機能を使用している場合に選択できます。
5. [使用するプロジェクトジェネレータ]でジェネレータを選択してください。デフォルトで選択されているものが適切ですが、複数のジェネレータが同じターゲットをサポートしている場合もあります。
6. [OK]ボタンをクリックしてください。生成プロセスを開始します。このプロセスは、選択したジェネレータに依存します。このとき、さらにターゲット設定オプションのダイアログボックスを表示することがあります。
7. 設定を完了すると、現在のプロジェクトに新しいセッションを追加します。追加されたセッションは、メインツールバーの[セッション]ドロップダウンリストボックスの中にあります。

新しいセッションを追加できます。入力したセッション名のファイルを新しく作成します。ファイルが既に存在する場合はエラーを表示します。

### 新しい空のセッションを追加するには

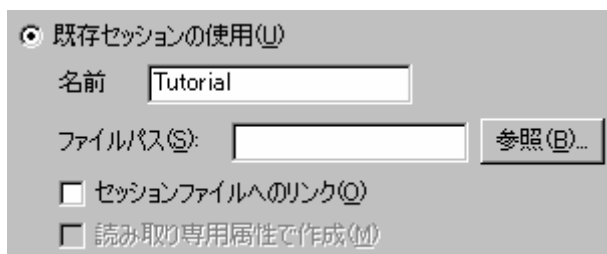
1. [デバッグ -> デバッグセッション]を選択してください。[デバッグセッション]ダイアログボックスが開きます。
2. [追加]ボタンをクリックしてください。[新規セッション追加]ダイアログボックスが開きます。
3. [新規セッションの追加]ラジオボタンをチェックしてください。
4. セッションの名前を入力してください。
5. [OK]ボタンをクリックしてください。



別のファイルからセッションデータをインポートして新しいセッションを作成できます。インポート元ファイルのデータはそのまますべてコピーします。

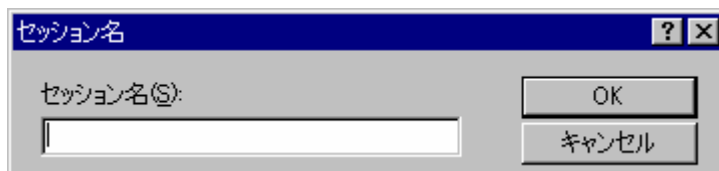
### 既存のセッションを新しいセッションファイルにインポートするには

1. [デバッグ -> デバッグセッション]を選択してください。[デバッグセッション]ダイアログボックスが開きます。
2. [追加]ボタンをクリックしてください。[新規セッション追加]ダイアログボックスが開きます。
3. [既存セッションの使用]ラジオボタンをチェックしてください。
4. セッションの名前を入力してください。
5. 現在のプロジェクトにインポートしたい既存のセッションファイルをブラウザしてください。
6. [OK]ボタンをクリックしてください。



これと同じ操作が、[ファイル -> セッションのインポート]を使ってもできます。

1. [ファイル -> セッションのインポート]を選択してください。[セッション名]ダイアログボックスが開きます。



2. 新しいセッションの名前を入力してください。
3. 新しいセッションにインポートしたいセッションファイルを選択してください。
4. [OK]ボタンをクリックしてください。ブラウザしたファイルと同じ設定で、新しい名前の付いた新しいセッションを追加します。

### 17.1.6.3 リンクをセッションにインポートする

新しいセッションを High-performance Embedded Workshop システムに追加するとき、セッションファイルをプロジェクトディレクトリにインポートまたはコピーしたりするのではなく、その場でセッションファイルにリンクできます。この方法は、ネットワーク環境で他のユーザとデバッグ情報を共有する場合に便利です。

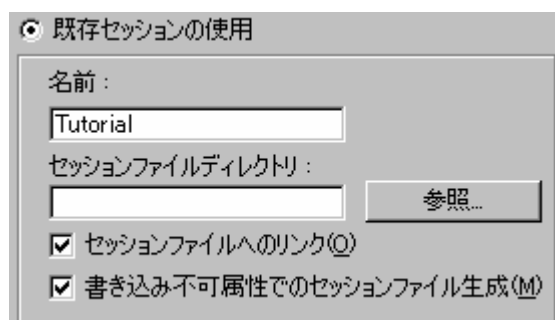
#### 既存のセッションファイルにリンクをインポートするには

1. [デバッグ -> デバッグセッション]を選択してください。[デバッグセッション]ダイアログボックスを開きます。
2. [追加]ボタンをクリックしてください。[新規セッション追加]ダイアログボックスを開きます。
3. [既存セッションの使用]ラジオボタンをチェックしてください。
4. セッションの名前を入力してください。
5. 現在のプロジェクトにインポートしたい既存のセッションファイルを参照してください。
6. [セッションファイルへのリンク]チェックボックスをオンにしてください。プロジェクトディレクトリに新しいセッションファイルを生成せず、High-performance Embedded Workshop は既存のセッションファイルにリンクします。このファイルの場所はステップ 5 で入力したもので、すべてのセッションデータをこの場所に保存します。
7. [OK]ボタンをクリックしてください。

リンクしたセッションファイルを読み取り専用で使用できます。デバッグ設定ファイルを共有していて、データを誤って変更してしまうのを防ぐ場合に便利です。

#### リンクを既存のファイルにインポートし、読み取り専用にするには

1. [デバッグ -> デバッグセッション]を選択してください。[デバッグセッション]ダイアログボックスを開きます。
2. [追加]ボタンをクリックしてください。[新規セッション追加]ダイアログボックスを開きます。
3. [既存セッションの使用]ラジオボタンをチェックしてください。
4. セッションの名前を入力してください。
5. 現在のプロジェクトにインポートしたい既存のセッションファイルを参照してください。
6. [セッションファイルへのリンク]チェックボックスをオンにしてください。
7. [書き込み不可属性でのセッションファイル生成]チェックボックスをオンにしてください。High-performance Embedded Workshop はこのセッションへの変更を保存することはできません。このセッションを開いて、データを読み取ることができます。





#### 17.1.6.4 セッションを削除する

##### セッションを削除するには

1. [デバッグ -> デバッグセッション]を選択してください。[デバッグセッション]ダイアログボックスを開きます。
2. 削除したいセッションを選択してください。
3. [削除]ボタンをクリックしてください。
4. [OK]ボタンをクリックしてください。

現在のセッションを削除することはできません。

#### 17.1.6.5 セッションを書き込み不可にする

##### セッションを書き込み不可にするには

書き込み不可(R)

1. [デバッグ -> デバッグセッション]を選択してください。[デバッグセッション]ダイアログボックスを開きます。
2. 書き込み不可にしたいセッションを選択してください。
3. [プロパティ]ボタンをクリックしてください。[セッションプロパティ]ダイアログボックスを開きます。
4. [書き込み不可]チェックボックスをオンにしてください。リンクを読み取り専用にします。これは、デバッガ設定ファイルを共有する場合、およびデータを誤って修正したくない場合に便利です。
5. [OK]ボタンをクリックしてください。

#### 17.1.6.6 セッション情報を保存する

##### セッションを保存するには

[ファイル -> セッションの保存]を選択してください。

[基本設定 -> オプション]を選択し、[オプション]ダイアログボックスの[ワークスペース]タブを表示してください。[セッション保存前に確認]チェックボックスがオンの場合、セッション保存の前に、セッションを保存するかを確認するダイアログボックスを表示させることができます。ここで、[いいえ]を選択するとセッションへの変更を保存しません。

##### セッションを別名で保存するには

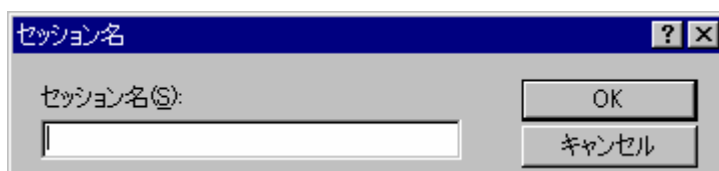
1. [デバッグ -> デバッグセッション]を選択してください。[デバッグセッション]ダイアログボックスを開きます。
2. 保存したいセッションを選択してください。
3. [名前を付けて保存]ボタンをクリックしてください。[セッションの保存]ダイアログボックスを開きます。
4. 新しいファイルの場所をブラウズしてください。
5. セッションファイルを別の場所へエクスポートしたい場合は、[プロジェクトとのリンク]チェックボックスをオンにしないでください。現在のセッションの場所の代わりにこの場所を High-performance

Embedded Workshop で使用したい場合は、[プロジェクトとのリンク]チェックボックスをオンにしてください。

6. [保存]ボタンをクリックしてください。

これと同じ操作が、[ファイル->名前を付けてセッションを保存]を使っても可能です。

1. [ファイル->名前を付けてセッションを保存]を選択してください。[セッション名]ダイアログボックスを開きます。



2. 新しいセッションの名前を入力してください。
3. [OK]ボタンをクリックしてください。

#### 17.1.6.7 セッション情報のリフレッシュ

**セッションを再び読み込むには**

[ファイル->セッションのリフレッシュ]を選択してください。

これにより現在のセッションに加えられている変更を廃棄して、セッション情報を再び読み込みます。

## 17.2 プログラムを表示する

この節では、プログラムをソースコードおよびアセンブリ言語モニターとして表示する方法を説明します。

ソースコードは、次の2種類のウィンドウにある[ソースモード]で表示します。主な特徴を以下に示します。

### エディタウィンドウ

- エディタウィンドウで表示するソースファイルは編集可能です。
- ソースコードをステップ実行などによりデバッグするとき、対応するソースファイル（現在のプログラムカウンタ(PC)のアドレスを含む）が変わると、別のウィンドウで自動的に開きます。対応するソースコード行がない場合にステップ実行すると、[ソースモード]は[逆アセンブリモード]に切り替わります。

### 逆アセンブリウィンドウ

- 逆アセンブリウィンドウで表示するソースファイルはすべて読み取り専用で編集できません。ソースファイルを編集するにはポップアップメニューからエディタウィンドウを開いて行きます。
- ソースコードをステップ実行などによりデバッグするとき、逆アセンブリウィンドウは対応するソースファイル（現在のPCアドレスを含む）に自動的に切り替わります。逆アセンブリウィンドウが開いている間、エディタウィンドウは開かずに、必要に応じて逆アセンブリウィンドウ内にソースファイルを表示します。対応するソースコード行がない場合にステップ実行すると、[ソースモード]は[逆アセンブリモード]に切り替わります。
- デバッグ中に逆アセンブリウィンドウのソースモード表示を優先する機能があります。この機能を使用した場合の動作を次に示します。

- 逆アセンブリウィンドウを[逆アセンブリモード]で開き、ソースのアドレスでブレークすると、逆アセンブリウィンドウは[ソースモード]に切り替わります。
- ソースのアドレスでブレークしても、逆アセンブリウィンドウが開いている場合はエディタウィンドウを開きません。
- プログラム実行前と後では、表示しているウィンドウの重なり順序を変更しません。
- エディタウィンドウで開いているソースファイルの表示位置を変更しません。

逆アセンブリウィンドウの[逆アセンブリモード]でステップを続ける場合は、[デバッグ->ステップモード->アセンブリ]を選択してください。

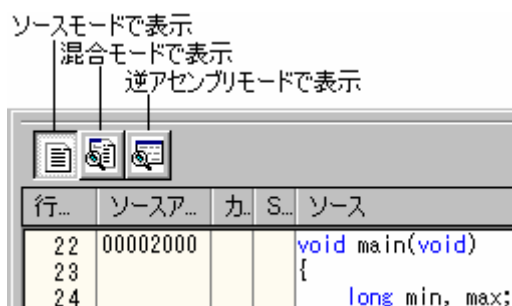
#### 注：

ブレークすると High-performance Embedded Workshop は PC 位置を表示します。プロジェクトがもともとのパスから移動した場合、ソースファイルを自動的に見つけることができない場合があります。この場合、High-performance Embedded Workshop はソースファイル参照ダイアログボックスを開くので、手動でファイルを探すことができます。このパスは、このデバッグプロジェクトの他のソースファイルを更新するために使用します。

### 17.2.1 エディタウィンドウを開く

ワークスペースウィンドウ上のソースファイルをダブルクリックするか、ソースファイル上で右ボタンをクリックしポップアップメニューを表示して[開く]を選択すると、エディタでファイルを開きます。

#### ツールバー



High-performance Embedded Workshop V.4.00 以降のエディタでは、逆アセンブリ表示を統合しています。

モードを切り替えるツールバーがあります。各モードが有効である場合、ボタンをクリックすると表示を切り替えます。

以下の3つのモードがあります。

モード	機能
ソースモード	このモードは、標準的な High-performance Embedded Workshop エディタで、ソースファイルを編集できます。また、ソースファイルを参照している場合は、キーワードを強調表示します。行番号、アドレス、ブレークポイント、ブックマーク、およびソースコードを参照できます。
混合モード	ソースファイルの混合モードは、逆アセンブリウィンドウの混合モードとは異なります。継続する逆アセンブリを示すかわりに、ソースコードの各行に関連する逆アセンブリを示します。このモードは表示を編集できません。モジュールがダウンロードされている場合のみ有効になります。行番号、ブレークポイント、アドレス、オブジェクトコード、ラベル、および混合（ソース・逆アセンブリ）を参照できます。
逆アセンブリモード	逆アセンブリモードは、実際に継続する逆アセンブリコードをアドレス順に示します。これは、[表示 -> 逆アセンブリ]を選択するのと同じです。ターゲットがセッションに接続している場合のみ有効になります。ブレークポイント、アドレス、オブジェクトコード、ラベル、および逆アセンブリコードを参照できます。

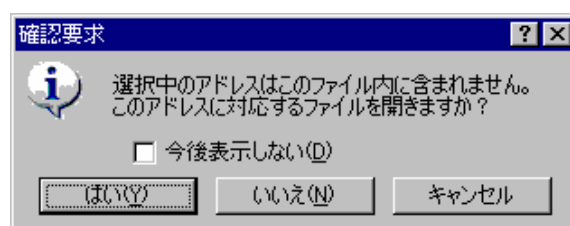
**注：**

以下の条件下で、[ソースモード]から[混合モード]へ切り替えることはできません。

1. ターゲットが現在のセッションに接続していない。
2. ダウンロードモジュールをダウンロードしていない。
3. 現在のプロジェクトに有効なデバッグ情報がない。
4. 現在表示しているファイルを編集したが、変更内容を保存していない。

[逆アセンブリモード]で、アドレスがそのソースファイルのアドレス範囲から外れ、別のソースファイルのアドレス範囲に変わった場合、[ソースモード]または[混合モード]へ切り替えると下記の確認ダイアログボックスが開きます。

アドレスに対応する別のファイルを開く場合は「はい」を選択してください。切り替える前の位置で表示するには「いいえ」を選択してください。他のモードへ切り替えない場合は、「キャンセル」を選択してください。



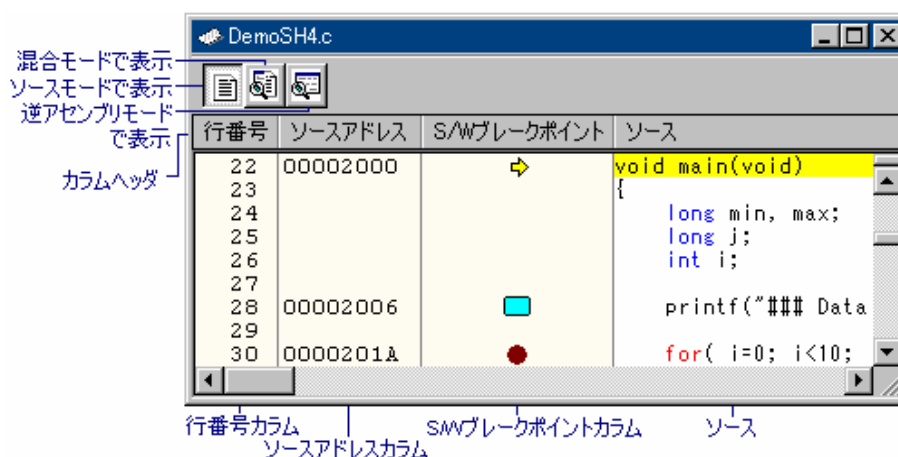
確認要求ダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。

**確認要求ダイアログボックスを再び表示するには**

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスが開きます。
2. [確認]タブを選択してください。
3. [表示モード切り替えで別ファイルを開く時の確認]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. [OK]をクリックしてください。

**17.2.1.1 ソースコードを表示する**

[ソースモードで表示]ボタンをクリックしたときにソースファイルのコードを表示します。

**ウィンドウの構成**

- 各モードが有効である場合、ツールバーのボタンをクリックすると表示を切り替えます。混合モードの逆アセンブリを表示するには、[混合モードで表示]ボタンをクリックしてください。逆アセンブリを表示するには、[逆アセンブリモードで表示]ボタンをクリックしてください。
- [ソース]領域の左側には以下に示すカラムがあります。

カラム名	内容
行番号	ソースファイルに対応する行番号を表示します。
ソースアドレス	モジュールをダウンロードすると、ソースファイルに対応するアドレスを表示します。
S/W ブレークポイント	PC 位置(👉)、PC ブレークポイント位置(●)、ブックマーク位置(■)を表示します。ダブルクリックにより PC ブレークポイントを設定または解除します。

- [ソース]領域はシンタックスを強調表示したコードを含みます。

## ウィンドウのオプション

[ソース]領域内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能	
コンパイル '<ファイル名>'	●	ファイルをコンパイルします。	
開く '<ファイル名>'	-	選択されたファイル名を元にファイルを開きます。	
切り取り	-	選択部分を削除し Windows® クリップボードに貼り付けます。	
コピー	-	選択部分をコピーし Windows® クリップボードに貼り付けます。	
貼り付け	-	Windows® クリップボードの内容をコピーしてアクティブウィンドウのカーソル位置に貼り付けます。	
プロジェクトにファイルの追加	-	ファイルをプロジェクトに追加します。	
定義へジャンプ '<ナビゲーション項目名>'	-	#define、C 関数、または C++ クラスなどのナビゲーション項目の定義位置を表示します。	
ナビゲーションリスト	-	スマートエディタ機能が有効な場合に、アルファベット順に C Define、C Function、または C++ Classes のナビゲーションリストをポップアップウィンドウで表示します。項目上をダブルクリックすると、カーソル位置にその文字列を貼り付けます。	
検索	-	現在のファイルのテキストを検索します。	
置換	-	テキストを置換します。	
ジャンプ	-	指定した行にジャンプします。	
括弧の呼応	-	括弧の組み合わせを見つけます。	
ブックマーク	ブックマークの挿入/削除	-	ブックマークを設定/解除します。
	次のブックマーク	-	次のブックマークにジャンプします。
	前のブックマーク	-	同じファイルの1つ前のブックマークに戻ります。
	すべてのブックマークの削除	-	ファイル内のすべてのブックマークを削除します。
テンプレート	テンプレートの定義	-	テンプレートを設定します。
	テンプレートの挿入	-	テンプレートを挿入します。
ブレークポイントの挿入/削除	●	標準のブレークポイント(PC ブレークポイント)を挿入または削除します。	
ブレークポイントの有効化/無効化	●	現在設定している PC ブレークポイントを有効または無効とします。	
表示カラムの設定	-	表示カラムを設定します。	
カラム	カラム名	-	表示カラムを設定します。
カラムヘッダの表示/非表示	-	カラムヘッダの表示/非表示を切り替えます。	
インスタントウォッチ *	-	変数名を現在のテキストカーソル(マウスカーソルではありません)の位置から抜き出して名前とともにインスタントウォッチダイアログボックスを開きます。ソースのある行のみ有効になります。	
ウォッチウィンドウに追加 *	-	選択した変数をウォッチウィンドウに登録します。	
カーソル位置まで実行	●	現在の PC アドレスからユーザプログラムの実行を開始します。プログラムは、PC がテキストカーソルの位置によって指定されたアドレスに達するか、別のブレーク条件が成立するまで実行します。	
カーソル位置に PC 値を設定	●	PC の値をテキストカーソルの位置によって指定されたアドレスに変更します。	

ポップアップメニューオプション	マクロの記録	機能
PC位置を表示	-	PC位置のソースファイルまたは逆アセンブリを表示します。
逆アセンブリ	-	現在のソース行に対応したアドレスの逆アセンブリウィンドウを開きます。
プロパティ	-	ファイルのプロパティを表示します。

**注：**

\*. 機能のサポートはデバッガに依存します。

### 17.2.1.2 アセンブリ言語コードを表示する

[逆アセンブリモード]の表示開始アドレスは、[ソースモード]のカーソル位置に対応するアドレスとなります。逆アセンブリコードの表示は、デバッガが接続されている場合のみ有効になります。

#### ウィンドウの構成



- 各モードが有効である場合、ツールバーのボタンをクリックすると表示を切り替えます。混合モードの逆アセンブリを表示するには、[混合モードで表示]ボタンをクリックしてください。ソースコードを表示するには、[ソースモードで表示]ボタンをクリックしてください。
- [逆アセンブリ]領域の左側には以下に示すカラムがあります。

カラム名	内容
S/W ブレークポイント - ASM	PC位置(→)、PCブレークポイント位置(●)を表示します。ダブルクリックによりPCブレークポイントを設定または解除します。
逆アセンブリアドレス	逆アセンブリアドレスを表示します。ダブルクリックにより[アドレス指定]ダイアログボックスを開きます。逆アセンブリの表示開始アドレスを入力してください。
オブジェクトコード	オブジェクトコードを表示します。
ラベル	ラベルを表示します。モジュールをダウンロードしていない場合このカラムは使用できません。

- [逆アセンブリモード]では、[逆アセンブリ]領域のアセンブリ言語コードをダブルクリックすると、[アセンブル]ダイアログボックスを開きます。アセンブリ言語コードを入力してください。

#### ウィンドウのオプション

[逆アセンブリ]領域内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能
リフレッシュ	-	逆アセンブリの最新情報を取得し表示内容を更新します。
リフレッシュ抑止	-	[逆アセンブリモード]のメモリ範囲がリフレッシュされないようにロックします。
ソースの表示	-	アドレスに対応するソースの位置でエディタを表示します。
表示アドレス設定	-	開始アドレスを指定します。



ポップアップメニューオプション	マクロの記録	機能
カーソル位置まで実行	●	現在の PC アドレスからプログラムを実行します。プログラムは、PC がテキストカーソルの位置によって指定されたアドレスに達するか、別のブレーク条件が成立するまで実行します。
PC 位置を表示	-	PC 位置の逆アセンブリを表示します。
カーソル位置に PC 値を設定	●	PC の値をテキストカーソルの位置によって指定されたアドレスに変更します。
編集	-	任意のアドレスの命令を変更します。
範囲で検索	-	指定した範囲で文字列を検索します。
コピー	-	選択部分をコピーし Windows® クリップボードに貼り付けます。
表示カラムの設定	-	表示カラムを設定します。
カラムヘッダの表示/非表示	-	カラムヘッダの表示、非表示を切り替えます。
テキストファイルに保存	-	指定した範囲をテキストで保存します。
印刷	-	指定した範囲を印刷します。
ブレークポイントの挿入/削除	●	標準のブレークポイント(PC ブレークポイント)を挿入または削除します。
ブレークポイントの有効化/無効化	●	現在設定している PC ブレークポイントを有効または無効とします。

### 17.2.1.3 逆アセンブリ表示をリフレッシュする

外部オペレーションが逆アセンブリ表示のメモリを変更する場合、High-performance Embedded Workshop はメモリの変更を検知できません。例えば、表示されている範囲のメモリに対して外部フラッシュユーティリティを使用してプログラミングしても、逆アセンブリは変更されません。

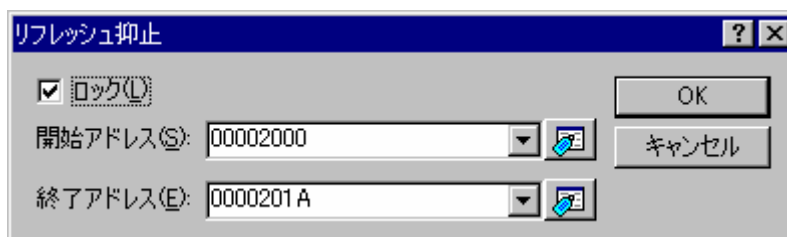
[逆アセンブリ]領域内 ([逆アセンブリモード]) で右クリックし、ポップアップメニューから[リフレッシュ]を選択してください。

逆アセンブリの最新情報を取得し表示内容を更新します。

### 17.2.1.4 逆アセンブリ表示のリフレッシュを抑制する

逆アセンブリ表示のメモリ範囲がリフレッシュされないようにロックします。この機能を逆アセンブリのリフレッシュ抑制と呼びます。

[逆アセンブリ]領域内 ([逆アセンブリモード]) で右クリックし、ポップアップメニューから[リフレッシュ抑制]を選択してください。[リフレッシュ抑制]ダイアログボックスを開きます。



[ロック]チェックボックスをオンにしてください。他のコントロールが有効になります。

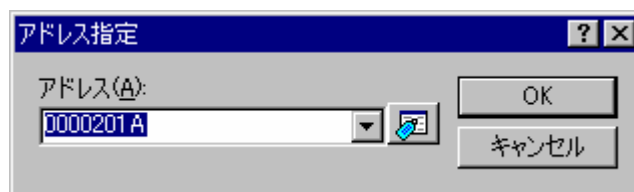
ロックおよびキャッシュする[開始アドレス]および[終了アドレス]を指定し[OK]ボタンをクリックしてください。ロックしたメモリ範囲のみを表示します。

### 17.2.1.5 特定のアドレスを見る

逆アセンブリ表示でプログラムを見ている場合、プログラム内の他の箇所を見たいときがあります。そのような場合、プログラム内のコードをスクロールせずに特定のアドレスにジャンプできます。



[逆アセンブリ]領域内 ([逆アセンブリモード]) で右クリックし、ポップアップメニューから[表示アドレス設定]を選択します。[アドレス指定]ダイアログボックスを開きます。



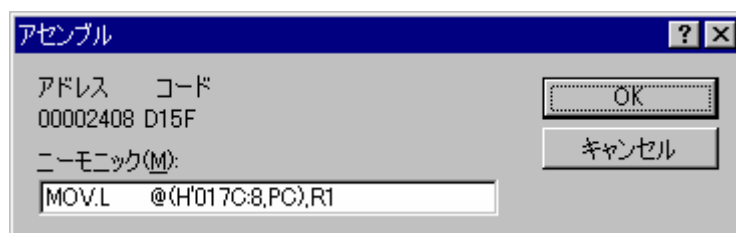
[アドレス]エディットボックスにアドレスを指定し、[OK]ボタンをクリックしてください。アドレスは、ラベル名で入力することも可能です。

逆アセンブリ表示を更新して入力したアドレスのコードを表示します。

オーバーロード関数またはクラス名を入力した場合、[関数選択]ダイアログボックスを開くので、関数を選択してください (機能のサポートはデバッガに依存します)。

### 17.2.1.6 アセンブリ言語コードを編集する

逆アセンブリ表示 ([逆アセンブリモード]) で編集したい命令をダブルクリックすることによって、アセンブリ言語コードを編集できます。[アセンブル]ダイアログボックスを開きます。



アドレス、オブジェクトコード、およびニーモニックを表示します。

[ニーモニック]に新しい命令を入力 (または古い命令を編集) します。

"Enter"キーを押すと、メモリ内容を新しいオブジェクトコードに書き換えて、次の命令に移ります。

[OK]ボタンをクリックすると、メモリ内容を新しい命令コードに書き換えて、ダイアログボックスを閉じます。

[キャンセル]ボタンをクリックするか"Esc"キーを押すと、メモリ内容を書き換えずに、ダイアログボックスを閉じます。

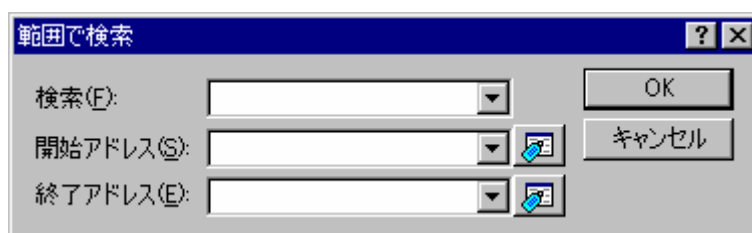
#### 注:

アセンブリ言語コードは、現在のメモリ内容から表示しています。メモリ内容を修正すると、[アセンブル]ダイアログボックスおよび逆アセンブリ表示では、新しいアセンブリ言語コードを表示します。しかしソース表示は変更しません。これはソースファイルにアセンブラを含む場合も同じです。

### 17.2.1.7 逆アセンブリで範囲検索する

逆アセンブリ表示の特定のアドレス範囲でテキスト文字列を検索できます。

[逆アセンブリ]領域内 ([逆アセンブリモード]) で右クリックし、ポップアップメニューから[範囲で検索]を選択してください。[範囲で検索]ダイアログボックスを開きます。

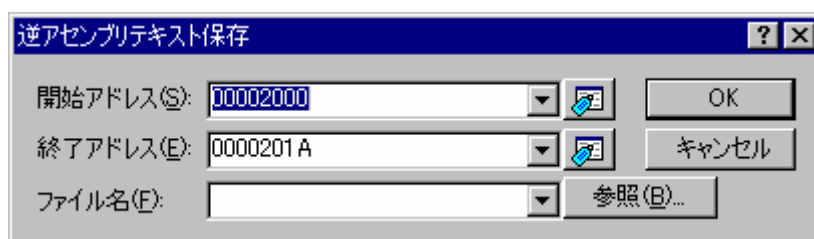


検索する文字列、開始アドレスおよび終了アドレスを指定してください。[OK]ボタンをクリックしてください。範囲にある最初にあらわれた該当文字列を選択します。

### 17.2.1.8 逆アセンブリテキストを保存する

特定のアドレス範囲の逆アセンブリ表示内容をテキストに保存できます。

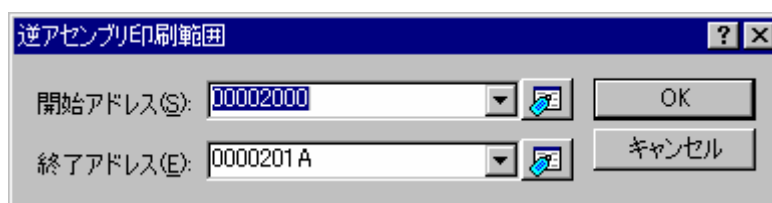
[逆アセンブリ]領域内 ([逆アセンブリモード]) で右クリックし、ポップアップメニューから[テキストファイルに保存]を選択してください。[逆アセンブリテキスト保存]ダイアログボックスを開きます。



保存する開始アドレス、終了アドレス、およびファイル名を指定してください。[OK]ボタンをクリックしてください。

### 17.2.1.9 逆アセンブリ表示を印刷する


逆アセンブリ表示を印刷するには、ウィンドウがアクティブな場合に[ファイル->印刷]または[印刷]ツールバーボタン(🖨️)を使用するか、または[逆アセンブリ]領域内 ([逆アセンブリモード]) のポップアップメニューにある[印刷]を使用します。[印刷]を選択すると、[逆アセンブリ印刷範囲]ダイアログボックスを開きます。特定のアドレス範囲の表示内容を印刷できます。



印刷する開始アドレスと終了アドレスを指定してください。[OK]ボタンをクリックすると、標準のプリントフォーマットを選択するダイアログボックスが開きます。ここで、使用するプリンタとページ設定のオプションを選択できます。[印刷]ボタンをクリックしてください。

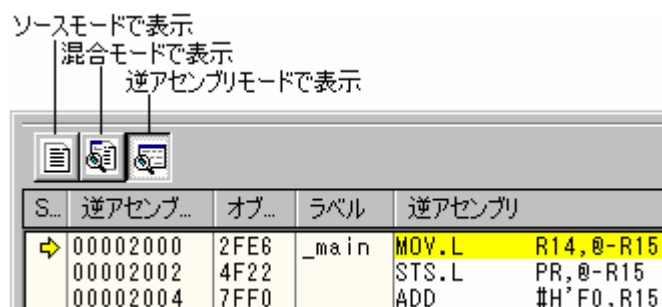
## 17.2.2 逆アセンブリウィンドウを開く

以下のいずれかの方法によりアセンブリ言語レベルでコードを表示できます。

- [表示 -> 逆アセンブリ]を選択する。
- [逆アセンブリ]ツールバーボタン(  )をクリックする。
- "Ctrl+D"キーを押す。

この場合、逆アセンブリウィンドウは現在のプログラムカウンタ (PC) の位置で開きます。

### ツールバー



High-performance Embedded Workshop V.4.02 以降の逆アセンブリウィンドウはソース表示を使用できます。

モードを切り替えるツールバーがあります。各モードが有効である場合、ボタンをクリックすると表示を切り替えます。

以下の3つのモードがあります。

モード	機能
ソースモード	現在の PC アドレスに対応するソースファイルがある場合、逆アセンブリウィンドウ内にソースファイルを開きます。 ステップ実行などによるプログラムカウンタ(PC)位置の変化に追従し、同一ウィンドウ上で PC に対応したソースファイルおよびコードを表示します。別のソースファイル領域へ移動しても、新たなウィンドウは開きません。このモードは、ソースファイルを読み取り専用で編集できません。モジュールがダウンロードされている場合のみ有効になります。 また、ソースファイルを参照している場合は、キーワードを強調表示します。行番号、アドレス、ブレークポイント、およびソースコードを参照できます。
混合モード	逆アセンブリウィンドウの混合モードは、ソースファイルの混合モードとは異なります。継続する逆アセンブリを示すかわりに、ソースコードの各行に関連する逆アセンブリを示します。このモードはアセンブリ言語コードを編集できます。モジュールがダウンロードされている場合のみ有効になります。 ブレークポイント、アドレス、オブジェクトコード、ラベル、および逆アセンブリコードを参照できます。
逆アセンブリモード	逆アセンブリモードは、実際に継続する逆アセンブリコードをアドレス順に示します。このモードはアセンブリ言語コードを編集できます。ターゲットがセッションに接続している場合のみ有効になります。 ブレークポイント、アドレス、オブジェクトコード、ラベル、および逆アセンブリコードを参照できます。

### 17.2.2.1 ソースコードを表示する

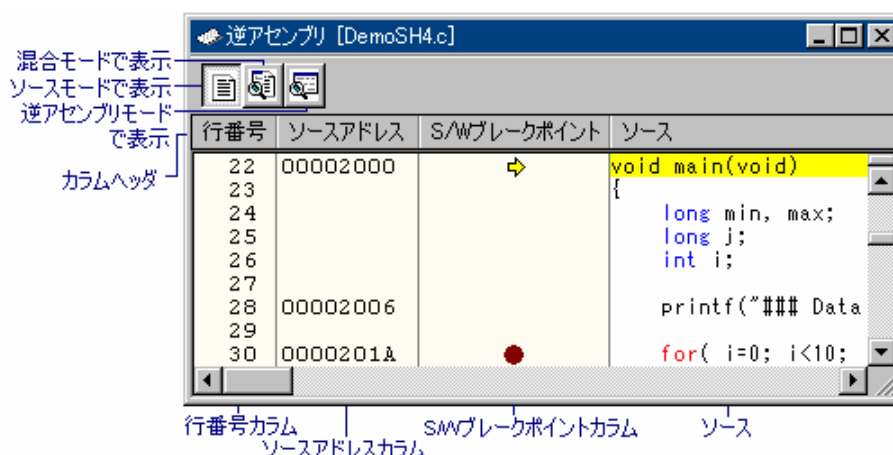
逆アセンブリウィンドウ内から[ソースモード]に切り替えるとき、現在の PC アドレスに対応するソースファイルがある場合、逆アセンブリウィンドウ内にソースファイルを開きます。

現在の PC アドレスに対応するソースファイルがない場合は、ソースファイルを開くためにダイアログボックスを開きます。ファイルを選択し[開く]ボタンをクリックしてください。逆アセンブリウィンドウのソースモードでは、テキストベースのファイルはプロジェクトのソースファイルに限らず、すべて選択できます。

このウィンドウで開くソースファイルは、すべて読み取り専用で編集できません。ソースファイルの表示とデバッグを目的とするためです。

この表示モードはモジュールがダウンロードされている場合のみ有効になります。

## ウィンドウの構成



- 各モードが有効である場合、ツールバーのボタンをクリックすると表示を切り替えます。混合モードの逆アセンブリを表示するには、[混合モードで表示]ボタンをクリックしてください。逆アセンブリを表示するには、[逆アセンブリモードで表示]ボタンをクリックしてください。

- [ソース]領域の左側には以下に示すカラムがあります。

カラム名	内容
行番号	ソースファイルに対応する行番号を表示します。
ソースアドレス	モジュールをダウンロードすると、ソースファイルに対応するアドレスを表示します。
S/W ブレイクポイント	PC 位置(➡)、PC ブレイクポイント位置(●)を表示します。ダブルクリックにより PC ブレイクポイントを設定または解除します。

- [ソース]領域はシンタックスを強調表示したコードを含みます。

## ウィンドウのオプション

[ソース]領域内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能
編集ソースの表示	-	編集するソースファイルをエディタウィンドウで開きます。
ファイルを開く	-	逆アセンブリウィンドウ内に表示するファイルを選択します。
コピー	-	選択部分をコピーし Windows® クリップボードに貼り付けます。
定義へジャンプ '<ナビゲーション項目名>	-	#define、C 関数、または C++クラスなどのナビゲーション項目の定義位置を表示します。
検索	-	現在のファイルのテキストを検索します。
ジャンプ	-	指定した行にジャンプします。
ブレイクポイントの挿入/削除	●	標準のブレイクポイント(PC ブレイクポイント)を挿入または削除します。
ブレイクポイントの有効化/無効化	●	現在設定している PC ブレイクポイントを有効または無効とします。
カラムヘッダの表示/非表示	-	カラムヘッダの表示/非表示を切り替えます。
表示カラムの設定	-	表示カラムを設定します。
カラム	カラム名	-
カラム	カラム名	-
表示カラムを設定します。	-	表示カラムを設定します。
インスタントウォッチ *	-	変数名を現在のテキストカーソル(マウスカーソルではありません)の位置から抜き出して名前とともにインスタントウォッチダイアログボックスを開きます。ソースのある行のみ有効になります。
ウォッチウィンドウに追加 *	-	選択した変数をウォッチウィンドウに登録します。
表示アドレス設定	-	開始アドレスを指定します。

ポップアップメニューオプション	マクロの記録	機能
カーソル位置まで実行	●	現在の PC アドレスからユーザプログラムの実行を開始します。プログラムは、PC がテキストカーソルの位置によって指定されたアドレスに達するか、別のブレーク条件が成立するまで実行します。
カーソル位置に PC 値を設定	●	PC の値をテキストカーソルの位置によって指定されたアドレスに変更します。
PC 位置を表示	-	PC 位置のソースファイルまたは逆アセンブリを表示します。
プロパティ	-	ファイルのプロパティを表示します。

**注：**

\*. 機能のサポートはデバッガに依存します。

### 17.2.2.2 編集するソースファイルを開く

逆アセンブリウィンドウのソースモードは、ソースを参照しデバッグが可能ですが編集はできません。

逆アセンブリウィンドウのソースモードで現在開いているソースファイルを編集できます。ポップアップメニューから[編集ソースの表示]を選択してください。エディタウィンドウで編集するファイルを開きます。

また、ワークスペースウィンドウ、または[ファイル->開く]メインメニューによって手動で編集するファイルを開くこともできます。

### 17.2.2.3 逆アセンブリウィンドウで別のファイルを開く

逆アセンブリウィンドウ内にソースファイルを開くことができます。ポップアップメニューから[ファイルを開く]を選択してください。このメニューオプションは、逆アセンブリウィンドウにあるすべての表示モードのポップアップメニューで利用できます。

このメニューオプションを選択すると、ソースファイルを開くためにファイルを参照するダイアログボックスを開きます。テキストベースのファイルはプロジェクトのソースファイルに限らず、すべて選択できます。

### 17.2.2.4 デバッグ中に逆アセンブリウィンドウのソースモード表示を優先する

デバッグ中に逆アセンブリウィンドウのソースモード表示を優先できます。

#### デバッグ中に逆アセンブリウィンドウのソースモード表示を優先するには

1. [基本設定->オプション]を選択し、[オプション]ダイアログボックスを開きます。
2. [デバッグ]タブを選択します。
3. [デバッグ中に逆アセンブリウィンドウのソースモード表示を優先]チェックボックスをオンにします。チェックボックスのデフォルトはオフです。
4. [OK]をクリックします。

この機能を使用した場合の動作を次に示します。

- 逆アセンブリウィンドウを[逆アセンブリモード]で開き、ソースファイルのアドレスでブレークすると、逆アセンブリウィンドウは[ソースモード]に切り替わります。
- ソースファイルのアドレスでブレークしても、逆アセンブリウィンドウが開いている場合はエディタウィンドウを開きません。
- プログラム実行前と後では、表示しているウィンドウの重なり順序を変更しません。
- エディタウィンドウで開いているソースファイルの表示位置を変更しません。

逆アセンブリウィンドウの[逆アセンブリモード]でステップを続けたい場合は、[デバッグ -> ステップモード -> アセンブリ]を選択してください。

### 17.2.2.5 アセンブリ言語コードを表示する

[逆アセンブリモード]の表示開始アドレスは現在の PC の位置で開きます。逆アセンブリコードの表示は、デバッグに接続されている場合のみ有効になります。

#### ウィンドウの構成



- 各モードが有効である場合、ツールバーのボタンをクリックすると表示を切り替えます。混合モードの逆アセンブリを表示するには、[混合モードで表示]ボタンをクリックしてください。ソースコードを表示するには、[ソースモードで表示]ボタンをクリックしてください。
- [逆アセンブリ]領域の左側には以下に示すカラムがあります。

カラム名	内容
S/W ブレークポイント - ASM	PC 位置(➡)、PC ブレークポイント位置(●)を表示します。ダブルクリックにより PC ブレークポイントを設定または解除します。
逆アセンブリアドレス	逆アセンブリアドレスを表示します。ダブルクリックにより[アドレス指定]ダイアログボックスを開きます。逆アセンブリの表示開始アドレスを入力してください。
オブジェクトコード	オブジェクトコードを表示します。
ラベル	ラベルを表示します。モジュールをダウンロードしていない場合このカラムは使用できません。

- [逆アセンブリモード]および[混合モード]では、[逆アセンブリ]領域のアセンブリ言語コードをダブルクリックすると、[アセンブル]ダイアログボックスを開きます。アセンブリ言語コードを入力してください。

#### ウィンドウのオプション

[逆アセンブリ]領域内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能
ファイルを開く	-	逆アセンブリウィンドウ内に表示するファイルを選択します。
リフレッシュ	-	逆アセンブリの最新情報を取得し表示内容を更新します。
リフレッシュ抑止	-	逆アセンブリウィンドウのメモリ範囲がリフレッシュされないようにロックします。
ソースの表示	-	アドレスに対応するソースの位置でエディタウィンドウを開きます。
表示アドレス設定	-	開始アドレスを指定します。
カーソル位置まで実行	●	現在の PC アドレスからプログラムを実行します。プログラムは、PC がテキストカーソルの位置によって指定されたアドレスに達するか、別のブレーク条件が成立するまで実行します。
PC 位置を表示	-	PC 位置の逆アセンブリを表示します。



ポップアップメニューオプション	マクロの記録	機能
カーソル位置に PC 値を設定	●	PC の値をテキストカーソルの位置によって指定されたアドレスに変更します。
編集	-	任意のアドレスの命令を変更します。
範囲で検索	-	指定した範囲で文字列を検索します。
コピー	-	選択部分をコピーし Windows® クリップボードに貼り付けます。
表示カラムの設定	-	表示カラムを設定します。
カラムヘッダの表示/非表示	-	カラムヘッダの表示、非表示を切り替えます。
テキストファイルに保存	-	指定した範囲をテキストで保存します。
印刷	-	指定した範囲を印刷します。
ブレークポイントの挿入/削除	●	標準のブレークポイント(PC ブレークポイント)を挿入または削除します。
ブレークポイントの有効化/無効化	●	現在設定している PC ブレークポイントを有効または無効とします。

### 17.2.2.6 逆アセンブリ表示をリフレッシュする

外部オペレーションが逆アセンブリ表示のメモリを変更する場合、High-performance Embedded Workshop はメモリの変更を検知できません。例えば、表示されている範囲のメモリに対して外部フラッシュユーティリティを使用してプログラミングしても、逆アセンブリは変更されません。

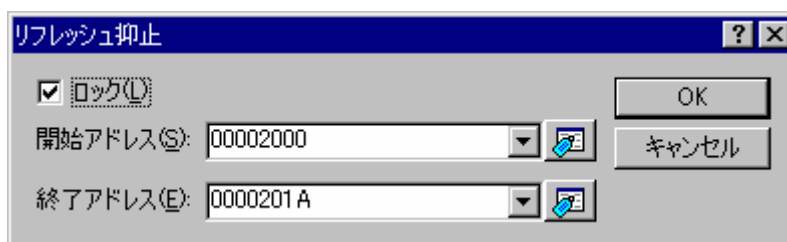
[逆アセンブリ]領域内 ([逆アセンブリモード]または[混合モード]) で右クリックし、ポップアップメニューから[リフレッシュ]を選択してください。

逆アセンブリの最新情報を取得し表示内容を更新します。

### 17.2.2.7 逆アセンブリ表示のリフレッシュを抑止する

逆アセンブリ表示のメモリ範囲がリフレッシュされないようにロックします。この機能を逆アセンブリのリフレッシュ抑止と呼びます。

[逆アセンブリ]領域内 ([逆アセンブリモード]または[混合モード]) で右クリックし、ポップアップメニューから[リフレッシュ抑止]を選択してください。[リフレッシュ抑止]ダイアログボックスを開きます。



[ロック]チェックボックスをオンにしてください。他のコントロールが有効になります。

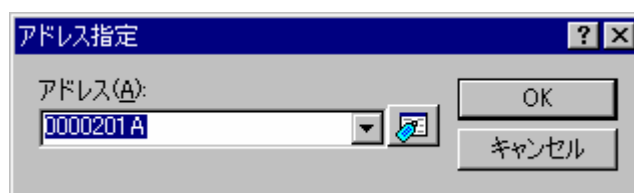
ロックおよびキャッシュする[開始アドレス]および[終了アドレス]を指定し[OK]ボタンをクリックしてください。ロックしたメモリ範囲のみを表示します。

### 17.2.2.8 特定のアドレスを見る

逆アセンブリウィンドウでプログラムを見ている場合、プログラム内の他の箇所を見たいときがあります。そのような場合、プログラム内のコードをスクロールせずに特定のアドレスにジャンプできます。

[ソース]領域内 ([ソースモード]) または[逆アセンブリ]領域内 ([逆アセンブリモード]または[混合モード]) で右クリックし、ポップアップメニューから[表示アドレス設定]を選択します。[アドレス指定]ダイアログボックスを開きます。





[アドレス]エディットボックスにアドレスを指定し、[OK]ボタンをクリックしてください。アドレスは、ラベル名で入力することも可能です。

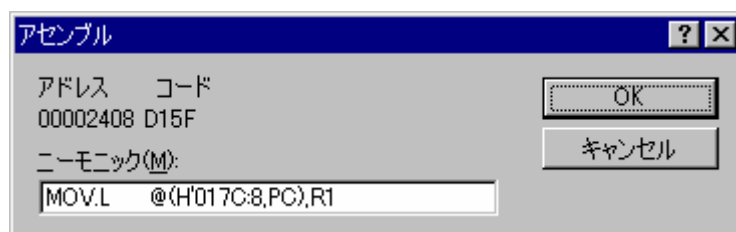
逆アセンブリ表示を更新して入力したアドレスのコードを表示します。

オーバーロード関数またはクラス名を入力した場合、[関数選択]ダイアログボックスを開くので、関数を選択してください（機能のサポートはデバッグに依存します）。

逆アセンブリウィンドウが[ソースモード]の場合に、別のソースに関連するが現在開いているソースファイルにないアドレスを入力すると、関連したソースファイルを[ソースモード]で自動的に開きます。ソースに関連しないアドレスを入力した場合、逆アセンブリウィンドウは指定されたアドレスを見るために[逆アセンブリモード]に切り替わります。

### 17.2.2.9 アセンブリ言語コードを編集する

逆アセンブリ領域内（[逆アセンブリモード]または[混合モード]）で編集したい命令をダブルクリックすることによって、アセンブリ言語コードを編集できます。[アセンブル]ダイアログボックスを開きます。



アドレス、オブジェクトコード、およびニーモニックを表示します。

[ニーモニック]に新しい命令を入力（または古い命令を編集）します。

"Enter"キーを押すと、メモリ内容を新しいオブジェクトコードに書き換えて、次の命令に移ります。

[OK]ボタンをクリックすると、メモリ内容を新しい命令コードに書き換えて、ダイアログボックスを閉じます。

[キャンセル]ボタンをクリックするか"Esc"キーを押すと、メモリ内容を書き換えずに、ダイアログボックスを閉じます。

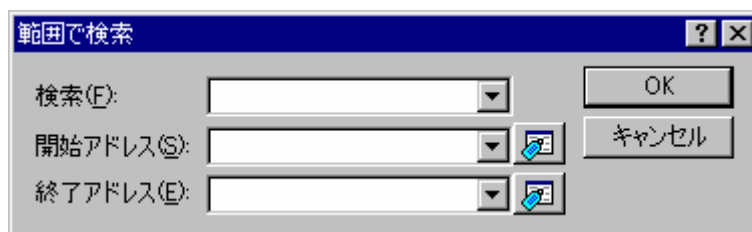
#### 注：

アセンブリ言語コードは、現在のメモリ内容から表示しています。メモリ内容を修正すると、[アセンブル]ダイアログボックスおよび逆アセンブリ表示では、新しいアセンブリ言語コードを表示します。しかしソース表示は変更しません。これはソースファイルにアセンブラを含む場合も同じです。

### 17.2.2.10 逆アセンブリで範囲検索する

逆アセンブリ表示の特定のアドレス範囲でテキスト文字列を検索できます。

[逆アセンブリ]領域内 ([逆アセンブリモード]または[混合モード]) で右クリックし、ポップアップメニューから[範囲で検索]を選択してください。[範囲で検索]ダイアログボックスを開きます。

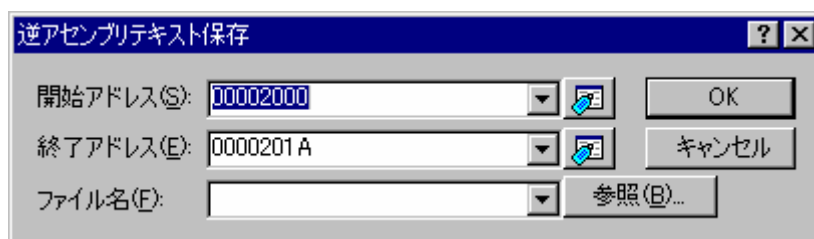


検索する文字列、開始アドレスおよび終了アドレスを指定してください。[OK]ボタンをクリックしてください。範囲の中で最初にあらわれた該当文字列を選択表示します。

### 17.2.2.11 逆アセンブリテキストを保存する

特定のアドレス範囲の逆アセンブリ表示内容をテキストに保存できます。

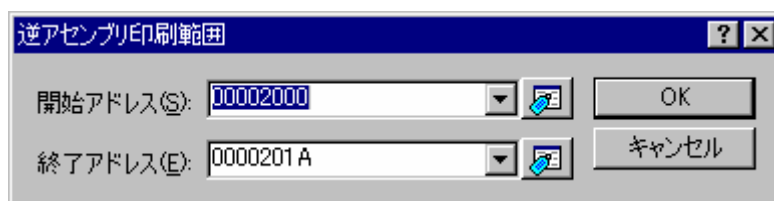
[逆アセンブリ]領域内 ([逆アセンブリモード]または[混合モード]) で右クリックし、ポップアップメニューから[テキストファイルに保存]を選択してください。[逆アセンブリテキスト保存]ダイアログボックスを開きます。



保存する開始アドレス、終了アドレス、およびファイル名を指定してください。[OK]ボタンをクリックしてください。


### 17.2.2.12 逆アセンブリ表示を印刷する

逆アセンブリ表示を印刷するには、ウィンドウがアクティブな場合に[ファイル->印刷]または[印刷]ツールバーボタン(🖨)を使用するか、または[逆アセンブリ]領域内 ([逆アセンブリモード]または[混合モード]) のポップアップメニューにある[印刷]を使用します。[印刷]を選択すると、[逆アセンブリ印刷範囲]ダイアログボックスを開きます。特定のアドレス範囲の表示内容を印刷できます。



印刷する開始アドレスと終了アドレスを指定してください。[OK]ボタンをクリックすると、標準のプリントフォーマットを選択するダイアログボックスが開きます。ここで、使用するプリンタとページ設定のオプションを選択できます。[印刷]ボタンをクリックしてください。

### 17.2.3 現在のPC位置を表示する

PC位置を表示するには、[PC位置の表示]ツールバーボタン()をクリックするか、[デバッグ->PC位置を表示]を選択してください。現在のPCの位置で、エディタウィンドウまたは逆アセンブリウィンドウを開きます。

### 17.2.4 PC位置の行を強調表示する

エディタウィンドウおよび逆アセンブリウィンドウでは、PCの位置は[S/Wブレイクポイント]カラムにアイコンで表示されます。PC位置の行のソースコードおよびアセンブリコードカラムを強調表示する機能があります。

#### PC位置の行を強調表示するには

1. [基本設定->オプション]を選択し、[オプション]ダイアログボックスを開きます。
2. [エディタ]タブを選択します。
3. [PCカーソル行のハイライト表示]チェックボックスをオンにします。チェックボックスのデフォルトはオンです。
4. [OK]ボタンをクリックします。

この行の色は、[表示形式]ダイアログボックスでカスタマイズできます。デフォルトの行の色は、テキスト(前面)が黒色で背景が黄色です。

#### PC位置の行の色を変更するには


1. [基本設定->表示の形式]を選択し、[表示形式]ダイアログボックスを開きます。
2. PC位置の行の色を変更したいアイテムをダイアログボックスの左側のツリーから選択し展開します。
  - エディタウィンドウおよび逆アセンブリウィンドウ(ソースモード)の場合は[Source]を選択し拡張する。
  - エディタウィンドウおよび逆アセンブリウィンドウ(混合モード、逆アセンブリモード)の場合は[逆アセンブリ]を選択し拡張する。
3. [PC Line Highlight]カテゴリを選択します。
4. [カラー]タブの[前面]リストと[背景]リストの選択を変更します。
5. [OK]ボタンをクリックします。

## 17.3 メモリを操作する

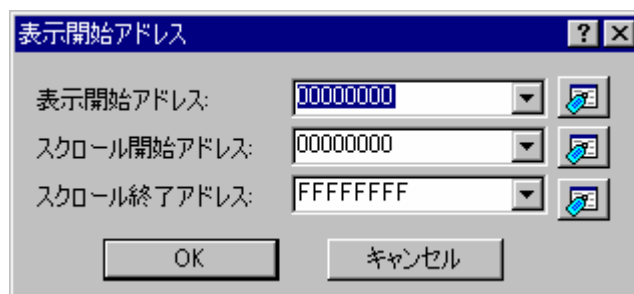
メモリ内容を操作する方法を説明します。操作の種類としては、色々なフォーマットでのメモリ内容の表示、メモリブロックのフィル/移動、およびロードモジュールファイルのロード/ベリファイなどがあります。

### 17.3.1 メモリウィンドウを開く

メモリウィンドウは、連続したメモリ内容をダンプ形式で表示するウィンドウです。

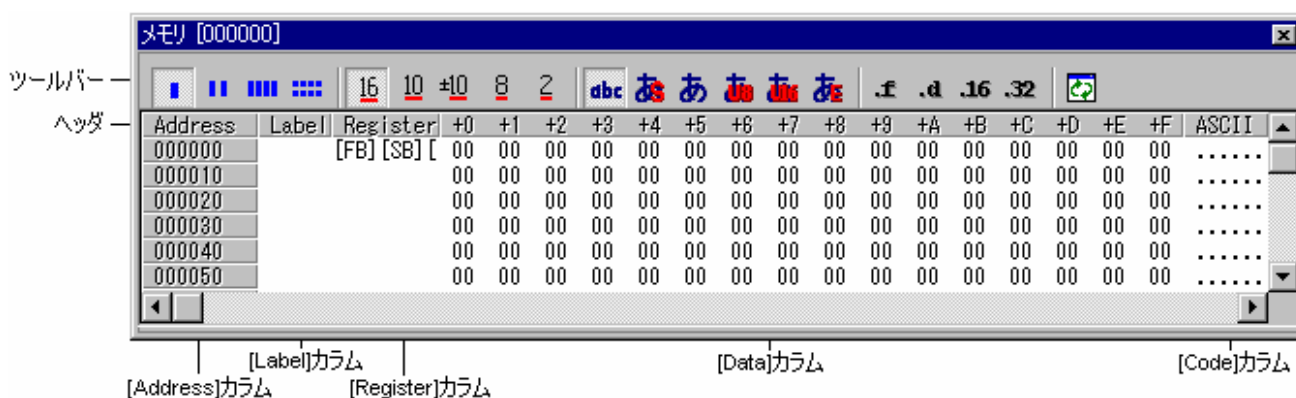
メモリウィンドウを開くには、[表示->CPU->メモリ]を選択するか、[メモリ]ツールバーボタン()をクリックしてください。

メモリウィンドウは、開くときに[表示開始アドレス]ダイアログボックスを表示します。[表示開始アドレス]、[スクロール開始アドレス]、および[スクロール終了アドレス]を指定してください。



[OK]ボタンをクリックするか、'Enter'キーを押すとダイアログボックスは閉じてメモリウィンドウが開きます。入力したスクロール開始アドレス、スクロール終了アドレス内でスクロールできます。

### ウィンドウの構成



- [Label]および[Register]カラムは、デフォルトで表示しません。
- [Label]カラムは、行頭のアドレスに割り付けられているラベル名を表示します。
- [Register]カラムは、行頭のアドレスに割り付けられているレジスタ名を表示します。
- [Data]カラムのヘッダの"+n"は、行頭のアドレスからのオフセット値です。
- [Code]カラムのヘッダにコード名を表示します。
- [Address]または[Label]カラムをダブルクリックすることにより、表示開始アドレスを変更するためのダイアログボックスが開きます。
- [Data]または[Code]カラムをダブルクリックすることにより、クリックしたアドレスのメモリ内容を変更するためのダイアログボックスが開きます。値の編集操作はマクロに記録できます。（「マクロの記録」）
- [Address]、[Data]、および[Code]カラムは、インプレース編集により値を変更できます。（「マクロの記録」）

### ウィンドウのオプション



ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	マクロの記録	機能
設定		●	任意のアドレスにデータを設定します。
フィル		●	任意のアドレス領域を一定データで充填します。

ポップアップメニューオプション		ツールバーボタン	マクロの記録	機能	
コピー			●	任意のアドレス領域のデータを別領域にコピーします。	
比較 *1			●	任意の2つのアドレス領域のデータを比較します。	
テスト *1			-	任意のアドレス領域をテストします。	
テキストファイルに保存			-	任意のアドレス領域をテキストファイルに保存します。	
検索 *1			-	任意のアドレス領域のデータを検索します。	
次を検索 *1			-	任意のアドレス領域の次のデータを検索します。	
開始アドレス			-	表示開始アドレスを指定します。	
スクロール範囲			-	スクロール範囲を指定します。	
レジスタ *1 (xxxxx)		-	-	指定したレジスタが示すアドレスから表示します。	
スタック追従 *1			-	スタックポインタ位置を追従します。	
スタートアップシンボル			-	ダウンロード時に表示するラベル位置を指定します。	
最新の情報に更新			-	ウィンドウの表示内容を手動で更新します。	
表示固定		-	-	ウィンドウの表示内容の更新を抑止します。	
自動更新 *2			-	ユーザプログラム実行中、ウィンドウの表示内容を定期的に更新します。	
更新間隔 *2			-	自動更新の間隔を設定します。	
データ長	1byte		-	1バイト単位で表示します。	
	2byte		-	2バイト単位で表示します。	
	4byte		-	4バイト単位で表示します。	
	8byte		-	8バイト単位で表示します。	
基数	16進数表示		-	16進数で表示します。	
	10進数表示		-	10進数で表示 (符号なし) します。	
	符号付10進数表示		-	10進数で表示 (符号付き) します。	
	8進数表示		-	8進数で表示します。	
	2進数表示		-	2進数で表示します。	
表示コード	ASCII			-	ASCIIコードで表示します。
	SJIS			-	SJISコードで表示します。
	JIS			-	JISコードで表示します。
	UNICODE	UTF-8		-	UTF-8コードで表示します。
		UTF-16		-	UTF-16コードで表示します。
	EUC			-	EUCコードで表示します。
	Float	Float		-	単精度浮動小数点値で表示します。
		Double		-	倍精度浮動小数点値で表示します。
	Complex	Float Complex		-	複素数を単精度浮動小数点値で表示します。
		Double Complex		-	複素数を倍精度浮動小数点値で表示します。
		Float Imaginary		-	虚数を単精度浮動小数点値で表示します。
		Double Imaginary		-	虚数を倍精度浮動小数点値で表示します。
	Fixed	16bit Fixed		-	16ビット固定値で表示します。
		32bit Fixed		-	32ビット固定値で表示します。
24 bit Accum		-	-	24ビット累算値で表示します。	
40 bit Accum		-	-	40ビット累算値で表示します。	
レイアウト	ラベル		-	[Label]カラムの表示/表示なしを切り替えます。	
	レジスタ		-	レジスタ表示部の表示/表示なしを切り替えます。	

ポップアップメニューオプション		ツールバーボタン	マクロの記録	機能
	コード		-	コード表示部の表示/表示なしを切り替えます。
カラム			-	表示桁数を指定します。
カバレッジ *1	有効		-	カバレッジ計測結果の表示/表示なしを切り替えます。
保存			●	任意のアドレス領域のメモリ内容を保存します。
読み込み			●	メモリにファイルをロードします。
分割		-	-	ウィンドウを分割表示します。
ツールバー表示		-	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ		-	-	ツールバーボタンをカスタマイズします。

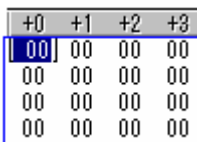
注：

- \*1. 機能のサポートはデバッガに依存します。
- \*2. 機能がサポートされている場合のみ表示されます。

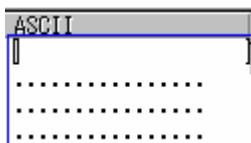
### 17.3.2 任意のアドレスにデータを設定する

メモリウィンドウで任意のアドレスにデータを設定するには次の方法があります。

- [Data]カラムでインプレース編集します。



- [Code]カラムでインプレース編集します。



- 以下のいずれかの方法で[メモリ設定]ダイアログボックスを開きます。
  - [Data]カラムをダブルクリックする。
  - [Code]カラムをダブルクリックする。
  - 変更するデータを選択し、ポップアップメニューの[設定]を選択する。



[データ]領域に設定するデータ（値または文字）を入力してください。

[ベリファイ]チェックボックスでメモリベリファイを指定します。ベリファイ機能のサポートはデバッガに依存します。

#### 値を設定する場合

[データ形式]グループの[値]ボタンをクリックしてください。[データサイズ]領域にデータ長を指定してください。

#### 文字を設定する場合

[データ形式]グループの[文字列]ボタンをクリックしてください。[コード]領域に文字コードを指定してください。

### 17.3.3 アドレス領域を選択する

メモリアドレス範囲がメモリウィンドウにある場合、最初のメモリユニット（メモリウィンドウディスプレイでの選択に従い）をクリックして最後のユニットまでマウスをドラッグすることによって領域を選択できます。選択した領域は強調表示します。

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

### 17.3.4 任意のアドレス領域を一定データで充填する

メモリウィンドウ上で充填するアドレス領域をドラッグして選択してください。ポップアップメニューから[フィル]を選択して[メモリフィル]ダイアログボックスを開きます。



[フィルデータ]へ充填するデータ(値または文字)を入力してください。

[ベリファイ]チェックボックスでメモリベリファイを指定します。ベリファイ機能のサポートはデバッガに依存します。



充填するアドレス領域をドラッグしていない場合は、開始アドレス、終了アドレスを入力する必要があります。終了アドレスには先頭に'+記号を付けることができ、この記号を入力すると開始アドレス+入力した値が終了アドレスになります。

#### 値を設定する場合

[データ形式]グループの[値]ボタンをクリックしてください。[データサイズ]領域でデータ長も指定してください。

#### 文字列を設定する場合

[データ形式]グループの[文字列]ボタンをクリックしてください。[コード]領域で文字コードも指定してください。

表示データ長が2バイトの場合は、2バイト分の文字が指定できます。

文字列を指定するには、[メモリ設定]ダイアログボックスを使用してください。

[メモリ設定]ダイアログボックスは、ポップアップメニューから[設定]を選択してください。

### 17.3.5 任意のアドレス領域のデータを別領域にコピーする

メモリウィンドウ上でコピー元アドレス領域をドラッグして選択してください。ポップアップメニューから[コピー]を選択して[メモリコピー]ダイアログボックスを開きます。



[コピー先アドレス]にコピー先の開始アドレスを入力してください。

[ベリファイ]チェックボックスでメモリベリファイを指定します。ベリファイ機能のサポートはデバッガに依存します。

コピー元のアドレス領域をドラッグしていない場合は、コピー元開始アドレス/終了アドレスを入力する必要があります。終了アドレスには先頭に'+記号を付けることができ、この記号を入力すると開始アドレス+入力した値が終了アドレスになります。

#### ドラッグアンドドロップ

ドロップデータの種類	動作
メモリウィンドウの[Data]カラムの選択範囲	選択範囲のデータ内容を、ドロップ位置からの領域にコピーします。

### 17.3.6 任意の2つのアドレス領域を比較する

メモリウィンドウ上で比較元アドレス領域をドラッグして選択してください。ポップアップメニューから[比較]を選択して[メモリ比較]ダイアログボックスを開きます。



[比較先アドレス]に比較先アドレス、[データサイズ]に比較フォーマットを指定してください。

比較元のアドレス領域をドラッグしていない場合は、比較元開始アドレス/終了アドレスを入力する必要があります。終了アドレスには先頭に'+記号を付けることができ、この記号を入力すると開始アドレス+入力した値が終了アドレスになります。

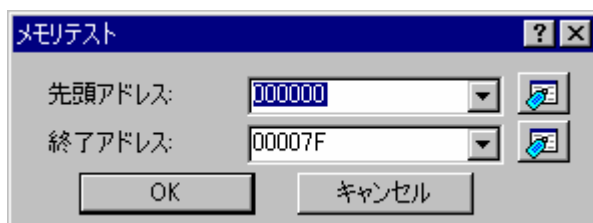
不一致個所があった場合は、そのアドレスをメッセージボックスに表示します。

比較したメモリの内容が一致した場合は、「比較成功」メッセージを表示します。

機能のサポートはデバッガに依存します。

### 17.3.7 任意のアドレス領域をテストする

メモリウィンドウ上でテストするアドレス領域をドラッグして選択してください。ポップアップメニューから[テスト]を選択して[メモリテスト]ダイアログボックスを開きます。



テストするアドレス領域をドラッグしていない場合は、テスト開始アドレス/終了アドレスを入力する必要があります。

終了アドレスには先頭に'+記号を付けることができ、この記号を入力すると開始アドレス+入力した値が終了アドレスになります。

機能のサポートはデバッガに依存します。

#### 注：

テストはデバッガに依存しますが、すべての場合において、メモリの現在の内容を上書きします。

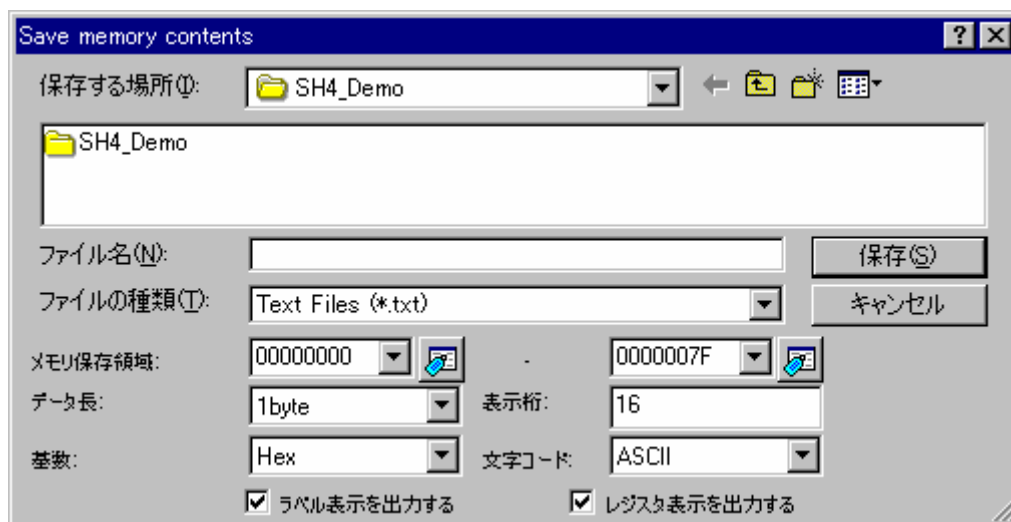
作成したプログラムまたはデータは、消去します。

### 17.3.8 任意のアドレス領域をテキストファイルに保存する

メモリウィンドウ上でテキストファイルに保存するアドレス領域をドラッグして選択してください。ポップアップメニューから[テキストファイルに保存]を選択して[メモリ内容保存]ダイアログボックスを開きます。

[メモリ保存領域]で出力範囲、[データ長]でデータ長、[表示桁]で桁数、[基数]で基数、[文字コード]でコードを選択してください。[ラベル表示を出力する]で[Label]カラムの表示/表示なし、[レジスタ表示を出力する]で[Register]カラムを表示/表示なしを切り替えてください。

保存するアドレス領域をドラッグしていない場合は、出力範囲を入力する必要があります。



### 17.3.9 任意のアドレス領域のデータを検索する

メモリウィンドウ上で検索するアドレス領域をドラッグして選択してください。ポップアップメニューから[検索]を選択して[メモリ検索]ダイアログボックスを開きます。



[検索データ]で検索するデータ値を入力し、[検索データ形式]ドロップダウンリストで検索フォーマットを選択してください。

[検索データ形式]で"pattern" (パターン検索)を選択すると、最大 256 バイト以内のバイト列(512 文字以内)が検索できます。その場合、検索データは 1 バイトにつき 2 文字単位で指定してください。

パターン検索以外では検索条件として、データ一致/不一致、検索方向を指定できます。パターン検索はデータ一致および順方向のみの検索となります。

検索するアドレス領域をドラッグしていない場合は、検索開始アドレス、終了アドレスを入力する必要があります。終了アドレスには先頭に '+' 記号を付けることができ、この記号を入力すると開始アドレス + 入力した値が終了アドレスになります。

データが見つかったら、見つかったデータを強調表示します。データが見つかった状態で、ポップアップメニューから[次を検索]を選択すると、次のアドレスから検索を続行します。

機能のサポートはデバッガに依存します。

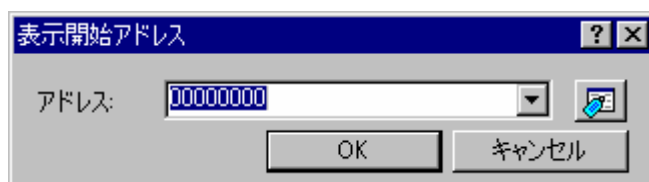
### 17.3.10 表示開始アドレスを変更する

スクロールバー、Page Up/Page Down キー、上下矢印キーで表示位置を変更してください。  
表示アドレスを直接指定するには次の方法があります。

- [Address]カラムでインプレース編集します。



- 以下のいずれかの方法で[表示開始アドレス]ダイアログボックスを開きます。[アドレス]に表示するアドレスを指定してください。
  - [Address]カラムをダブルクリックする。
  - ポップアップメニューの[開始アドレス]を選択する。



### ドラッグアンドドロップ

操作	動作
メモリウィンドウの Address 領域のアドレスを選択して、別のメモリウィンドウの[Address]カラムにドロップ	ウィンドウの表示開始アドレスを、そのアドレスに変更します。
変数名（テキスト文字列）を選択して、メモリウィンドウの[Address]カラムにドロップ	ウィンドウの表示開始アドレスを、その変数のアドレスに変更します。

### 17.3.11 スクロール範囲を変更する

メモリウィンドウのポップアップメニューから[スクロール範囲]を選択してください。[スクロール範囲]ダイアログボックスが開きます。



表示するスクロール範囲を指定してください。

デフォルトのスクロール範囲は、0h~MCUの最大アドレスです。

### 17.3.12 指定したレジスタが示すアドレスから表示する

メモリウィンドウのポップアップメニュー[レジスタ]以下でレジスタを選択してください。

機能のサポートはデバッガに依存します。

### 17.3.13 スタックポインタ位置を追従する

メモリウィンドウは、スタックポインタ位置を追従して表示アドレスを変更する機能を持っています(デフォルトは、スタックポインタ位置を追従しません)。スタックポインタ位置を追従するには、メモリウィンドウのポップアップメニューから[スタック追従]を選択してください。[追従スタックポインタ]ダイアログボックスが開きます。

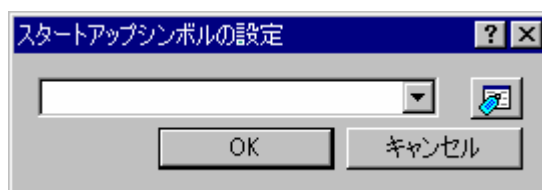


ドロップダウンリストで追従するスタックポインタを選択してください。メモリウィンドウは、そのスタックポインタ位置を追従して表示アドレスを自動的に変更します。[追従スタックポインタ]ダイアログボックスで[Not Following]を選択するとスタックポインタ位置の追従を終了します。

機能のサポートはデバッガに依存します。

### 17.3.14 ダウンロード時に表示するラベル位置を変更する

メモリウィンドウのポップアップメニューから[スタートアップシンボル]を選択してください。[スタートアップシンボルの設定]ダイアログボックスが開きます。



ドロップダウンリストで開始シンボルを指定してください。

### 17.3.15 ウィンドウの表示内容を手動で更新する

メモリウィンドウの内容を強制的に更新できます。

メモリウィンドウのポップアップメニューから[最新の情報に更新]を選択してください。

### 17.3.16 ウィンドウの表示内容の更新を抑止する

プログラムの実行停止時などに、自動的にメモリウィンドウ内容を更新しないようにできます。

メモリウィンドウのポップアップメニューから[表示固定]を選択してください。表示内容の更新を抑止しているときは、メモリウィンドウの内容をグレーで表示します。

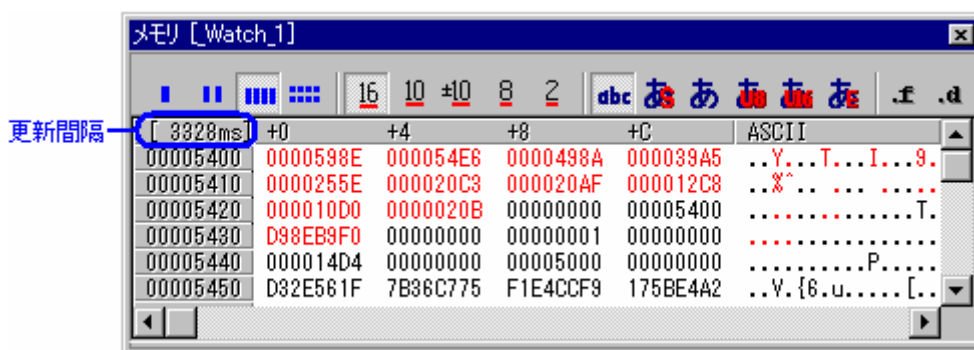
メモリウィンドウの[自動更新]機能は、この機能と同時に指定できません。

### 17.3.17 ウィンドウの表示内容を定期的に更新する

メモリウィンドウのポップアップメニューから[自動更新]を選択すると、ユーザプログラム実行中にメモリウィンドウの表示内容を定期的に更新します。

ユーザプログラム実行中、実際の更新間隔をメモリウィンドウのヘッダの左端のカラムに表示します。

機能のサポートはデバッガに依存します。



ユーザプログラム実行中の自動更新間隔を設定するには、メモリウィンドウのポップアップメニューから[更新間隔]を選択します。機能がサポートされている場合のみ表示されます。

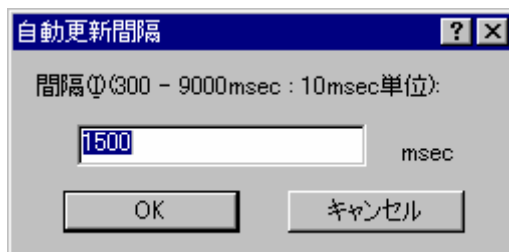
メモリウィンドウの[表示固定]機能は、この機能と同時に指定できません。

### 17.3.18 更新間隔を設定する

メモリウィンドウの表示内容をユーザプログラム実行中に自動的に更新する間隔を設定できます。機能のサポートはデバッガに依存します。

#### 更新間隔を設定するには

1. ウィンドウ内を右クリックしポップアップメニューを表示します。
2. [更新間隔]を選択します。[自動更新間隔設定]ダイアログボックスが開きます。



3. 更新間隔を 10ms 単位で入力します。更新間隔のデフォルト値、および更新間隔の指定可能範囲は、デバッガに依存します。
4. [OK]ボタンをクリックします。

更新間隔は、各メモリウィンドウで個別に指定できます。

メモリウィンドウの[自動更新]機能を有効にすると、ユーザプログラム実行中、実際の更新間隔をメモリウィンドウのヘッダの左端のカラムに表示します。

ユーザプログラム実行時の状態によっては、実際の更新間隔が設定した値よりも大きくなる場合があります。

### 17.3.19 表示データ長を変更する

メモリウィンドウのポップアップメニュー[データ長]以下で表示バイト長を指定してください。

以下のいずれかが選択できます。

[1byte]	1 バイト単位表示 (デフォルト)
[2byte]	2 バイト単位表示
[4byte]	4 バイト単位表示
[8byte]	8 バイト単位表示

### 17.3.20 表示基数を変更する

メモリウィンドウのポップアップメニュー[基数]以下で表示基数を指定してください。

以下のいずれかが選択できます。

[16 進数表示]	16 進数で表示 (デフォルト)
[10 進数表示]	10 進数で表示 (符号なし)
[符号付 10 進数表示]	10 進数で表示 (符号付き)
[8 進数表示]	8 進数で表示
[2 進数表示]	2 進数で表示

### 17.3.21 表示コードを変更する

メモリウィンドウのポップアップメニュー[表示コード]以下で表示コードを指定してください。

以下のいずれかが選択できます。

[ASCII]		ASCII コードで表示 (デフォルト)
[SJIS]		SJIS コードで表示
[JIS]		JIS コードで表示
[UNICODE]	[UTF-8]	UTF-8 コードで表示
	[UTF-16]	UTF-16 コードで表示
[EUC]		EUC コードで表示
[Float]	[Float]	単精度浮動小数点値で表示
	[Double]	倍精度浮動小数点値で表示
[Complex]	[Float Complex]	複素数を単精度浮動小数点値で表示
	[Double Complex]	複素数を倍精度浮動小数点値で表示
	[Float Imaginary]	虚数を単精度浮動小数点値で表示
	[Double Imaginary]	虚数を倍精度浮動小数点値で表示
[Fixed]	[16 bit Fixed]	16 ビット固定値で表示
	[32 bit Fixed]	32 ビット固定値で表示
	[24 bit Accum]	24 ビット累算値で表示
	[40 bit Accum]	40 ビット累算値で表示

### 17.3.22 レイアウトを設定する

メモリウィンドウのポップアップメニュー[レイアウト]以下で表示するレイアウトを設定してください。以下を選択できます。

[ラベル]	[Label]カラムの表示/表示なしの切り替え
[レジスタ]	[Register]カラムの表示/表示なしの切り替え
[コード]	[Code]カラムの表示/表示なしの切り替え



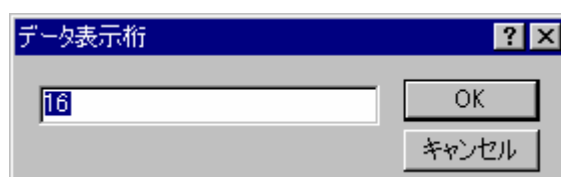
表示している場合はオプションにチェックマークが付いています。

[Label]、[Register]、および[Code]カラムを表示しない場合:

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

### 17.3.23 表示桁数を変更する

メモリウィンドウのポップアップメニューから[カラム]を選択すると、[データ表示桁]ダイアログボックスが開きます。



表示桁数を指定してください (1 ~ 256)。

### 17.3.24 カバレッジ計測結果の表示/表示なしを切り替える

メモリウィンドウは、カバレッジ計測表示がデフォルトで無効になっています。

有効にするには、メモリウィンドウのポップアップメニュー[カバレッジ]以下の[有効]を選択してください。

カバレッジ計測時、コード実行部分のデフォルトの色は、前面が黒色で背景が水色です。コード未実行部分のデフォルトの色は、前面が黒色で背景がグレーです。

カバレッジ計測時のコード実行部分および未実行部分の色は、[表示形式]ダイアログボックスでカスタマイズできます。詳細は、「17.3.29 既存の色をカスタマイズする」を参照してください。

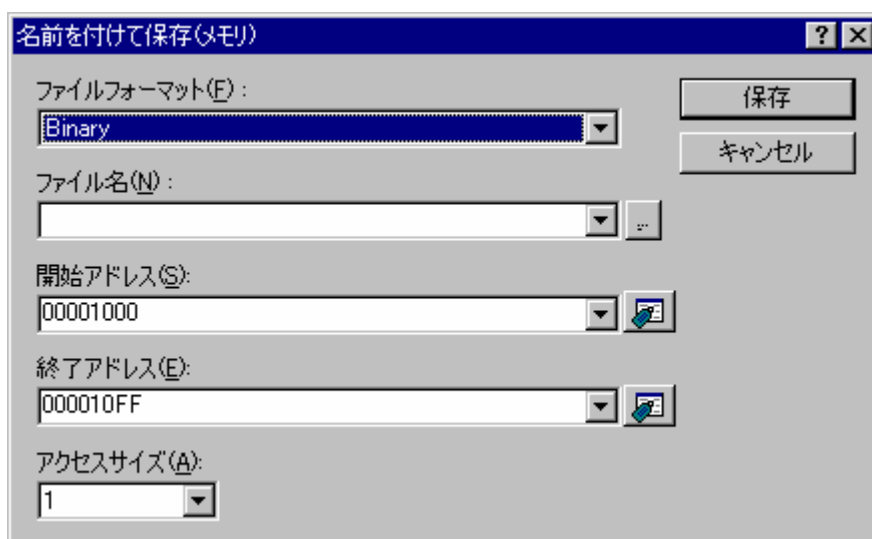
+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

機能のサポートはデバッガに依存します。

### 17.3.25 任意のアドレス領域を保存する

メモリウィンドウ上で保存するアドレス領域をドラッグして選択してください。ポップアップメニューから[保存]を選択して[名前を付けて保存(メモリ)]ダイアログボックスを開きます。

([デバッグ -> メモリの保存]を選択して表示することもできます)



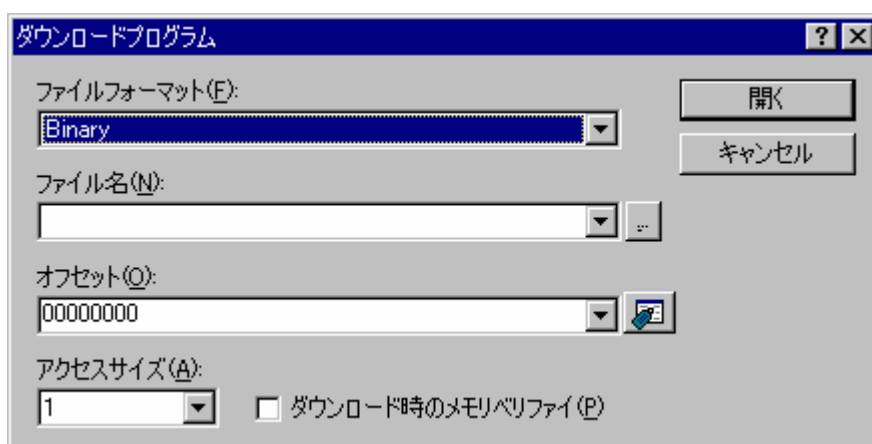
保存するアドレス領域をドラッグしていない場合は、保存開始アドレス/終了アドレスを入力する必要があります。終了アドレスには先頭に'+記号を付けることができ、この記号を入力すると開始アドレス+入力した値が終了アドレスになります。

[ファイルフォーマット]にファイルフォーマット、[ファイル名]にファイル名、[アクセスサイズ]にアクセスサイズを入力します。[ファイル名]には、メモリを保存するために使用した過去4ファイルのファイル名を表示します。また[...]ボタンをクリックすると、標準の[名前を付けて保存]ダイアログボックスを開きます。[アクセスサイズ]から、保存時のアクセスサイズを選択できます。保存するメモリがリトルエンディアンの時は、アクセスサイズによってデータの並びが異なります。

ファイルの保存が完了すると、確認のメッセージボックスを表示する場合があります。

### 17.3.26 メモリにファイルをロードする

デバッガのメモリにファイルをロードできます。メモリウィンドウのポップアップメニューから[読み込み]を選択して、[ダウンロードプログラム]ダイアログボックスを開きます。



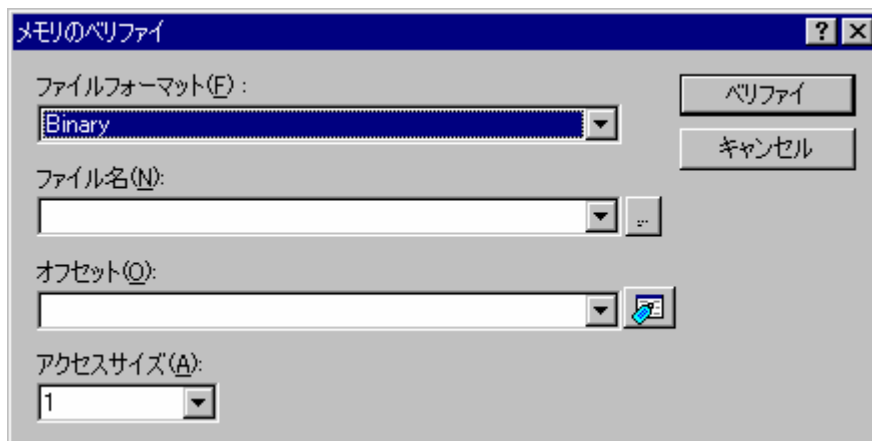
[ファイルフォーマット]にファイルフォーマット、[ファイル名]にファイル名、[オフセット]にオフセット値、[アクセスサイズ]にアクセスサイズ、[ダウンロード時のメモリバリエーション]でメモリバリエーションを指定します。[オフセット]には、ロードアドレス値を変更するときはオフセット値、変更しないときは0を指定します。

### 17.3.27 ウィンドウを分割表示する

メモリウィンドウを上下2分割で表示したいときは、ポップアップメニューから[分割]を選択後、分割バーを移動して分割します。ポップアップメニューから[分割]を選択すると分割表示を解除します。

### 17.3.28 アドレス領域を検証する

メモリとファイル内容をベリファイします。[デバッグ -> メモリのベリファイ]を選択して[メモリのベリファイ]ダイアログボックスを開きます。



[ファイルフォーマット]にファイルフォーマット、[ファイル名]にファイル名、[オフセット]にオフセット、[アクセスサイズ]にアクセスサイズを指定します。

ベリファイが成功した場合は、「File verified OK」メッセージを表示します。

ベリファイの不一致箇所があった場合は、「Verify failed」メッセージを表示します。

機能のサポートはデバッグに依存します。

### 17.3.29 テキストの色を変更する

メモリウィンドウはデータおよびコードを表示しているテキストの色およびフォントを他のウィンドウと同じように[表示形式]ダイアログボックスでカスタマイズできます（下表のカテゴリの"通常"）。

さらに、メモリウィンドウのデータおよびコードカラムで、メモリアクセスおよびメモリの値が変更された部分の色を、[表示形式]ダイアログボックスでカスタマイズできます。

#### 既存の色をカスタマイズするには

1. [基本設定 -> 表示の形式]を選択し、[表示形式]ダイアログボックスを開きます。
2. [メモリ]アイテムをダイアログボックスの左側のツリーから選択し展開します。
3. 色を変更したいカテゴリを選択します。次の表に示すカテゴリが選択できます。

カテゴリ	カスタマイズ対象	カラータブの前面の色 (デフォルト)	カラータブの背景の色 (デフォルト)
通常	テキスト	黒	白
アクセス履歴あり *	カバレッジ計測：コード実行部分	黒	水色
アクセス履歴なし *	カバレッジ計測：コード未実行部分	黒	グレー
カバレッジ計測領域外 *	カバレッジ計測：領域外	黒	白
メモリ領域外 *	メモリ領域外	グレー	グレー
変更	値が変更された部分	赤	白

**注：**


- \*. 機能のサポートはデバッガに依存します。
- 4. [カラー]タブの[前面]リストと[背景]リストの選択を変更します。
- 5. [OK]ボタンをクリックします。

## 17.4 メモリ内容を画像形式で表示する

画像ウィンドウを使用すると、メモリ内容を画像形式で表示できます。

機能のサポートはデバッガに依存します。

### 17.4.1 画像ウィンドウを開く

[表示 -> グラフィック -> 画像]を選択するか、[画像]ツールバーボタン(  )をクリックすると、[画像プロパティ]ダイアログボックスが開きます。



[画像プロパティ]ダイアログボックスでは画像ウィンドウの表示方法を指定します。

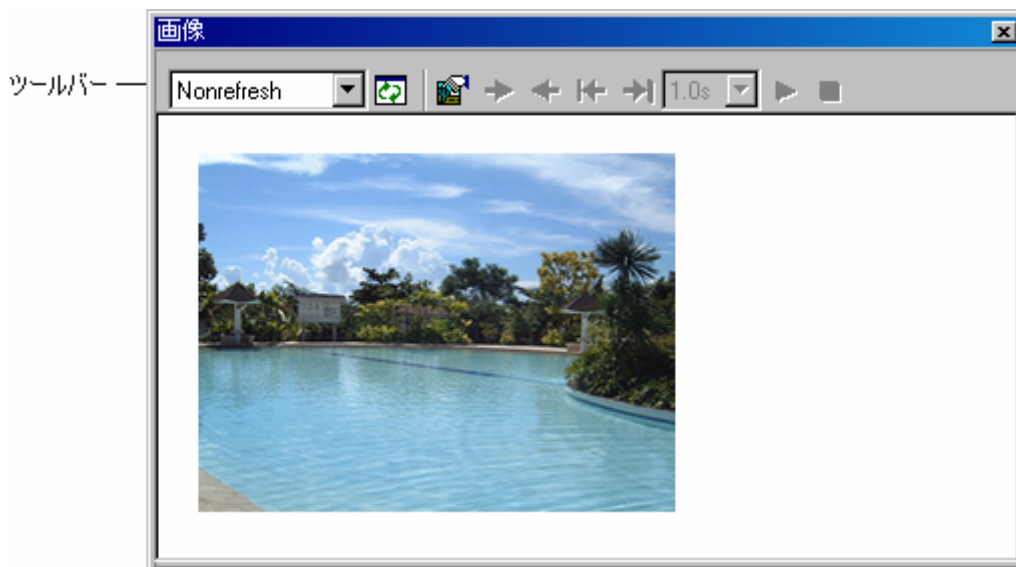
- [色情報] 表示する画像のカラー情報を指定します。
- [モード] フォーマットを指定します。
- [モノクロ] 白黒で表現します。
- [RGB] R(赤)、G(緑)、B(青)で表現します。
- [BGR] B(青)、G(緑)、R(赤)で表現します。
- [YCbCr] Y(輝度)、Cb(青色の色差)、Cr(赤色の色差)で表現します。
- [ビット/ピクセル] 選択した[モード]によって、ビット/ピクセルを指定します(RGB/BGR 選択時有効)。
- [サンプリング] サンプリングのフォーマットを指定します(YCbCr 選択時有効)。  
サンプリング例を以下に示します。

サンプリング比率	元のデータ	サンプリングデータ	画像(4*4 ピクセル)のデータ量
4:4:4	11,12,13,14 21,22,23,24 31,32,33,34 41,42,43,44	11,12,13,14 21,22,23,24 31,32,33,34 41,42,43,44	16 バイト
4:2:2		11,11,13,13 21,21,23,23 31,31,33,33 41,41,43,43	8 バイト
4:1:1		11,11,13,13 11,11,13,13 31,31,33,33 31,31,33,33	4 バイト
4:2:0		11,11,11,11 21,21,21,21 31,31,31,31 41,41,41,41	4 バイト

- [フォーマット] 点順次/面順次を指定します(YCbCr 選択時有効)。
- フォーマット**                      **各データの順序**
- 点順次 (4:2:0 選択時無効)    Y,Cb,Cr,Y,Cb,Cr,Y,Cb,Cr,...,Y,Cb,Cr
- 面順次                                    Y,Y,Y,...,Y,Cb,Cb,Cb,...,Cb,Cr,Cr,Cr,...,Cr
- 面順次 2                                Y,Y,Y,...,Y,Cb,Cr,Cb,Cr,Cb,Cr,...,Cb,Cr
- [バッファ情報] データの格納場所、サイズ、パレットのアドレスを指定します。
- [データアドレス] 表示する画像データのメモリ開始アドレスを指定します (16 進数表示)。
- [パレットアドレス] カラーパレットデータのメモリ開始アドレスを指定します (16 進数表示)。(RGB/BGR の 8 ビット選択時有効)
- [幅/高さ情報(ピクセル)] 画像の幅と高さを指定します。
- [幅]                                      画像の幅を指定します (接頭辞省略時は 10 進数で入力、10 進数で表示)。
- [高さ]                                    画像の高さを指定します (接頭辞省略時は 10 進数で入力、10 進数で表示)。
- [バッファ                                幅と高さから画像のバッファサイズを表示します (16 進サイズ(16 進) 数表示)。
- [連続表示の有効] 連続表示が有効になります。
- [イメージ情報] 連続表示するフレーム数を指定します。
- [フレーム数]                            フレーム数を指定します。
- [全バッファ                            幅と高さからフレーム数から画像のバッファサイズを表示      サイズ(16 進) します (16 進数表示)。
- [表示情報] 画像全体中の表示部分の位置、サイズ、データ開始位置を指定します。(16 進数表示、RGB/BGR の 8 ビット選択時有効)
- [表示モード]                            画像の全体表示/部分表示を指定します。

	[全面表示]	画像を全体表示します。
	[部分表示]	画像を部分表示します。
[開始位置]		データの表示開始位置を指定します。
	[上部]	左上からデータを表示します。
	[下部]	左下からデータを表示します。
[座標]		部分表示する画像の開始位置を指定します([部分表示]選択時有効)。
	[X 座標]	開始位置の X 座標を指定します ( 接頭辞省略時は 10 進数で入力、10 進数で表示 )。
	[Y 座標]	開始位置の Y 座標を指定します ( 接頭辞省略時は 10 進数で入力、10 進数で表示 )。
[幅/高さ情報(ピクセル)]		部分表示する画像の幅と高さを指定します。
	[幅]	表示の幅を指定します ( 接頭辞省略時は 10 進数で入力、10 進数で表示 )。
	[高さ]	表示の高さを指定します ( 接頭辞省略時は 10 進数で入力、10 進数で表示 )。

[画像プロパティ]ダイアログボックスに設定後、[OK]ボタンをクリックすると画像ウィンドウが開きます。画像ウィンドウ表示後もポップアップメニューの[プロパティ]を選択することで本ダイアログボックスを表示し、表示内容を変更できます。メモリの内容を画像で表示します。



- ウィンドウ内をダブルクリックすると、マウスポインタの位置のピクセル情報を[ピクセル情報]ダイアログボックスに表示します。

### ウィンドウのオプション

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション		ツールバーボタン	マクロの記録	機能
自動更新	更新しない		-	ウィンドウの表示内容を自動更新しません。
	停止		-	ユーザプログラム実行停止時にウィンドウ内容を更新します。
	リアルタイム		-	ユーザプログラム実行中、ウィンドウの表示内容を定期的に更新します。
更新			-	ウィンドウを更新します。
更新間隔 *			-	自動更新間隔を設定します。
次の画像			-	次の画像を表示します。
前の画像			-	前の画像を表示します。
先頭の画像			-	先頭の画像を表示します。
最後の画像			-	最後の画像を表示します。
連続表示の更新 間隔	0.5 秒		-	0.5 秒間隔で画像を連続表示します。
	1.0 秒		-	1.0 秒間隔で画像を連続表示します。
	2.0 秒		-	2.0 秒間隔で画像を連続表示します。
	4.0 秒		-	4.0 秒間隔で画像を連続表示します。
画像の連続表示			-	画像を連続表示します。
連続表示の停止			-	連続表示を停止します。
プロパティ			-	[画像プロパティ]ダイアログボックスを開きます。
ツールバー表示		-	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ		-	-	ツールバーボタンをカスタマイズします。

**注：**

\*. 機能がサポートされている場合のみ表示されます。

**17.4.2 ウィンドウの表示内容を定期的に更新する**

ポップアップメニューから[自動更新 -> 更新しない]を選択すると、ウィンドウの表示内容を自動更新しません。

ポップアップメニューから[自動更新 -> 停止]を選択すると、ユーザプログラム実行停止時にウィンドウの表示内容を更新します。

ポップアップメニューから[自動更新 -> リアルタイム]を選択すると、ユーザプログラム実行中にウィンドウの表示内容を定期的に更新します。

ユーザプログラム実行中の自動更新間隔を設定するには、ポップアップメニューから[更新間隔]を選択します。機能がサポートされている場合のみ表示されます。

**17.4.3 ウィンドウの表示内容を手動で更新する**

画像ウィンドウのポップアップメニューから[更新]を選択すると、直ちにウィンドウの表示内容を更新します。

**17.4.4 更新間隔を設定する**

画像ウィンドウの表示内容をユーザプログラム実行中に自動的に更新する間隔を設定できます。

機能のサポートはデバッガに依存します。

**更新間隔を設定するには**

1. ウィンドウ内を右クリックしポップアップメニューを表示します。
2. [更新間隔]を選択します。[更新間隔]ダイアログボックスが開きます。





3. 更新間隔を 10ms 単位で入力します。更新間隔は 10～10000ms の範囲で指定できます。デフォルト値は 1000ms です。
4. [OK]ボタンをクリックします。

指定した更新間隔は、画像ウィンドウ間で共通に使用します。


プログラム実行時の状態によっては、実際の更新間隔が設定した値よりも大きくなる場合があります。

#### 17.4.5 連続フレームで画像を見る

画像ウィンドウは、フレームの操作により複数の画像の切り替え表示ができます。この機能により、複数の画像に各々アドレスを設定する必要がなく画像を参照できます。

同じサイズのフレームとして画像を連続したアドレスのメモリに読み込んでおくと、フレームを順番に切り替えることにより画像を参照できます。





##### 連続したフレームで複数の画像を表示する機能を有効にするには

1. 以下のいずれかの操作を選択し[画像プロパティ]ダイアログボックスを開きます。
  - 現在画像ウィンドウを開いている場合は、ポップアップメニューから[プロパティ]を選択する。
  - 新しく画像ウィンドウを開く場合は、[表示 -> グラフィック -> 画像]を選択するか、[画像]ツールバーボタンをクリックする。
2. [画像プロパティ]ダイアログボックスで[連続表示の有効]チェックボックスをオンにします。
3. [フレーム数]エディットボックスに画像を連続表示するフレームの数(2フレーム以上)を入力します。
4. OK ボタンをクリックします。

この機能を有効にすると、画像のフレームを手動または定期的に切り替えて参照できます。

##### 17.4.5.1 画像のフレームを手動で切り替える

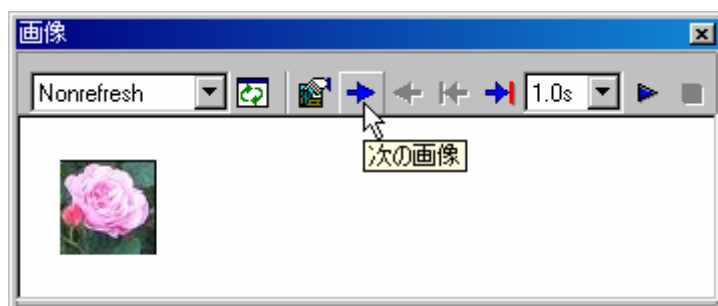
連続したフレームで複数の画像を表示する機能が有効の場合、以下に示すウィンドウのポップアップメニューオプション、またはウィンドウのツールバーボタンにより、画像のフレームを手動で切り替えることができます。

ポップアップメニューオプション	ツールバーボタン	機能
次の画像		次の画像を表示します。
前の画像		前の画像を表示します。
先頭の画像		先頭の画像を表示します。
最後の画像		最後の画像を表示します。

現在表示しているフレームの位置により、ウィンドウのポップアップメニューオプション、またはウィンドウのツールバーボタンのアクティブ状態は異なります。

### 画像のフレームを手動で切り替えるには

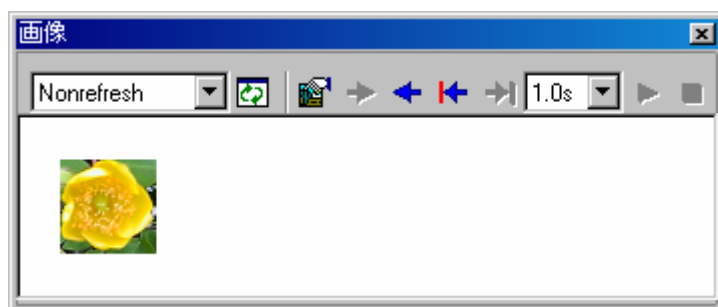
[次の画像]をクリックすると、次の画像のフレームへ切り替えます。例えば、連続した3フレーム分の画像を参照する場合は以下のように操作します。



1. [次の画像]ツールバーボタンをクリックすると、2フレーム目の画像を表示します。



2. [次の画像]ツールバーボタンをクリックすると、3フレーム目の画像を表示します。



#### 17.4.5.2 画像のフレームを定期的に切り替える

連続したフレームで複数の画像を表示する機能が有効の場合、以下に示すウィンドウのポップアップメニューオプション、またはウィンドウのツールバーボタンにより、画像のフレームを定期的に切り替えることができます。

ポップアップメニューオプション		ツールバーボタン	機能
連続表示の更新間隔 *	0.5 秒	1.0s	0.5 秒間隔で画像を連続表示します。
	1.0 秒		1.0 秒間隔で画像を連続表示します。
	2.0 秒		2.0 秒間隔で画像を連続表示します。
	4.0 秒		4.0 秒間隔で画像を連続表示します。
画像の連続表示			画像を連続表示します。
連続表示の停止			連続表示を停止します。

**注：**

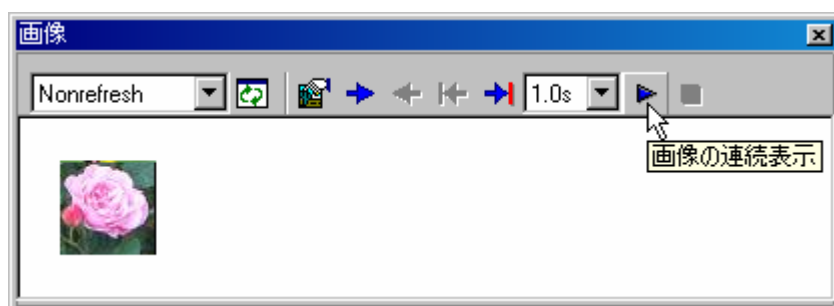
\*. 使用するデバッガまたは画像サイズにより更新間隔時間内に表示を更新できないことがあります。

現在表示しているフレームの位置により、ウィンドウのポップアップメニューオプション、またはウィンドウのツールバーボタンのアクティブ状態は異なります。

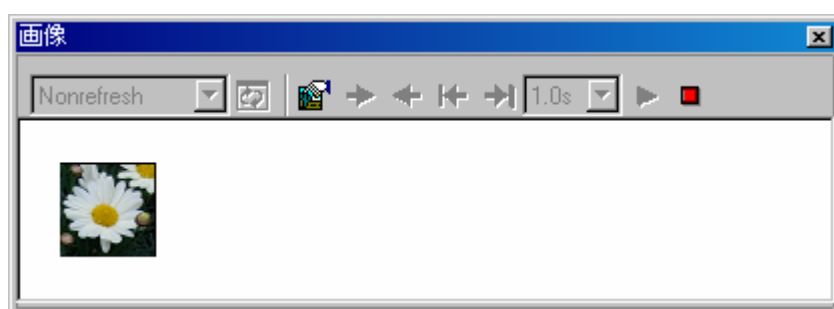
**画像のフレームを定期的に切り替えるには**

[画像の連続表示]を使用すると、画像のフレームを定期的に切り替えます。例えば、連続した3フレーム分の画像を参照する場合は以下のように操作します。

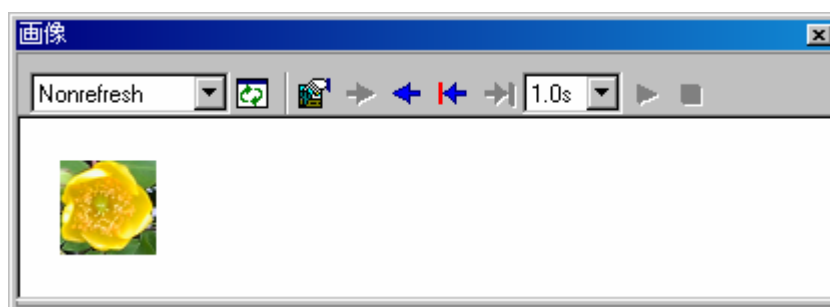
- 1 フレーム目の画像で、[連続表示の更新間隔]ドロップダウンリストから画像の表示更新間隔を選択し、[画像の連続表示]ツールバーボタンをクリックします。



2. 選択した時間が経過すると、自動的に2フレーム目の画像を表示します。



3. 選択した時間が経過すると、自動的に3フレーム目の画像を表示します。画像の連続表示が終了します。



#### 17.4.6 ピクセル情報を表示する

ウィンドウ内をダブルクリックするとマウスポインタの位置のピクセル情報を[ピクセル情報]ダイアログボックスに表示します。



カーソル位置のピクセル情報を表示します。

- [カラーモード] 画像のフォーマットを表示します。
- [ピクセル] カーソル位置のカラー情報を表示します (10進数表示)。
- [位置] カーソル位置を X 座標、Y 座標で表示します (10進数表示)。
  - [X] カーソル位置の X 座標を表示します。
  - [Y] カーソル位置の Y 座標を表示します。
- [バッファサイズ] バッファサイズを表示します (10進数表示)。
  - [幅] バッファの幅を表示します。
  - [高さ] バッファの高さを表示します。
- [画像サイズ] 表示の幅と高さを表示します (10進数表示)。
  - [幅] 幅を表示します。
  - [高さ] 高さを表示します。

### 17.5 メモリ内容を波形形式で表示する

波形ウィンドウを使用すると、メモリ内容を波形形式で表示します。

機能のサポートはデバッガに依存します。

#### 17.5.1 波形ウィンドウを開く

[表示 -> グラフィック -> 波形]を選択するか、[波形]ツールバーボタン(📊)をクリックすると、[波形プロパティ]ダイアログボックスが開きます。

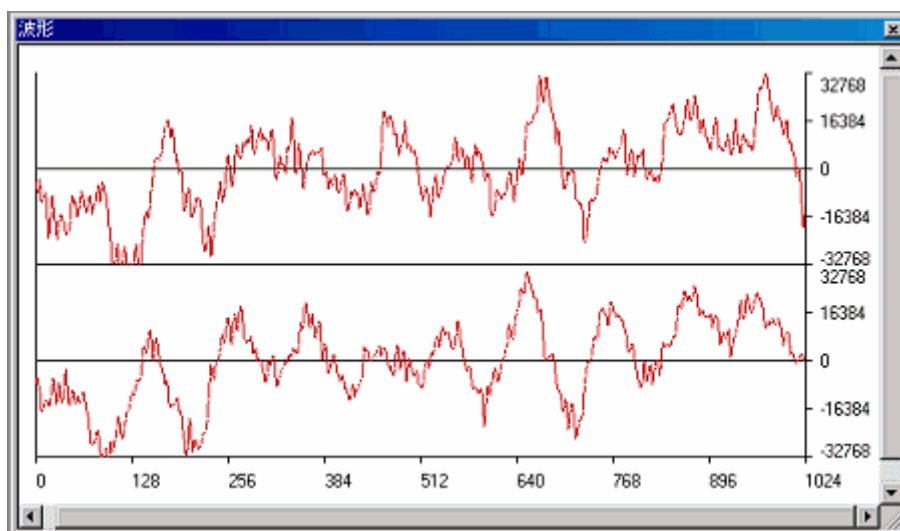


表示する波形形式を指定します。下記項目を指定できます。

- [データアドレス]                      データのメモリ開始アドレスを指定します（16進数表示）。
- [データサイズ]                        8ビット/16ビットを指定します。
- [チャンネル]                            モノラル/ステレオを指定します。
- [バッファサイズ(16進)]               データのバッファサイズを指定します（16進数表示）。

[波形プロパティ]ダイアログボックスに設定後、[OK]ボタンをクリックすると波形ウィンドウが開きます。波形ウィンドウ表示後もポップアップメニューの[プロパティ]を選択することで[波形プロパティ]ダイアログボックスを開き表示内容を変更できます。

メモリ内容を波形で表示します。横軸(X)にサンプリングデータ数、縦軸(Y)にサンプリング値を表示します。



- グラフ内でサンプリング情報を表示したい座標をダブルクリックすると、[サンプリング情報]ダイアログボックスが開きます。

### ウィンドウのオプション

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション		マクロの記録	機能
自動更新	更新しない	-	ウィンドウ内容を自動更新しません。
	停止	-	ユーザプログラム実行停止時にウィンドウ内容を更新します。
	リアルタイム	-	ユーザプログラム実行中、ウィンドウの表示内容を定期的に更新します。
更新		-	ウィンドウを更新します。

ポップアップメニューオプション		マクロの記録	機能
更新間隔 *		-	自動更新間隔を設定します。
伸張		-	拡大表示します。
圧縮		-	縮小表示します。
元に戻す		-	最初のサイズに戻します。
圧縮・伸張倍率	X2	-	拡大/縮小倍率を 2 倍に設定します。
	X4	-	拡大/縮小倍率を 4 倍に設定します。
	X8	-	拡大/縮小倍率を 8 倍に設定します。
スケール	128	-	横軸のサイズを 128 ピクセルに設定します。
	256	-	横軸のサイズを 256 ピクセルに設定します。
	512	-	横軸のサイズを 512 ピクセルに設定します。
カーソル削除		-	カーソルを非表示にします。
サンプリング情報		-	サンプリング情報を表示します。
プロパティ		-	[波形プロパティ]ダイアログボックスを開きます。

**注：**

\*. 機能がサポートされている場合のみ表示されます。

### 17.5.2 ウィンドウの表示内容を定期的に更新する

ポップアップメニューから[自動更新 -> 更新しない]を選択すると、ウィンドウの表示内容を自動更新しません。

ポップアップメニューから[自動更新 -> 停止]を選択すると、ユーザプログラム実行停止時にウィンドウの表示内容を更新します。

ポップアップメニューから[自動更新 -> リアルタイム]を選択すると、ユーザプログラム実行中にウィンドウの表示内容を定期的に更新します。

ユーザプログラム実行中の自動更新間隔を設定するには、ポップアップメニューから[更新間隔]を選択します。機能がサポートされている場合のみ表示されます。

### 17.5.3 ウィンドウの表示内容を手動で更新する

波形ウィンドウのポップアップメニューから[更新]を選択すると、直ちにウィンドウの表示内容を更新します。

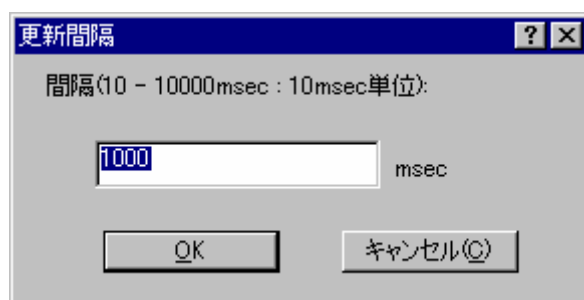
### 17.5.4 更新間隔を設定する

波形ウィンドウの表示内容をユーザプログラム実行中に自動的に更新する間隔を設定できます。

機能のサポートはデバッガに依存します。

#### 更新間隔を設定するには

1. ウィンドウ内を右クリックしポップアップメニューを表示します。
2. [更新間隔]を選択します。[更新間隔]ダイアログボックスが開きます。



- 更新間隔を 10ms 単位で入力します。更新間隔は 10～10000ms の範囲で指定できます。デフォルト値は 1000ms です。
- [OK]ボタンをクリックします。

指定した更新間隔は、波形ウィンドウ間で共通に使用します。

プログラム実行時の状態によっては、実際の更新間隔が設定した値よりも大きくなる場合があります。

#### 17.5.5 拡大表示する

ポップアップメニューから[伸張]を選択すると、横軸を拡大して表示します。

#### 17.5.6 縮小表示する

ポップアップメニューから[圧縮]を選択すると、横軸を縮小して表示します。

#### 17.5.7 最初のサイズに戻す

ポップアップメニューから[元に戻す]を選択すると、最初のサイズに戻して表示します。

#### 17.5.8 拡大/縮小倍率を設定する

ポップアップメニューの[圧縮・伸張倍率]サブメニューで拡大/縮小倍率を 2、4、または 8 倍から選択します。

#### 17.5.9 横軸のサイズを設定する

ポップアップメニューの[スケール]サブメニューで横軸のサイズを 128、256、または 512 ピクセルから選択します。

#### 17.5.10 カーソルを非表示にする

ポップアップメニューの[カーソル削除]をチェックすると、カーソルを非表示にします。

#### 17.5.11 サンプリング情報を表示する

グラフ内をクリックするとカーソル（緑色の縦線）が表示されます。カーソルは右矢印キーおよび左矢印キーで左右に移動できます。ウィンドウ内で右クリックし、ポップアップメニューから[サンプリング情報]を選択すると、[サンプリング情報]ダイアログボックスが開きます。



また、グラフ内でサンプリング情報を表示したい座標をダブルクリックすると、[サンプリング情報]ダイアログボックスが開きます。



波形ウィンドウのカーソル位置のサンプリング情報を表示します。下記情報を表示します。

- [データサイズ] 8ビット/16ビットを表示します。
- [チャンネル] データのチャンネルを表示します。
- [データ値] [X] カーソル位置の X 座標を表示します。
- [Y] カーソル位置の Y 座標を表示します(ステレオ選択時は上下 2 つの Y 座標を表示します)。


## 17.6 I/Oレジスタを見る

CPU および ROM/RAM と同様、マイクロコントローラには内蔵周辺モジュールがあります。デバイスによって周辺モジュールの数および型は異なりますが、代表的なモジュールとしては、DMA コントローラ、シリアルコミュニケーションインタフェース、A/D コンバータ、インテグレートドタイマユニット、バスステートコントローラおよびウォッチドッグタイマなどがあります。マイクロコントローラのアドレス空間にマッピングしたアクセスレジスタは、内蔵周辺モジュールを制御します。

組込みマイクロコントローラアプリケーションでは、内蔵周辺レジスタを使用した設定が重要であるため、これらのレジスタの内容を見やすく表示しています。

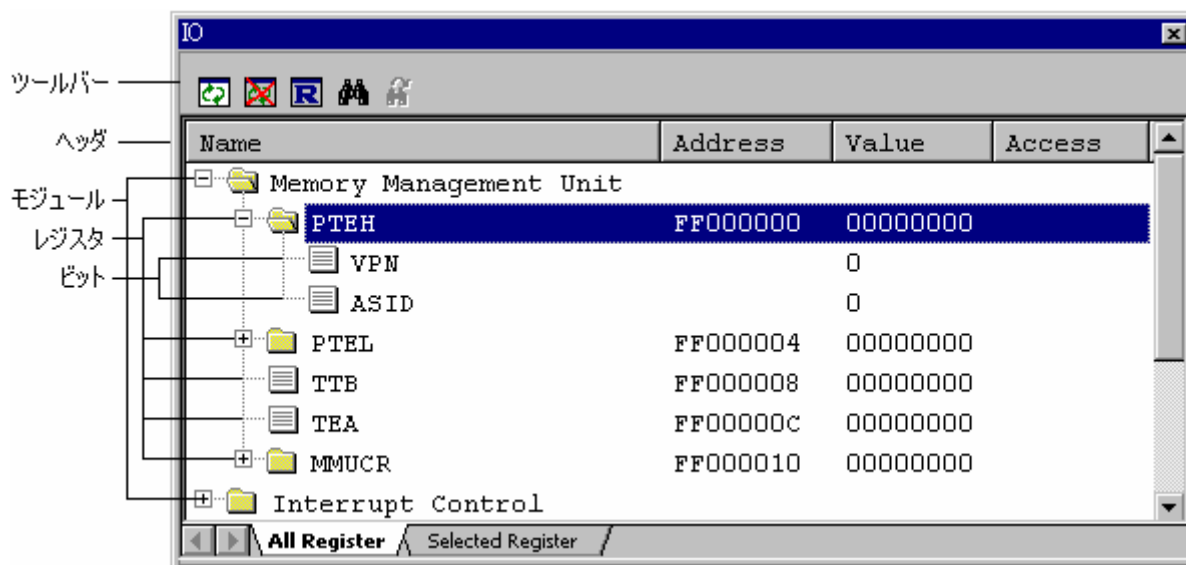
メモリウィンドウは、連続したメモリアドレスのデータをバイト、ワード、ロングワード、単精度浮動小数点、倍精度浮動小数点、または ASCII 値として表示できます。これに対し I/O レジスタは、非連続なメモリアドレスに異なるサイズでレジスタが割り付いているので、High-performance Embedded Workshop はこれらのレジスタを簡単に確認したり設定したりできるように[I/O]ウィンドウを提供します。

### 17.6.1 IOウィンドウを開く

[IO]ウィンドウを開くには、[表示 -> CPU -> IO]を選択するか、[IO の表示]ツールバーボタン()をクリックします。内蔵周辺と一致するモジュールが I/O レジスタ情報を構成します。

[IO]ウィンドウには、2つのタブ ([All Register]および[Selected Register])があります。[IO]ウィンドウを最初に関くと、[All Register]タブにモジュール名の一覧のみを表示します。

## ウィンドウの構成



### タブ

[All Register]  
[Selected Register]

### 説明

すべての I/O レジスタを表示します。  
選択した I/O レジスタを表示します。デフォルトでは、I/O レジスタを表示しません。

- I/O レジスタの表示を展開できます。
- I/O レジスタの値に更新があった場合、該当行のテキストを赤色で表示します。
- I/O レジスタまたはビット表示行をダブルクリックすると、値を設定するダイアログボックスが開きます。値の編集によりマクロを記録します。（「マクロの記録」）
- [Value]カラムでは、インプレース編集により I/O レジスタまたはビットの値を設定できます。（「マクロの記録」）

## ウィンドウのオプション

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	マクロの記録	機能
最新の情報に更新		-	ウィンドウの表示内容を手動で更新します。
表示固定		-	ウィンドウの表示内容の更新を抑制します。
IO ファイルのロード		-	[IO]ウィンドウに表示する制御レジスタの情報を I/O ファイルからロードします。
I/O レジスタの選択		-	[Selected Register]タブに表示する I/O レジスタを選択します。
印刷		-	現在の表示内容を印刷します。
ファイルに保存		-	現在の表示内容をテキストファイルに保存します。
検索		-	I/O レジスタを検索します。
次を検索		-	検索条件に一致する次の I/O レジスタを検索します。
ツールバー表示	-	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	-	ツールバーボタンをカスタマイズします。

### 17.6.2 I/Oレジスタ表示を展開する

I/Oレジスタの名前、アドレス、および値を表示するには、モジュール名の左側にあるプラス(+)の展開記号をクリックするか、クリックまたはカーソルを使用することによってモジュール名を選択し、'→'キーを押します。モジュールの表示が展開され、その周辺モジュールのそれぞれのレジスタおよびその名前、アドレス、および値を表示します（プラス記号がマイナス記号に変わります）。マイナス記号(-)の展開記号をクリックする（または'←'キーを押す）と、表示しているI/Oレジスタを閉じます。

ビットレベルで表示するには、同様にI/Oレジスタを展開します。

#### 注：

エミュレータデバッガを使用している場合は、I/Oレジスタからのリードがユーザプログラムの動作に影響することがあります。例えば、データレジスタのリードで割り込みをキャンセルできる場合があります。[IO]ウィンドウでI/Oモジュールを展開するとレジスタ値を表示するためにデータをリードします。I/Oモジュールを閉じている場合は、データをリードしないためユーザプログラムの動作に影響しません。また、メモリウィンドウ(または、逆アセンブリウィンドウ)でI/O領域を開くと同じ影響があることに注意してください。

[IO]ウィンドウの[表示固定]機能を使用すると、I/Oレジスタ表示を展開している場合もデータをリードしません。

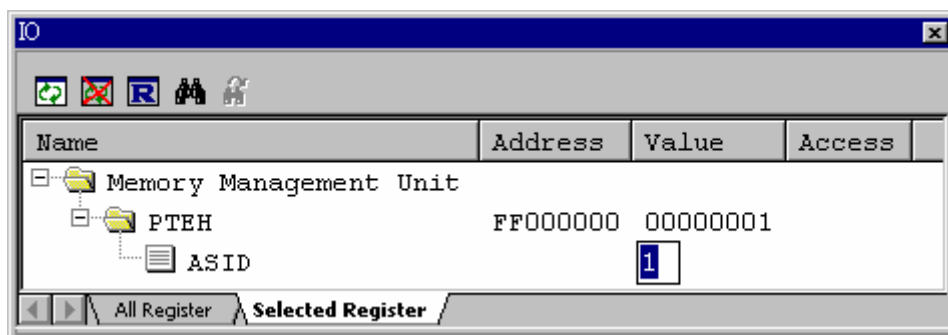
### 17.6.3 I/Oレジスタの内容を設定する

[IO]ウィンドウでI/Oレジスタの内容を設定します。

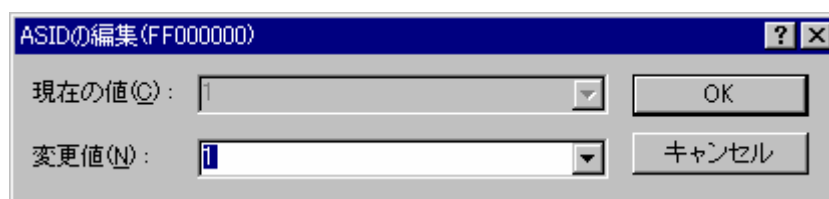
#### I/Oレジスタの内容を設定するには

[IO]ウィンドウでI/Oレジスタの内容を設定するには、次の方法があります。

- [Value]カラムでインプレース編集します。



- 変更するI/Oレジスタまたはビット表示行をダブルクリックし、I/Oレジスタ値を編集するダイアログボックスを開きます。[変更値]に値を入力してください。



### 17.6.4 ウィンドウの表示内容を手動で更新する

[IO]ウィンドウの表示内容を強制的に更新できます。

[IO]ウィンドウのポップアップメニューから[最新情報に更新]を選択してください。

### 17.6.5 ウィンドウの表示内容の更新を抑止する

ユーザプログラムの実行停止時などに、自動的に[IO]ウィンドウの表示内容を更新しないようにできます。

[IO]ウィンドウのポップアップメニューから[表示固定]を選択してください。

[IO]ウィンドウの内容をグレーで表示します。ただし、値が変更された I/O レジスタのテキストの色は変更しません。

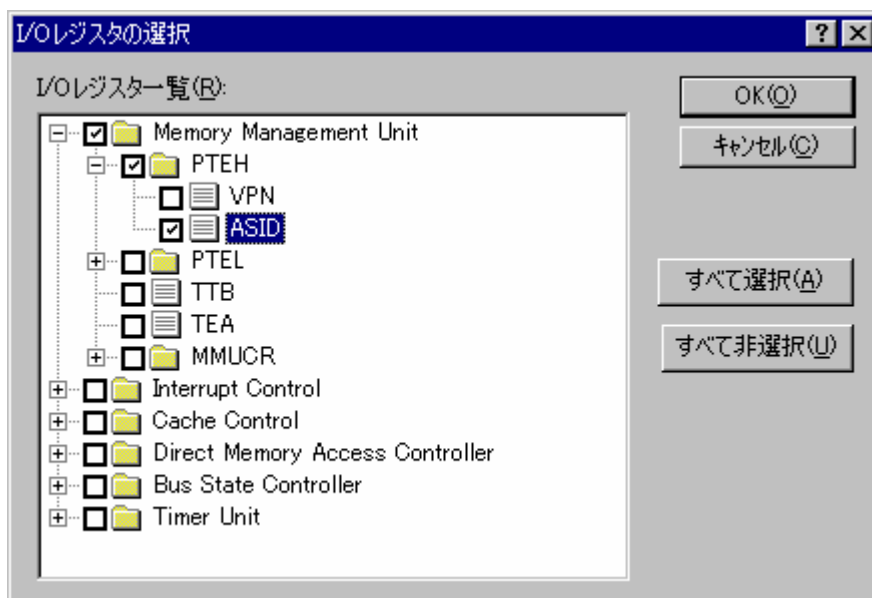
この機能が有効な場合、[IO]ウィンドウの[All Register]および[Selected Register]タブに表示している I/O レジスタの展開状態は変更できません。

### 17.6.6 表示するI/Oレジスタを選択する

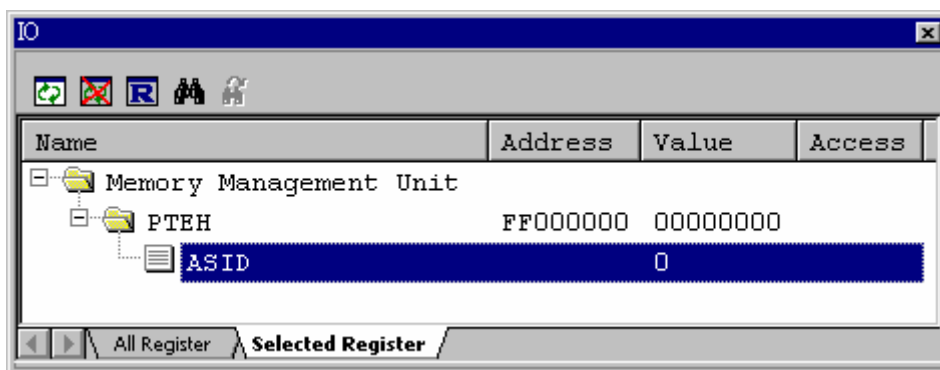
[IO]ウィンドウの[Selected Register]タブに I/O レジスタを選択して表示します。

#### 表示する I/O レジスタを選択するには

1. ウィンドウ内を右クリックしポップアップメニューを表示します。
2. [I/O レジスタの選択]を選択します。[I/O レジスタの選択]ダイアログボックスが開きます。以下の図は、SH-4 シミュレータデバッガを使用している場合の[I/O レジスタの選択]ダイアログボックスです。



3. [I/O レジスタ一覧]の各項目にはチェックボックスを表示します。チェックボックスはデフォルトでオフです。'+ 'が表示されている項目は、'+ 'のクリックによって展開できます。また、'- 'が表示されている項目は、'- 'のクリックによって折りたたむことができます。'+ または'- 'が表示されている項目のチェックボックスをオンにすると、その項目の要素のチェックボックスはすべてオンになります。オフにすると、その項目の要素のチェックボックスはすべてオフになります。表示する I/O レジスタのチェックボックスをオンにしてください。
4. OK をクリックすると、[I/O レジスタの選択]ダイアログボックスを閉じ、選択した I/O レジスタを [IO]ウィンドウの[Selected Register]タブに表示します。

**注：**


この機能はユーザプログラム実行中に使用できません。

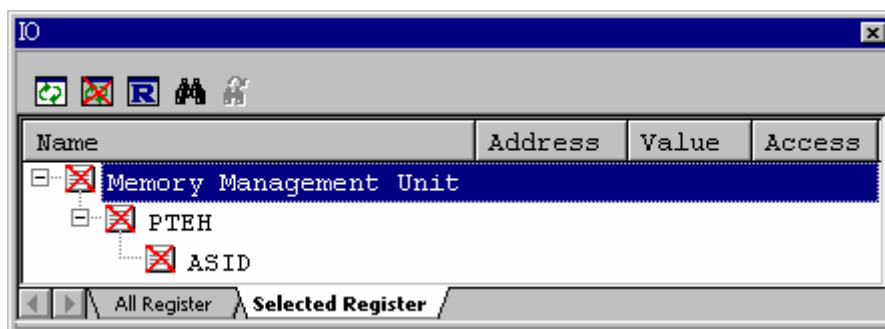
### 17.6.7 I/Oファイルをロードする

[IO]ウィンドウに I/O ファイルを手動でロードできます。

#### I/O ファイルをロードするには

1. ウィンドウ内で右クリックしポップアップメニューを表示します。
2. [I/O ファイルのロード]を選択します。[I/O ファイルを指定]ダイアログボックスが開きます。
3. [I/O ファイル]にロードしたい I/O ファイルを指定します。[I/O ファイル]にプレースホルダを挿入するには、プレースホルダボタンをクリックしポップアップメニューからプレースホルダを選択します。ファイルを参照するには[参照]ボタンをクリックします。
4. [I/O ファイルのパス名を保存する]チェックボックスをオフにすると、セッションを保存するときに [I/O ファイル]のパス名を保存しません。次回このセッションを選択したとき、[IO]ウィンドウへのロード内容はデバッガに依存します。自動的に I/O ファイルがロードされる場合は、このままこのチェックボックスをオフにしてください。CPU 選択に依存した I/O ファイルがデバッガにより自動的にロードされています。  
[I/O ファイルのパス名を保存する]チェックボックスをオンにすると、セッションを保存するときに [I/O ファイル]のパス名を保存します。次回このセッションを選択したときは前回ロードした I/O ファイルを [IO]ウィンドウにロードします。  
このチェックボックスのデフォルトはオフです。
5. [OK]ボタンをクリックします。

ロードした I/O ファイルのレジスタ名と、[IO]ウィンドウの[Selected Register]タブに表示しているレジスタ名が一致しない場合、以下のように  アイコンで示します。



I/O ファイルのフォーマットは「リファレンス 6. I/O ファイルフォーマット」を参照してください。

### 17.6.8 IOウィンドウの表示内容を印刷する

現在選択している[IO]ウィンドウのタブに表示している内容をテキスト形式で印刷できます。[IO]ウィンドウのポップアップメニューから[印刷]を選択してください。

### 17.6.9 IOウィンドウの表示内容を保存する

現在選択している[IO]ウィンドウのタブに表示している内容をテキストファイルに保存できます。[IO]ウィンドウのポップアップメニューから[ファイルに保存]を選択してください。

### 17.6.10 I/Oレジスタを検索する

[IO]ウィンドウに表示している I/O レジスタを検索できます。

#### I/O レジスタを検索するには

1. ウィンドウ内で右クリックしポップアップメニューを表示します。
2. [検索]を選択します。[検索]ダイアログボックスが開きます。



3. [検索カラム]から I/O レジスタを検索するカラムを選択してください。

Address	I/O レジスタアドレス
Name	I/O レジスタ名
4. 選択したカラムで検索するテキストを[検索値]に入力してください。大文字と小文字は区別しません。
5. [検索値]に入力したテキストと一致する I/O レジスタを検索する場合は、[一致検索]チェックボックスをオンにしてください。このオプションをチェックしない場合は、[検索値]に入力したテキストと一部でも一致する I/O レジスタを検索します。
6. [OK]ボタンをクリックすると先頭行から検索を開始します。該当行が強調表示されます。

次の該当箇所を検索する場合は、[IO]ウィンドウのポップアップメニューから[次を検索]を選択してください。

### 17.6.11 次を検索する


[IO]ウィンドウで I/O レジスタが検索できた後、ポップアップメニューから[次を検索]を選択すると、検索条件に一致する次の I/O レジスタを検索します。

## 17.7 レジスタ内容を見る

アセンブリ言語レベルでデバッグする場合に、CPUの汎用レジスタの内容を簡単に見ることができます。これは、レジスタウィンドウを使用して行います。

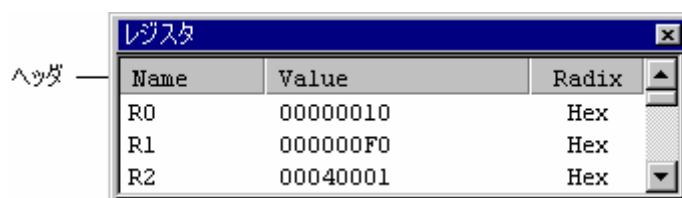
### 17.7.1 レジスタウィンドウを開く

レジスタウィンドウは、レジスタの内容やフラグの内容を表示するウィンドウです。ウィンドウからレジスタ/フラグの値を変更できます。

レジスタウィンドウを開くには、[表示 -> CPU -> レジスタ]を選択するか、[レジスタ]ツールバーボタン(  )をクリックします。レジスタウィンドウが開き、レジスタおよびその値を表示します。

#### ウィンドウの構成


以下の図は、SH-4用デバッガのレジスタウィンドウです。



- レジスタ表示行をダブルクリックすると、レジスタ値を設定するダイアログボックスが開きます。（「マクロの記録」）
- [Value]カラムでは、インプレース編集によりレジスタ値を設定できます。（「マクロの記録」）
- フラグに対応したボタンをクリックすることにより、フラグの値を切り替えることができます。（「マクロの記録」）

#### ウィンドウのオプション

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション		マクロの記録	機能
基数	16進数表示	-	16進数で表示します。
	10進数表示	-	10進数で表示します。
	8進数表示	-	8進数で表示します。
	2進数表示	-	2進数で表示します。
バンク 0 *		-	レジスタバンク 0 のレジスタを表示します。
バンク 1 *		-	レジスタバンク 1 のレジスタを表示します。
レイアウト	基数	-	基数カラムの表示/表示なしを切り替えます。
	フラグ	-	フラグ表示部の表示/表示なしを切り替えます。
	設定	-	表示するレジスタを選択します。
編集			レジスタの内容を設定します。
最新情報に更新		-	ウィンドウの表示内容を手動で更新します。
表示固定		-	ウィンドウの表示内容の更新を抑制します。
分割		-	ウィンドウを分割表示します。
ファイルに保存		-	レジスタウィンドウの表示内容を保存します。

#### 注：

\*. 機能のサポートはデバッガに依存します。



### 17.7.2 レジスタの表示基数を変更する

レジスタごとに表示基数が変更できます。

レジスタウィンドウのポップアップメニュー[基数]以下で表示基数を選択してください。

以下のいずれかが選択できます。

[16進数表示]	16進数で表示
[10進数表示]	10進数で表示
[8進数表示]	8進数で表示
[2進数表示]	2進数で表示

### 17.7.3 レジスタバンクを切り替える

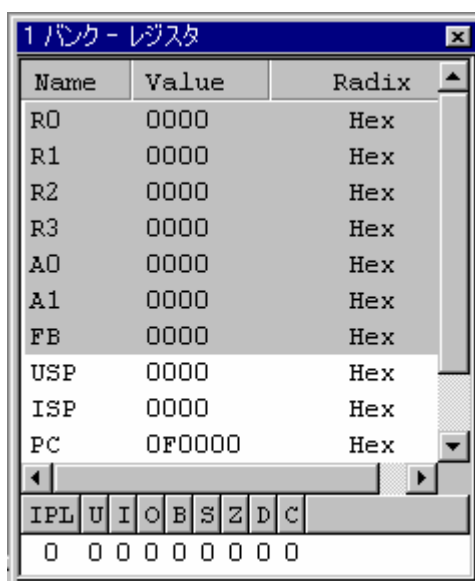
レジスタウィンドウを開いた直後は、フラグの値に対応したバンクのレジスタ内容を表示します。

バンクの切り替えは、レジスタウィンドウのレジスタ表示部上で右クリックすると表示されるポップアップメニュー、またはフラグのフラグ切り替えにより可能です(フラグの値を変更した場合は、その値に応じてレジスタバンクが切り替わります)。

バンク 1 のレジスタ内容を参照する場合は、ポップアップメニューから[バンク 1]を選択してください。

バンク 1 固有のレジスタは、背景がグレー表示になります。

以下の図は、M16C ファミリ用デバッガのレジスタウィンドウです。



バンク 0 のレジスタ内容を参照する場合は、ポップアップメニューから[バンク 0]を選択してください(オプション[バンク 0]、[バンク 1]の操作では、フラグの値は変化しません)。

機能のサポートはデバッガに依存します。

### 17.7.4 レイアウトを設定する

レイアウトを設定するには、レジスタウィンドウのポップアップメニュー[レイアウト]以下で選択してください。

下記から選択できます。

[基数]	[Radix]カラムの表示/表示なしの切り替え
[フラグ]	フラグ表示部の表示/表示なしの切り替え

表示している場合はオプションにチェックマークが付いています。

[Radix]カラム、およびフラグ表示部を表示する場合:

レジスタ									
Name	Value	Radix							
R0	00000010	Hex							
R1	000000F0	Hex							
R2	00040001	Hex							
MD	RB	BL	FD	M	Q	IMASK.I3	IMASK.I2		
1	1	1	1	1	1	1	1		
IMASK.I1	IMASK.I0	S	T	FR	SZ	PR	DN		
1	1	0	1	0	0	0	1		
CAUSE.E	CAUSE.V	CAUSE.Z	CAUSE.O	CAUSE.U					
0	0	0	0	0					
CAUSE.I	ENABLE.V	ENABLE.Z	ENABLE.O						
0	0	0	0						
ENABLE.U	ENABLE.I	FLAG.V	FLAG.Z	FLAG.O					
0	0	0	0	0					
FLAG.U	FLAG.I	RMI	RMO						
0	0	0	1						

### 17.7.5 表示するレジスタを選択する

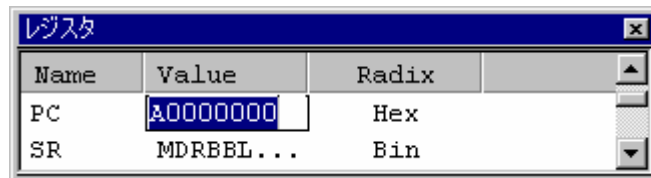
レジスタウィンドウのポップアップメニューから[レイアウト]以下で[設定]を選択してください。以下の図は、SH-4用デバッガの[設定]ダイアログボックスです。



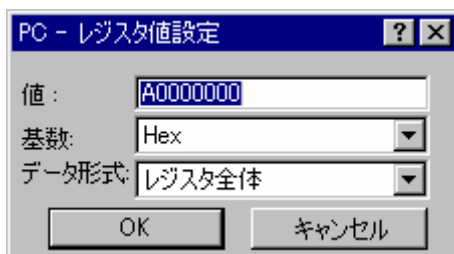
### 17.7.6 レジスタの内容を設定する

レジスタの内容を設定するには、次の方法があります。

- [Value]カラムでインプレース編集する。



- 以下のいずれかの方法でレジスタ変更ダイアログボックスを開き、変更する値を入力する。
  - 変更するレジスタ行をダブルクリックする。
  - 変更するレジスタを選択して、ポップアップメニューの[編集]を選択する。



[値]には数字または C/C++ の式を入力できます。

[基数]ドロップダウンリストから基数を選択できます。

[データ形式]ドロップダウンリストからオプションを選択して、レジスタの内容のすべて、マスクした領域、フローティングビットまたはフラグビットを修正できます(このリスト内容は、CPU および選択したレジスタによって異なります)。

新しい数字または式を入力したら、[OK]ボタンをクリックするか'Enter'キーを押します。ダイアログボックスは閉じて、新しい値をレジスタに書き込みます。

### 17.7.7 フラグの値を設定する

#### フラグ表示している場合:

変更するフラグのボタンをクリックしてください。クリックごとにフラグの状態 (0/1) が切り替わります。

複数ビットで構成されるフラグの場合は、変更値を入力するダイアログボックスが開きます。

以下の図は、SH-4 用デバッガのレジスタウィンドウです。

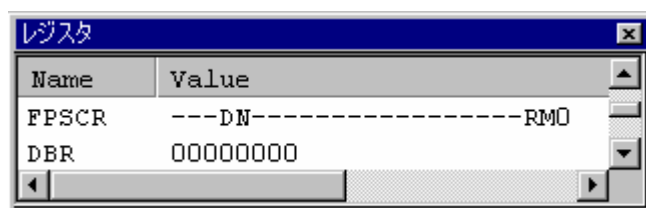


#### フラグをレジスタ表示している場合:

フラグ行をダブルクリックしてください。ダイアログボックスが開きます。

変更する値を入力してください。

以下の図は、SH-4用デバッガのレジスタウィンドウです。（フラグ行（FPSCR））



### 17.7.8 ウィンドウの表示内容を手動で更新する

レジスタウィンドウの表示内容を強制的に更新できます。

レジスタウィンドウのポップアップメニューから[最新情報に更新]を選択してください。

### 17.7.9 ウィンドウの表示内容の更新を抑止する

ユーザプログラムの実行停止時などに、自動的にレジスタウィンドウの表示内容を更新しないようにできます。

レジスタウィンドウのポップアップメニューから[表示固定]を選択してください。

レジスタウィンドウの内容をグレーで表示します。

ウィンドウを2枚開き、一方のウィンドウでこの機能を使用することにより、レジスタやフラグの値を比較できます。

### 17.7.10 ウィンドウを分割表示する

レジスタウィンドウを上下2分割で表示したいときは、ポップアップメニューから[分割]を選択後、分割バーを移動して分割します。

ポップアップメニューから[分割]を選択すると分割表示を解除します。

### 17.7.11 レジスタウィンドウの表示内容を保存する

レジスタウィンドウに表示しているデータをテキスト形式でファイルへ保存するには、レジスタウィンドウのポップアップメニューから[ファイルに保存]を選択します。[名前を付けて保存]ダイアログボックスを開きます。ファイル名を指定してください。

### 17.7.12 レジスタの内容を使用する

逆アセンブリまたはメモリウィンドウのアドレス指定など、値を入力するときにレジスタ値を使用できます。"#PC"または"%PC"のように、レジスタ名のプレフィックスに"#記号または"%記号を付けます。サポートするプレフィックスはデバッガに依存します。

### 17.7.13 テキストの色を変更する

レジスタウィンドウはレジスタおよびフラグの値を表示しているテキストの色およびフォントを他のウィンドウと同じように[表示形式]ダイアログボックスでカスタマイズできます。

レジスタウィンドウではこれに加えて、レジスタおよびフラグの値が変更された部分の色を、[表示形式]ダイアログボックスでカスタマイズできます。レジスタおよびフラグの値が変更された部分のデフォルトの色は、前面が赤色で背景が白色です。背景色は、[Text]カテゴリ選択時の背景色と同じ色になります。

#### 値が変更された部分の色をカスタマイズするには

1. [基本設定 -> 表示の形式]を選択し、[表示形式]ダイアログボックスを開きます。
2. [レジスタ]をダイアログボックスの左側のツリーから選択し展開します。
3. [変更]カテゴリを選択します。
4. [カラー]タブの[前面]リストの選択を変更します。
5. [OK]ボタンをクリックします。





### 17.8 基数を設定する

ダイアログボックス、ウィンドウ、またはコマンドで、値を数値で入力するときに"0x"などのプレフィックスを付けなかった場合、入力した数値を処理する基数を設定します。


#### 注：

- ダイアログボックスまたはウィンドウ内に基数用のフィールドがある場合は、その基数の指定に従います。
- ダイアログボックス、ウィンドウ、またはコマンドによっては、入力した数値を処理する基数が固定されている場合があります。

基数を設定するには、以下に示す[基本設定->基数]のサブメニューを選択するか、[デバッグ]ツールバーのボタンをクリックしてください。


[基本設定 -> 基数]サブメニュー	[デバッグ]ツールバー	機能
16 進数		入力した数値を処理する基数を 16 進数に設定します。(デフォルト)
10 進数		入力した数値を処理する基数を 10 進数に設定します。
8 進数		入力した数値を処理する基数を 8 進数に設定します。
2 進数		入力した数値を処理する基数を 2 進数に設定します。

### 17.9 ターゲットマイコンをリセットする

ターゲットマイコンをリセットするには、[CPU リセット]ツールバーボタン()をクリックするか、[デバッグ]メニューから[CPU のリセット]を選択します。

ターゲットマイコンをリセットすることにより、内蔵 I/O レジスタの初期化、およびリセットベクタに設定されたアドレスを PC に設定します。

### 17.10 カーソル位置にPC値を設定する

PC を現在のテキストカーソルの位置によって示されるアドレスに設定するには、[カーソル位置に PC 設定]ツールバーボタン()をクリックするか、[デバッグ -> カーソル位置に PC 値を設定]を選択します。


### 17.11 デバッグを初期化する

[デバッグ -> 初期化]を選択します。


現在開いているウィンドウを閉じ、デバッグとの接続を解除します。そして、デバッグの再接続をします。

### 17.12 デバッグを接続/接続解除する

**デバッグを接続するには**

1. [接続]ツールバーボタン()をクリックするか、[デバッグ]メニューから[接続]を選択します。

**デバッグを接続解除するには**


1. [接続解除]ツールバーボタン()をクリックするか、[デバッグ]メニューから[接続解除]を選択します。

機能のサポートはデバッグに依存します。


### 17.13 プログラムを実行する


この節では、作成したプログラムコードの実行方法について説明します。ここでは、プログラムを連続して実行させたり、シングルステップ実行を行ったり、同時に複数の命令を実行させたりします。

#### 17.13.1 実行を継続する

作成したプログラムが停止すると、High-performance Embedded Workshop は CPU の現在のプログラムカウンタ (PC) アドレス値に対応するエディタおよび逆アセンブリウィンドウの行の左余白に黄色の矢印 (  ) を表示します。ステップ実行を行った場合、または実行を続けた場合、この命令を次に実行します。


**現在の PC アドレスから実行を継続するには**


- [実行]ツールバーボタン()をクリックする。
- [デバッグ -> 実行]を選択する。

前回停止時と異なるアドレスから実行を継続するには、下記の方法で PC を変更後に[実行]ツールバーボタン () をクリックするか、[デバッグ -> 実行]を選択します。

- レジスタウィンドウ上で変更する。
- エディタウィンドウまたは逆アセンブリウィンドウ上で、テキストカーソル(マウスカーソルではありません)を変更したい行に移動してポップアップメニューから[カーソル位置に PC 値を設定]を選択する。

#### 17.13.2 リセットから実行を開始する

ターゲットマイコンをリセットしてリセットベクタアドレスからプログラムを実行させるには、[リセット後実行]ツールバーボタン() をクリックするか、[デバッグ -> リセット後実行]を選択します。

プログラムは、ブレークポイントにヒットするまで、またはブレーク条件が成立するまで実行を続けます。プログラムの実行はいつでも手動で停止できます。その方法としては、[停止]ツールバーボタン()をクリックするか、[デバッグ -> プログラムの停止]を選択します。


注：

プログラムはリセットベクタ位置に格納したアドレスから実行を開始します。したがって、この位置に自分のスタートアップコードのアドレスを含んでいることを確認することが重要です。

### 17.13.3 ブレークポイントを無視して実行する

アプリケーションのデバッグ中、続けて実行を開始するとき、ブレークポイントの設定を保持したい場合があります。フリー実行機能を使用することにより、すべてのブレークポイントを無効に設定したとみなしてコードを直ちに実行できます。

現在の PC アドレスからブレークポイント (ソフトウェアとハードウェア) を無視して実行するには

- [フリー実行]ツールバーボタン()をクリックする。
- [デバッグ -> フリー実行]を選択する。

注：

機能のサポートはデバッグに依存します。

### 17.13.4 カーソルまで実行する


アプリケーションを実行している途中で、シングルステップ実行を複数回行うだけの比較的小さいセクションコードのみを実行したいと考える場合があります。この場合特定のポイントまで実行できると便利です。これは、[カーソル位置まで実行]機能を使用して行うことができます。

[カーソル位置まで実行]を使用するには

1. エディタウィンドウまたは逆アセンブリウィンドウが開いていて、プログラムを停止するアドレスを表示していることを確認します。
2. プログラムを停止するアドレスの行にテキストカーソルを置いてください。
3. ポップアップメニューから[カーソル位置まで実行]を選択します。

デバッグは作成したコードを現在の PC 値から実行し、カーソル位置が示すアドレスまで実行します。

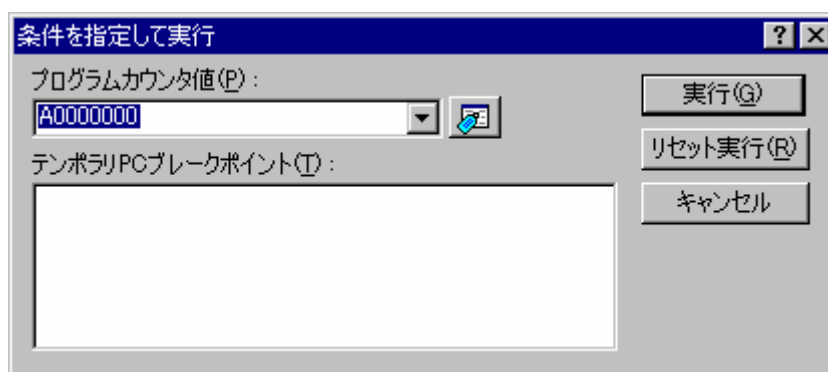
注：

- 作成したプログラムがこのアドレスのコードを実行しない場合、プログラムは停止しません。その場合、コードの実行を中止するには、'Esc キー'を押すか、[デバッグ -> プログラムの停止]を選択するか、[停止]ツールバーボタン()をクリックします。
- [カーソル位置まで実行]はテンポラリブレークポイント機能を利用しています。既にブレークポイントを最大数利用している場合はこの機能を利用できません。

### 17.13.5 条件を指定して実行する

[条件を指定して実行]ダイアログボックスを利用すると、任意のアドレスから命令を実行できます。[条件を指定して実行]ダイアログボックスを開くには、[デバッグ -> 条件を指定して実行]を選択します。





[条件を指定して実行]ダイアログボックスでは命令実行の条件を設定します。

[プログラムカウンタ値] 実行を開始する命令アドレスを設定します。初期値は現在の PC 値となります。

[テンポラリ PC ブレークポイント] 一時的な PC ブレークポイントを設定します。ダイアログボックスによる命令実行が停止すると、このブレークポイントは解除されます。

[実行]ボタンをクリックすると、設定した内容にしたがって命令を実行します。

[リセット実行]ボタンをクリックすると、リセットベクタから命令を実行します。

[キャンセル]ボタンをクリックすると、命令を実行しないで、ダイアログボックスを閉じます。

#### 注：

[テンポラリ PC ブレークポイント]はテンポラリブレークポイント機能を利用しています。既にブレークポイントを最大数利用している場合はこの機能を利用できません。

### 17.13.6 リセット時にmain関数の先頭までプログラムを実行する

High-performance Embedded Workshop は、リセット時にソフトウェアブレークポイントを設定せずに main 関数の先頭までプログラムを実行する機能があります。

この機能は main 関数に一時的にソフトウェアブレークポイントを設定します。プログラムの実行停止後にソフトウェアブレークポイントを解除します。

#### リセット時に main 関数の先頭までプログラムを実行するには

1. [基本設定 -> オプション]を選択し、[オプション]ダイアログボックスを開きます。
2. [デバッグ]タブを選択します。
3. [リセット時に main 関数の先頭まで実行]チェックボックスをオンにします。デフォルトはオフです。
4. ターゲットマイコンをリセットします。以下のいずれかの操作を選択してください。
  - [CPU リセット]ツールバーボタン(🔧)をクリックする。
  - [デバッグ -> CPU のリセット]を選択する。

#### 注：


[リセット時に main 関数の先頭まで実行]はテンポラリブレークポイント機能を利用しています。既にブレークポイントを最大数利用している場合はこの機能を利用できません。

### 17.13.7 シングルステップ

作成したコードをデバッグするために、一度に1行だけまたは1つの命令だけステップ実行して、この命令がシステムにどのように影響するかを確認したい場合、ソース表示では、ソースライン1行だけをステップ実行します。逆アセンブリ表示では、アセンブリ言語命令単位にステップ実行します。命令が他の関数またはサブルーチンを呼び出す場合、オプションでその関数にステップインまたはステップオーバできます。命令が呼び出しを実行しない場合は、いずれのオプションでも、その命令を実行し、次の命令で停止します。


関数にステップイン実行を選択した場合は、関数の呼び出しを実行しその関数の最初の行または命令で停止します。

#### 関数にステップイン実行するには

- [ステップイン]ツールバーボタン()をクリックする。
- [デバッグ -> ステップイン]を選択する。


関数をステップオーバ実行を選択した場合は、呼び出しおよびその関数(およびその関数が実行するすべての関数呼び出し)を実行して、呼び出し元の関数の次の行または命令で停止します。

#### 関数をステップオーバ実行するには

- [ステップオーバ]ツールバーボタン()をクリックする。
- [デバッグ -> ステップオーバ]を選択する。

関数内の確認したい命令の実行が終了した場合や、誤って関数にステップイン実行した場合に、ステップアウト機能を使用すると関数内の残りのコードをすべてステップイン実行することなく呼び出し元の関数に戻ることができます。

#### 関数からステップアウト実行するには

- [ステップアウト]ツールバーボタン()をクリックする。
- [デバッグ -> ステップアウト]を選択する。

#### ステップモードを選択するには

[デバッグ->ステップモード]から次のサブメニューを選択します。

サブメニュー	機能
[自動] (デフォルト)	エディタウィンドウがアクティブの場合はソースライン1行だけをステップ実行します。逆アセンブリウィンドウがアクティブの場合はアセンブリ言語命令単位にステップ実行します。
[アセンブリ]	アセンブリ言語命令単位にステップ実行します。
[ソース]	ソースライン1行だけをステップ実行します。

High-performance Embedded Workshop エディタのソース表示でステップインしている際、デバッグ情報がないライブラリ関数 (printf 等) はソース表示のままステップしたい場合があります。デフォルトでは逆アセンブリウィンドウへジャンプし、アセンブリ言語命令単位にステップします。

High-performance Embedded Workshop エディタのソース表示でデバッグ情報がないアドレスにステップインした場合、アセンブリ言語命令単位にステップ実行しないようにできます。

### デバッグ情報のないアドレスへステップインしないようにするには

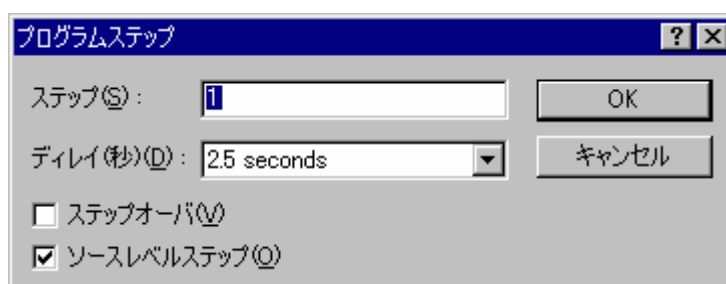
1. [基本設定 -> オプション]を選択し、[オプション]ダイアログボックスを開きます。
2. [デバッグ]タブを選択します。
3. [デバッグ情報のないアドレスにステップインしない]チェックボックスをオンにします。チェックボックスのデフォルトはオフです。

注：

機能のサポートはデバッガに依存します。

### 17.13.8 複数のステップ


[プログラムステップ]ダイアログボックスを使用することにより、一度に複数のステップ実行ができます。ステップ間の時間差を選択し、自動的にステップ実行するよう設定できます。[デバッグ -> ステップ]を選択してください。[プログラムステップ]ダイアログボックスを開きます。





[ステップ]	連続実行するステップ数を設定します。デフォルトの値は、1です。
[ディレイ (秒)]	コードを自動的にステップ実行するときのステップ間の遅延を選択します。デフォルトの値は、2.5 seconds です。 [No Refresh] (各ウィンドウの更新をしない) または 0~3 秒まで、0.5 秒単位で選択できます。
[ステップオーバ]	チェックすると関数呼び出しをステップオーバします。チェックボックスのデフォルトはオフです。
[ソースレベルステップ]	チェックするとソースレベルでステップします。チェックボックスのデフォルトはオンです。

[OK]ボタンをクリックするか、'Enter'キーを押してステップ実行を開始します。


### 17.14 プログラムを停止する

この節では、作成したプログラムの実行を停止する方法を説明します。停止手段として、[停止]ツールバーボタン(  )を使用して停止する方法、および作成したコードの特定の場所にブレークポイントを設定することによって停止する方法について説明します。

#### 17.14.1 停止ツールバーボタンによる停止

作成したプログラムが実行中の場合、[停止]ツールバーボタン(  )を使用できます。プログラムが停止している場合は、[停止]ツールバーボタン(  )はグレー表示となり使用できません。

## プログラムを停止するには


- [停止]ツールバーボタン()をクリックする。
- [デバッグ -> プログラムの停止]を選択する。

プログラムが停止すると、停止要因などの情報がアウトプットウィンドウの[Debug]タブに表示されます。

### 17.14.2 標準のブレークポイント (PCブレークポイント)

作成したプログラムをデバッグする場合、PCブレークポイントにより指定した行または命令でプログラムの実行を停止させることができます。標準的なPCブレークポイントを設定したり解除したりする方法を以下に示します。

#### エディタウィンドウ上でPCブレークポイントを設定するには

1. PCブレークポイントを設定する位置のエディタウィンドウまたは逆アセンブリウィンドウが開いていることを確認します。
2. プログラムを停止したい行でポップアップメニューの[ブレークポイントの挿入/削除]を選択するか、"F9"キーを押します。左余白に赤丸()を表示します。これは、PCブレークポイントを設定したことを示します。
3. ポップアップメニューの[ブレークポイントの有効化/無効化]を選択すると、現在設定しているブレークポイントの有効/無効の切り替えができます。
4. 作成したプログラムを実行してPCブレークポイントを設定したアドレスに達すると、"PC Breakpoint"というメッセージをアウトプットウィンドウの[Debug]タブに表示し、実行を中断し、エディタウィンドウまたは逆アセンブリウィンドウを更新し、PC位置の行をハイライト表示します。

#### 補足:

テンプレート関数の実体が複数存在する場合、同一ソース行に対して複数のアドレスにブレークポイントを設定できます。

```
template <typename T> T1 tempfunc(T t1)
{
    g_iVal += t1; -> ここにブレークポイントを設定すると、tempfunc(c)とtempfunc(s)
                   に対してブレークポイントを設定できます。
    return g_iVal;
}
void main(void)
{
    char c;
    short s;
    ...
    tempfunc(c);
    tempfunc(s);
}
```

ブレークポイントを設定するときに、すべてのアドレスにブレークポイントを設定するかどうかを確認するダイアログボックスが開きます。すべてのアドレスにブレークポイントを設定する場合は「はい」を選択してください。設定しない場合は「いいえ」を選択してください。

確認要求ダイアログボックスを表示しない場合、[今後表示しない]チェックボックスをオンにしてください。

#### 確認要求ダイアログボックスを再び表示するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスが開きます。

2. [確認]タブを選択してください。
3. [複数ブレークポイントの設定]チェックボックスをオンにしてください。このチェックボックスのデフォルトはオンです。
4. OK をクリックしてください。

#### 注：

ブレーク発生時には、PC ブレークポイントを設定した行または命令を実行する直前で停止します。その PC ブレークポイントで停止した後に Go または Step を選択した場合、矢印で表示した行から実行します。

#### [ブレークポイント]ダイアログボックスを使用して PC ブレークポイントの設定を編集するには

PC ブレークポイントの設定を編集するには、[編集 -> ソースブレークポイント]メニューを選択し、[ブレークポイント]ダイアログボックスを開きます。

- このダイアログボックスに現在設定している PC ブレークポイントを表示します。
- [コードの編集]ボタンにより PC ブレークポイントが存在するソースを見ることができます。
- [削除]、[すべて削除]ボタンにより、PC ブレークポイント 1 つまたはすべてを削除できます。
- 各ブレークポイントのチェックボックスにより有効 / 無効の切り替えができます。

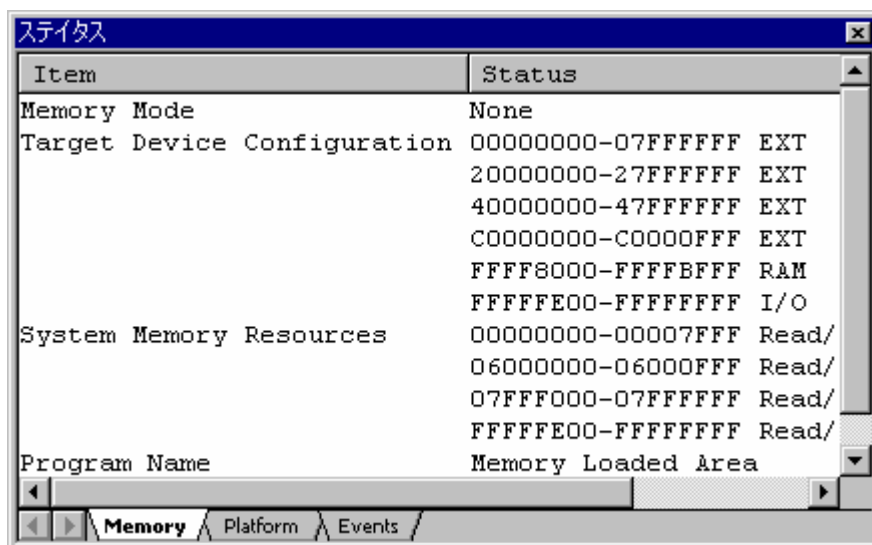
#### PC ブレークポイントを切り替えるには

PC ブレークポイントを設定したり、解除したりするには、PC ブレークポイントを設定している行の[BP]カラムをダブルクリックするか、行にカーソルを置き、"F9"キーを使用します。その行で使用可能な標準のブレークポイントが有る場合、色の付いた丸印(●)を左余白に表示します。

### 17.15 現在の状態を表示する

デバッガの現在の状態を知るにはステータスウィンドウを使用します。ステータスウィンドウを開くには、[表示 -> CPU -> ステータス]を選択するか、[ステータスの表示]ツールバーボタン(📄)をクリックします。ステータスウィンドウには、3 つのタブがあります。

#### ウィンドウの構成




タブ	説明
[Memory]	メモリマッピングおよび現在ロードしたオブジェクトファイルが使用するメモリエリアなど、現在のメモリステータスに関する情報を含んでいます。
[Platform]	CPU シリーズおよび動作モードなど、デバッガのステータス情報、実行状態および実行統計情報を含んでいます。
[Events]	リソース情報およびブレークポイント等のイベント情報に関する情報を含んでいます。

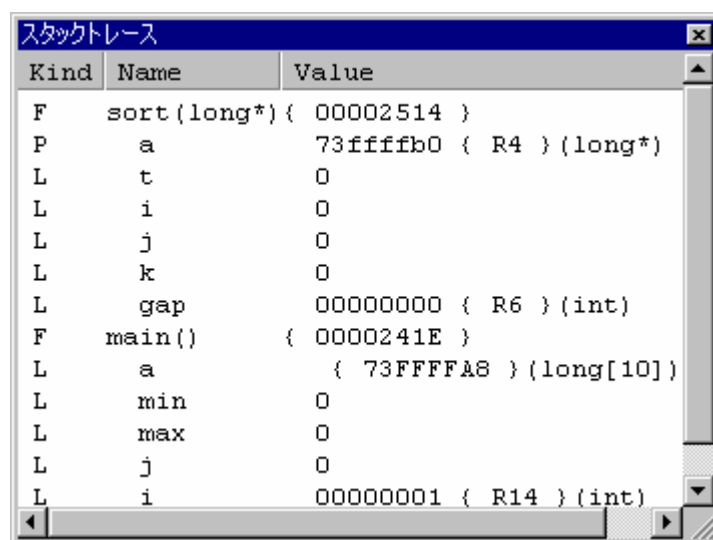
## 17.16 関数呼び出し履歴を見る

関数呼び出し履歴を表示する場合は、スタックトレースウィンドウを使用します。

### 17.16.1 スタックトレースウィンドウを開く

スタックトレースウィンドウを開くには、[表示 -> コード -> スタックトレース]を選択するか、[スタックトレース]ツールバーボタン(  )をクリックします。

#### ウィンドウ構成



表示する項目は以下の通りです。

[Kind]	該当シンボルのシンボル種別を示します。 F: 関数 P: 関数パラメータ * L: ローカル変数 *
[Name]	シンボル名を示します。
[Value]	シンボルの値、アドレス、型を示します。

#### 注:

\*. 機能のサポートはデバッガに依存します。

#### ウィンドウのオプション

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

ポップアップメニューオプション	マクロの記録	機能
ソースファイル表示	-	選択した関数に該当するソースコードを表示します。
表示設定	-	スタックトレースウィンドウの表示形式を設定します。
コピー	-	選択部分をコピーし Windows® クリップボードに貼り付けます。

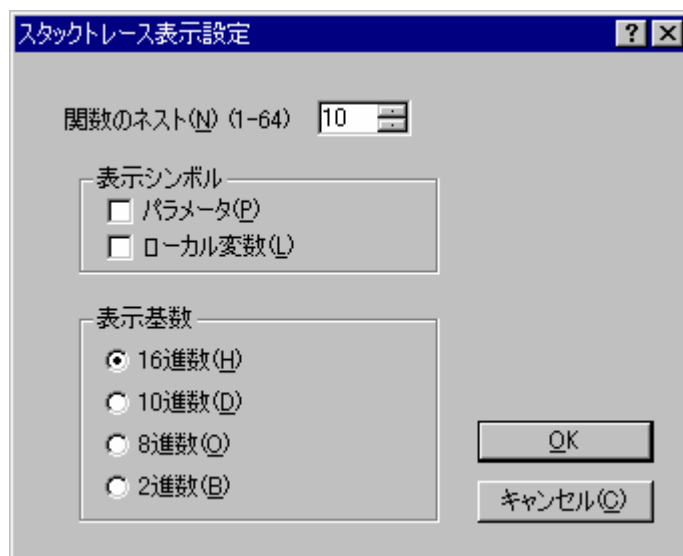
### 17.16.2 ソースコードを表示する

関数を選択した状態で、ポップアップメニューから[ソースファイル表示]を選択すると、エディタウィンドウを開いて選択した関数に該当するソースコードを表示します。

### 17.16.3 表示形式を設定する

ポップアップメニューから[表示設定]を選択すると、[スタックトレース表示設定]ダイアログボックスを開きます。

スタックトレースウィンドウの表示形式を設定します。



[関数のネスト]           スタックトレースウィンドウに表示する関数コールネスト数を指定します。

[表示シンボル] \*       関数以外に表示するシンボルを指定します。

[表示基数] \*           スタックトレースウィンドウの表示基数を指定します。

注：

\*. 機能のサポートはデバッガに依存します。

### 17.16.4 エンコード形式を設定する

変数値が文字、文字列の場合に UTF-8 として表示するには[基本設定 -> オプション]を選択し、[オプション]ダイアログボックスを開きます。

[デバッグ]タブを選択し、[エンコード形式]を[UTF-8]にします。

これで、スタックトレースウィンドウは変数の値を UTF-8 として表示します。

[エンコード形式]を[ローカルコードページ] (デフォルト) にした場合は、ASCII およびシフト JIS コードとして表示します。

注:

スタックトレースウィンドウではウィンドウ内に表示するフォントのデフォルトが"Courier New"となっており、UTF-8 やシフト JIS コードで表現される日本語文字等に対応していません。

これらの文字コードを表示する場合は[表示形式]ダイアログボックスで日本語文字に対応したフォント(MS Pゴシックなど)に変更してご使用ください。



## 17.17 外部デバッガを使う


High-performance Embedded Workshop は外部デバッガツールを起動できます。他のデバッガを使用する場合は、[ツール]メニューに追加する必要があります。

[カスタマイズ]ダイアログボックスの[デバッガ]タブに外部デバッガに関連する情報を設定します。一部のターゲットが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できます。[基本設定 -> カスタマイズ]を選択してダイアログボックスを開き、[デバッガ]タブを選択してください。

最初に、[外部デバッガの選択]ドロップダウンリストで使用する外部デバッガを選択します。

外部デバッガを使用する場合、そのデバッガ固有の情報を構築する必要があります。

[Hitachi Debugger Interface (version 4.x or greater)]	日立デバッグインタフェースを High-performance Embedded Workshop と統合して構築する
[Renesas PD debugger]	PD デバッガを High-performance Embedded Workshop と統合して構築する
[Other external debugger]	日立デバッグインタフェースまたは PD デバッガ以外の外部デバッガを High-performance Embedded Workshop と統合して構築する
[Non selected]	外部デバッガを使用しない

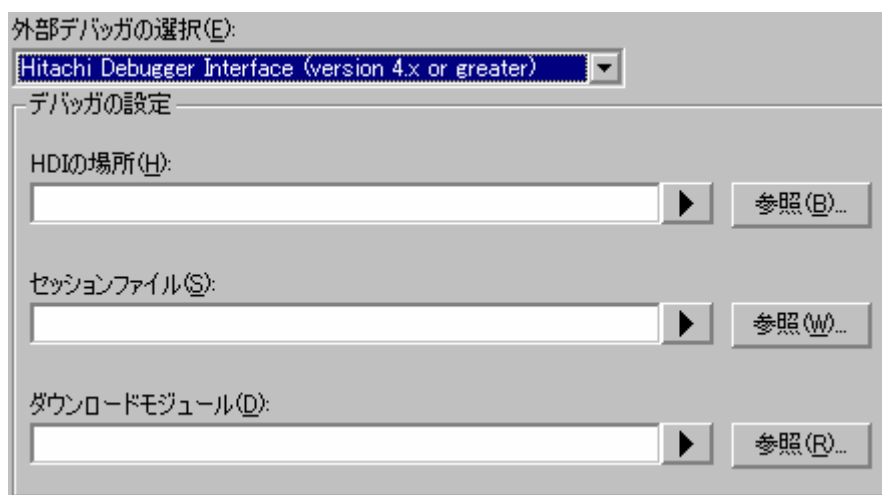
外部デバッガを起動するには、[外部デバッガの実行]ツールバーボタン(  )をクリックしてください。

ビルド後にダウンロードモジュールが変更された場合、High-performance Embedded Workshop から外部デバッガに切り替わり、すぐデバッグできるようになります。

外部デバッガを使用しているときにエディタウィンドウのどれかをダブルクリックすると High-performance Embedded Workshop に切り替わり、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。[Other external debugger]を選択した場合は、この操作を行うことはできません。

### 17.17.1 日立デバッグインタフェースをHigh-performance Embedded Workshopと統合して構築する

日立デバッグインタフェースを High-performance Embedded Workshop と統合し、High-performance Embedded Workshop の外部デバッガのオプションで起動できるように設定するために必要な情報を以下に示します。



### 日立デバッグインタフェースを High-performance Embedded Workshop と統合して構築するには

1. 日立デバッグインタフェースの場所を指定します。日立デバッグインタフェース 4.0 以降のバージョンでない場合この動作は保証されません。
2. セッションファイルの場所を指定します。起動時にどのセッションをロードするか日立デバッグインタフェースに伝えます。
3. ダウンロードモジュールの場所を指定します。

### 17.17.2 PD デバッガを High-performance Embedded Workshop と統合して構築する

PD デバッガを High-performance Embedded Workshop と統合し、High-performance Embedded Workshop の外部デバッガのオプションで起動できるように設定するために必要な情報を以下に示します。

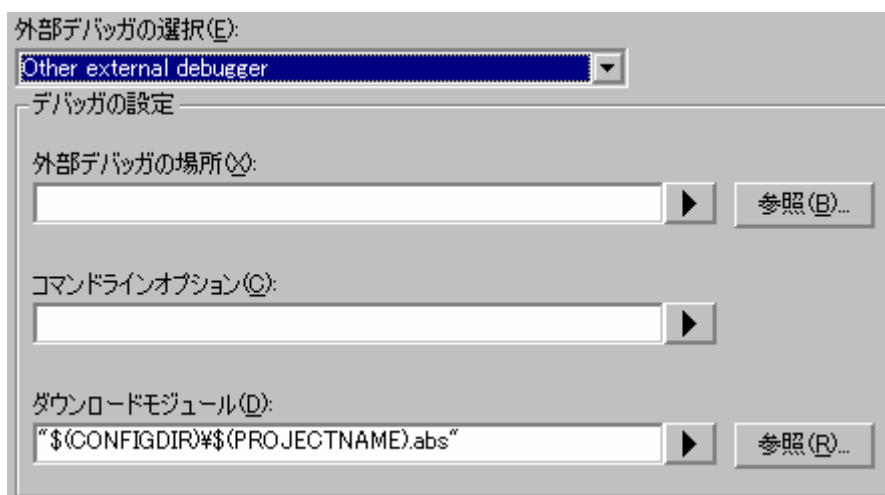


### PD デバッガを High-performance Embedded Workshop と統合して構築するには

1. PD デバッガの場所を指定します。
2. PD Profile ファイルの場所を指定します。起動時にどのセッションをロードするか PD デバッガに伝えます。
3. コマンドラインオプションです。PD デバッガの動作を変更するオプションを指定できます。
4. ダウンロードモジュールの場所を指定します。

### 17.17.3 外部デバッガを High-performance Embedded Workshop と統合して構築する

日立デバッグインタフェースまたは PD デバッガ以外の外部デバッガを High-performance Embedded Workshop と統合し、High-performance Embedded Workshop の外部デバッガのオプションで起動できるように設定するために必要な情報を以下に示します。



### 外部デバッガを High-performance Embedded Workshop と統合して構築するには

1. 外部デバッガの場所を指定します。
2. コマンドラインオプションです。外部デバッガの動作を変更するオプションを指定できます。
3. ダウンロードモジュールの場所を指定します。

## 17.18 デバッガに依存する機能

この節に記載している機能は、連携ツールの製品パッケージにバンドルされている High-performance Embedded Workshop で共通のデバッグ機能ではありません。旧バージョンのユーザーズマニュアルおよびヘルプに記載していたデバッグ機能です。

ご使用のデバッガで利用可能な機能の詳細説明は、エミュレータまたはシミュレータのユーザーズマニュアルおよびヘルプを参照してください。

### 17.18.1 ラベルを見る

デバッグ情報を含んだユーザプログラムをロードした時に、ラベルを同時に登録します。ラベルを追加することも可能です。

逆アセンブリウィンドウにおいては、対応するアドレスの代わりとして、また命令オペランドの一部として、ラベルの最初の 8 文字を表示します。

#### 注：

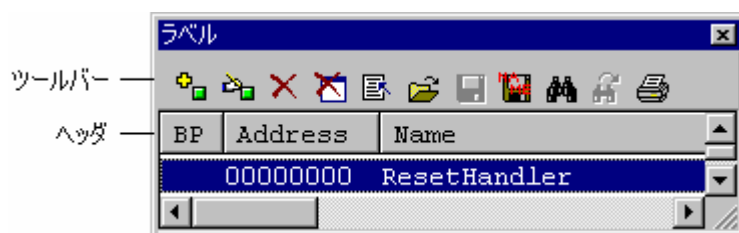
オペランドとラベルの値が一致すれば、命令のオペランドはラベル名に置き換わります。同じ値を持つラベルが 2 つ以上ある場合、アルファベット順で先に来るラベルを表示します。

ダイアログボックスで、アドレスまたは値を入力できる場合には、ラベルを使用することもできます。

#### 17.18.1.1 ラベルを一覧にする

現在のデバッグセッションに定義したラベルのすべてを見るには、[表示 -> シンボル -> ラベル]を選択するか、[ラベルの表示]ツールバーボタン(🔍)をクリックします。

## ウィンドウ構成



- 各カラムのヘッダをクリックするとアルファベット順（ASCIIコードによって）またはアドレス値でソートしたシンボルを表示できます。
- [BP]カラムでダブルクリックすると、関数の入り口でソフトウェアブレークポイントを設定したり解除したりできます。（「マクロの記録」）

## ウィンドウのオプション

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

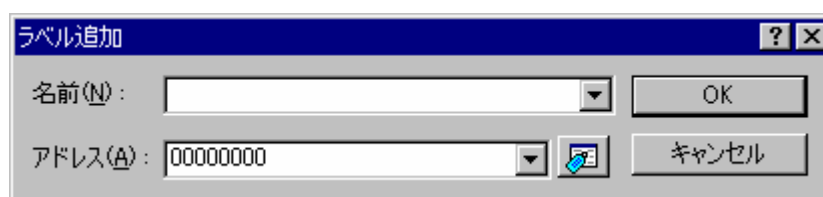
本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	マクロの記録	機能
追加		●	ラベルを追加します。
編集		-	ラベルを編集します。
削除		●	ラベルを削除します。
すべてを削除		●	すべてのラベルを削除します。
ロード		●	ラベルをファイルからロードします。
上書き保存		-	ラベルをシンボルファイルに上書き保存します。
名前を付けて保存		-	ラベルをファイルに保存します。
検索		-	ラベルを検索します。
次を検索		-	ラベルが検索できた後、検索条件に一致する次のラベルを検索します。
ソースを表示		-	ラベルに対応するソースプログラムを表示します。
印刷		-	表示内容を印刷します。
ツールバー表示	-	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	-	ツールバーボタンをカスタマイズします。

### 17.18.1.2 ラベルを追加する

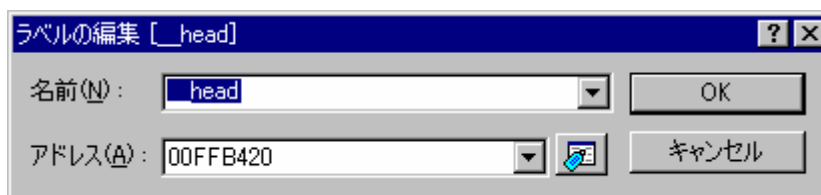
ラベルを追加するには、ポップアップメニューから[追加]を選択し[ラベル追加]ダイアログボックスを開きます。



新しいラベル名を[名前]に入力し、対応する値を[アドレス]に入力して[OK]ボタンを押します。[ラベル追加]ダイアログボックスが閉じ、ラベルリストに新しいラベルを追加、更新します。多重定義関数やクラス名を入力したときは、[関数選択]ダイアログボックスが開くので、関数を選択して[アドレス]を設定します。詳細は「17.18.2.3 複数ラベルをサポートする」を参照してください。

### 17.18.1.3 ラベルを編集する

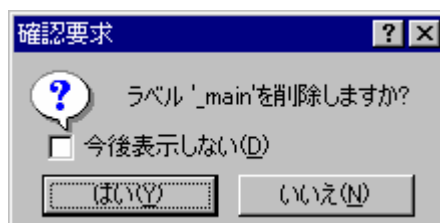
ラベルを編集するには、ポップアップメニューから[編集]を選択し[ラベルの編集]ダイアログボックスを開きます。



ラベル名と対応する値を編集して、[OK]ボタンを押すとラベルリストに編集を反映し、保存します。多重定義関数やクラス名を入力したときは、[関数選択]ダイアログボックスが開くので、関数を選択して[アドレス]を設定します。詳細は「17.18.2.3 複数ラベルをサポートする」を参照してください。

### 17.18.1.4 ラベルを削除する

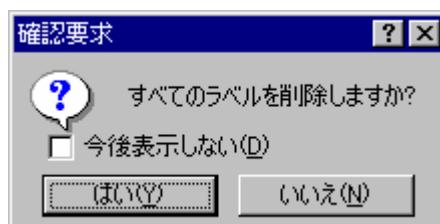
削除したいラベルを選択した状態で、ポップアップメニューから[削除]を選択します。この際に確認メッセージボックスを表示します。



[OK]ボタンを押すとラベルリストから削除し、ウィンドウを更新します。このメッセージボックスの表示が不要のときは、[今後表示しない]チェックボックスをオンにしてください。

### 17.18.1.5 すべてのラベルを削除する

ポップアップメニューから[すべてを削除]を選択すると、リストからすべてのラベルを削除します。この際、確認メッセージボックスを表示します。



[OK]ボタンを押すと、すべてのラベルを High-performance Embedded Workshop のシンボルテーブルから削除し、リスト表示もクリアします。メッセージボックスの表示が不要のときは、[今後表示しない]チェックボックスをオンにしてください。

### 17.18.1.6 ラベルをファイルからロードする

シンボルファイルをロードして現在の High-performance Embedded Workshop のシンボルテーブルに結合できます。ポップアップメニューから[ロード]を選択すると、シンボルをロードするダイアログボックスを開きます。ダイアログボックスは、Windows® 標準のファイルを開くダイアログボックスと同様です。ファイルを選択し、[開く]ボタンを押すとロードを開始します。シンボルファイルの標準拡張子は".sym"です。

### 17.18.1.7 ラベルをファイルに保存する

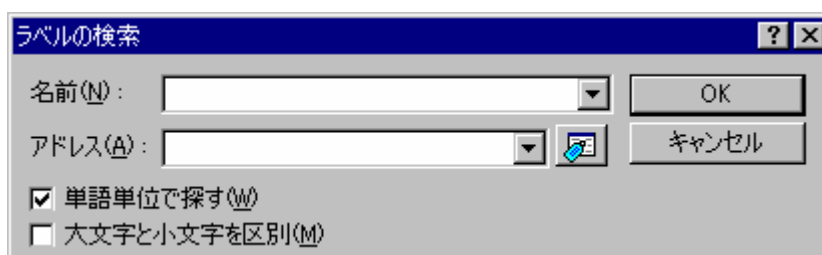
ポップアップメニューから[名前を付けて保存]を選択すると、シンボルを保存するダイアログボックスを開きます。ダイアログボックスは Windows® 標準の[名前を付けて保存]ダイアログボックスと同様に操作できます。[ファイル名]にファイル名を入力し[保存]ボタンを押すと、シンボルファイルにラベルリストを保存します。標準ファイル拡張子は".sym"です。

シンボルフォーマットについては「リファレンス 7. シンボルファイルフォーマット」を参照してください。

一度[名前を付けて保存]メニューでファイルに保存すると、以後はポップアップメニューの[上書き保存]で、現在のシンボルテーブルを同一シンボルファイルに保存できます。

### 17.18.1.8 ラベルを検索する

ポップアップメニューから[検索]を選択すると、[ラベルの検索]ダイアログボックスが開きます。



1. [名前]に検索するラベル名を入力するか、ドロップダウンリストボックスから以前検索したラベル名を選択してください。
2. [アドレス]に検索するラベルのアドレスを入力するか、ドロップダウンリストボックスから以前検索したアドレスを選択してください。
3. [名前]に入力した名前と一致するラベルを検索する場合は、[単語単位で探す]チェックボックスをオンにしてください。このオプションをチェックしない場合は、[名前]に入力した名前と一部でも一致するラベルを検索します（[アドレス]は指定できません）。
4. ラベル名の大文字と小文字を区別する場合は、[大文字、小文字を区別する]チェックボックスをオンにしてください。
5. [OK]ボタンをクリックすると検索を開始します。

#### 注：

ラベルは、はじめの 1024 文字分しか情報を保持していません。したがって、ラベル名のはじめの 1024 文字分は重複しないようにしてください。

### 17.18.1.9 次を検索する

ラベルが検索できた後、ポップアップメニューから[次を検索]を選択すると、検索条件に一致する次のラベルを検索します。

### 17.18.1.10 ラベルに対応するソースコードを表示する

ラベルを選択した状態で、ポップアップメニューから[ソースを表示]を選択すると、ラベルのアドレスに対応するソースファイルを開きます。

## 17.18.2 Elf/Dwarf2 のサポート

High-performance Embedded Workshop は、C/C++およびアセンブリ言語で書いたアプリケーションのデバッグのために Elf/Dwarf2 オブジェクトファイルフォーマットをサポートします。

### 主な特長：

- ソースレベルデバッグ
- C/C++演算子
- C/C99 \*/C++の式 (キャスト、ポインタ、参照など)
- あいまいな関数名
- オーバレイメモリロード
- ウォッチ-ローカル、およびユーザ定義
- スタックトレース

### 注：

\*. C99 言語仕様をサポートしたコンパイラの使用時に指定できます。

### 17.18.2.1 C/C++演算子

以下の C/C++ 言語演算子を使用できます。

```
+, -, *, /, &, |, ^, ~, !, >>, <<, %, (, ), <, >, <=, >=, ==, !=, &&, ||
Buffer_start + 0x1000
#R1 | B'10001101
((pointer + (2 * increment_size)) & H'FFFF0000) >> D'15
!(flag ^ #ER4)
```

### 17.18.2.2 C/C99/C++の式

式の例:

式	内容
Object.value	メンバの直接参照を指定します(C/C++)
p_Object->value	メンバの間接参照を指定します(C/C++)
Class::value	クラスを持つメンバの参照を指定します(C++)
*value	ポインタを指定します(C/C++)
&value	参照を指定します(C/C++)
array[0]	配列を指定します(C/C++)
Object.*value	ポインタを持つメンバの参照を指定します(C++)



式	内容
::g_value	グローバル変数の参照を指定します(C++)
Class::function(short)	メンバ関数を指定します(C++)
(struct STR) *value	キャスト動作を指定します(C/C++)
[+-]浮動小数点定数(+/-)浮動小数点定数*I	複素数あるいは虚数を指定します。 浮動小数点定数として、10進浮動小数点定数での指定以外に16進浮動小数点定数による指定ができます。(C99言語仕様をサポートしたコンパイラの使用時に指定できます)

### 17.18.2.3 複数ラベルをサポートする

プログラム言語の中の、例えばC++オーバーロード関数などでは、1つのラベルが複数のアドレスを表す場合があります。各ダイアログボックスでこのようなラベル名を入力した場合、High-performance Embedded Workshopは[関数選択]ダイアログボックスを使用してオーバーロード関数およびメンバ関数を表示します。



[関数選択]ダイアログボックスでは、オーバーロード関数またはメンバ関数を選択します。通常、一度に1つの関数を選択します。ただし、ブレークポイントを設定する場合においてのみ、複数の関数を選択できます。このダイアログボックスには3つの領域があります。

[関数名の選択] 同じ名前をもつ関数またはメンバ関数、およびその詳細情報を表示します。

[名前] 関数名を表示します。

[アドレス] 関数アドレスを表示します。

[関数名の指定] 設定する関数およびそれらの詳細情報を表示します。

[名前] 関数名を表示します。

[アドレス] 関数アドレスを表示します。

[カウンタ] 同じ名前をもつ関数またはメンバ関数の数を表示します。

[全関数] 同じ名前をもつ関数またはメンバ関数を表示します。

[選択関数] [関数名の選択]リストボックスに表示する関数の数を表示します。

[指定関数] [関数名の指定]リストボックスに表示する関数の数を表示します。

#### (1) 関数を選択する

[関数名の選択]リストボックスから選択したい関数をクリックして、[>]ボタンをクリックします。

選択した関数を[関数名の指定]リストボックスに表示します。[関数名の選択]リストの関数すべてを選択するには、[>>]ボタンをクリックします。

## (2) 関数の選択を解除する

[関数名の指定]リストボックスから選択を解除する関数をクリックして、[<]ボタンをクリックします。すべての関数の選択を解除するには、[<<]ボタンをクリックします。選択を解除した関数は、[関数名の指定]リストボックスから[関数名の選択]リストボックスへ戻します。

## (3) 関数を設定する

[OK]ボタンをクリックして、[関数名の指定]リストボックスに表示した関数を設定します。関数を設定し、[関数選択]ダイアログボックスを閉じます。

[キャンセル]ボタンをクリックすると、関数を設定せずにダイアログボックスを閉じます。

### 17.18.2.4 オーバレイプログラムのデバッグ

オーバレイ機能を利用するプログラムをデバッグできます。ここではオーバレイ機能を使用するための設定について説明します。

#### (1) セクショングループを表示する

オーバレイ（いくつかのセクショングループを同じアドレス範囲に割り当てる）を使用すると、アドレス範囲およびセクショングループを[オーバレイ]ダイアログボックスに表示します。

[デバッグ->オーバレイの構成]を選択して[オーバレイ]ダイアログボックスを開きます。

このダイアログボックスには2つの領域があります。[アドレス]リストボックスおよび[セクション名]リストボックスです。



[アドレス]リストボックスは、オーバレイプログラムが使用するアドレス範囲を表示します。アドレス範囲の1つをクリックして[アドレス]リストボックスのアドレス範囲を選択します。

[セクション名]リストボックスは、選択したアドレス範囲に割り当てたセクショングループを表示します。



#### (2) セクショングループを設定する

オーバレイ関数を使用するときは、最も優先度の高いセクショングループを[オーバレイ]ダイアログボックスで選択していなければ、High-performance Embedded Workshopは正しく動作しません。

まず[アドレス]リストボックスに表示したアドレス範囲の1つをクリックします。選択したアドレス範囲に割り当てたセクショングループを[セクション名]リストボックスに表示します。

表示しているセクショングループの中から最も優先度の高いセクショングループをクリックします。



セクショングループを選択したら、[OK]ボタンをクリックして優先度の設定を保存して、ダイアログボックスを閉じます。[キャンセル]ボタンをクリックすると、優先度の設定を保存せずにダイアログボックスを閉じます。

#### 注：

オーバーレイ関数を使用するアドレス範囲内では[オーバーレイ]ダイアログボックスに指定したセクションのデバッグ情報を参照します。したがって、現在ロードしているプログラムと同じセクションを[オーバーレイ]ダイアログボックスで選択します。

### 17.18.3 変数を表示する

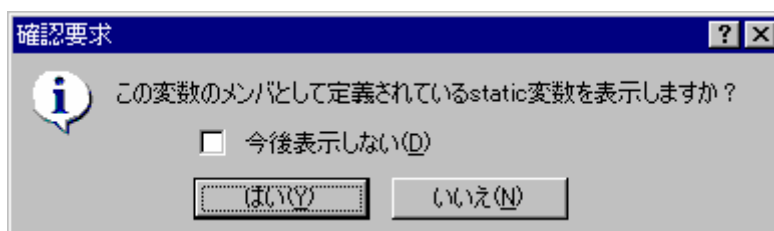
ソースプログラム上の変数の値を表示する方法について説明します。

変数を表示する際に以下に示す注意事項があります。

ELF/DWARF2 フォーマットのロードモジュールをデバッグする場合、ウォッチウィンドウやローカルウィンドウ上で構造体を展開したときに static メンバ変数を通常のメンバ変数と同じように表示できます。しかし、この機能はデバッグ情報量の多さによって変数表示に時間がかかることがあります。

#### ウォッチ/ローカルで変数展開時に static メンバを表示しないようにするには

1. [基本設定 -> オプション]を選択し、[オプション]ダイアログボックスを開きます。
2. [確認]タブを選択します。
3. [ウォッチ/ローカルで変数展開時に static メンバを表示]チェックボックスをオンにします。デフォルトはチェックボックスがオフです。
4. 次の確認ダイアログボックスが表示されたときに[いいえ]を選択してください。



この操作により変数展開時に static メンバ変数を表示しなくなり、表示時間を短縮できます。( [いいえ] を選択した後も static メンバ変数を直接ウォッチウィンドウに追加することで static メンバ変数の参照は可能です )

### 17.18.3.1 ツールチップウォッチ

プログラムに定義された変数の値を参照するための機能です。値を参照したい変数を含むプログラムをエディタウィンドウまたは逆アセンブリウィンドウ（ソースモード）に表示し、変数名の上にマウスカーソルを置くと、ツールチップとしてポップアップウィンドウにウォッチ情報を表示します。

#### ツールチップウォッチ機能を使用するには

1. [基本設定 -> オプション]を選択してください。[オプション]ダイアログボックスを開きます。
2. [エディタ]タブを選択してください。
3. [ツールチップウォッチ]チェックボックスをオンにしてください。
4. [OK]ボタンをクリックしてください。

ツールチップウォッチ

#### エディタウィンドウまたは逆アセンブリウィンドウ（ソースモード）にツールチップウォッチを表示するには

1. 参照したい変数を表示しているエディタウィンドウまたは逆アセンブリウィンドウ（ソースモード）を開きます。
2. 参照したい変数名の上にマウスカーソルを置きます。ツールチップとしてポップアップウィンドウに変数のウォッチ情報を表示します。

```

sort(a);
printf("*** Sorting results
for( i=0; i<10; i++ ){
    printf("a[%d]=%ld\n",i,a
}
min = a[0];
max = a[9];
min = 0;
max = 0;
change(a);
min = a[9];
max = a[0];

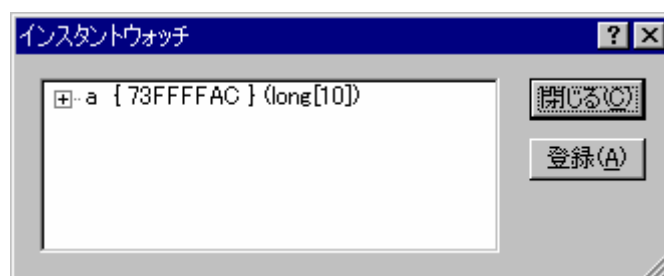
```

### 17.18.3.2 インスタントウォッチ

参照したい変数を含むソースファイルをエディタウィンドウまたは逆アセンブリウィンドウ（ソースモード）に表示します。

参照したい変数名の上にマウスのカーソルを置いてポップアップメニューから[インスタントウォッチ]を選択します。

[インスタントウォッチ]ダイアログボックスが開き、カーソル上の変数を表示します。




変数名の左側の '+' 記号はクリックすれば情報を展開表示できることを、 '-' 記号は情報を縮小表示できることを示します。[登録] ボタンをクリックすると、変数をウォッチウィンドウに登録して、ダイアログボックスを閉じます。[閉じる] ボタンをクリックすると、変数をウォッチウィンドウに登録しないで、ダイアログボックスを閉じます。

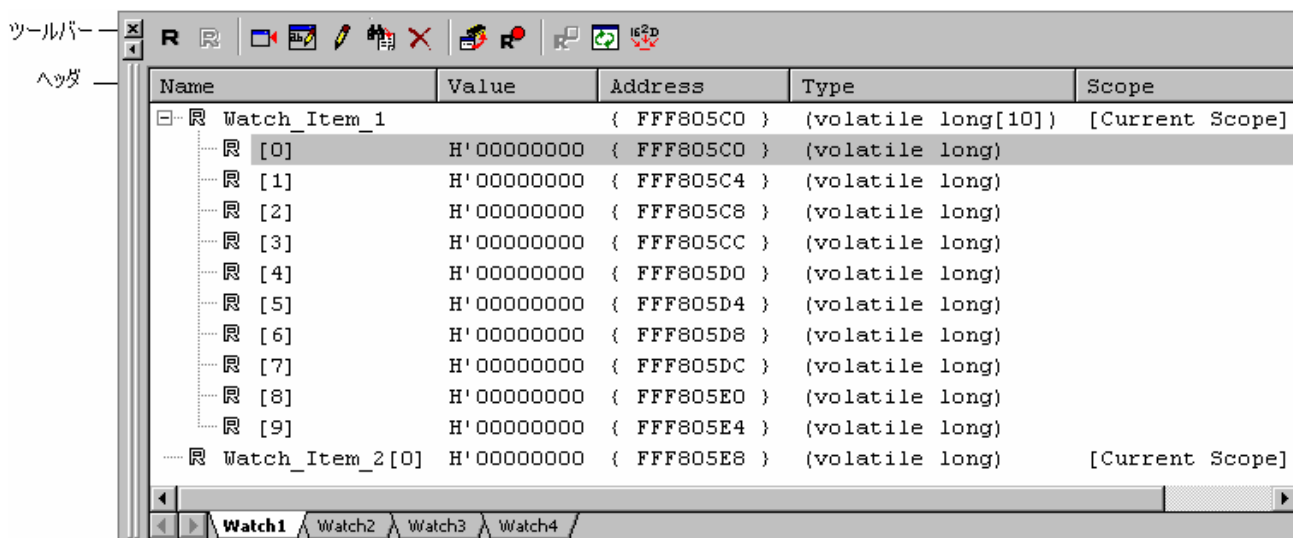
### 17.18.3.3 ウォッチウィンドウ

ウォッチウィンドウを開くことにより、任意の変数について値を参照できます。

#### (1) ウォッチウィンドウを開く

ウォッチウィンドウを開くには、[表示 -> シンボル -> ウォッチ] を選択するか、[ウォッチ] ツールバーボタン (  ) をクリックします。ウォッチウィンドウが開きます。

#### ウィンドウの構成



- ウィンドウの内容は、ロードモジュール(\*.abs等)内のデバッグ情報から、C/C++ソースプログラムの情報がある場合にのみ表示します。コンパイラ等の最適化により、ソースプログラムの情報としてデバッグ情報がない場合は表示できません。また、マクロ宣言されたものについても表示できません。
- エディタウィンドウおよび逆アセンブリウィンドウからウォッチウィンドウへのドラッグアンドドロップ機能により、変数を登録できます。また、エディタウィンドウおよび逆アセンブリウィンドウのソースモードで、[ウォッチウィンドウに追加]メニューオプションにより変数を登録できます。
- ドラッグアンドドロップ機能により変数の表示順序を変更できます。
- 変数をメモリウィンドウの[Address]カラムヘドラッグアンドドロップすると、メモリウィンドウの表示アドレスを変数のアドレスに簡単に変更できます。
- "R"マークはその変数がリアルタイムに更新できることを示します。"R"マークが太字のとき、その変数の値をユーザプログラムの実行時にリアルタイム更新します。
- [Name]カラムをダブルクリックすると、変数名を変更する[シンボル名の編集]ダイアログボックスが開きます。変数名の編集によりマクロを記録します。(「マクロの記録」)
- [Value]カラムをダブルクリックすると、変数の値を変更する[値の編集]ダイアログボックスが開きます。値の編集によりマクロを記録します。(「マクロの記録」)
- [Value]カラムでは、インプレース編集により値を変更できます。(「マクロの記録」)
- [Name]、または[Address]カラムのヘッダをクリックすると変数をソートします。

- スコープ別に変数を参照できます。参照先は、[Current Scope](現在有効な変数)、[Global](グローバル変数)、スコープを指定する各ファイル (ファイルスコープ内の static 変数) から参照できます。
- [Scope]カラムをダブルクリックすると、スコープを設定する[スコープ設定]ダイアログボックスが開きます。\*
- [Scope]カラムでは、インプレース編集によりスコープを設定できます。\*

注：

\*. スコープの設定機能のサポートは、デバッガ、および[デバッグの設定]ダイアログボックスの[デバッグフォーマット]の設定に依存します。サポートしていない場合、常に[Current Scope]を表示し変更できません。

## ウィンドウのオプション

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	マクロの記録	機能
自動更新有効化		●	選択している変数の "R" マークが太字になり、リアルタイム更新します。
全シンボル自動更新有効化		●	すべての "R" マークが太字になり、リアルタイム更新します。
自動更新無効化		●	選択している変数の "R" マークが中抜きになり、リアルタイム更新を解除します。
全シンボル自動更新無効化		●	すべての "R" マークが中抜きになり、リアルタイム更新を解除します。
更新間隔設定 *1		-	自動更新の間隔を設定します。
値更新記録	記録開始	●	[記録設定]ダイアログボックスを開きます。値更新の記録を開始します。
	記録終了	●	値更新の記録を終了します。
シンボル登録		●	[シンボル登録]ダイアログボックスを開きます。ウォッチする変数を入力します。
シンボル名の編集		●	[シンボル名の編集]ダイアログボックスを開きます。変数名を編集します。
値の編集		●	[値の編集]ダイアログボックスを開きます。変数の値を変更します。
スコープ設定 *2		-	選択されている変数のスコープを設定します。
削除		●	テキストカーソルの位置によって示された変数をウォッチウィンドウから削除します。
全シンボル削除		●	すべて変数をウォッチウィンドウから削除します。
基数	16 進数表示	●	16 進数で表示します。
	10 進数表示	●	10 進数で表示します。
	8 進数表示	●	8 進数で表示します。
	2 進数表示	●	2 進数で表示します。
	初期表示の設定	-	変数の初期表示基数を設定します。
コピー		-	選択部分をコピーし Windows® クリップボードに貼り付けます。
保存		-	表示内容をファイルに保存します。
メモリウィンドウへ移動		-	メモリウィンドウを開きます。
ツールバー表示	-	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	-	ツールバーボタンをカスタマイズします。



**注：**

- \*1. 機能がサポートされている場合のみ表示されます。
- \*2. 機能のサポートは、デバッガ、および[デバッグの設定]ダイアログボックスの[デバッグフォーマット]の設定に依存します。

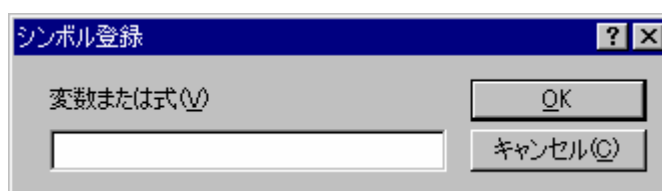
**(2) 変数を登録する**

ウォッチウィンドウに変数を登録します。

**変数を登録するには**

ウォッチウィンドウに変数を登録するには次の方法があります。

- エディタウィンドウまたは逆アセンブリウィンドウからウォッチウィンドウ上へ変数をドラッグアンドドロップします。
- エディタウィンドウまたは逆アセンブリウィンドウのソースモードで変数を選択し、ポップアップメニューから[ウォッチウィンドウに追加]を選択します。
- エディタウィンドウまたは逆アセンブリウィンドウのソースモードで、インスタントウォッチ機能を使用して登録します。
- 以下のいずれかの方法で[シンボル登録]ダイアログボックスを開きます。
  - ウォッチウィンドウを選択した状態で"Insert"キーを押す。
  - ウォッチウィンドウ内を右クリックしポップアップメニューを表示する。[シンボル登録]を選択する。



[変数または式]に変数名を入力して[OK]ボタンをクリックします。

ウォッチウィンドウに変数が登録されます。登録した変数のスコープは、デフォルトで[Current Scope]です。ウォッチウィンドウで登録した変数はセッションファイルに保存されます。

[変数または式]に入力する配列の要素番号には、2進数、8進数、10進数、または16進数を使用できます。ただし、プレフィックスについて以下の入力規則があります。

- 2進数： プレフィックス"0b"、または"0B"が必要です。
- 8進数： プレフィックス"0"が必要です。
- 10進数： プレフィックスは不要です。
- 16進数： プレフィックス"0x"、または"0X"が必要です。

℞	a[0b00011110]	H'00000202	{ 000011A4 }	(long)
℞	a[0B00011110]	H'00000202	{ 000011A4 }	(long)
℞	a[036]	H'00000202	{ 000011A4 }	(long)
℞	a[30]	H'00000202	{ 000011A4 }	(long)
℞	a[0x1e]	H'00000202	{ 000011A4 }	(long)
℞	a[0X1E]	H'00000202	{ 000011A4 }	(long)



**注：**

登録した変数がローカル変数で現在のスコープにない場合、ウォッチウィンドウに登録しますが、値は"Not available now."を表示します。

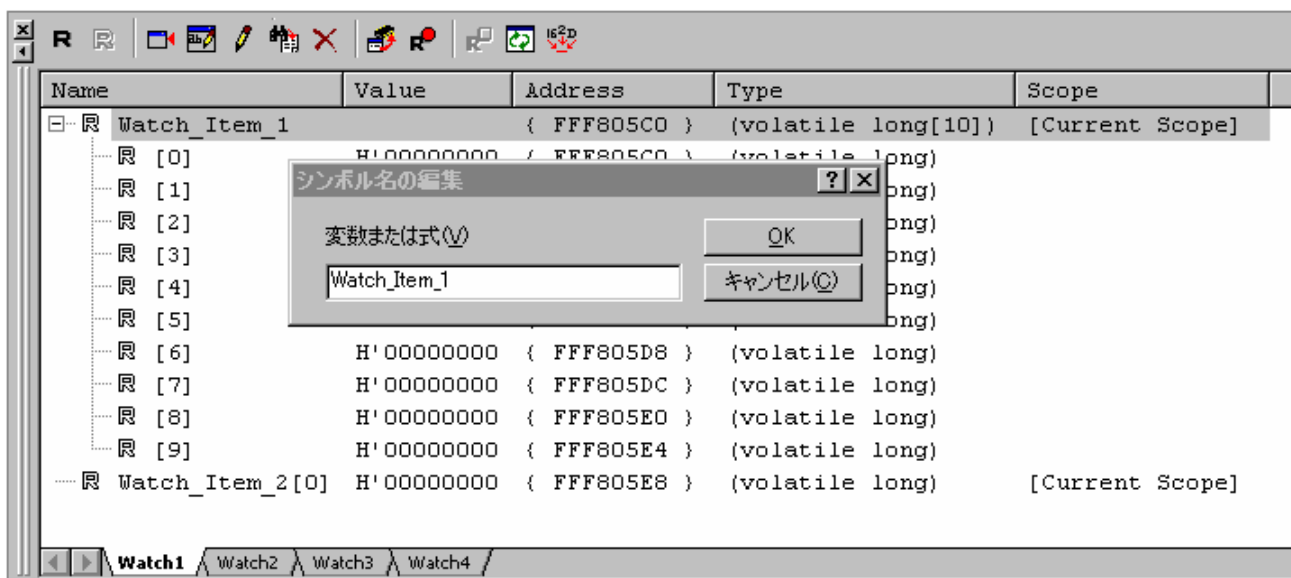
**(3) 変数名を編集する**

ウォッチウィンドウで変数名を編集します。構造体やポインタなどのように展開表示が可能な変数の場合、展開したときに表示されるサブアイテムの名前は編集できません。

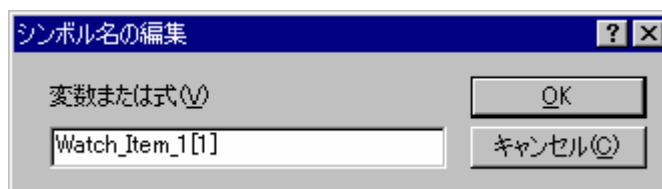
**変数名を編集するには**

以下のいずれかの方法で[シンボル名の編集]ダイアログボックスを開きます。

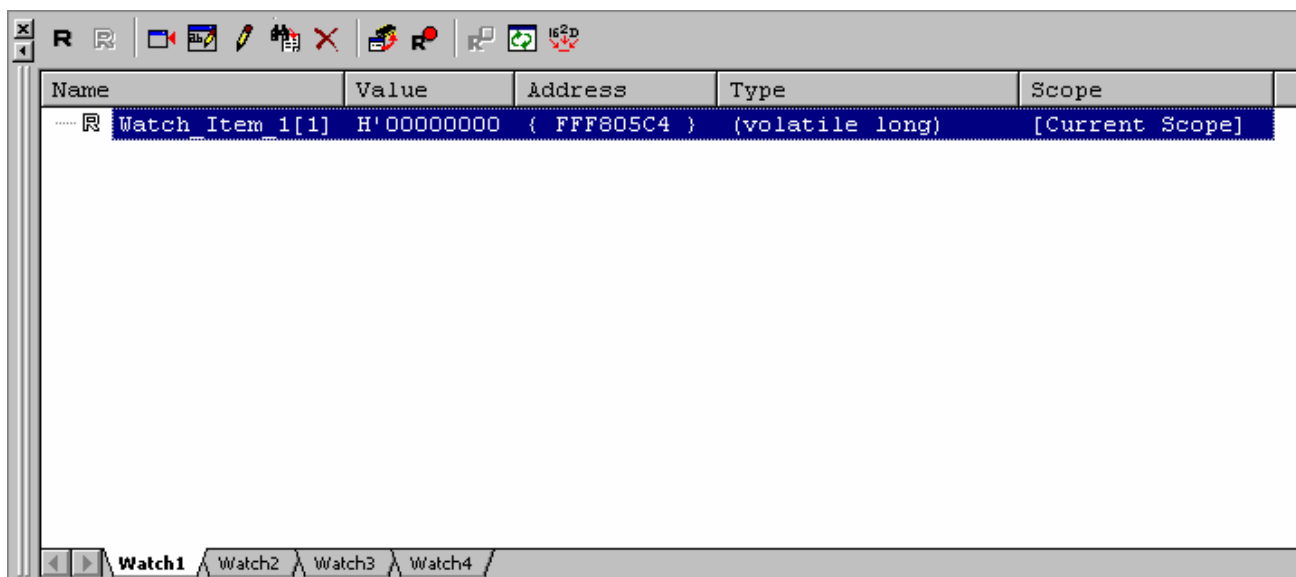
- [Name]カラムをダブルクリックします。
- 変数名を表示している変数行を右クリックしポップアップメニューを表示します。[シンボル名の編集]を選択します。



[変数または式]に変数名を入力して[OK]ボタンをクリックします。



ウォッチウィンドウを更新し、新しい変数名を表示します。

**注：**

以下の場合、変数名は編集できません。

- ユーザプログラムを実行している。
- 複数の変数を選択している。

**(4) 変数を展開する**

変数がポインタ、配列、または構造体の場合、その名前の左側にプラス(+)<sup>1</sup>の展開記号を表示します。変数を展開できることを意味します。変数を展開するには、プラス記号(+)<sup>1</sup>をクリックします。展開した変数は、タブ 1 つ分インデントを付けて、その要素(構造体または配列の場合)またはデータ値(ポインタの場合)を表示し、プラス記号がマイナス記号に変わります。変数が要素にポインタ、構造体、または配列を含む場合、その横に展開記号を表示します。

展開した変数を元に戻すには再び展開記号をクリックします。その要素は元に戻りマイナス記号はプラス記号に戻ります。

また、変数を選択した状態で数字キー (1~9) を押すと、その数字の階層分変数を展開できます。

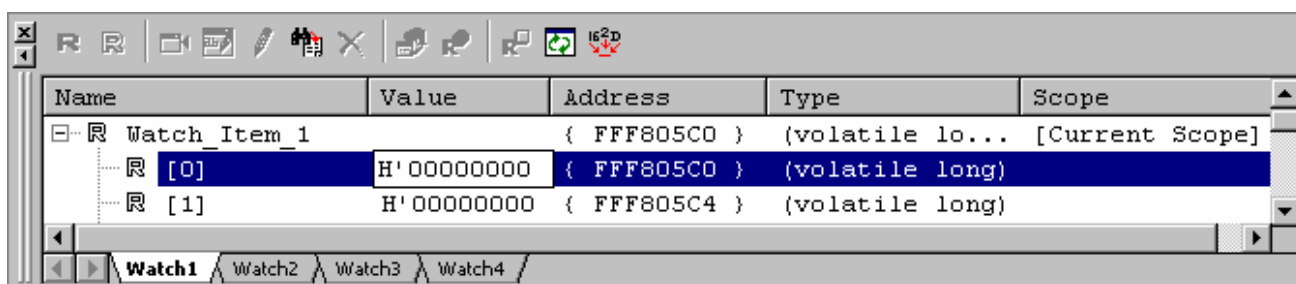
**(5) 変数の値を編集する**

ウォッチウィンドウで変数の値を編集します。

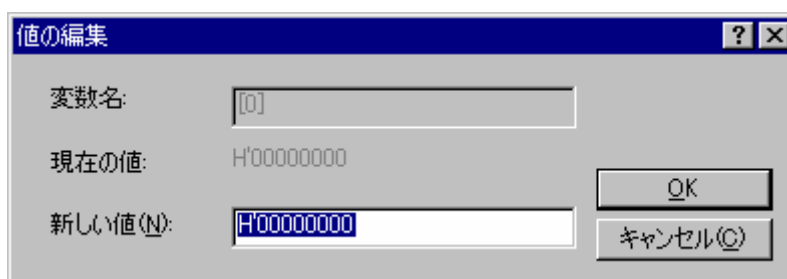
**変数の値を編集するには**

ウォッチウィンドウで変数の値を編集するには次の方法があります。

- [Value]カラムでインプレース編集します。



- 以下のいずれかの方法で[値の編集]ダイアログボックスを開きます。
  - [Value]カラムをダブルクリックする。
  - 値を表示している変数行で"Enter"キーを押す。
  - 値を表示している変数行を右クリックしポップアップメニューを表示する。[値の編集]を選択する。



[新しい値]に値を入力して[OK]ボタンをクリックします。

ウォッチウィンドウを更新し、新しい値を表示します。

#### (6) 変数のスコープを設定する

ウォッチウィンドウは、登録したシンボルのスコープを[Scope]カラムに表示します。変数登録時はスコープに[Current Scope]を表示します。変数のサブアイテムはルートアイテムのスコープに準じており、スコープを表示しません。

[Scope]カラムでは、登録した変数を参照するスコープを指定できます。同名変数がある場合、スコープを指定してウォッチする変数を区別できます。

スコープの設定機能のサポートは、デバッグ、および[デバッグの設定]ダイアログボックスの[デバッグフォーマット]の設定に依存します。サポートしていない場合、常に[Current Scope]を表示し変更できません。

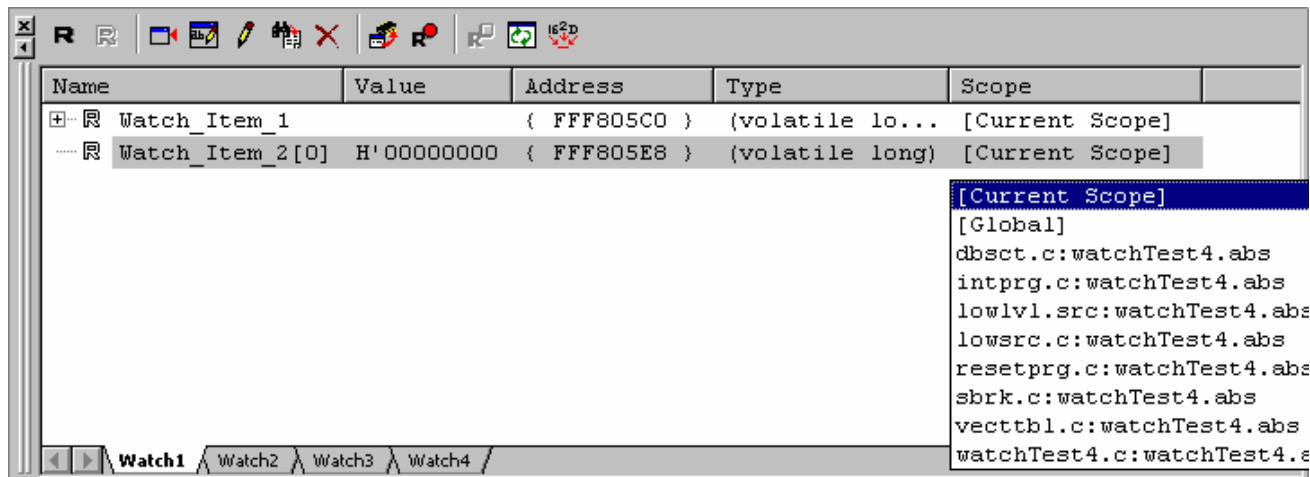
選択できるスコープの種類を以下に示します。

[Current Scope]	プログラムカウンタの指すアドレスから参照可能な範囲をスコープとします。 (デフォルト)
[Global]	グローバル変数を対象とし、全モジュールをスコープとします。 static で宣言したグローバル変数は検索範囲対象外とします。
ファイルスコープ名:	static 変数を対象とし、選択しているファイルスコープをスコープとします。
ロードモジュール名	関数内で宣言した static 変数、および[Global]で検索対象となるグローバル変数は検索範囲対象外とします。

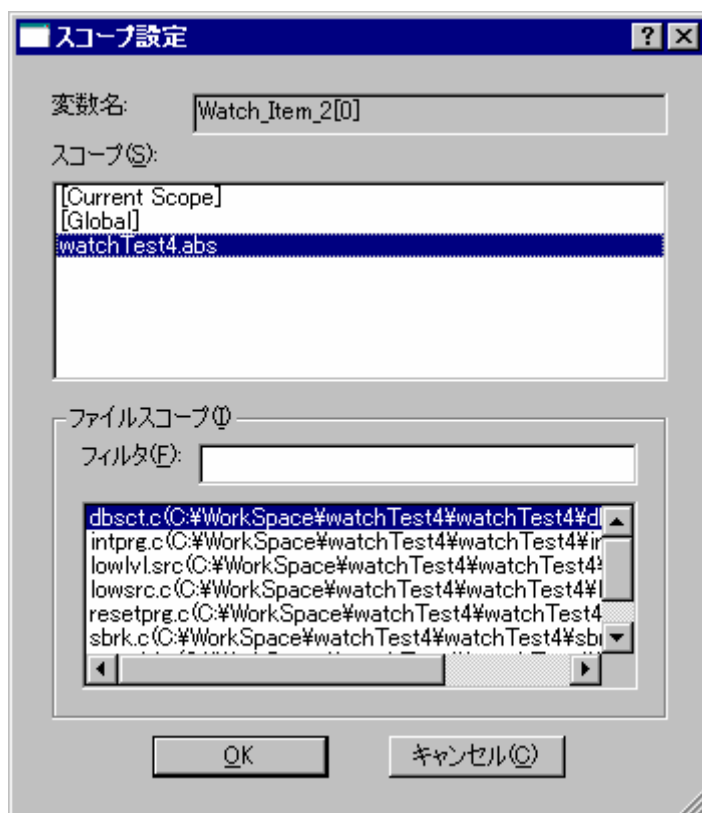
### 変数のスコープを設定するには

ウォッチウィンドウで変数のスコープを設定するには次の方法があります。

- [Scope]カラムでインプレース編集します。リストボックスの右側にあるボタンをクリックし、リストを表示します。リストからスコープを選択します。



- 以下のいずれかの方法で[スコープ設定]ダイアログボックスを開きます。
  - [Scope]カラムをダブルクリックする。
  - スコープを表示している変数行を右クリックしポップアップメニューを表示する。[スコープ設定]を選択する。



1. [変数名]に表示している変数のスコープを[スコープ]から選択します。

2. [スコープ]でロードモジュール名を選択した場合、[ファイルスコープ]からファイルスコープを選択します。[フィルタ]に入力した文字列（大文字と小文字を区別しません）を含むファイルスコープを抽出してから選択できます。

**注：**

ウォッチウィンドウで指定されている変数のスコープがダウンロードモジュールと一致しない場合、一致しない変数の[Scope]カラムがハイライト表示されます。ハイライト表示された変数のスコープを再設定してください。

**(7) 変数を削除する**

ウォッチウィンドウで特定の変数を削除するには、その変数を選択しポップアップメニューから[削除]を選択します。変数を削除し表示内容を更新します。

すべての変数を削除するには、ポップアップメニューから[全シンボル削除]を選択します。すべての変数を削除し表示内容を更新します。

**(8) 表示基数を変更する**

ウォッチウィンドウで、変数を選択してポップアップメニューから[基数]を選択すると、サブメニューで変数の表示基数を変更できます。

以下のいずれかが選択できます。

[16進数表示]	16進数で表示（デフォルト）
[10進数表示]	10進数で表示
[8進数表示]	8進数で表示
[2進数表示]	2進数で表示

**(9) 初期表示基数を設定する**

ウォッチウィンドウで、変数を選択してポップアップメニューから[基数 -> 初期表示の設定]を選択すると、[初期表示基数]ダイアログボックスが開きます。



リストボックスで基数を選択してください。次回登録時より、指定した基数で変数が表示されます。

**(10) 表示内容をファイルに保存する**

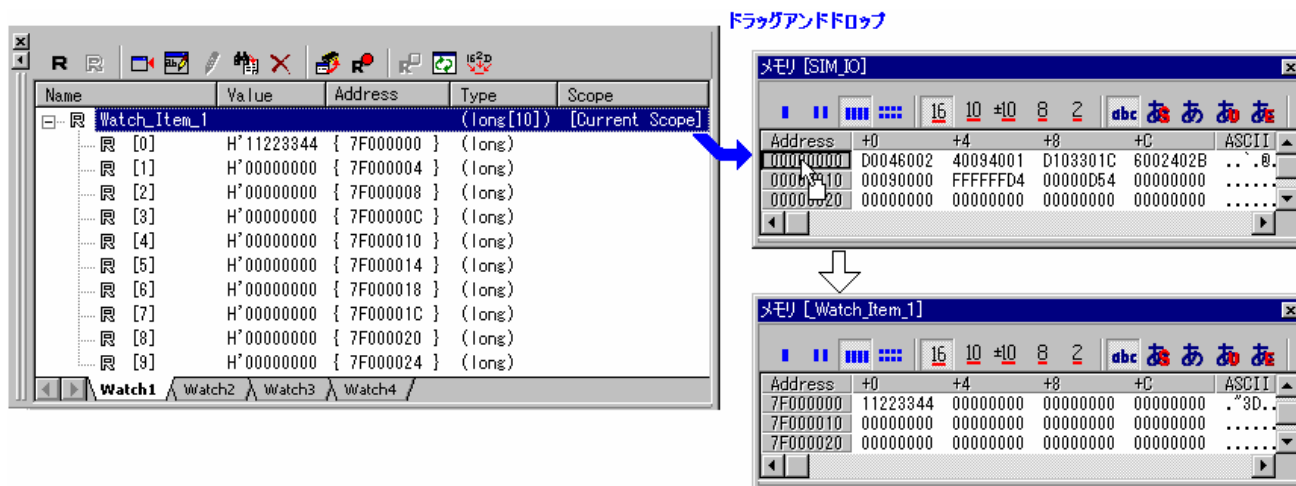
ウォッチウィンドウの表示内容をファイルに保存するには、ポップアップメニューの[保存]を選択します。

[保存]を選択すると、[名前を付けて保存]ダイアログボックスを開きます。ファイル名を指定し、ウォッチウィンドウに表示している内容を保存します。[Append]チェックボックスをチェックすると追加書き込みし、チェックしないと上書きします。

### (11) メモリウィンドウを表示する

選択している変数が割り付いているメモリ領域をメモリウィンドウに表示できます。ウォッチウィンドウのポップアップメニューから[メモリウィンドウに移動]を選択すると、メモリウィンドウを開きます。

ウォッチウィンドウに表示している変数をメモリウィンドウの[Address]カラムへドラッグアンドドロップすると、メモリウィンドウの表示アドレスを変数のアドレスに簡単に変更できます。



異なるスコープに同名の変数が定義されている場合、メモリウィンドウで参照するメモリ領域は、プログラムカウンタの指すアドレスから参照可能な範囲になります。

### (12) リアルタイム更新を設定する

ウォッチウィンドウの変数で、変数名の左に表示している"R"マークは、その変数がリアルタイムに更新できることを示します。

リアルタイム更新はウォッチウィンドウのポップアップメニューで設定します。

自動更新有効化                      選択している変数の"R"マークが太字になり、リアルタイム更新します。

全シンボル自動更新有効化        すべての"R"マークが太字になり、リアルタイム更新します。

自動更新無効化                      選択している変数の"R"マークが中抜きになり、リアルタイム更新を解除します。

全シンボル自動更新無効化        すべての"R"マークが中抜きになり、リアルタイム更新を解除します。

プログラム実行中、リアルタイム更新を有効に設定した変数の値を編集できます。

### (13) 更新間隔を設定する

ウォッチウィンドウの表示内容をユーザプログラム実行中に自動的に更新する間隔を設定できます。

機能のサポートはデバッガに依存します。

#### 更新間隔を設定するには

1. ウィンドウ内を右クリックしポップアップメニューを表示します。
2. [更新間隔設定]を選択します。[表示更新間隔設定]ダイアログボックスが開きます。



3. 更新間隔を 10ms 単位で入力します。更新間隔のデフォルト値は 100ms です。更新間隔の指定可能範囲はデバッガに依存します。
4. [OK]ボタンをクリックします。

指定した更新間隔は、ウォッチウィンドウのタブ間で共通に使用します。

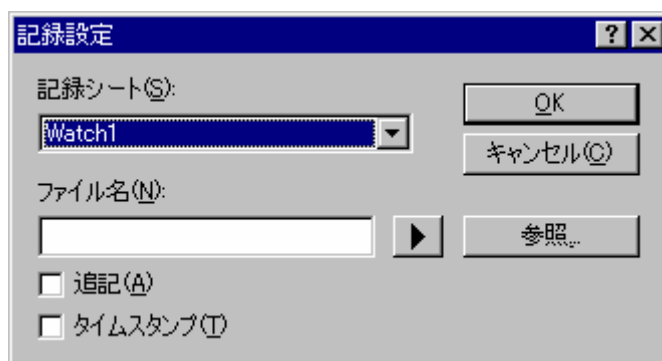
プログラム実行時の状態によっては、実際の更新間隔が設定した値よりも大きくなる場合があります。

#### (14) 値更新を記録する

ウォッチウィンドウでリアルタイム更新を設定した変数は、値の変化をファイルに記録できます。

#### 値更新の記録を開始するには

1. ウィンドウ内を右クリックしポップアップメニューを表示します。
2. [値更新記録]を選択し、サブメニューから[記録開始]を選択します。[記録設定]ダイアログボックスが開きます。



3. [記録設定]ダイアログボックスでは、記録方法を指定します。
4. [記録シート]で値更新を記録するシートを選択してください。同時に記録できるシートは 1 枚です。なお、[記録シート]には[記録開始]ダイアログボックス表示時に自動更新が設定された変数が存在するシート名のみが表示されます。
5. [ファイル名]で記録するファイル名を入力してください。[ファイル名]にプレースホルダを挿入するには、プレースホルダボタンをクリックし、ポップアップメニューからプレースホルダを選択します。ファイルを参照するには[参照]ボタンをクリックします。
6. [追記]チェックボックスをオンにすると、入力したファイルに追記します。デフォルトはオフです。
7. [タイムスタンプ]チェックボックスをオンにすると、タイムスタンプ (ms 単位) をファイルに記録します。デフォルトはオフです。
8. [OK]ボタンをクリックすると、記録状態になります。ただし、次の場合は記録状態になりません。



- [追記]チェックボックスがオンの場合、[ファイル名]に入力したファイルに既に記録されている変数と、[記録シート]領域で選択したシートの記録対象変数が一致しない。または、それらの表示基数が一致しない。

記録状態では、以下のタイミングで変数の値をファイルに記録します。

- 実行中の自動（リアルタイム）更新
- 実行停止時の値更新
- 値の編集時
- ウィンドウを操作したことによる表示の更新時

### リアルタイム更新の記録を終了するには

1. ウィンドウ内を右クリックしポップアップメニューを表示します。
2. [値更新記録]を選択し、サブメニューから[記録終了]を選択します。

また次の場合、自動的に記録を停止します。

- ダウンロード、およびアンロード時
- 値更新記録の対象となっている変数の展開が閉じた。
- 値更新記録の対象となっている変数の表示基数を変更した。
- 値更新記録の対象となっているシートで新たにリアルタイム更新対象変数を追加した。
- 値更新記録の対象となっている変数を並び替えた。
- 値更新記録の対象となっている変数のリアルタイム更新を解除した。

### (15) テキストの色を変更する

ウォッチウィンドウは、表示しているテキストの色およびフォントを[表示形式]ダイアログボックスでカスタマイズできます（下表のカテゴリの"Text"）。

さらに、ウォッチウィンドウの[Value]カラムのアクセス属性別または値の変更により、該当部分の色を[表示形式]ダイアログボックスでカスタマイズできます。

### 既存の色をカスタマイズするには

1. [基本設定 -> 表示の形式]を選択し、[表示形式]ダイアログボックスを開きます。
2. [ウォッチ]アイテムをダイアログボックスの左側のツリーから選択し展開します。
3. 色を変更したいカテゴリを選択します。次の表に示すカテゴリから選択できます。

カテゴリ	カスタマイズ対象	カラータブの前面の色 (デフォルト)	カラータブの背景の色 (デフォルト)
Text	テキスト	黒	白
書き込み *	ライトアクセスされたメモリ	黒	赤
読み込み *	リードアクセスされたメモリ	黒	緑
変更 *	変数の値、または割付け位置に変更があった場合の色	赤	白

#### 注：

- \*. 機能のサポートはデバッガに依存します。
4. [カラー]タブの[前面]リストと[背景]リストの選択を変更します。
  5. [OK]ボタンをクリックします。

### (16) エンコード形式を設定する

変数値が文字、文字列の場合に UTF-8 として表示するには[基本設定 -> オプション]を選択し、[オプション]ダイアログボックスを開きます。

[デバッグ]タブを選択し、[エンコード形式]を[UTF-8]にします。

これで、ウォッチウィンドウは変数の値を UTF-8 として表示します。

[エンコード形式]を[ローカルコードページ] (デフォルト) にした場合は、ASCII およびシフト JIS コードとして表示します。

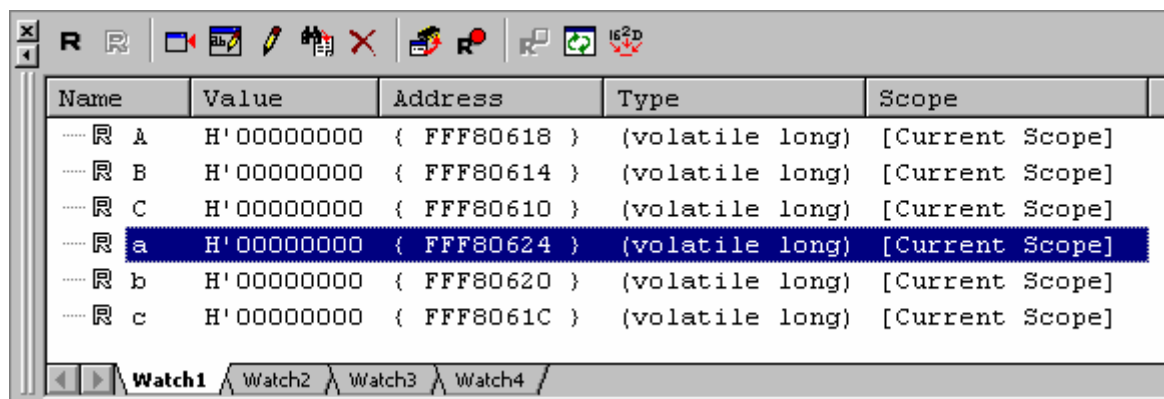
#### 注:

ウォッチウィンドウではウィンドウ内に表示するフォントのデフォルトが"Courier New"となっており、UTF-8 やシフト JIS コードで表現される日本語文字等に対応していません。

これらの文字コードを表示する場合は[表示形式]ダイアログボックスで日本語文字に対応したフォント(MS Pゴシックなど)に変更してご使用ください。

### (17) 変数をソートする


[Name]、または[Address]カラムのヘッダをクリックすると変数をソートします。1 回目のクリックで昇順に、2 回目のクリックで降順にソートします。以後、交互に繰り返します。



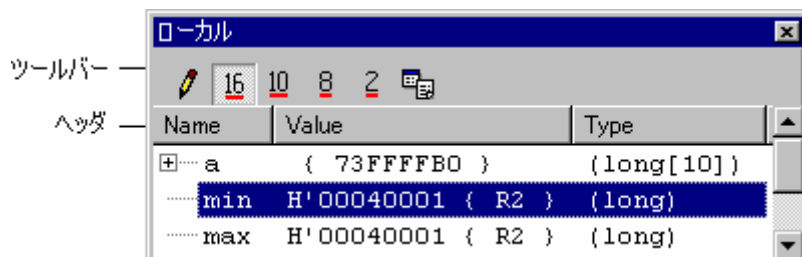
### 17.18.3.4 ローカルウィンドウ

ローカルウィンドウを開くことにより、ローカル変数の一覧とそれらの値を参照できます。

#### (1) ローカルウィンドウを開く

ローカルウィンドウを開くには、[表示 -> シンボル -> ローカル]を選択するか、[ローカル]ツールバーボタンをクリックします。

#### ウィンドウの構成



- 現在のローカル変数を表示します。プログラムをデバッグしていくに従い、実行から停止またはブレークした後、ローカルウィンドウを更新して現在のローカル変数とその値を表示します。ローカル変数を定義した時点で初期化していないと、ローカル変数に値を代入するまでローカルウィンドウの値は不定となります。
- アブソリュートファイル(\*.abs)に含まれるデバッグ情報によって、現在の PC 位置から関数内にあるローカル変数に関連付けることができない場合は空白となります。
- [Value]カラムをダブルクリックすると、変数の値を変更するダイアログボックスが開きます。（「マクロの記録」）
- [Value]カラムでは、インプレース編集により変数の値を変更できます。（「マクロの記録」）

## ウィンドウのオプション

ウィンドウ内で右クリックすると、使用可能なオプションをポップアップメニューで表示します。

本ウィンドウは、基本的なオプションをツールバーに割り付けています。

[ツールバー表示]、[ツールバーのカスタマイズ]は、ツールバー内で右クリックしても、ポップアップメニューを表示します。

ポップアップメニューオプション	ツールバーボタン	マクロの記録	機能
値の編集			選択しているローカル変数の値を変更するダイアログボックスを開きます。
基数	16 進数表示	16	16 進数で表示します。
	10 進数表示	10	10 進数で表示します。
	8 進数表示	8	8 進数で表示します。
	2 進数表示	2	2 進数で表示します。
コピー		-	選択部分をコピーし Windows® クリップボードに貼り付けます。
ツールバー表示	-	-	ツールバー表示/表示なしを切り替えます。
ツールバーのカスタマイズ	-	-	ツールバーボタンをカスタマイズします。

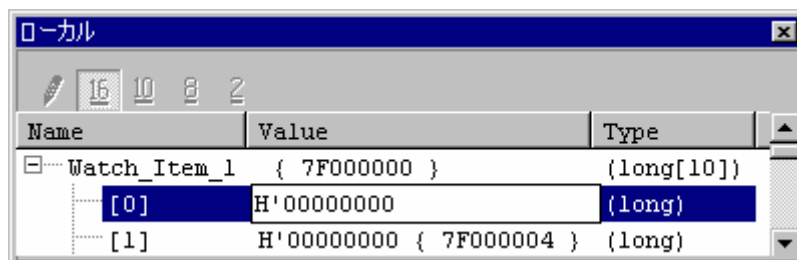
## (2) ローカル変数の値を編集する

ローカルウィンドウでローカル変数の値を変更できます。

### ローカル変数の値を編集するには

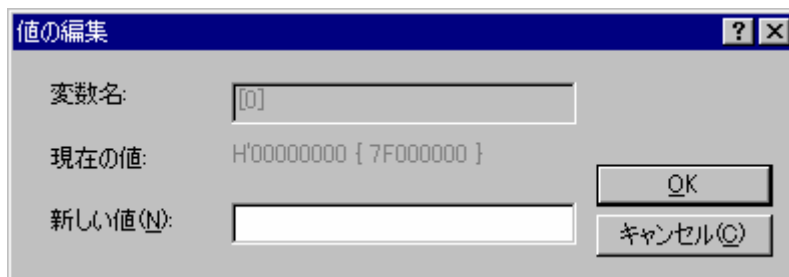
ローカルウィンドウでローカル変数の値を編集するには次の方法があります。

- [Value]カラムでインプレース編集します。



- 以下のいずれかの方法で[値の編集]ダイアログボックスを開きます。
  - [Value]カラムをダブルクリックする。

- 値を表示しているローカル変数行を右クリックしポップアップメニューを表示する。ポップアップメニューから[値の編集]を選択する。



[新しい値]に値を入力して[OK]ボタンをクリックします。

ローカルウィンドウを更新し、新しい値を表示します。

### (3) 表示基数を変更する

ローカルウィンドウで、変数を選択してポップアップメニューから[基数]を選択すると、サブメニューで変数の表示基数を変更できます。

以下のいずれかが選択できます。

[16進数表示]	16進数で表示 (デフォルト)
[10進数表示]	10進数で表示
[8進数表示]	8進数で表示
[2進数表示]	2進数で表示

### (4) エンコード形式を設定する

変数値が文字、文字列の場合に UTF-8 として表示するには[基本設定 -> オプション]を選択し、[オプション]ダイアログボックスを開きます。

[デバッグ]タブを選択し、[エンコード形式]を[UTF-8]にします。

これで、ローカルウィンドウは変数の値を UTF-8 として表示します。

[エンコード形式]を[ローカルコードページ] (デフォルト) にした場合は、ASCII およびシフト JIS コードとして表示します。

#### 注:

ローカルウィンドウではウィンドウ内に表示するフォントのデフォルトが"Courier New"となっており、UTF-8 やシフト JIS コードで表現される日本語文字等に対応していません。

これらの文字コードを表示する場合は[表示形式]ダイアログボックスで日本語文字に対応したフォント(MS Pゴシックなど)に変更してご使用ください。

## 18. 同期デバッグ機能

マルチコアマイコンのデバッグのために同期デバッグ機能をサポートします。

本機能を用いることで、マルチコアでプログラムを実行し、コア間で共通のデバッグ動作(実行、ステップ、プログラムの停止等)を同期できます。[同期デバッグ]ダイアログボックスを使用して、同期デバッグシステムの構成や起動を行うことができます。

同期デバッグに関する設定は SDO ファイルに保存されます。SDO ファイルは以下の"Sync"ディレクトリに置かれます。

```
"<Local setting directory>%Renesas%HEW%hew_001%Sync"
```

<Local setting directory>は通常、Windows® により以下のように設定されます。

Windows® 2000 operating system および Windows® XP operating system:

```
"C:%Documents and Settings%<ユーザ名>%Local Settings%Application Data"
```

Windows Vista® operating system:

```
"C:%Users%<ユーザ名>%AppData%Local"
```

### 注：

日本語フォルダを含むフォルダ階層に High-performance Embedded Workshop をインストールしている場合、同期デバッグ機能は使用できません。

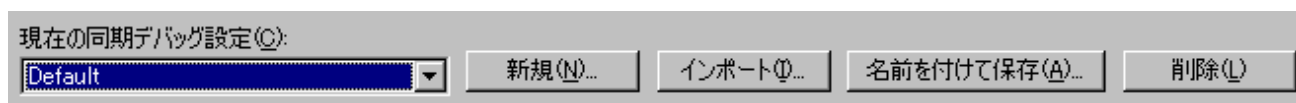
### 18.1 同期デバッグダイアログボックス

[デバッグ]メニューから[同期デバッグ]を選択し、[同期デバッグ]ダイアログボックスを開きます。

High-performance Embedded Workshop の起動時やワークスペースを開いているときにもこのダイアログを開くことができます。また、High-performance Embedded Workshop の[ようこそ!]ダイアログボックスから本ダイアログボックスを開くこともできます。



### 18.1.1 同期デバッグ設定を管理する



#### [現在の同期デバッグ設定]ドロップダウンリスト

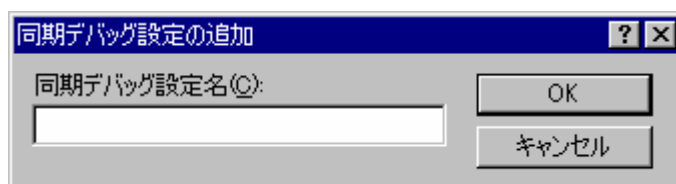
[同期デバッグ]ダイアログボックスの最上部にあるドロップダウンリストには、Sync ディレクトリにある同期デバッグ設定ファイルが表示されます。

ダイアログボックスの設定を別の同期デバッグ設定に変更するには、ドロップダウンリストでその同期デバッグ設定を選択してください。同期デバッグ設定を選択した場合、現在の設定は自動的に保存されます。

#### 新しい同期デバッグ設定を作成するには

以下のようにして新しい同期デバッグ設定を作成できます。

1. [新規]ボタンをクリックします。



2. [同期デバッグ設定の追加]ダイアログボックスが開くので、同期デバッグ設定ファイルの名前を入力します。Windows のファイル名として使用可能な文字のみを入力できます。ファイル名には 100 文字の制限があります。
3. OK ボタンをクリックします。

新しい同期デバッグ設定用に ".SDO" ファイルが自動的に作成され、Sync ディレクトリに保存されます。新しい同期デバッグ設定は [現在の同期デバッグ設定] ドロップダウンリストに追加され、現在の同期デバッグ設定として選択されます。ダイアログボックスの他のすべてのコントロールはデフォルト状態になります。

#### 既存の同期デバッグ設定を別のファイル名で保存するには

「名前を付けて保存」動作と同じ働きをします。現在のダイアログボックスの設定を新しい同期デバッグ設定として保存するには以下のようにしてください。

1. [名前を付けて保存]ボタンをクリックします。
2. 同期デバッグ設定ファイルの新しい名前を入力します。
3. [保存]ボタンをクリックします。

同期デバッグ設定を Sync ディレクトリに保存した場合、その同期デバッグ設定は [現在の同期デバッグ設定] ドロップダウンリストに追加され、新しい同期デバッグ設定と同様に選択されますが、ダイアログボックスのコントロールはデフォルトに戻りません。また、ファイルを別の場所に保存することもできます(例えば、ネットワークドライブに保存して他の High-performance Embedded Workshop ユーザと共有することも可能です)。ただし、この場合、同期デバッグ設定はドロップダウンリストに追加されません。

### 同期デバッグ設定をインポートするには

High-performance Embedded Workshop に同期デバッグ設定をインポートするには、([同期デバッグ]ダイアログボックスを開く前に)同期デバッグ設定の SDO ファイルを Sync ディレクトリにコピーするか、ダイアログボックスで以下の操作を行ないます。

1. [インポート]ボタンをクリックします。
2. インポートしたい同期デバッグ設定ファイルをブラウズします。
3. [インポート]ボタンをクリックします。

ファイルが Sync ディレクトリにコピーされ、同期デバッグ設定はドロップダウンリストに追加されます。ドロップダウンリストでインポートした同期デバッグ設定が選択され、読み込まれます。

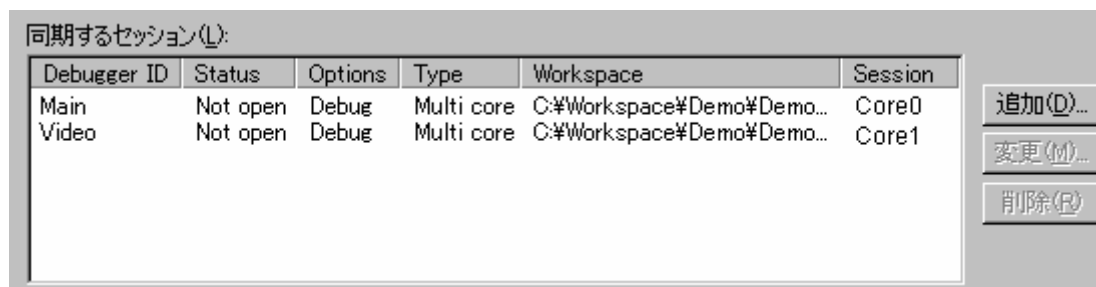
### 同期デバッグ設定を削除するには

同期デバッグ設定を削除するには、([同期デバッグ]ダイアログボックスを開く前に)Sync ディレクトリから同期デバッグ設定のファイルを削除するか、以下のように操作します。

1. ドロップダウンリストから削除したい同期デバッグ設定を選択します。
2. [削除]ボタンをクリックします。

同期デバッグ設定はドロップダウンリストから削除され、ファイルが Sync ディレクトリから削除されます。ドロップダウンリスト中の次の同期デバッグ設定(アルファベット順)が選択され、読み込まれます。

## 18.1.2 同期するセッションを定義する



### 同期するセッションリスト

ダイアログボックスの中央にあるリストボックスは同期するセッション(通常、High-performance Embedded Workshop ワークスペース内のデバッグセッションに対応します)についての情報を示します。

ダイアログボックスを初めて開いた時には、リストはローカル PC 上で動作しているすべての High-performance Embedded Workshop アプリケーションで開いているすべてのセッションを表示します。同期デバッグ設定の選択を変更したり、新しい同期デバッグ設定を作成したりした時も同様の設定になります。これは、マルチコアの同期デバッグ設定のセットアップを簡単にするためです。

例えば、異なる High-performance Embedded Workshop アプリケーションで複数のセッションを開いた上で、ある 1つのアプリケーションで[同期デバッグ]ダイアログボックスを開き、各セッションを個別にセッションリストに追加することなく、同期デバッグを開始することが可能です。

プラットフォーム(またはマルチコアプラットフォーム上のコア)は、High-performance Embedded Workshop のデバッグセッションを最初に作成しないと同期できません。



**注:**

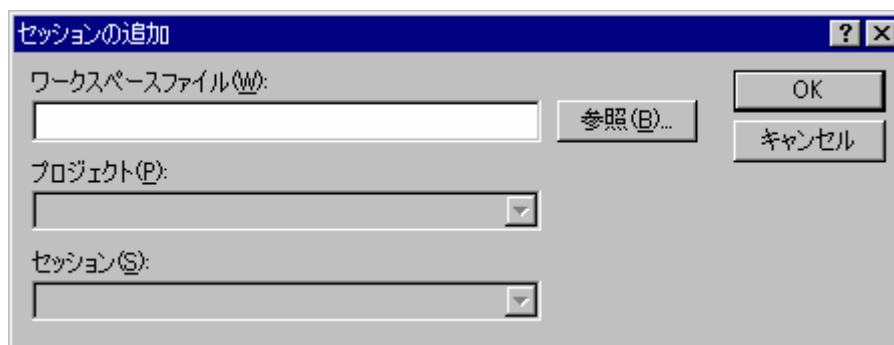
セッションリスト中のセッションがプラットフォームと接続中だった場合、セッションリストは読み出し専用になります。セッションの追加や変更、削除はできません。また、プラットフォームとすでに接続しているセッションはリストに自動的に追加されません。

リストの各エントリは以下の情報を含みます。

<b>Debugger ID:</b>	独自の ID。デバッグセッションを特定できるものに設定してください。
<b>Status:</b>	下記のいずれかになります。
<b>Not open:</b>	セッションは現在 High-performance Embedded Workshop アプリケーションで開いていません。[同期]ボタンがクリックされたときに開きます。
<b>Not connected:</b>	セッションは現在プラットフォームと接続していません。
<b>Break:</b>	セッションはプラットフォームと接続しています。ターゲットプログラムは動作していません。
<b>Running:</b>	セッションはプラットフォームと接続しています。ターゲットプログラムが実行中です。
<b>Options:</b>	同期デバッグを開始した時に、セッションがどのように使われるかを指定するプラットフォーム固有のオプションです。通常のデバッグ時には、"Debug"を選択してください。
<b>Type:</b>	セッションと関連するプラットフォームの型。下記のいずれかになります。
<b>Single core:</b>	通常のシングルコアプラットフォーム
<b>Multi core:</b>	マルチコアプラットフォーム
<b>Workspace:</b>	セッションを含むワークスペースファイルの絶対パスファイル名
<b>Session:</b>	ワークスペース内のセッション名

**リストにセッションを追加するには**

1. [追加]ボタンをクリックします。



2. 有効な High-performance Embedded Workshop ワークスペースファイルを入力するかブラウズします。
3. ワークスペース内のプロジェクトを選択します。
4. プロジェクト内のセッションを選択します。
5. OK ボタンをクリックします。

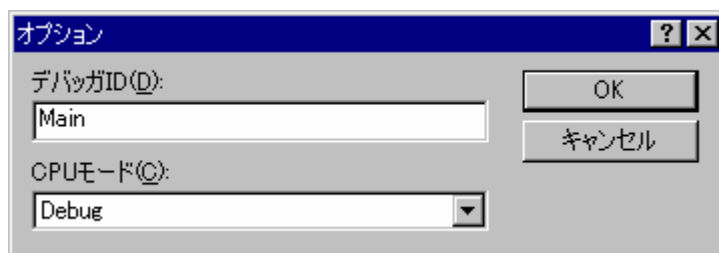
ワークスペースとプロジェクトとセッションの組み合わせを指定します。これは同期デバッグダイアログボックスのセッションリストに追加されます。セッションにはプラットフォーム名(通常シングルコアプラットフォームの場合)やコア名(マルチコアプラットフォームの場合)に基づいた Debugger ID が与えられます。

**注：**

指定されたワークスペースは、最新版の High-performance Embedded Workshop のバージョンで保存されている必要があります。また、保存した後に新しい位置にコピーされてはいけません。旧バージョンで保存されたファイル、あるいは保存後にコピーされたファイルはこのダイアログボックスで使用可能にするには High-performance Embedded Workshop (ワークスペースファイルをアップデートするように)でそのようなワークスペースを開いて、閉じなければなりません。

**Debugger ID を変更するには**

1. 変更するセッションをリストで選択します。
2. [変更]ボタンをクリックします。



3. セッションの新しいユニークな ID を入力します。
4. OK ボタンをクリックします。

リスト上の Debugger ID が更新されます。

同期デバッグ中は、Debugger ID は変更できません。

**セッションのオプションを変更するには**

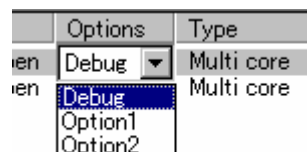
セッションのオプションを変更する方法は2つあります。

**方法1：**

1. リストからセッションを選択します。
2. [変更]ボタンをクリックします。
3. [CPU モード]ドロップダウンリストから新しいオプションを選択します。
4. OK ボタンをクリックします。

**方法2：**

1. リストからセッションを選択します。
2. [Options]のエントリをクリックします。ドロップダウンリストが表示されます。



3. ドロップダウンリストから新しいオプションを選択します。

同期デバッグ中は、セッションのオプションは変更できません。

### セッションをリストから削除するには

1. 削除するセッションを選択します。
2. [削除]ボタンをクリックします。

セッションはリストから削除されます。

### 18.1.3 同期オプションを設定する

#### 概要

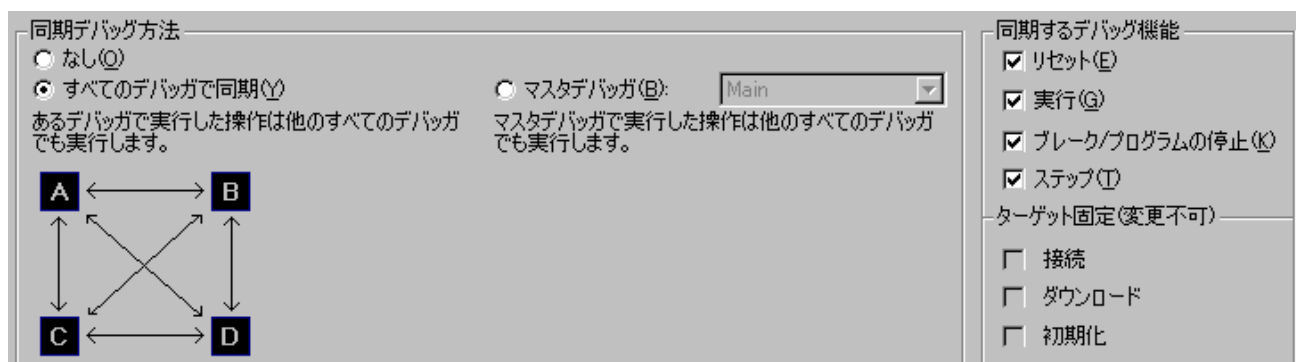
特定のデバッグ動作(例えば、[実行])が実行されたセッションから別のセッションへ、その動作を同期させる設定が有効な場合、その動作は両方のセッションで実行されます。どのセッションでその動作を実行する必要があるかを理解するために同期設定は複数のセッションでチェックされます。同期設定には単方向と双方向があります。

#### 例 1:

2つのセッションがそれぞれ「CORE0」と「CORE1」というIDで同期デバッグ中とします。「CORE0」から「CORE1」への[実行]デバッグ動作の単方向同期設定が一つだけあります。[実行]が「CORE0」セッションで実行された場合、[実行]は両方のセッションで実行されます。[実行]が「CORE1」セッションで実行された場合、[実行]は「CORE1」でしか実行されません。

#### 例 2:

4つのセッションがそれぞれ「CORE0」、「CORE1」、「CORE2」、「CORE3」というIDで同期デバッグ中とします。[実行]デバッグ動作の双方向同期設定がすべてのセッション間で存在します。[実行]がどのセッションで実行されても、すべてのセッションで[実行]は発生します。



### 同期デバッグ方法

ダイアログボックスで3つの同期デバッグ方法を指定できます。

<b>なし:</b>	同期はありません。
<b>すべてのデバッガで同期:</b>	チェックされた動作に関して、すべてのセッション間で双方向の同期を設定します。
<b>マスタデバッガ:</b>	チェックされた動作に関して、指定されたマスタデバッガから他のすべてのデバッガへの単方向の同期を設定します。ドロップダウンリストでマスタデバッガとなるデバッガを指定します。

## 同期するデバッグ機能

ここでのチェックボックスは同期するデバッグ動作を示します。[リセット]、[実行]、[ブレーク/プログラムの停止]、[ステップ]は必要に応じて同期・非同期を設定できます。[接続]、[ダウンロード]、[初期化]の同期・非同期はデバッグするプラットフォームによって決まります。非マルチコアプラットフォームでの状態は常にオフです。

## プラットフォームの設定

マルチコアプラットフォームのデバッグ時、通常はユーザが指定できる複数の同期が自動的に有効になっていることもあります。オプションのチェックボックスがチェックされてグレーになっている場合、その動作が常に同期されることがプラットフォームから指定されています。

注:

- プラットフォームによる設定は SDO(同期デバッグ設定)ファイルを書き換えることで無効にはできません。
- プラットフォームによる設定は選択した同期デバッグ方法より優先します。これらの動作は、「なし」や「マスタデバッガ」が選択されていても、すべてのデバッガで常に同期化されます。

### 18.1.4 メモリ更新オプションを設定する

メモリ変更時にすべてのデバッガの表示を更新(P)

このオプションがチェックされている場合、全セッションのメモリデータ（メモリウィンドウ、ウォッチウィンドウ等）を表示するすべての High-performance Embedded Workshop のウィンドウは、同期デバッグ中のセッションのいずれかのメモリが変更された時、必ず更新されます（すなわち、プラットフォームからメモリを読み出します）。

オプションが選択されていない場合、メモリが変更された時にはローカルセッションのメモリウィンドウのみが更新されます。セッション間でメモリを共有している場合、他のセッションのメモリ関連ウィンドウで正しいメモリ内容を表示するには、それらのウィンドウでマニュアルリフレッシュを実行する必要があります。

セッション間でメモリを共有するのは通常マルチコアプラットフォームをデバッグする時のみです。セッション間でメモリを共有しない場合（複数のシングルコアプラットフォームをデバッグする場合等）、あるセッションにおける動作は他のセッションのメモリを変更しないため、本チェックボックスの状態は関係ありません。

### 18.1.5 同期デバッグモード



デバッグ時のモードを選択してください。

**Internal:** 指定されたセッションは1つの High-performance Embedded Workshop アプリケーションで開かれます。このモードはすべてのセッションが同じ High-performance Embedded Workshop ワークスペース中にある場合にのみ選択可能です。

**Parallel:** 指定されたセッションは異なる High-performance Embedded Workshop アプリケーションで開かれます。

**注：**

Internal モードは、同期するセッションのリスト中のすべてのセッションが同じワークスペースにある場合のみ選択可能です。異なるワークスペースからのセッションが選択された場合には、Internal モードは使用できません。

### 18.1.6 同期デバッグを開始する

同期デバッグの設定が終了したら、[同期]ボタンをクリックしてデバッグを開始してください。

Parallel モードが選択されている場合、[同期デバッグ]ダイアログボックスを開いていた High-performance Embedded Workshop から 1 つのセッションが開かれ、残りのセッションのために新しい High-performance Embedded Workshop が起動します。

Internal モードを選択した場合、すべてのセッションに共通するワークスペースが開きます。

同期デバッグ時の High-performance Embedded Workshop の使用方法の詳細については、「18.2 同期デバッグ時に High-performance Embedded Workshop を使用する」を参照してください。

### 18.1.7 同期デバッグを更新する

同期デバッグ中に[同期デバッグ]ダイアログボックスを開いた時、[同期]ボタンの代わりに[更新]ボタンがあります。[同期デバッグ]ダイアログボックスの設定を変更した場合、[更新]ボタンをクリックすると、新しい設定にしたがって更新されます。Parallel モード時で追加の High-performance Embedded Workshop アプリケーションを開くことも含みます。

同期デバッグ時の High-performance Embedded Workshop の使用方法の詳細については、「18.2 同期デバッグ時に High-performance Embedded Workshop を使用する」を参照してください。

### 18.1.8 同期デバッグを停止する

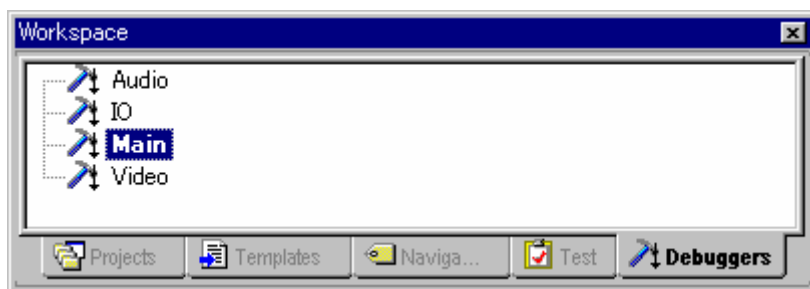
同期デバッグ中に[同期デバッグ]ダイアログボックスを開いた時、[非同期]ボタンが有効になっています。同期デバッグ機能を停止するには、[非同期]ボタンをクリックしてください。同期化されていた、あらゆる High-performance Embedded Workshop アプリケーションはそれぞれのプラットフォームから接続を切断され、再び独立して機能します。Debuggers タブはワークスペースウィンドウから削除されます。セッションの動作は非同期になります。また、どの High-performance Embedded Workshop アプリケーションも終了しません。

## 18.2 同期デバッグ時に High-performance Embedded Workshop を使用する

同期デバッグモードで High-performance Embedded Workshop を使用する主な目的は、通常マルチコアプラットフォームに接続された異なる High-performance Embedded Workshop セッションで主要なデバッグ操作を同期化することです。




High-performance Embedded Workshop の機能は、Parallel モードと Internal モードで共通する部分と相違する部分があります。本項ではそれらについて説明します。

## 18.2.1 共通の機能



### Debuggers タブ

同期デバッグを開始すると、Debuggers タブがワークスペースウィンドウに追加されます。同期するセッションごとに異なる項目があります。各項目のテキストは[同期デバッグ]ダイアログボックスで指定された Debugger ID です。各項目のアイコンはセッションの現在の状態を示します。以下のステータスアイコンが使用可能です。

-  セッションはプラットフォームと接続されていません。
-  セッションは現在停止している（つまり、動作していない）プラットフォームと接続されています。
-  セッションは現在動作しているプラットフォームと接続されています。

Debugger ID のアイコンを右クリックすると、以下のメニュー項目が表示されます。

- CPU のリセット
- 実行
- リセット後実行
- ステップイン
- ステップアウト
- ステップオーバ
- プログラムの停止
- ドッキングビュー
- 非表示
- プロパティ

メニュー上でデバッグ操作を選択した時、選択した（すべての）セッションでのみ実行されます。デバッグ操作を Debuggers タブで実行した場合、すべての同期設定は無視されます。マルチコアプラットフォームのデバッグ時、いくつかのデバッグ操作は無効になります。これらは、ハードウェアにより常に同期されるデバッグ操作です。つまりいくつかのデバッグ操作はプラットフォームのコアで個々に実行することができないことを意味します。

選択したセッションでその動作を実行できない場合は、メニュー項目も無効になります。例えば、動作しているセッションを右クリックした場合、[プログラムの停止]メニュー項目のみが有効になります。

[プロパティ]メニュー項目を使ってセッションの情報を取得できます。選択した場合、以下の情報を含んだダイアログボックスが表示されます。

：ワークスペース名、プロジェクト名、セッション名。

[アクティブセッションに設定]メニュー項目は Internal モードでのデバッグ時にのみ有効になります。選択した場合、選択したセッションは現在アクティブなセッションになります。詳細については、「18.2.3 Internal モードの機能」を参照してください。



## セッションの動作

同期できる動作を下の表に示します。同期する動作は同期デバッグダイアログボックスの設定により決まります。表の各項目の意味は以下のとおりです。

動作：	動作名。
オプション：	動作を同期化するために、同期デバッグダイアログボックスでチェックしておくべきチェックボックスのオプション（単数または複数）。
方法：	動作が High-performance Embedded Workshop でどのように実行されるかの説明。
効果：	動作を実行した効果（すべての同期するセッションにおいて）。

[初期プラットフォーム]とは、動作が開始したセッションに接続したプラットフォームを意味します。実行する動作によって、本セッションにおける効果は他のセッションでの効果と異なります。

動作	オプション	方法	効果
Reset CPU	リセット	"デバッグ->CPUのリセット"メニュー。 CPUリセットツールバーボタン。 "reset"コマンド。 "ダウンロード後にCPUリセット"オプションが有効な時はダウンロード後。	すべての同期化されたプラットフォームをリセットします。
Go	実行	"デバッグ->実行"メニュー。 実行ツールバーボタン。 "go"コマンド。	すべての同期化されたプラットフォームを実行します。
Go Until	実行	"デバッグ->条件を指定して実行"メニュー。 "リセット時にmain関数の先頭まで実行"オプションが有効な時はリセット後。 エディターポップアップ"カーソル位置まで実行"メニュー。	一時的なブレークポイント群を設定した状態で初期プラットフォームを実行します。 他のすべての同期化されたプラットフォームは通常通りに実行します。
Reset Go	リセットと実行	"デバッグ->リセット後実行"メニュー。 リセット後実行ツールバーボタン。 "go_reset"コマンド。	すべての同期化されたプラットフォームをリセットした後で実行します。 (【注】ユーザのプラットフォームによっては、本動作はリセットの後に実行を個別に実行した場合と同じでない可能性もあります。)
Break	ブレーク/プログラム停止	ブレークポイントに達したとき。	動作しているすべての同期化されたプラットフォームを停止します。
Halt	ブレーク/プログラム停止	"デバッグ->停止"プログラムメニュー。 停止ツールバーボタン。 "halt"コマンド。	動作しているすべての同期化されたプラットフォームを停止します。
Step In	ステップ	"デバッグ->ステップイン"メニュー。 ステップインツールバーボタン。 "step"コマンド。	すべての同期化されたプラットフォームに対してステップインを実施します。
Step Over	ステップ	"デバッグ->ステップオーバー"メニュー。 ステップオーバーツールバーボタン。 "step_over"コマンド。	すべての同期化されたプラットフォームに対してステップオーバーを実施します。
Step Out	ステップ	"デバッグ->ステップアウト"メニュー。 ステップアウトツールバーボタン。 "step_out"コマンド。	すべての同期化されたプラットフォームに対してステップアウトを実施します。
Connect	接続	"デバッグ->接続"メニュー。 接続ツールバーボタン。 "connect"コマンドやプラットフォーム固有のコマンド。	すべての同期するセッションのプラットフォームに接続します。
Download (file load)	ダウンロード	"デバッグ->ダウンロード"メニュー項目。 ワークスペースタブ"Download modules"フォルダ。 "file_load"コマンド。 "ターゲット接続時にダウンロード"オプションが設定されている時は接続時。 "ビルド後のダウンロード"オプションが設定されている時はビルド後。 モジュールが High-performance Embedded Workshop の外部で変更された時。	すべての同期化されたプラットフォームにモジュールをダウンロードします。モジュールのファイル名は各セッションにおいて同じでなければなりません。
Initialize	初期化	"デバッグ->初期化"メニュー。 "initialize"コマンド。	すべての同期化されたプラットフォームは初期化されます。



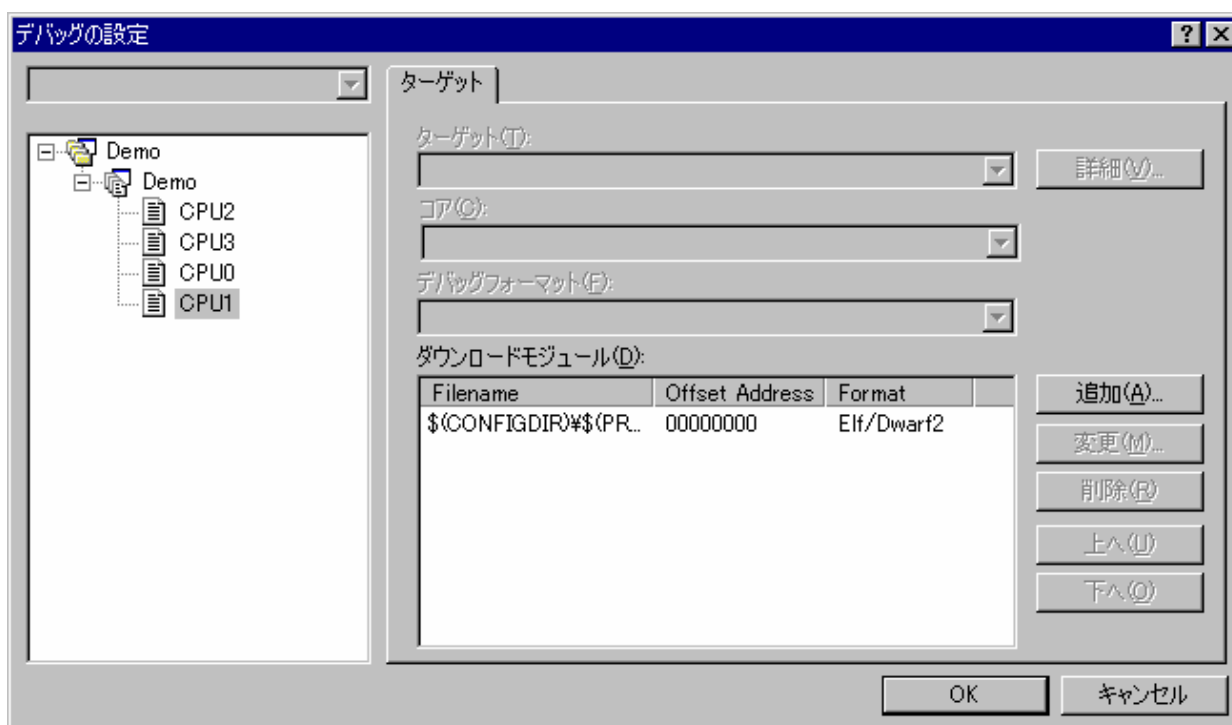
ステップに関しては、同期セッションでのステップのスタイル（ソースか命令か）はプログラムカウンタ、ステップモード、そして（[ステップイン]ボタンをクリックしたセッションでなく）ステップを実行するセッションでのエディタのカーソル位置で決まります。

通常の High-performance Embedded Workshop のステップ実行（例えば、ツールバーボタンの[ステップイン]ボタンをクリックしたように）でステップ操作がローカルに実行されたかのように、それぞれの個々の Debugger は動作します。

ダウンロードに関しては、同期処理を開始するセッションのモジュールと同じファイル名を持つモジュールを持つセッションでのみ動作が同期化されます。セッションが1つ以上のモジュールを持つ場合は、一致するファイル名を持つモジュールだけがダウンロードされます。

表に記載した以外の High-performance Embedded Workshop の操作は、同期デバッグ時に行うことが可能です。しかし、それらの動作は、同期処理を開始するセッションでのみ実行され、他のセッションでは同期実行されません。

### デバッグの設定ダイアログボックス



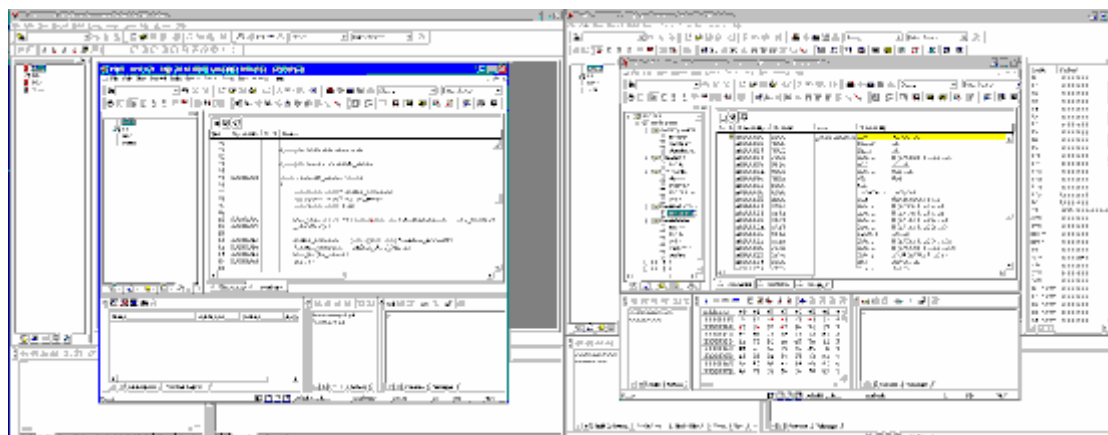
同期デバッグ時、[デバッグの設定]ダイアログボックスの機能が変わります。通常デバッグ時に使用可能なオプションを全部設定することはできず、セッションごとに設定する Download modules のみを変更できます。ダイアログボックスの左側で[Multiple]を使用して複数のセッションの設定を同時に変更することが可能です。これにより、同期ダウンロードの準備が整った複数のセッションに同じモジュールを容易に追加できます。

ダウンロードモジュールを選択し、[変更]ボタンをクリックすると、[ダウンロードモジュール]ダイアログボックスが開きます。このダイアログで[デバッグ情報のみのダウンロード]を設定することができます。

マルチコアデバッグですべてのコアが同じメモリ領域を共有している場合、メモリイメージは1つのセッションで一度だけダウンロードし、他のセッションではデバッグ情報だけダウンロードすることを推奨します。

ダイアログボックスのボタンの機能性は通常デバッグ時と同じです。詳細については、「17.1.4.3 デバッグの設定を変更する」を参照してください。

## 18.2.2 Parallelモードの機能



### 一般的な動作

Parallelモード動作時、各セッションは異なる High-performance Embedded Workshop アプリケーションで開かれます。いくつかのセッションの動作が同期するだけで、通常デバッグ時と同じようにこれらのセッションは独自に使用できます。

### High-performance Embedded Workshop ウィンドウのタイトル

High-performance Embedded Workshop ウィンドウのタイトルはアプリケーション中で開いているセッションの表示 ID から始まります。アプリケーションが同期実行中、タイトルは変わりません。

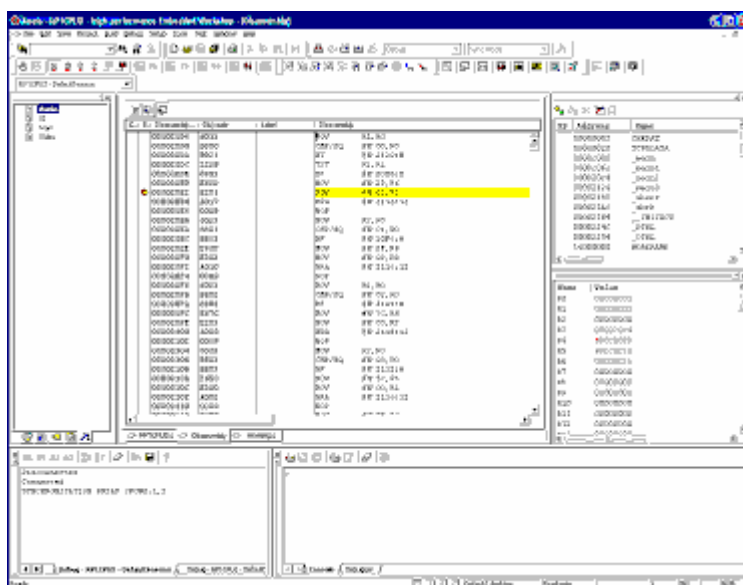
### 同期デバッグの終了

現在のセッションを変更する動作を実行すると、High-performance Embedded Workshop アプリケーションは同期デバッグシステムから削除され、他の High-performance Embedded Workshop アプリケーションと非同期になります。

セッションを変更する動作は以下のとおりです。

- 新規ワークスペースの作成
- ワークスペースを開く
- ワークスペースを閉じる
- アクティブプロジェクトに設定
- セッションの変更

### 18.2.3 Internalモードの機能



#### 概要

Internalモード動作時、すべてのセッションは1つのHigh-performance Embedded Workshopアプリケーションで開かれます。別の開いたセッションをデバッグするには、セッションを（様々な方法で）切り替えます。セッションを切り替えて、あるセッションを選んだ時、デバッグビューはそのセッションのデータを表示するために変更され、それ以降実行される動作はそのセッション（「アクティブセッション」）上で実行されます。これにより、一つのHigh-performance Embedded Workshopアプリケーションを用いて、複数のプラットフォーム、またはマルチコアプラットフォーム上の複数のコアをデバッグできます。

#### High-performance Embedded Workshop ウィンドウのタイトル

High-performance Embedded Workshop ウィンドウのタイトルは現在のセッションの表示IDから始まります。アクティブセッションが変わる度にタイトルは変わります。

#### アクティブセッションの切替え

アクティブセッションを切り替える方法はいくつかあります。

Debuggers タブ:

タブ中のセッションを右クリックして、"アクティブセッションに設定"を選択することで、アクティブセッションに設定できます。

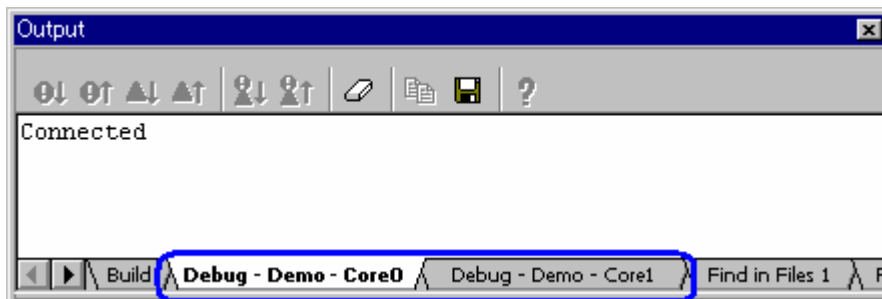
アプリケーションツールバー:

Internalモードでデバッグ時、アプリケーションツールバーにドロップダウンリストが追加されます。このドロップダウンリストは、同期するセッションを説明するプロジェクト/セッションの組み合わせを含みます。ある特定のセッションをドロップダウンリストから選択することで、アクティブセッションをそのセッションに変更できます。

ショートカットキーの組み合わせ: "Ctrl+Shift+N"キーを同時に押すことにより、アクティブセッションをDebuggersタブ中の次のセッションに変更できます。

## Output ウィンドウ

Internal モード使用時、Output ウィンドウは同期するセッションごとに異なる[Debug]タブを表示します。各タブには、対応するセッションが認識できるように、セッションのプロジェクト名とセッション名があります。



## 制限される動作

Internal モードでは、以下の機能は使用できません。

- アクティブプロジェクトに設定
- プロジェクトの削除
- プロジェクトのアンロード
- セッションの変更
- コンフィグレーションの変更

同期デバッグ時、[ビルド]は現在のプロジェクトとコンフィグレーションをビルドします。

非アクティブのプロジェクトとコンフィグレーションをビルドする場合は、[複数のビルド]ダイアログボックスを使用してください。

## 18.3 同期デバッグ時にコマンドラインウィンドウを使用する

### 概要

同期デバッグ機能を使用時、2つの新しいコマンドが使えます。これらのコマンドにより、あるセッションで実行されたコマンドファイルが別のセッションでコマンドを実行できるように、コマンドを実行しているセッションを変えることができます。

### SET\_CORE コマンド

**短縮形:** SMC

**説明:** 現在のセッションをコマンドラインウィンドウから設定します。これ以降のすべてのコマンドは要求されたセッションに送られます。

**注:** GUI から呼び出された動作は影響されません。

**シンタックス:** smc <Debugger ID>

パラメータ	型	説明
<Debugger ID>	文字列	セッションの ID。空の場合、コントロールはローカルセッションに戻されます。

**例:** smc "SH-X (core 1)"

これ以降、コマンドは"SH-X (コア 1)"セッションに送られます。

## SHOW\_ALL\_CORE コマンド

短縮形: SAC

説明: 使用可能なセッションのリストを表示します。このコマンドは前回のコマンドの有効な入力のリストを取得するために使われます。表示されるセッションのリストは Debuggers タブに表示されているものと同じです。

シンタックス: sac

パラメータ	型	説明
なし	なし	使用可能なセッションのリストを表示します。

### 例外

「SET\_CORE」コマンドを使用した場合でも、いくつかのコマンドは「SET\_CORE」コマンドで設定したセッションで実行されず、ローカルなセッションで実行されます。以下のコマンドには、この制限があります:

```
"ASSEMBLE"
"CLOSE_TEST_SUITE"
"COMPARE_TEST_DATA"
"MEMORY_EDIT"
"OPEN_TEST_SUITE"
"REFRESH_SESSION"
"RUN_TEST"
"SET_CORE"
"SHOW_ALL_CORE"
"SUBMIT"
```

### 注:

「SET\_CORE」コマンドで設定したセッションで、コマンドパラメータにファイル名を含むコマンドを実行する場合、ファイル名は以下のように中括弧「{」、「}」で囲んで指定してください。

```
file_load Elf/Dwarf2 {$(CONFIGDIR)\$(PROJECTNAME).abs} 0x0 p byte
```

## 18.4 用語の解説

用語	定義内容
コア	マルチコアプラットフォーム上の個別の CPU コア。
デバッグ	特定の対象をデバッグするために High-performance Embedded Workshop アプリケーションから開くことができます。ワークスペースファイルとプロジェクト名、セッション名の組み合わせで指定されます。同期デバッグモードで動作時、同期化されている High-performance Embedded Workshop アプリケーション（Parallel モードの場合）またはセッション（Internal モードの場合）を指す総称です。
デバッグ ID	High-performance Embedded Workshop GUI でセッションを独自に特定するための文字列。
同期を開始するセッション	動作が実行されたセッション。
Internal モード	1つの High-performance Embedded Workshop アプリケーションですべてのセッションを開いた状態で同期デバッグ機能を使用しているモード。
Parallel モード	別々の High-performance Embedded Workshop アプリケーションで各セッションを開いた状態で同期デバッグ機能を使用しているモード。
ローカルセッション	ユーザが現在やりとりしているセッション。
プラットフォーム	High-performance Embedded Workshop に接続されているハードウェアまたはシミュレータ。シングルコアプラットフォームまたはマルチコアプラットフォーム上の個別のコアを意味することもあります。
マルチコアプラットフォーム	2つ以上の CPU コアを持つハードウェアまたはシミュレータデバイス。同期デバッグ機能の主な目的はユーザのマルチコアプラットフォームのデバッグをサポートすることです。
SDO ファイル	同期デバッグ設定のファイル。同期デバッグセッションを起動させるのに High-performance Embedded Workshop が必要とするすべての情報を含みます。同期デバッグダイアログボックスで保存されます。

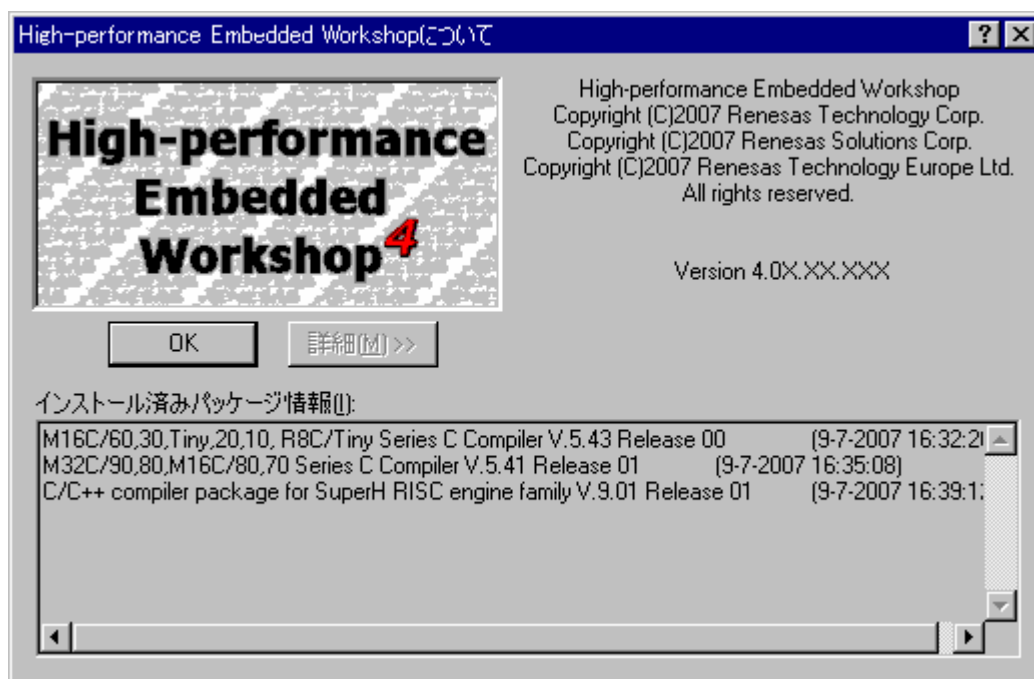
## 19. テクニカルサポート

### 19.1 バージョン情報の参照

High-performance Embedded Workshop のバージョン情報を参照できます。

#### バージョン情報を参照するには

1. [ヘルプ -> High-performance Embedded Workshop のバージョン情報]を選択してください。[High-performance Embedded Workshop について]ダイアログボックスが開きます。著作権、バージョン番号などを表示します。
2. さらに[詳細>>]ボタンをクリックすると、[インストール済みパッケージ情報]に使用するコンピュータへインストールしている High-performance Embedded Workshop と連携する Renesas 製品パッケージの情報（バージョン番号など）を表示します。



3. [OK]ボタンをクリックすると、ダイアログボックスを閉じます。

### 19.2 アップデートの確認

High-performance Embedded Workshop 製品のバージョンアップやサービスパックを確認するには

1. [ヘルプ -> テクニカルサポート -> アップデートの Web 確認]を選択してください。
2. デフォルトのウェブブラウザが起動して、お客様の地域の High-performance Embedded Workshop ダウンロードページを表示します。
3. High-performance Embedded Workshop のバグの修正や新規機能の追加の更新情報を確認してください。





## リファレンス

## 1. メインメニュー

### 1.1 ファイルメニューオプション

メニュー	メニューオプション	ショートカットキー	ツールボタン	マクロの記録*	機能
ファイル	新規作成	Ctrl+N		-	新しいエディタを作成します。
	開く	Ctrl+O		-	ファイルを開きます。
	閉じる	Ctrl+F4	-	-	ファイルを閉じます。
	新規ワークスペース	-	-	-	新規にワークスペースを作成します。
	ワークスペースを開く	-	-	●	ワークスペースを開きます。
	ワークスペースの保存	-	-	●	ワークスペースを保存します。
	旧バージョン形式でワークスペースを保存	-	-	-	別名または旧フォーマットでワークスペースを保存します。
	ワークスペースを閉じる	-	-	●	ワークスペースを閉じます。
	新規セッション	-	-	●	新しいセッションを作成します。
	セッションのインポート	-	-	●	既存のセッションをインポートします。
	セッションの保存	-	-	●	現在のセッションを保存します。
	名前を付けてセッションを保存	-	-	-	現在のセッションを別名で保存します。
	セッションのリフレッシュ	-	-	●	現在のセッションを再ロードします。
	ダウンロードモジュールの追加	-	-	●	新しくモジュールをダウンロードします。
	上書き保存	Ctrl+S		-	エディタウィンドウの内容を保存します。
	すべて保存	Ctrl+Shift+S		-	すべての開いているエディタウィンドウの内容を保存します。
	名前を付けて保存	-	-	-	エディタウィンドウの名前を新しい名前で保存します。
	ページレイアウトの設定	-	-	-	ページレイアウトを設定します。
	印刷	Ctrl+P		-	ファイルを印刷します。
	最近使ったファイル	-	-	-	最近使ったファイルを開きます。
	最近使ったワークスペース	-	-	●	最近使ったワークスペースを開きます。
	最近使ったテストスイート	-	-	-	最近使ったテストスイートを開きます。
	最近ダウンロードしたモジュール	-	-	●	最近ダウンロードしたモジュールをダウンロードします。
アプリケーションの終了	-	-	-	High-performance Embedded Workshop を終了します。	

#### 注：

\*. 一部のメニューオプションに対する操作は、マクロ生成支援機能を使用し、High-performance Embedded Workshop コマンドラインのコマンドを記録できます。このリストでは、メニューオプションの「マクロの記録」欄にマクロの記録マーク(●)を表示し、マクロファイルへ記録可能かどうかを示しています。これらの詳細は「15.5.1 マacroを記録できる機能 ( High-performance Embedded Workshop 共通 ) 」を参照してください。

### 1.2 編集メニューオプション



メニュー	メニューオプション	ショートカットキー	ツールボタン	マクロの記録*	機能
編集	元に戻す	Ctrl+Z	-	-	最新の編集操作を取り消します。
	やり直し	Ctrl+Y	-	-	最新の取り消した編集操作をやり直します。
	切り取り	Ctrl+X		-	選択部分を削除し Windows® クリップボードに貼り付けます。
	コピー	Ctrl+C		-	選択部分をコピーし Windows® クリップボードに貼り付けます。

メニュー	メニューオプション	ショートカットキー	ツールボタン	マクロの記録*	機能
	貼り付け	Ctrl+V		-	Windows® クリップボードの内容をコピーしてアクティブウィンドウのカーソル位置に貼り付けます。
	削除	Delete	-	-	選択部分を削除します (Windows® クリップボードに貼り付けません)。
	すべて選択	Ctrl+A	-	-	アクティブウィンドウの内容をすべて選択します。
	検索	Ctrl+F		-	現在のファイルの文字列を検索します。
	ファイルから検索	F4		-	複数のファイルで文字列を検索します。
	置換	Ctrl+H	-	-	現在のファイルの文字列を置換します。
	ジャンプ	Ctrl+G	-	-	指定した行にジャンプします。
	括弧の呼応	Ctrl+Shift+M		-	括弧の組み合わせを見つけます。
ブックマーク	ブックマークの挿入/削除	Ctrl+F2		-	ブックマークを設定/解除します。
	次のブックマーク	F2		-	次のブックマークにジャンプします。
	前のブックマーク	Shift+F2		-	同じファイルの1つ前のブックマークに戻ります。
	すべてのブックマークの削除	-		-	ファイル内のすべてのブックマークを削除します。
テンプレート	テンプレートの定義	-		-	テンプレートを設定します。
	テンプレートの挿入	Ctrl+Shift+T		-	テンプレートを挿入します。
	ブレークポイントの挿入/削除	F9		●	標準のブレークポイント(PC ブレークポイント)を挿入/削除します。
	ブレークポイントの有効化/無効化	Ctrl+F9		●	現在設定している PC ブレークポイントの有効/無効の切り替えができます。
	表示カラムの設定	-	-	-	表示カラムを設定します。
	ソースブレークポイント	Ctrl+B	-	-	[ブレークポイント]ダイアログボックスを開きます。
	式の評価	-	-	-	式を評価します。

## 注：

\*. 一部のメニューオプションに対する操作は、マクロ生成支援機能を使用し、High-performance Embedded Workshop コマンドラインのコマンドを記録できます。このリストでは、メニューオプションの「マクロの記録」欄にマクロの記録マーク(●)を表示し、マクロファイルへ記録可能かどうかを示しています。これらの詳細は「15.5.1 マacroを記録できる機能 (High-performance Embedded Workshop 共通)」を参照してください。

## 1.3 表示メニューオプション


メニュー	メニューオプション	ショートカットキー	ツールボタン	マクロの記録*1	テストイメージデータの保存*2	機能
表示	差分	-		-	-	差分ウィンドウを開きます。
	マップ*3	-		-	-	2種類のマップウィンドウを開きます。
	コマンドライン	Ctrl+L		-	-	コマンドラインウィンドウを開きます。
	TCL ツールキット	Ctrl+Shift+K		-	-	詳細は別冊の「TCL/TK 添付資料」を参照してください。
	ワークスペース	Alt+K		●	-	ワークスペースウィンドウを開きます。
	アウトプット	Alt+O		-	 *4	アウトプットウィンドウを開きます。
	ステータスバー	Alt+A		-	-	ステータスバーの表示/非表示を切り替えます。
	逆アセンブリ	Ctrl+D		●	-	逆アセンブリウィンドウを開きます。
	CPU	レジスタ	Ctrl+R		●	
メモリ		Ctrl+M		●		メモリウィンドウを開きます。
IO		Ctrl+I		●		[IO]ウィンドウを開きます。
ステイタス		Ctrl+U		-		ステイタスウィンドウを開きます。

メニュー	メニューオプション	ショートカットキー	ツールボタン	マクロの記録 *1	テストイメージデータの保存 *2	機能
グラフィック	画像 *3	Ctrl+Shift+G		-	-	画像ウィンドウを開きます。
	波形 *3	Ctrl+Shift+V		-	-	波形ウィンドウを開きます。
コード	スタックトレース	Ctrl+K		-		スタックトレースウィンドウを開きます。

**注：**

\*1. 一部のウィンドウに対する特定の操作は、マクロ生成支援機能を使用し、High-performance Embedded Workshop コマンドラインのコマンドを記録できます。このリストでは、メニューオプションの「マクロの記録」欄にマクロの記録マーク(●)を表示し、マクロファイルへ記録可能かどうかを示しています。これらの詳細は「15.5.1 マクロを記録できる機能 (High-performance Embedded Workshop 共通)」を参照してください。

このリストに記載していないデバッガに依存するウィンドウでもコマンドが記録できるものがあります。これらの詳細は「15.5.2 マクロを記録できる機能 (デバッガ依存)」を参照してください。

\*2. 一部のウィンドウのデータは、テスト支援機能を使用し、テストイメージデータとしてテストイメージファイルに保存できます。このリストでは、テストイメージファイルに保存できるウィンドウのメニューオプションの「テストイメージデータの保存」欄に[ファイルの保存]マーク()を記載しています。これらの詳細は「16.6.1 テストイメージファイルに保存できる機能 (High-performance Embedded Workshop 共通)」を参照してください。

このリストに記載していないデバッガに依存するウィンドウでもテストイメージファイルに保存できるものがあります。これらの詳細は「16.6.2 テストイメージファイルに保存できる機能 (デバッガ依存)」を参照してください。

\*3. 機能のサポートはデバッガに依存します。

\*4. [Build]タブまたは[Debug]タブのデータが保存可能です。

## 1.4 プロジェクトメニューオプション

メニュー	メニューオプション	ショートカットキー	ツールボタン	マクロの記録 *1	機能
プロジェクト	アクティブプロジェクトに設定	-	-	●	プロジェクトを現在のプロジェクトに設定します。
	プロジェクトの挿入	-	-	●	ワークスペースにプロジェクトを追加します。
	依存関係	-	-	-	依存プロジェクトを生成します。
	構成の編集 *2	-	-	●	プロジェクトの構成を編集します。
	プロジェクトタイプの生成	-	-	-	現在のプロジェクトの設定を利用してプロジェクトのテンプレートを作成します。
	ファイルの追加	-	-	-	プロジェクトにファイルを追加します。
	ファイルの削除	-	-	-	プロジェクトからファイルを削除します。
	ファイルの拡張子	-	-	-	ファイルの拡張子を定義します。
	コンポーネント	-	-	-	オンデマンドのコンポーネントを設定します。

**注：**

\*1. 一部のメニューオプションに対する操作は、マクロ生成支援機能を使用し、High-performance Embedded Workshop コマンドラインのコマンドを記録できます。このリストでは、メニューオプションの「マクロの記録」欄にマクロの記録マーク(●)を表示し、マクロファイルへ記録可能かどうかを示しています。これらの詳細は「15.5.1 マクロを記録できる機能 (High-performance Embedded Workshop 共通)」を参照してください。

\*2. 機能のサポートはデバッガに依存します。

## 1.5 ビルドメニューオプション

メニュー	メニューオプション	ショートカットキー	ツールボタン	マクロの記録 *1	機能
ビルド *2	Toolchain	-	-	-	ファイルのビルド方法を設定します。
	コンパイル	Ctrl+F7		●	ファイルをコンパイルします。
	ビルド	F7		●	プロジェクトをビルドします。
	すべてをビルド	-		●	プロジェクトを全ビルドします。
	複数ビルド	-	-	●	複数のプロジェクトをビルドします。
	クリーン アクティブプロジェクト	-	-	●	現在のコンフィグレーションの中間ファイルおよび出力ファイルを削除します。
	クリーン 全プロジェクト	-		●	ワークスペースのすべてのプロジェクトのすべてのコンフィグレーションを削除します。
	すべての依存関係を更新	-	-	-	プロジェクトのファイル依存関係を更新します。
	ツールの中止	Ctrl+Break		-	ビルド中のツールの実行を中止します。
	ビルドから除外/除外の解除	-	-	-	ファイルをビルドから除外または除外を解除します。
	ビルドフェーズ	-	-	-	ビルドフェーズを設定します。
	ビルドの構成	-	-	●	ビルドコンフィグレーションを設定します。
	リンク順の設定	-	-	-	リンク順をカスタマイズします。
	Make ファイルの生成	-	-	-	Make ファイルを生成します。

## 注：

- \*1. 一部のメニューオプションに対する操作は、マクロ生成支援機能を使用し、High-performance Embedded Workshop コマンドラインのコマンドを記録できます。このリストでは、メニューオプションの「マクロの記録」欄にマクロの記録マーク(●)を表示し、マクロファイルへ記録可能かどうかを示しています。これらの詳細は「15.5.1 マクロを記録できる機能 (High-performance Embedded Workshop 共通)」を参照してください。
- \*2. 本メニューは、High-performance Embedded Workshop V.4.01 以降で作成したデバッグ専用プロジェクトを使用する場合は表示されません。

## 1.6 デバッグメニューオプション

メニュー	メニューオプション	ショートカットキー	ツールボタン	マクロの記録 *1	機能
デバッグ	同期デバッグ *2	-	-	-	同期デバッグを設定します。
	デバッグセッション	-	-	●	デバッグセッションの一覧表示、および追加/削除等が可能な[デバッグセッション]ダイアログボックスを開きます。
	デバッグの設定	-	-	-	デバッグ時の条件やダウンロードモジュール等の設定を行う[デバッグの設定]ダイアログボックスを開きます。
	CPUのリセット	-		●	ターゲットマイコンをリセットします。
	実行	F5		●	現在の PC からユーザプログラムを実行します。
	リセット後実行	Shift+F5		●	リセットベクタアドレスからユーザプログラムを実行します。
	フリー実行 *2	-		●	ブレークポイントを無視してユーザプログラムを実行します。
	カーソル位置まで実行	-		●	現在の PC アドレスからユーザプログラムの実行を開始します。プログラムは、PC がテキストカーソルの位置によって指定されたアドレスに達するか、別のブレーク条件が成立するまで実行します。
	カーソル位置を PC 値に設定	-		●	PC の値をテキストカーソルの位置によって指定されたアドレスに変更します。
	条件を指定して実行	-	-	●	実行時の PC や PC ブレークポイントの設定が可能な[条件を指定して実行]ダイアログボックスを開きます。
	PC 位置を表示	Ctrl+Shift+Y		-	PC 位置を表示します。
	ステップイン	F11		●	ユーザプログラムの 1 ブロックを実行して停止します。





メニュー	メニューオプション	ショートカットキー	ツールボタン	マクロの記録 *1	機能	
	ステップオーバー	F10		●	ユーザプログラムの1ブロックを実行して停止しますが、サブルーチン呼び出し時は、サブルーチンには入りません。	
	ステップアウト	Shift+F11		●	現在の関数の終わりに到達するまでユーザプログラムを実行します。	
	ステップ	-	-	●	ステップ動作の設定が可能な[プログラムステップ]ダイアログボックスを開きます。	
	ステップモード	自動	-	-	●	エディタウィンドウがアクティブの場合はソースライン1行だけをステップ実行します。逆アセンブリウィンドウがアクティブの場合はアセンブリ言語命令単位にステップ実行します。
		アセンブリ	-	-	●	アセンブリ言語命令単位にステップ実行します。
		ソース	-	-	●	ソースライン1行だけをステップ実行します。
	プログラムの停止	-		●	ユーザプログラムの実行を停止します。	
	初期化	-	-	●	デバッグを切断し、再接続します。	
	接続 *2	-		●	デバッグを接続します。	
	接続解除 *2	-		●	デバッグを接続解除します。	
	メモリの保存	-	-	●	任意のアドレス領域を保存します。	
	メモリのベリファイ *2	-	-	●	アドレス領域を検証します。	
	ダウンロード	-	-	●	オブジェクトプログラムをロードします。	
	アンロード	-	-	●	オブジェクトプログラムをアンロードします。	

## 注：

\*1. 一部のメニューオプションに対する操作は、マクロ生成支援機能を使用し、High-performance Embedded Workshop コマンドラインのコマンドを記録できます。このリストでは、メニューオプションの「マクロの記録」欄にマクロの記録マーク(●)を表示し、マクロファイルへ記録可能かどうかを示しています。これらの詳細は「15.5.1 マクロを記録できる機能 (High-performance Embedded Workshop 共通)」を参照してください。

\*2. 機能のサポートはデバッグに依存します。

## 1.7 基本設定メニューオプション

メニュー	メニューオプション	ショートカットキー	ツールボタン	マクロの記録 *	機能	
基本設定	カスタマイズ	-	-	-	High-performance Embedded Workshop アプリケーションの設定をカスタマイズします。	
	オプション	-	-	-	High-performance Embedded Workshop アプリケーションのオプションを設定します。	
	表示の形式	-	-	-	ウィンドウの表示色、フォント、キーワードなどを設定します。	
	基数	16進数	-		●	入力した数値を処理する基数を16進数に設定します。
		10進数	-		●	入力した数値を処理する基数を10進数に設定します。
8進数		-		●	入力した数値を処理する基数を8進数に設定します。	
2進数		-		●	入力した数値を処理する基数を2進数に設定します。	

## 注：

\*. 一部のメニューオプションに対する操作は、マクロ生成支援機能を使用し、High-performance Embedded Workshop コマンドラインのコマンドを記録できます。このリストでは、メニューオプションの「マクロの記録」欄にマクロの記録マーク(●)を表示し、マクロファイルへ記録可能かどうかを示しています。これらの詳細は「15.5.1 マクロを記録できる機能 (High-performance Embedded Workshop 共通)」を参照してください。

## 1.8 ツールメニューオプション

メニュー	メニューオプション	ショートカットキー	ツールボタン	機能	
ツール	アドミニストレーション	-	-	アドミニストレーションを設定します。	
	ツールチェインバージョンを変更	-	-	ツールチェインバージョンを変更します。	
	バージョン管理	選択	-	-	バージョン管理システムを選択します。
		構成	-	-	構成を設定します。
		ファイルの追加	-		バージョン管理に選択したファイルを追加します。
		ファイルの削除	-		バージョン管理から選択したファイルを削除します。
		ファイルの取得	-		バージョン管理から選択したファイルの読み取り専用ローカルファイルを取得します。
		ファイルのチェックアウト	-		バージョン管理から選択したファイルの書き込み可能なローカルファイルを取得します。
		ファイルのチェックイン	-		バージョン管理に選択したファイルのローカルコピーを戻します（更新します）。
		ファイルの状態	-		選択したファイルの状態を表示します。
	外部デバッガの起動	-		外部デバッガを起動します。	
	マクロの設定	-	-	マクロの設定ダイアログボックスを開きます。	
	マクロの記録	-		マクロの記録を開始します。	
	マクロの実行	-		マクロを実行します。	
	マクロの停止	-		マクロを停止します。	
各種システムツール	-		システムツール（[Renesas Call Walker]など）を起動します。		

## 注：

- \*1. [マクロの停止]とシステムツール用のメニューオプションの間には、記録したマクロをメニューに割り当てたときのメニューオプションを表示します。
- \*2. システムツール用のメニューオプションの下には、[ツール]メニューのカスタマイズにより追加した外部ツール用のメニューオプションを表示します。

## 1.9 テストメニューオプション

メニュー	メニューオプション	ショートカットキー	ツールボタン	機能
テスト	テストスイートの作成	-	-	新規に空のテストスイートを作成します。テストの追加、実行を開始できます。
	テストスイートを開く	-	-	既存のテストスイートを開きます。
	テストスイートの編集	-	-	現在のテストスイートを編集します。テストスイートへのテストの追加および削除ができます。
	テストスイートを閉じる	-	-	現在開いているテストスイートを閉じます。
	テストイメージファイルの作成	-	-	テストイメージファイル (*.HIF) に保存するテストイメージデータの設定およびカスタマイズが可能です。
	テストイメージファイルの比較	-	-	テストイメージファイルを、現在の High-performance Embedded Workshop システム、またはディスクにある別のテストイメージファイルと比較します。結果はテストブラウザに表示されます。
	テストを実行	-	-	テストスイートで定義した複数のテストを実行し、比較した結果をテストブラウザで見ることができます。テストの実行方法を指定する様々なオプションがあります。
	テスト結果ブラウザ	-	-	前回実行したテストの結果を表示します。テストの成功/失敗と、失敗した要因の詳細を見ることができます。



## 1.10 ウィンドウメニューオプション

メニュー	メニューオプション	ショートカットキー	ツールボタン	機能
ウィンドウ	重ねて表示	-	-	開いているすべてのウィンドウを重ねて並べます。
	上下に並べて表示	-	-	ウィンドウを上下に並べます。
	左右に並べて表示	-	-	ウィンドウを左右に並べます。
	アイコンの整列	-	-	アイコンを整列します。
	すべて閉じる	-	-	開いているすべてのウィンドウを閉じます。
	バーチャルデスクトップ	デスクトップマネージャ Default1-4	-	-

## 1.11 ヘルプメニューオプション

メニュー	メニューオプション	ショートカットキー	ツールボタン	機能	
ヘルプ	トピック	-	-	High-performance Embedded Workshop ヘルプウィンドウを開きます。	
	テクニカルサポート	不具合レポートの作成	-	-	High-performance Embedded Workshop 不具合レポートを作成します。
		アップデートの web 確認	-	-	High-performance Embedded Workshop 製品のバージョンアップやサービスパックを確認します。
	High-performance Embedded Workshop のバージョン情報	-	-	High-performance Embedded Workshop のバージョン情報を参照するための[High-performance Embedded Workshop について]ダイアログボックスを開きます。	
	デバグヘルプ	-	-	デバグ接続時エミュレータまたはシミュレータのヘルプウィンドウを開きます。	

## 2. ウィンドウ

ウィンドウ名	ウィンドウ表示メニュー
差分	[表示 -> 差分]
Map Section Information *1	[表示 -> マップ]
Map Symbol Information *1	
コマンドライン	[表示 -> コマンドライン]
Console *2	[表示 -> TCL ツールキット]
Workspace	[表示 -> ワークスペース]
Output	[表示 -> アウトプット]
逆アセンブリ	[表示 -> 逆アセンブリ]
レジスタ	[表示 -> CPU -> レジスタ]
メモリ	[表示 -> CPU -> メモリ]
IO	[表示 -> CPU -> IO]
ステータス	[表示 -> CPU -> ステータス]
画像 *1	[表示 -> グラフィック -> 画像]
波形 *1	[表示 -> グラフィック -> 波形]
スタックトレース	[表示 -> コード -> スタックトレース]
テストブラウザ	[テスト -> テスト結果ブラウザ]

### 注：

\*1. 機能のサポートはデバuggaに依存します。

\*2. 別冊の「TCL/TK 添付資料」を参照してください。

## 3. コマンド

### 3.1 コマンド一覧（アルファベット順）

コマンド名	短縮形	説明
!	-	コメント
ADD_FILE	AF	カレントプロジェクトへのファイル追加
ASSERT	-	コンディションのチェック
AUTO_COMPLETE	AC	オートコンプリートの切り替え
BUILD *1	BU	カレントプロジェクトのビルド
BUILD_ALL *1	BL	カレントプロジェクトのすべてのビルド
BUILD_FILE *1	BF	ファイルのコンパイル
BUILD_MULTIPLE *1	BM	複数プロジェクトのビルド
CACHE *2	-	メモリアクセスの有効化/無効化
CHANGE_CONFIGURATION	CC	コンフィギュレーションの設定
CHANGE_PROJECT	CP	プロジェクトの設定
CHANGE_SESSION	CS	セッションの設定
CLEAN	CL	ビルドの中間ファイルおよび出力ファイルの削除
CLEAR_OUTPUT_WINDOW	COW	アウトプットウィンドウの各タブの表示内容クリア
CLOSE_TEST_SUITE	CTS	テストスイートを閉じる
CLOSE_WORKSPACE	CW	ワークスペースを閉じる
COMPARE_TEST_DATA	CTD	テストデータの比較
CONNECT *2	CN	デバッグの接続
DEFAULT_OBJECT_FORMAT	DO	デフォルトオブジェクトフォーマットの設定
DISCONNECT *2	DN	デバッグの切断
ERASE	ER	コマンドラインウィンドウの内容クリア
EVALUATE	EV	式の計算
FILE_LOAD	FL	オブジェクト(プログラム)ファイルのロード
FILE_LOAD_ALL	LA	すべてのオブジェクト(プログラム)ファイルのロード
FILE_SAVE	FS	メモリ内容のファイルセーブ
FILE_UNLOAD	FU	オブジェクト(プログラム)ファイルのアンロード
FILE_UNLOAD_ALL	UA	すべてのオブジェクト(プログラム)ファイルのアンロード
FILE_VERIFY *2	FV	ファイル内容とメモリ内容の比較
FREE_GO *2	FG	ブレークポイントを無効にしたユーザプログラムの実行
GENERATE_MAKE_FILE *1	GM	カレントワークスペースの make ファイル作成
GO	GO	ユーザプログラムの実行
GO_RESET	GR	リセットベクタからのユーザプログラムの実行
GO_TILL	GT	テンポラリブレークポイントまでのユーザプログラムの実行
HALT	HA	ユーザプログラムの停止
HELP	HE	コマンドラインヘルプ
INITIALIZE	IN	デバッグの初期化
LOG	LO	ロギングファイルの操作
MEMORY_COMPARE *2	MC	メモリ内容の比較
MEMORY_DISPLAY	MD	メモリ内容の表示
MEMORY_EDIT	ME	メモリ内容の変更
MEMORY_FILL	MF	指定データによるメモリ内容の一括変更
MEMORY_FIND *2	MI	メモリ内容の検索
MEMORY_MOVE	MV	メモリブロックの移動
MEMORY_TEST *2	MT	メモリブロックのテスト
OPEN_TEST_SUITE	OTS	テストスイートを開く
OPEN_WORKSPACE	OW	ワークスペースを開く
QUIT	QU	High-performance Embedded Workshop の終了
RADIX	RA	入力した数値を処理する基数の設定
REFRESH_SESSION	RSE	セッション情報の再読み込み

コマンド名	短縮形	説明
REMOVE_FILE	REM	カレントプロジェクトからのファイル削除
RESET	RE	CPUのリセット
RUN_TEST	RT	テストの実行
SAVE_SESSION	SE	現在のセッションの保存
SAVE_WORKSPACE	SW	ワークスペースの保存
SET_DISASSEMBLY_SOFT_BREAK	SDB	逆アセンブリレベルのソフトウェアブレークポイントの設定または解除
SET_SOURCE_SOFT_BREAK	SSB	ソースレベルのソフトウェアブレークポイントの設定または解除
SLEEP	-	コマンド実行の遅延
STATE_DISASSEMBLY_SOFT_BREAK	TDB	逆アセンブリレベルのソフトウェアブレークポイントの有効化または無効化
STATE_SOURCE_SOFT_BREAK	TSB	ソースレベルのソフトウェアブレークポイントの有効化または無効化
STEP	ST	ステップ実行(命令単位またはソース行単位)
STEP_MODE	SM	ステップモードの設定
STEP_OUT	SP	現在の関数を終了するまでのステップ実行
STEP_OVER	SO	ステップオーバ実行
STEP_RATE	SR	ステップ実行速度の設定、表示
SUBMIT	SU	コマンドファイルの実行
TCL	-	TCLの有効化/無効化
TOOL_INFORMATION	TO	ツール情報の出力
UPDATE_ALL_DEPENDENCIES *1	UD	カレントプロジェクトのすべての依存関係更新

**注：**

- \*1. ツールチェーンが登録されていない場合は、使用できません。
- \*2. 機能のサポートはデバグガに依存します。

各コマンドのシンタックスについてはオンラインヘルプを参照してください。

## 3.2 コマンド一覧（機能別）

### High-performance Embedded Workshop アプリケーション制御コマンド

コマンド名	短縮形	説明
ADD_FILE	AF	カレントプロジェクトへのファイル追加
CHANGE_CONFIGURATION	CC	コンフィグレーションの設定
CHANGE_PROJECT	CP	プロジェクトの設定
CHANGE_SESSION	CS	セッションの設定
CLEAR_OUTPUT_WINDOW	COW	アウトプットウィンドウの各タブの表示内容クリア
CLOSE_WORKSPACE	CW	ワークスペースを閉じる
EVALUATE	EV	式の計算
OPEN_WORKSPACE	OW	ワークスペースを開く
QUIT	QU	High-performance Embedded Workshop の終了
RADIX	RA	入力した数値を処理する基数の設定
REFRESH_SESSION	RSE	セッション情報の再読み込み
REMOVE_FILE	REM	カレントプロジェクトからのファイル削除
SAVE_SESSION	SE	現在のセッションの保存
SAVE_WORKSPACE	SW	ワークスペースを保存する
TOOL_INFORMATION	TO	ツール情報の出力

## ビルド関連（ツールチェーンを登録している場合に使用可能）

コマンド名	短縮形	説明
BUILD	BU	カレントプロジェクトのビルド
BUILD_ALL	BL	カレントプロジェクトのすべてのビルド
BUILD_FILE	BF	ファイルのコンパイル
BUILD_MULTIPLE	BM	複数プロジェクトのビルド
CLEAN	CL	ビルドの中間ファイルおよび出力ファイルの削除
GENERATE_MAKE_FILE	GM	カレントワークスペースの make ファイル作成
UPDATE_ALL_DEPENDENCIES	UD	カレントプロジェクトのすべての依存関係更新

## コマンドライン操作関連

コマンド名	短縮形	説明
!	-	コメント
ASSERT	-	コンディションのチェック
AUTO_COMPLETE	AC	オートコンプリートの切り替え
ERASE	ER	コマンドラインウィンドウの内容クリア
HELP	HE	コマンドラインヘルプ
LOG	LO	ロギングファイルの操作
SLEEP	-	コマンド実行の遅延
SUBMIT	SU	コマンドファイルの実行
TCL	-	TCL の有効化/無効化

## テスト支援機能関連

コマンド名	短縮形	説明
CLOSE_TEST_SUITE	CTS	テストスイートを閉じる
COMPARE_TEST_DATA	CTD	テストデータの比較
OPEN_TEST_SUITE	OTS	テストスイートを開く
RUN_TEST	RT	テストの実行

## 実行関連（デバッガ接続時に使用可能）

コマンド名	短縮形	説明
FREE_GO *	FG	ブレークポイントを無効にしたユーザプログラムの実行
GO	GO	ユーザプログラムの実行
GO_RESET	GR	リセットベクタからのユーザプログラムの実行
GO_TILL	GT	テンポラリブレークポイントまでのユーザプログラムの実行
HALT	HA	ユーザプログラムの停止
RESET	RE	CPU のリセット
STEP	ST	ステップ実行(命令単位またはソース行単位)
STEP_MODE	SM	ステップモードの設定
STEP_OUT	SP	現在の関数を終了するまでのステップ実行
STEP_OVER	SO	ステップオーバ実行
STEP_RATE	SR	ステップ実行速度の設定、表示

## 注：

\*. 機能のサポートはデバッガに依存します。

**メモリ操作関連 (デバッガ接続時に使用可能)**

コマンド名	短縮形	説明
CACHE *	-	メモリアクセスの有効化/無効化
FILE_LOAD	FL	オブジェクト(プログラム)ファイルのロード
FILE_LOAD_ALL	LA	すべてのオブジェクト(プログラム)ファイルのロード
FILE_SAVE	FS	メモリ内容のファイルセーブ
FILE_UNLOAD	FU	オブジェクト(プログラム)ファイルのアンロード
FILE_UNLOAD_ALL	UA	すべてのオブジェクト(プログラム)ファイルのアンロード
FILE_VERIFY *	FV	ファイル内容とメモリ内容の比較
MEMORY_COMPARE *	MC	メモリ内容の比較
MEMORY_DISPLAY	MD	メモリ内容の表示
MEMORY_EDIT	ME	メモリ内容の変更
MEMORY_FILL	MF	指定データによるメモリ内容の一括変更
MEMORY_FIND *	MI	メモリ内容の検索
MEMORY_MOVE	MV	メモリブロックの移動
MEMORY_TEST *	MT	メモリブロックのテスト

**注：**

\*. 機能のサポートはデバッガに依存します。

**ソフトウェアブレーク設定関連 (デバッガ接続時に使用可能)**

コマンド名	短縮形	説明
SET_DISASSEMBLY_SOFT_BREAK	SDB	逆アセンブリレベルのソフトウェアブレークポイントの設定または解除
SET_SOURCE_SOFT_BREAK	SSB	ソースレベルのソフトウェアブレークポイントの設定または解除
STATE_DISASSEMBLY_SOFT_BREAK	TDB	逆アセンブリレベルのソフトウェアブレークポイントの有効化または無効化
STATE_SOURCE_SOFT_BREAK	TSB	ソースレベルのソフトウェアブレークポイントの有効化または無効化

**その他デバッグ関連 (デバッガ接続時に使用可能)**

コマンド名	短縮形	説明
CONNECT *	CN	デバッガの接続
DISCONNECT *	DN	デバッガの切断
INITIALIZE	IN	デバッガの初期化

**注：**

\*. 機能のサポートはデバッガに依存します。

各コマンドのシンタックスについてはオンラインヘルプを参照してください。

## 4. 正規表現

High-performance Embedded Workshop エディタでは検索、置換、およびファイルから検索操作の際、文字列に特殊文字を指定できます。

下表に指定できる文字とその詳細を示します。

文字	機能
?	改行文字以外の任意の一文字と一致します。 例えば、t?p は"top"や"tip" と一致するが"trap"とは一致しません。
*	改行文字以外の文字列（0 個以上の文字列）と一致します。2 行にまたがって一致することはありません。* 文字は残りのパターンが一致するための最短文字列と一致します。 例えば、*o は"too"の"to", "trowel"の"tro", "sporty orange"の"ty o"と一致しますが、 "smart orange" とは一致しません（*文字は 2 行にまたがって一致することはないため）。
\t	タブと一致します。 例 1 \t8 タブの次に 8 がある場合を検索します。 例 2 init\t init の次にタブがある場合を検索します。
[ ]	括弧内の文字または文字の範囲の中の任意の一文字と一致します。括弧をネストすることはできません。 [-]は文字の範囲を示します。例[a-z]、[0-9]。範囲の始まりの文字は終わりの文字より小さい ASCII コードでなければなりません。 [~]は[~ と]の間の文字以外の一文字と一致します。改行文字がこの括弧内になければ、改行文字とも一致します。 例 1 [AEIOU] 大文字の母音をすべて検索します。 例 2 [<>?] 記号<, >, ?を検索します。 例 3 [A-Za-z0-9_] 大文字、小文字、数字、下線を検索します。 例 4 [~0-9] 数字以外の文字を検索します。 例 5 [ \t\n] 空白、タブ、改行文字を検索します。 例 6 [\\] \のすぐ後に]を置くと、記号] を検索します。
\	正規表現を無視する特殊文字です。バックスラッシュの次の文字が正規表現のとき、通常の文字として扱います。バックスラッシュの次が非正規表現のとき、バックスラッシュは無視されます。 例 1 \ アスタリスクをすべて検索します。 例 2 \\ バックスラッシュをすべて検索します。

注：

バックスラッシュ文字\  
は日本語 Windows® では円記号として表示されます。



## 5. プレースホルダ

プレースホルダは、High-performance Embedded Workshop の複数のツールによって提供される機能です。ここではプレースホルダの使い方を説明します。

### 5.1 プレースホルダとは？

プレースホルダとは一時的にテキストに挿入される特殊文字列です。後で実際の値に置き換えます。例えば、High-performance Embedded Workshop のプレースホルダの1つに、\$(FULLFILE)があります。これは、すべてのパス付きのファイルを示します。

パラメータとしてファイルを編集できるエディタが c:¥myedit¥myeditor.exe であるとします。c:¥files ディレクトリのファイル FILE1.C を開き、このエディタを起動するには、以下のように直接指定することもできます。

```
c:¥myedit¥myeditor.exe c:¥files¥FILE1.C
```

しかし、このエディタを介して任意のファイルを開きたいとき、上記コマンドは c:¥files¥file1.c を開くだけのものであるため、問題が起きます。指定したエディタを使うときにその時点で選択したファイルを開くことができるようにするには、特定のファイル名を一般的なプレースホルダに置き換えます。以下に例を示します。

```
c:¥myedit¥myeditor.exe $(FULLFILE)
```

これで、High-performance Embedded Workshop がエディタでファイルを開くとき、\$(FULLFILE)を選択されたファイル名で置き換えます。

### 5.2 プレースホルダを挿入する

プレースホルダを挿入するには以下に示す方法があります。

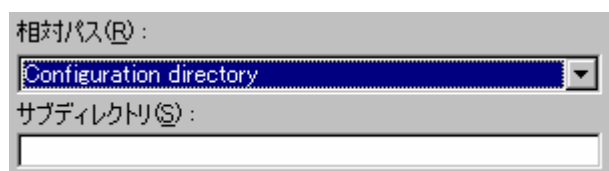
#### 例 1

1. プレースホルダを挿入したい場所にカーソルを置いてください。
2. 次に、必要なプレースホルダを編集フィールドの右のプレースホルダボタンをクリックしてください。
3. 必要なプレースホルダをポップアップメニューから選択してください。



#### 例 2

1. "Custom directory"以外のプレースホルダをドロップダウンリストボックスから選択し、プレースホルダによって示されたディレクトリからの相対サブディレクトリを指定してください。
2. "Custom directory"を選択した場合、[サブディレクトリ]に絶対ディレクトリパスを指定してください。



## 例 3

1. プレースホルダを挿入したい場所にカーソルを置いてください。
2. 必要なプレースホルダをドロップダウンリストボックスから選択してください。
3. [挿入]ボタンをクリックしてください。



## 例 4

1. フィールドにプレースホルダを直接入力してください。大文字で入力して、"\$(" で開始し)" で終了してください。

正しい

\$(FILEDIR)

誤り

\$(Filedir)

\$( FILEDIR )

\$FILEDIR

## 5.3 使用できるプレースホルダ

以下の表に、使用できるプレースホルダと意味、またその使用例を示します。

プレースホルダ	意味	プレースホルダの展開例
\$(FULLFILE)	ファイル名 (絶対パスを含む)	c:\workspace\project\file.src
\$(FILEDIR)	ファイルディレクトリ	c:\workspace\project
\$(FILENAME)	ファイル名 (パスを除き拡張子を含む)	file.src
\$(FILELEAF)	ファイル名 (パスと拡張子を除く)	file
\$(EXTENSION)	ファイルの拡張子	src
\$(WORKSPDIR)	ワークスペースディレクトリ	c:\workspace
\$(WORKSPNAME)	ワークスペース名	workspace
\$(PROJDIR)	プロジェクトディレクトリ	c:\workspace\project
\$(PROJECTNAME)	プロジェクト名	project
\$(CONFIGDIR)	コンフィグレーションディレクトリ	c:\workspace\project\debug
\$(CONFIGNAME)	コンフィグレーション名	debug
\$(HEWDIR)	High-performance Embedded Workshop インストールディレクトリ	c:\Program Files\Renesas\Hew
\$(TCINSTALL)	ツールチェインインストールディレクトリ (オプション設定ダイアログボックス、および[新規ビルドフェーズ]ダイアログボックス (ステップ 3) 上)	c:\Program Files\Renesas\Hew\Tools\renesas\sh\9_0_1_1
\$(TOOLDIR)	ツールインストールディレクトリ (Tools Administration 上)	c:\Program Files\Renesas\Hew\Tools\renesas\sh\9_0_1_1
\$(TEMPDIR)	テンポラリディレクトリ	c:\temp
\$(WINDIR)	Windows® ディレクトリ	c:\windows
\$(WINSYSDIR)	Windows® システムディレクトリ	c:\windows\system
\$(EXEDIR)	コマンドディレクトリ	v:\vc\win32
\$(USERNAME)	ユーザログイン (バージョン管理)	JHARK
\$(PASSWORD)	ユーザパスワード (バージョン管理)	214436
\$(VCDIR)	「仮想的」バージョン管理ディレクトリ	"c:\project"は"x:\vc\project"へマッピングしている。
\$(COMMENT)	コメント (バージョン管理)	[コメント入力]ダイアログボックスを開く。
\$(LINE)	エラー/ウォーニングの行番号	12

上記の表では以下のことを仮定しています。

- ファイルパスは "c:¥workspace¥project¥file.src"。
- ワークスペース名 "workspace" の位置は "c:¥workspace"。
- プロジェクト名 "project" の位置は "c:¥workspace¥project"。
- コンフィグレーション名 "debug" にはコンフィグレーションディレクトリがあり、位置は "c:¥workspace¥project¥debug"。
- HEW2.EXE を "c:¥Program Files¥Renesas¥Hew" にインストールしている。
- ツールチェイン (コンパイラ、アセンブラ、リンカージェディタ) の \*.HRF ファイル の位置は "c:¥Program Files¥Renesas¥Hew¥Tools¥renesas¥sh¥9\_0\_1\_1"  
このディレクトリは[ビルド]メニューのオプション設定ダイアログボックス、および[新規ビルドフェーズ]ダイアログボックス (ステップ 3) 上では\$(TCINSTALL)として参照され、[ツールアドミニストレーション]ダイアログボックス上では\$(TOOLDIR)として参照される。
- Windows® operating system を c:¥Windows にインストールしており、Windows® システムファイルを c:¥Windows¥System にインストールしている。
- バージョン管理実行可能パスが v:¥vc¥win32¥ss.exe である。バージョン管理システムにログインするユーザ名は JHARK でパスワードが 214436 であり、バージョン管理実行可能ファイルへのコマンドラインには \$(COMMENT)を指定している。c:¥project は[ツール -> バージョン管理 -> 構成]を選択すると開く[バージョン管理設定]ダイアログボックスの[Projects]タブの x:¥vc¥project にマッピングしている。
- コンパイラまたはアセンブラのエラーが 12 行目で発生した。

#### 注：

どのフィールドでもすべてのプレースホルダを使用できるとは限りません。例えば、プレースホルダ\$(LINE)は依存ファイル位置を指定するときには意味をもちません。プレースホルダ\$(USERNAME)、\$(PASSWORD)、\$(VCDIR)、\$(COMMENT)はバージョン管理でのみ受け付けます。各編集フィールドで、使用できないプレースホルダを指定すると、警告メッセージを表示する場合があります。

## 5.4 プレースホルダを使うにあたって

プレースホルダによって、システムが使用する様々なファイルへのパスをフレキシブルに指定できます。

- パスまたはファイル名を入力する編集フィールドの右にプレースホルダボタン(▶)がある場合、プレースホルダをどのように使用してパスやファイルの指定をフレキシブルにできるかご考慮ください。
- いくつかの構成を使うとき、プレースホルダ\$(CONFIGDIR)を使用すると、現在の構成のディレクトリからファイルへの書き込みやそのファイルから現在の構成のディレクトリへの書き込みができます。
- できるだけプレースホルダを使用してください。プレースホルダは後で削除したり追加したりできます。

## 6. I/Oファイルフォーマット

High-performance Embedded Workshop は、I/O レジスタ定義ファイルで取得する情報に基づいて、[IO]ウィンドウを作成します。デバッグを選択すると、High-performance Embedded Workshop は、選択したデバイスに対応する "<device>.IO" ファイルを検索し、存在する場合にはこのファイルをロードします。これは、I/O モジュール、およびそのレジスタのアドレスやサイズを記述するフォーマット済みテキストファイルです。テキストエディタでこのファイルを編集し、ユーザアプリケーションに特有のメモリマップレジスタや周辺レジスタ（例えば、マイコンのアドレス空間にマップした ASIC デバイスのレジスタ）のサポートを追加できます。

### ファイルフォーマット

各モジュール名を [Modules] 定義セクションで定義し、モジュールの番号を、順番に付けていなければなりません。各モジュールはレジスタ定義セクションに対応しており、セクション内のエントリは、I/O レジスタを定義します。

[BaseAddress] はデバイスのための定義であり、そのデバイスでは、CPU モードによってアドレス空間の I/O レジスタの場所が移動します。この場合、[BaseAddress] 値は、ある特有モードの I/O レジスタのベースアドレスです。また、レジスタ定義で使用するアドレスは、同じモードにおけるレジスタのアドレス位置です。I/O レジスタファイルを実際に使用する場合、定義したレジスタアドレスから [BaseAddress] 値を引き、その結果のオフセットを選択したモードのベースアドレスに加算します。

各モジュールにはセクションがあり、オプションの依存性によって形成するレジスタを定義します。依存性は、モジュールが有効かどうかを確認するためにチェックします。各レジスタ名をセクションで定義し、レジスタの番号を、順番に付けていなければなりません。依存性は、dep=<reg> <bit> <value> のようにセクションに入力します。

1. <reg> は依存性のレジスタ ID です。
2. <bit> はレジスタのビット位置です。
3. <value> は値で、ビットは、有効であるモジュールに使用しなければなりません。

[Register] 定義エントリは、id=<name> <address> [<size> [<absolute> [<format> [<bitfields>]]]] のフォーマットで入力します。

1. <name> は表示するレジスタ名です。
2. <address> はレジスタのアドレスです。
3. <size> は、B がバイトサイズ、W がワードサイズ、L がロングワードサイズを意味します（デフォルトはバイトです）。
4. <absolute> は、レジスタが絶対アドレスにある場合、A と設定します。これは、異なるモードの CPU によって I/O 空間アドレス範囲が移動する場合のみ関連します。レジスタが絶対アドレスにあると定義すると、ベースアドレスオフセットは計算せず、指定したアドレスを直接使用します。
5. <format> はレジスタを出力するためのフォーマットです。有効な値は、16 進数の場合は H、10 進数は D、2 進数は B です。
6. <bitfields> セクションは、レジスタのビットを定義します。

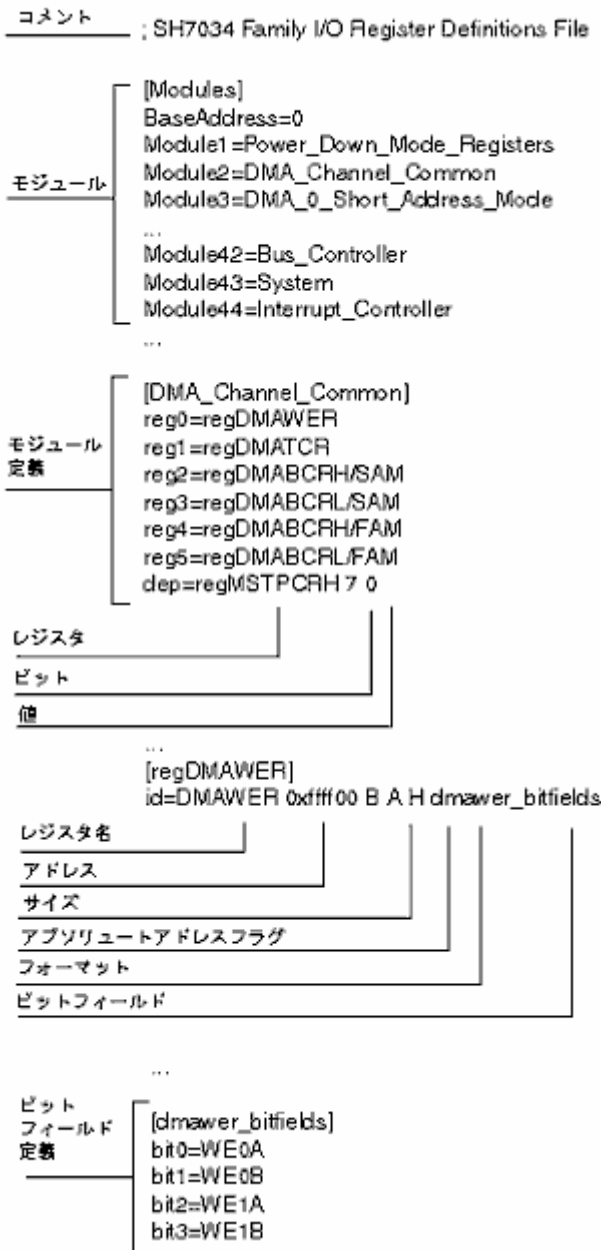
ビットフィールドセクションは、各エントリが bit<no>=<name> タイプのレジスタ内のビットを定義します。

1. <no> はビット番号です。
2. <name> はビットのシンボル名です。

コメント行を入れる場合、";" で始めなければなりません。

次に例を示します。

例:



## 7. シンボルファイルフォーマット

シンボルファイルを正確にデコードするためには、Pentica-B ファイルのフォーマットに基づいてファイルを作成する必要があります。

1. ファイルは、簡単な ASCII テキストファイルでなければなりません。
2. ファイルは、"BEGIN"で始めなければなりません。
3. 各シンボルは、個々の行で、まず、"H"で終了する 16 進数の値から始まり、次にスペース、シンボルテキストの順でなければなりません。
4. ファイルは、"END"で終了しなければなりません。

**例：**

```
BEGIN
11FAH Symbol_name_1
11FCH Symbol_name_2
11FEH Symbol_name_3
1200H Symbol_name_4
END
```

**注：**

機能のサポートはデバッガに依存します。

## 8. キーボードショートカット

High-performance Embedded Workshop のアプリケーションでは、主要なコマンドはすべてキーボードから操作できます。アプリケーションでのキーボードのコマンドを以下に示します。

機能キー	キー	機能
Alt	0-9	キーボードショートカットに割り当てたマクロを実行する。
Alt	A	ステータスバーの表示/非表示を切り替える。
Alt	K	ワークスペースウィンドウを開く。
Alt	O	アウトプットウィンドウを開く。
Alt	F4	アプリケーションを終了する。
Alt	Backspace	元に戻る。(または、エディタで Ctrl+Z)
Ctrl	0-9	テンプレートで予約されています。
Ctrl	A	エディタですべてを選択する。(他のウィンドウでも有効な場合あり)
Ctrl	B	[ブレークポイント]ダイアログボックスを開く。
Ctrl	C	コピーする。
Ctrl	D	逆アセンブリウィンドウを開く。
Ctrl	F	[検索]ダイアログボックスを開く。
Ctrl	G	ジャンプする。
Ctrl	H	[置換]ダイアログボックスを開く。
Ctrl	I	[IO]ウィンドウを開く。
Ctrl	K	スタックトレースウィンドウを開く。
Ctrl	L	コマンドラインウィンドウを開く。
Ctrl	M	メモリウィンドウを開く。
Ctrl	N	エディタで新しいソースファイルを開く。
Ctrl	O	[ファイルを開く]ダイアログボックスを開く。
Ctrl	P	印刷する。
Ctrl	R	レジスタウィンドウを開く。
Ctrl	S	現在のファイルを保存する。
Ctrl	U	ステータスウィンドウを開く。
Ctrl	V	クリップボードの内容を貼り付ける。
Ctrl	X	選択範囲を切り取る。
Ctrl	Y	取り消した操作をやり直す。
Ctrl	Z	直前の操作を元に戻す。
Ctrl	F2	ブックマークを設定、解除する。
Ctrl	F3	エディタで選択されている文字列の次を検索、または[検索]ダイアログボックスを開く。
Ctrl	F4	エディタウィンドウを閉じる。
Ctrl	F6	ウィンドウを切り替える。
Ctrl	F7	ファイルをコンパイルする。
Ctrl	F9	ブレークポイントの設定、解除する。
Ctrl	Spacebar	エディタにポップアップウィンドウとしてナビゲーションリストを表示する。
Ctrl	Backspace	単語を削除する。
Ctrl	Break	ビルドを中止する。
Ctrl	Insert	コピーする。
Ctrl	Tab	ウィンドウを切り替える。
Ctrl	Home	エディタのカーソルを現在のファイルの先頭に戻す。
Ctrl	End	エディタのカーソルを現在のファイルの最後に移動する。
Ctrl	↑	ウィンドウを上へスクロールする。(カーソル位置はエディタの同じ場所)
Ctrl	↓	ウィンドウを下へスクロールする。(カーソル位置はエディタの同じ場所)
Ctrl	←	エディタで、カーソルを1つ前の単語に移動する。
Ctrl	→	エディタで、カーソルを次の単語に移動する。
Ctrl+Alt	1-4	バーチャルデスクトップを切り替える。
Ctrl+Alt	PageUp	ウィンドウの次のタブに移動する。(例: アウトプットウィンドウ、ワークスペースウィンドウ)



機能キー	キー	機能
Ctrl+Alt	PageDown	ウィンドウの1つ前のタブに移動する。(例: アウトプットウィンドウ、ワークスペースウィンドウ)
Ctrl+Shift	8	エディタで、半角スペース、タブ記号を表示する。
Ctrl+Shift	G	画像ウィンドウを開く。
Ctrl+Shift	K	TCL ツールキットを開く。
Ctrl+Shift	L	エディタで1行削除する。
Ctrl+Shift	M	括弧の組み合わせを見つける。
Ctrl+Shift	N	アクティブセッションを Debuggers タブ中の次のセッションに変更する。(同期デバッグ機能)
Ctrl+Shift	S	ワークスペースの変更されたファイルをすべて保存する。
Ctrl+Shift	T	テンプレートを挿入する。
Ctrl+Shift	U	エディタで選択された文字列をすべて大文字にする。
Ctrl+Shift	V	波形ウィンドウを開く。
Ctrl+Shift	Y	PC 位置を表示する。
Ctrl+Shift	F3	エディタで選択されている文字列の前を検索、または[検索]ダイアログボックスを開く。
Ctrl+Shift	Spacebar	エディタで開き括弧を入力すると、ポップアップウィンドウとして関数名とパラメータを表示する。
Ctrl+Shift	Tab	ウィンドウを切り替える。
Ctrl+Shift	Home	カーソル位置からファイルの先頭までを選択する。
Ctrl+Shift	End	カーソル位置からファイルの最後までを選択する。
Ctrl+Shift	←	エディタで、1つ前の単語を選択する。
Ctrl+Shift	→	エディタで、次の単語を選択する。
Shift	F1	状況依存ヘルプを表示する。
Shift	F2	前のブックマークへジャンプする。
Shift	F3	前を検索する。
Shift	F5	リセット後実行する。
Shift	F8	前のビルドエラーまたはウォーニングが発生したエディタを表示する。
Shift	F10	コンテキストメニューを表示する。(ポップアップメニューでのマウスの右クリックと同様)
Shift	F11	関数からステップアウトする。
Shift	F12	すべてのウィンドウをリフレッシュする。
Shift	Delete	エディタで選択範囲を切り取る。(またはエディタで Ctrl+X)
Shift	ESC	アウトプットウィンドウを非表示にする。
Shift	Tab	エディタで、タブの位置を戻す。
Shift	Home	エディタで、カーソル位置から現在の行の先頭までを選択する。
Shift	End	エディタで、カーソル位置から現在の行の最後までを選択する。
Shift	↑	現在の行より上を選択する。
Shift	↓	現在の行より下を選択する。
Shift+Alt	Backspace	取り消した操作をやり直す。(または、エディタで Ctrl+Y)
なし	F1	状況依存ヘルプを表示する。
なし	F2	次のブックマークへジャンプする。
なし	F3	次の文字列を検索する。
なし	F4	複数のファイルで文字列を検索する。
なし	F5	プログラム実行する。
なし	F7	プロジェクトをビルドする。
なし	F8	次のビルドエラーまたはウォーニングが発生したエディタを表示する。
なし	F9	ブレークポイントを挿入/削除する。
なし	F10	ステップオーバー実行する。
なし	F11	ステップイン実行する。
なし	F12	ウィンドウの表示内容を更新する。
なし	Delete	クリアする。
なし	Insert	挿入モードと上書きモードを切り替える。
なし	Enter	エディタでキャリッジリターンを挿入する。
なし	Tab	エディタでタブを挿入する。
なし	ESC	プログラムの実行を停止する。
なし	Home	エディタで、カーソルを現在の行の先頭に戻す。

機能キー	キー	機能
なし	End	エディタで、カーソルを現在の行の最後に移動する。
なし	PageUp	エディタのページを上移動する。
なし	PageDown	エディタのページを下移動する。
なし	↑	エディタで、カーソルを上移動する。
なし	↓	エディタで、カーソルを下移動する。
なし	←	エディタで、カーソルを左移動する。
なし	→	エディタで、カーソルを右移動する。
なし	Alt + マウスで選択	エディタで列を選択する。

**注：**

機能のサポートはデバッガに依存します。

## 9. デバッガでドラッグアンドドロップする

High-performance Embedded Workshop のデバッガを使用中、各デバッグコンポーネントを他のコンポーネントと相互作用させることができます。他の表示への対象のドラッグで実現します。

以下に例を示します。

1. ラベル表示から他のデバッグ表示へラベルをドラッグできます。例えば、逆アセンブリウィンドウにラベルをドラッグした場合、そのラベルのあるアドレスへスクロールします。
2. 変数をエディタからウォッチウィンドウへドラッグできます。変数がウォッチウィンドウに追加されます。
3. 関数名をエディタから逆アセンブリへドラッグすると、逆アセンブリ表示はラベルの位置へジャンプします。

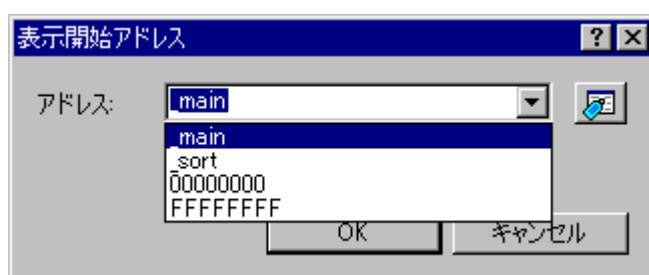
## 10. ラベルを使用してコードを参照する

デバッグモジュール内を検索するときにラベルを使用すると便利です。アドレスを入力できるエディットフィールドであれば、ラベルを使用できます。このようなフィールドにラベルを入力すると、内蔵の式処理がラベルをチェックし、アドレスに変換します。"\_main+100"や"\_MyFunction+100"などの評価を入力できます。

ラベルを使用すると評価する対象のアドレスを固定しません。この方法は特に、コマンドラインバッチファイルを使ってブレークポイントを設定するときに便利です。コマンドラインバッチファイルで常に関数にブレークポイントを設定する必要がある場合は、ラベルの使用により解決できます。

ラベルを使用すれば、コマンドラインバッチファイルの内容に影響を与えずにコードを変更できます。

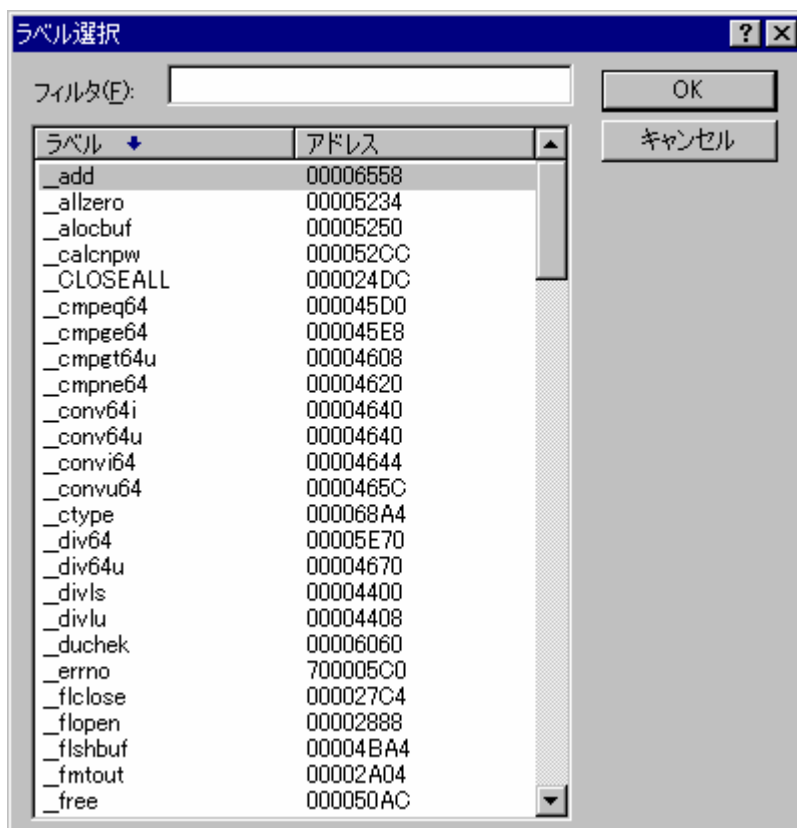
High-performance Embedded Workshop V.3.1 以降では、他にもラベルを使用する簡単な方法をサポートしています。



上記のダイアログボックスに、ラベルを選択するリストも表示します。High-performance Embedded Workshop のアプリケーション全体を通して、アドレスエディットフィールドに入力された過去 20 件を、このリストに保存します。同じラベルを何度も入力する場合、この最近使用した[アドレス]のドロップダウンリストを使用すれば、より効率的に入力できます。入力を評価するアドレスエディットフィールドの場合にこの操作が可能です。

アドレスを入力できるエディットフィールドの右にラベルブラウズボタン()が付いています。

これをクリックすると、[ラベル選択]ダイアログボックスが開き、現在のラベルのリストからラベルを選択できます。



リストは、ラベル名の昇順でソートされ右側にはそのアドレス値を表示しています。各々のカラムヘッダをクリックすると、クリックしたカラムでソートします。

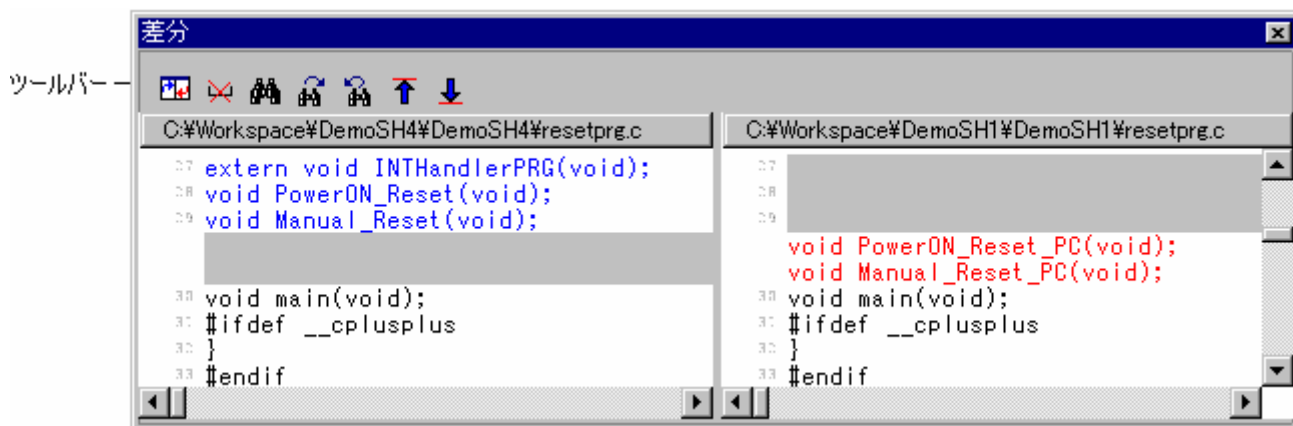
リストにある特定のラベルを検索するには、[フィルタ]エディットボックスを使用します。この内容を変更すると、入力した文字列（大文字と小文字を区別しません）を含むラベルだけをリストに表示します。例えば、上記のダイアログボックスで[フィルタ]に「conv6」を入力すると、リストには「\_\_conv64i」と「\_\_conv64u」のみを表示します。プロジェクトに多数のラベルがある場合にこの機能は有用です。

## 11. ウィンドウのツールバー

High-performance Embedded Workshop V.4.0 以降のコンポーネントのウィンドウでは、クライアント領域にツールバーを表示しています。ツールバーボタンをクリックすると、ウィンドウの主な機能へアクセスできます。

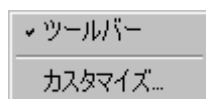
この機能は、High-performance Embedded Workshop の様々なウィンドウにあります。

例として、差分ウィンドウを以下に示します。

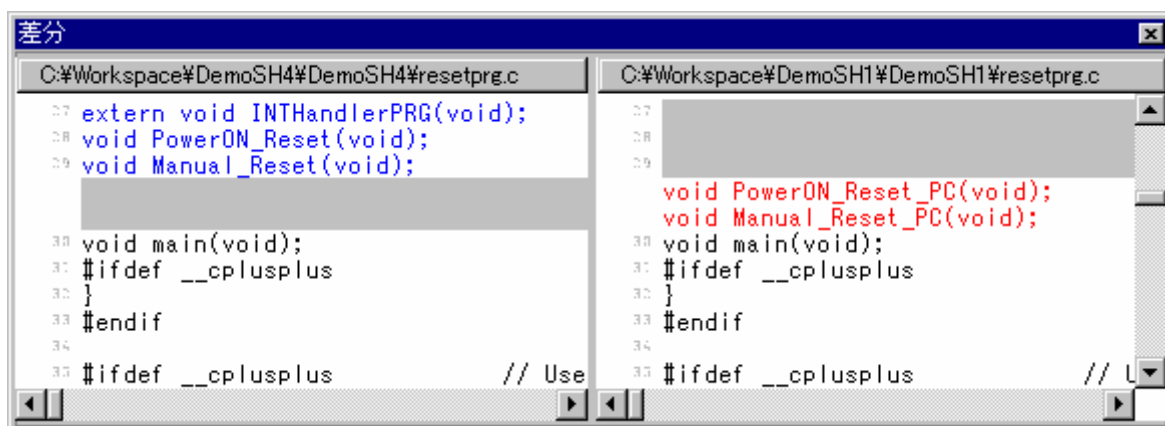


ツールバーをカスタマイズすることも可能です。

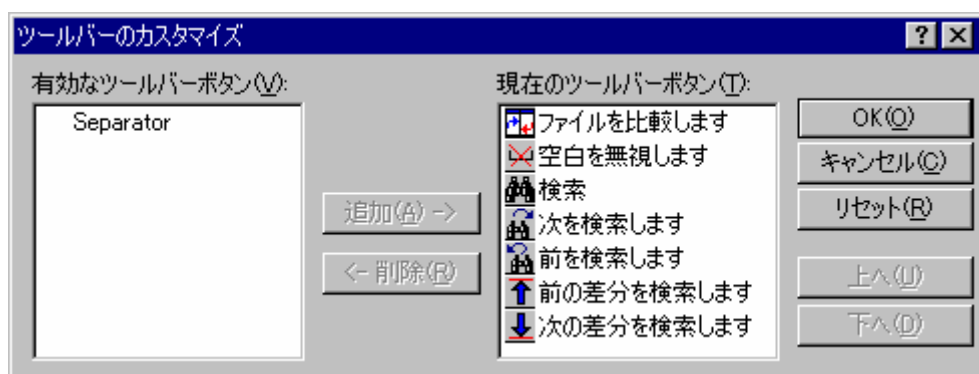
ツールバー上で右クリックすると、以下のメニューを表示します。



[ツールバー]メニューは、ツールバーの表示/表示なしの切り替えができます。（下の図は表示なしの場合）



[カスタマイズ]メニューは、[ツールバーのカスタマイズ]ダイアログボックスを開きます。これを以下に示します。



このダイアログボックスでは、表示するボタンを選択したり、ボタンの表示順序を変更できます。[有効なツールバーボタン]は、ツールバーに現在表示されていないボタンをすべて表示します。[現在のツールバーボタン]は、ツールバーに現在表示されているボタンをすべて表示します。

#### ツールバーにボタンを追加するには

1. [有効なツールバーボタン]リストから、追加したいツールバーボタンを選択してください。
2. [追加]ボタンをクリックしてください。
3. [OK]ボタンをクリックしてください。

#### ツールバーのボタンを移動するには

1. [現在のツールバーボタン]リストから、移動したいツールバーボタンを選択してください。
2. 移動させたい位置まで、[上へ]あるいは[下へ]ボタンをクリックしてください。
3. [OK]ボタンをクリックしてください。

#### ツールバーからボタンを削除するには

1. [現在のツールバーボタン]リストから、削除したいツールバーボタンを選択してください。
2. [削除]ボタンをクリックしてください。
3. [OK]ボタンをクリックしてください。

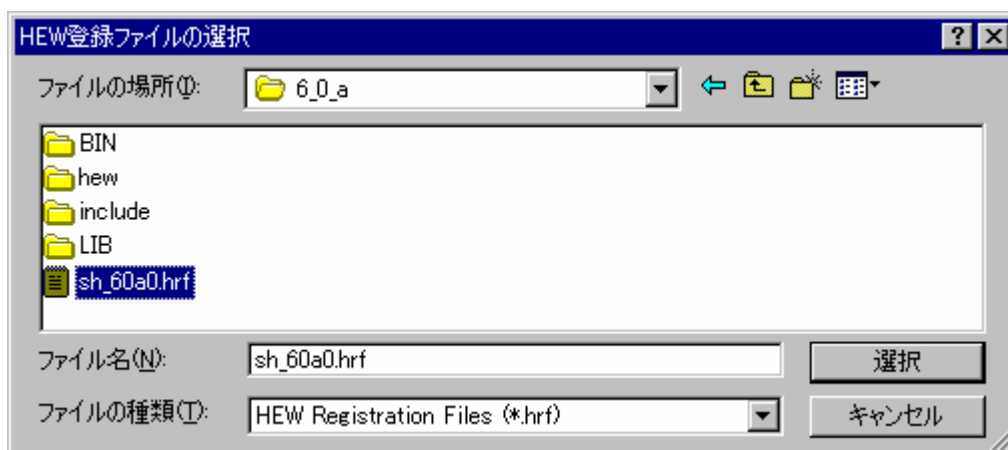


## 12. High-performance Embedded Workshop V.1.x用のツールチェーンでビルドする

High-performance Embedded Workshop V.1.x で作成したプロジェクトを、High-performance Embedded Workshop V.2.x 以降に登録されている新しいツールチェーンにアップグレードしないで使用する場合、該当する旧バージョンのツールチェーンに登録する必要があります。[ツール->アドミニストレーション]の[登録]ボタンで旧バージョンのツールチェーンの"HRF"ファイルを選択してください(下の図参照)。High-performance Embedded Workshop V.2.x 以降で、旧バージョンのツールチェーンを使ったビルド実行が可能になります。

ただし、一度 High-performance Embedded Workshop V.2.x 以降で開いたワークスペースは、High-performance Embedded Workshop V.1.x では開けませんのでご注意ください。

また、High-performance Embedded Workshop V.2.x 以降では、新規に旧バージョンのプロジェクトは作成できません。High-performance Embedded Workshop V.1.x にある旧バージョンのプロジェクトを作成する場合には、High-performance Embedded Workshop V.1.x をご使用ください。



## 13. HMAKEユーザガイドライン

### 13.1 コマンドライン

ファイルで hmake プログラムを実行するときに使うコマンドラインを説明します。

#### 基本構成

コマンドラインのシンタックスを以下に示します。

```
hmake <メイクファイル> <パラメータリスト>
```

拡張子なしでファイルを指定した場合、.mak を付加します。パラメータリストには、次の節に示す複数のパラメータを指定できます。また、パラメータは省略できます。パラメータリストはメイクファイル名の前に指定してもかまいません。各パラメータは、1つ以上の空白文字で区切ってください。パラメータは、大文字と小文字を区別しません。パラメータとファイル名を指定しなかった場合、ヘルプ情報を表示します。

#### Exit コード

実行中のメイクファイルでシンタックスエラー発生時、またはメイクファイル実行中の処理で不当エラーコードが返されたとき、hmake はコード 1 で終了します。その他の場合、hmake はコード 0 で終了します（ファイルのシンタックスや exit コード条件の指定方法は、以下を参照してください）。

#### パラメータ

指定できるパラメータとその機能を示します。

パラメータ	機能
/A	入出力ファイル状態にかかわらず全コマンドを実行する。全ビルドと同じ
/N	入出力ファイルの使用状態により、（ノーマルで）実行すべきコマンドを判断して、表示するが、実行はしない
/?	ヘルプ情報の表示

### 13.2 ファイルのシンタックス

hmake ファイルでは、基本的に 4 種類の文（変数宣言、記述部、コメント、メッセージコメント）を使用します。hmake ファイルの作成にあたって、これらの文はどのような順序でも使用できます。ただし、変数を記述部や他の変数宣言に使用する前に、変数宣言をしてください。nmake ファイルで使用する最初の "all" 文は、hmake ファイルでは必要ありません。メイクファイルでは、コマンドを出現する順序で実行します。"→" 文字は、メイクファイルのシンタックスを正しくするためにタブ文字を使用すべきところを示すことに注意してください。

#### 変数宣言

変数宣言をした後、その変数は、hmake ファイルのすべての部分で使うことができます。変数宣言のシンタックスを以下に示します。

```
<変数名> = <値>
```

変数名と '=' 符号の間や、値と '=' 符号の間に、複数の空白文字を入れることができます。値は、\" 文字で区切って、数行にわたって書くことができます。メインテキスト内で値に \" 文字を含む場合は文字どおり解釈されます。\" 文字の後に新しい行が続くときのみ、値が次の行にわたっているとみなされます。

変数宣言の例をいくつか示します。

```
EXECUTABLE = c:\dir\prog.exe
OUTPUT = c:\dir2\file1.out
INPUT = c:\dir2\file1.c
DEPEND = c:\dir2\file2.h \
        c:\dir2\file3.h \
        c:\dir2\file4.h
```

hmake ファイルの後半で変数を使うには、変数名の前に"\$("を、後ろに")"を加えてください。"\$()"文字付きの変数名を変数の値で置き換えます。次の記述部の節にその例を示します。変数名には、半角英数文字と半角下線を使用できます。別の変数宣言の中で変数を使うことができますが、すべての変数は使う前に宣言します。

### 13.3 記述部

#### 概要

記述部には、ターゲット、依存ファイル、そしてターゲットが依存ファイルより新しい場合に実行するコマンド一覧を指定します。ターゲット、依存ファイルのどちらか、またはいずれもない場合、コマンドを常に実行します。コマンドを常に実行したいときには、依存ファイルを指定する必要はありません。記述部のシンタックスを以下に示します。

```
<ターゲット 1> <ターゲット 2> ... : <依存ファイル 1> <依存ファイル 2> ...
→<コマンド 1>
→<コマンド 2>
→...
→<コマンド n>
```

最後のターゲットと"! "文字の間や、最初の依存ファイルと"! "文字の間に、複数の空白文字を入れることができます。最初のターゲットの前に空白文字を入れしないでください。各ターゲットと各依存ファイルは、少なくとも1つの空白文字で区切ってください。タブ文字はコマンドを含む行のはじめに必要です。変数宣言の後で、変数を上記シンタックスで記述部に使うことができます。

記述部の例（変数宣言の後に変数を使う例）を以下に示します。

```
c:\dir1\file1.obj : c:\dir1\file1.c c:\dir1\file1.h
gcc c:\dir1\file1.c
$(OUTPUT) : $(INPUT) $(DEPEND)
$(EXECUTABLE) $(INPUT)
```

## 特殊なコマンド

記述ブロックで使用できる特殊なコマンドが2つあります。cd コマンドはカレントディレクトリを変更し、set コマンドはメイクファイル実行の間に使用する環境変数を設定します。両方とも、相当する DOS コマンドと同じ方法で使用します。

これらのコマンドを使用した有効な記述ブロックの例を、以下に示します。

```
CHANGEDIR :
    cd c:\dir1\dir2
SETENV:
    set VAR1=value1
    set VAR2=value2
    set VAR3=value3
```

\$(EXECUTABLE)を置き換える CHANGEDIR と SETENV がファイル名でなくてもかまいません。これらは存在しないファイルとして扱われてコマンドを常に実行します。

## サブコマンドファイル

hmake でサブコマンドファイルを生成するには、記述部のコマンド部分を以下のように指定してください(上記の<コマンド n>を置き換えます)。

```
→<コマンド開始> <<
→<サブコマンド 1>,
→<サブコマンド 2>,
→...
→<サブコマンド n>
→<<<コマンド終了>
```

これで、Windows®の一時的ファイルに、<サブコマンド 1>、<サブコマンド 2>などの行を含むサブコマンドファイルを生成します。このコマンドファイルはメイク処理が終了すると削除します。コマンドファイル名は2つの"<<"の間のすべてのテキストで置き換えられます。サブコマンドファイル名は、自動的に hmake により付けられます。

例：

```
c:\dir1\file1.obj : c:\dir1\file1.c c:\dir1\file1.h
gcc @"<<
-c -o c:\dir1\file1.obj c:\dir1\file1.c
<<"
```

生成されたサブコマンドファイル名が"c:\temp\hmk111.cmd"の場合、hmake は以下を実行します。

(c:\dir1\file1.obj が更新されるとき)：

```
gcc @"c:\temp\hmk111.cmd"
```

サブコマンドファイル(c:\temp\hmk111.cmd)には以下を含みます。

```
-c -o c:\dir1\file1.obj c:\dir1\file1.c
```

記述部に1つ以上のコマンドを含むことができ、標準とサブコマンドファイルコマンドを混在して使用できます。

## 13.4 コメント

#文字はコメントを示します。この文字が行の最初にあると、（次の改行文字があるまで）その行を無視します。コメントの例を次に示します。

```
# My hmake file
# Variable declaration
OUTPUT= c:\dir1\file1.obj
# Descriptor
$(OUTPUT) : c:\dir1\file1.c c:\dir1\file1.h
    set VAR1=value1
    gcc c:\dir1\file1.c
```

hmake ファイルではコメントには専用の行が必要です。コメントを他の文の最後に付けることはできません。

## 13.5 メッセージコマンド

メッセージコマンドは、メイクファイル実行中にテキストの行を標準出力に出力します。これらのテキスト行は、適切に実行されている実行ファイルの出力の中で、メイクファイルに出現する順序で出力します。出力テキストのバッファリングは行いません。メッセージコマンドのシンタックスを以下に示します。

```
!MESSAGE <出力するテキスト>
```

改行文字が<出力するテキスト>の最後の文字の後にあると想定します。!MESSAGE と<出力するテキスト>の間の空白文字を無視します。メッセージコマンドの例を以下に示します。

```
!MESSAGE Executing C Compiler
```

---

High-performance Embedded Workshop V.4.07  
ユーザーズマニュアル

発行年月日 2009年09月01日 Rev.1.00

発行 株式会社 ルネサス テクノロジ 営業統括部  
〒100-0004 東京都千代田区大手町2-6-2

編集 株式会社 ルネサス ソリューションズ ツール開発部

---

© 2009. Renesas Technology Corp. and Renesas Solutions Corp., All rights reserved. Printed in Japan.

# High-performance Embedded Workshop V.4.07 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

RJJ10J2550-0100