

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い申し上げます。

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.4.04 リリースノート

弊社製品をご使用いただき厚く御礼申し上げます。ルネサス統合開発環境 High-performance Embedded Workshop を使用するにあたり注意事項がございます。ご留意いただけますようお願い申し上げます。

目次

1	注意事項.....	3
1.1	ネットワーク上のドライブに関する注意事項.....	3
1.1.1	PC間の時計のずれ.....	3
1.1.2	ネットワーク上プロジェクトのビルド.....	3
1.1.3	プロジェクトへのファイルの追加.....	3
1.2	C/C++言語コメント文に関する注意事項.....	3
1.2.1	エディタのシンタックスカラーリング機能.....	3
1.2.2	ファイル依存関係.....	4
1.3	ナビゲーション機能の注意事項.....	5
1.3.1	C言語関数定義ナビゲーション.....	5
1.3.2	ナビゲーション機能の"C++ Classes"のデフォルト設定.....	5
1.3.3	ナビゲーション機能の"C++ Classes"のウォーニングメッセージ.....	5
1.4	式の指定に関する注意事項.....	6
1.5	エラーメッセージが出たファイルをエディタで開くときの注意事項.....	6
1.6	オプションダイアログボックス上での相対ディレクトリ指定に関する注意事項.....	6
1.7	ワークスペースの作成言語に関する注意事項.....	6
1.8	日本語表示に関する注意事項.....	6
1.9	エディタウィンドウの文字セットに関する注意事項.....	6
1.10	デバッグに関する注意事項(共通).....	6
1.10.1	ソースレベル実行.....	6
1.10.2	ロードモジュール作成後のソースファイル位置移動.....	7
1.10.3	エディタウィンドウ.....	7
1.10.4	逆アセンブリウィンドウ.....	7
1.10.5	スタックトレースウィンドウ.....	7
1.10.6	メモリウィンドウのフォント.....	7
1.10.7	エントリポイント.....	7
1.10.8	レジスタウィンドウ.....	7
1.11	デバッグに関する注意事項(SuperH/H8ファミリデバッガ).....	7
1.11.1	ラインアセンブル.....	7
1.11.2	SYSROF形式ファイルのロード.....	7
1.11.3	Hitachi Debugging Interfaceセッションファイル.....	8
1.11.4	プロファイル.....	8
1.11.5	ウォッチウィンドウおよびローカルウィンドウ.....	8
1.11.6	オーバーレイの構成ダイアログボックス.....	8
1.11.7	シミュレータ・デバッガ.....	9
1.12	ウィンドウおよびダイアログボックスに関する注意事項.....	11
1.12.1	スクロール.....	11
1.12.2	ウィンドウ位置の保存.....	11
1.13	ヘルプに関する注意事項.....	11
1.14	マクロ生成支援機能に関する注意事項.....	12
1.15	コマンドに関する注意事項.....	12
1.15.1	MEMORY_EDITコマンド.....	12
1.15.2	コマンドバッチファイル実行タイミング(デバッグの設定ダイアログボックス).....	12
1.15.3	コマンドバッチファイル実行順序(デバッグの設定ダイアログボックス).....	12
1.15.4	コマンドラインのファイル指定.....	12
1.15.5	CACHEコマンド.....	13

1.15.6	コマンド短縮形	13
1.16	TCL コマンドと High-performance Embedded Workshop コマンドの親和性の拡張に伴う注意事項 13	13
1.17	TCL/TK コマンド入力の注意事項	13
1.18	TCL ツールキットとコマンドラインのコマンドについての注意事項	14
2	ツールチェーン仕様補足	15
2.1	ファイル拡張子	15
3	ツールチェーンアップグレード	16

1 注意事項

1.1 ネットワーク上のドライブに関する注意事項

1.1.1 PC 間の時計のずれ

ソースファイルや出力ファイルの更新日付は、保存した PC の時間で更新されます。ネットワークで共有しているプロジェクトやソースファイルをビルドするとき、PC 間の時計のずれが原因で再ビルドが正確にできない場合があります。このような場合は、PC 間の時計を合わせるか、または[ビルド -> すべてをビルド]でプロジェクトをビルドしてください。

1.1.2 ネットワーク上プロジェクトのビルド

ネットワーク経由で開いているプロジェクトは、ビルド中にネットワークの状態に依存して、ファイルが開けないなどのエラーを発生することがあります。例えば、C/C++コンパイラは次に示すエラーメッセージを出すことがあります。

```
C3019(F) Cannot open source file
```

このような場合、エラーが出たファイルを再度ビルドしてください。

1.1.3 プロジェクトへのファイルの追加

ネットワーク経由で、リモートドライブ上にあるファイルをプロジェクトに[プロジェクト -> ファイルの追加]などで追加するときに、ネットワークの状態に依存して High-performance Embedded Workshop がアプリケーションエラーで終了することがあります。このような場合、再度追加を試みてください。アプリケーションエラーが何度も起きる場合は、ファイルをローカルドライブにコピーしてから追加することをご検討ください。

1.2 C/C++言語コメント文に関する注意事項

1.2.1 エディタのシンタックスカラーリング機能

High-performance Embedded Workshop のエディタには、ソースプログラムファイルのコメント文を色付けするシンタックスカラーリング機能があります。しかし、ソースプログラムの書き方によって、シンタックスカラーリング機能が正しく動作しない場合があります。この場合、シンタックスカラーリング機能を外してください。シンタックスカラーリング機能を外するには、[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の[エディタ]タブで[シンタックスカラーリング]のチェックを外し[OK]ボタンをクリックしてください。次のような場合、シンタックスカラーリング機能が正しく動作しません。

1) コメントをネストした場合

例：/* /* */ */ 下線部がコメントの色になる。

2) 文字列内に “ /* ” や “ */ ” を記述した場合

例：/* 下線部がコメントの色になる。
char A[] = “*/”; 下線部がコメントの色になる。
*/

なお、C/C++ソースプログラムで *文字や、/文字をコメント以外の目的で隣接して記述することを避けることで回避できます。

1.2.2 ファイル依存関係

High-performance Embedded Workshop は、ソースファイルがインクルードしているファイルを検索し、ワークスペースウィンドウの[Projects]タブに依存ファイルとして表示します。この依存ファイルは、ファイル依存関係としてビルド可否の判断に利用されます。ソースプログラムがシンタックス上正しく書かれていても、依存ファイルの検索中にアウトプットウィンドウの[Build]タブにエラーメッセージが出力された場合は、インクルードしているファイルが依存ファイルとして正しく検索されていないことがあります。このような場合、(a) [ビルド -> すべてをビルド] を選択してすべてのファイルをビルドするか、または、(b) ワークスペースウィンドウの [Projects] タブで当該ソースファイル名を選択し、[ビルド -> コンパイル ファイル名]で当該ソースファイルをコンパイルしてください。

C/C++言語ソースファイルで次のような場合、依存ファイルが正しく検索されません。

1) 文字列内に “ /* ” や “ */ ” がある場合

例: char A[] = "/*";	コメント開始とみなされる。
#include "file.h"	インクルードファイルが検索されない。
char B[] = "*/";	コメント終了とみなされる。

High-performance Embedded Workshop で、#if、#ifdef、#define などのプリプロセッサ文をサポートするためには、[ビルド]メニューからツールチェインのオプションダイアログボックスを表示し、[全般]タブで[依存関係検索にてプリプロセッサ文サポート]をチェックします。

表 1.1 にサポートしているプリプロセッサ文を示します。

表 1.1 プリプロセッサ文

プリプロセッサ文	説明
#define	識別子を定義する。プリプロセッサ文が識別子を含んでいると、識別子を定義された文字列で置き換える。 例. #define NICE_FILE "nice.h" #include NICE_FILE
#undef	定義された識別子の置き換えを無効にする。
#include	指定されたファイルが現在のソースファイルに依存していることを示す。
#if~#else~#endif #elif	#if <式>は、式を満たすソースコードの範囲のみ評価する。 式の結果が 0 以外のときは、#if から#else の範囲。 式の結果が 0 のときは、#else から#endif の範囲。
defined マクロ	defined(<識別子>)文は、識別子が定義されていれば 1 を返す。識別子が定義されていなければ 0 を返す。通常、#if defined(MACRO)として記述される。
#ifdef #ifndef	#ifdef <識別子>文は、識別子が定義されていれば#if 1 と等価になる。識別子が定義されていなければ#if 0 と等価になる。 #ifndef <識別子>文は、識別子が定義されていなければ#if 0 と等価になる。識別子が定義されていなければ#if 1 と等価になる。
#line	無視する。
#error	無視する。
#pragma	無視する。
#	無視する。

プリプロセッサ文に<式>がある場合、依存関係の検索では以下の 10 種類の演算子をサポートしています。これ以外を使用すると依存関係の検索は正しく動作しません。

() !、<、<=、>、>=、==、!=、&&、||

1.3 ナビゲーション機能の注意事項

1.3.1 C 言語関数定義ナビゲーション

High-performance Embedded Workshop は、C/C++言語ソースファイルから C 言語関数定義を検索してワークスペースウィンドウの Navigation タブに表示します。検索された C 言語関数定義は、ダブルクリックにより該当ファイルの該当行がエディタで開かれます。C 言語関数定義の検索は、プリプロセッサ文を無視しています。例えば、ワークスペースウィンドウの Navigation タブに次の 2 つの func()関数定義が表示されます。

```
#define DEF 1
#ifdef DEF
void func(void)
{
}
#else
int func(int a)
{
}
```

1.3.2 ナビゲーション機能の"C++ Classes"のデフォルト設定

ナビゲーション機能の"C++ Classes"のデフォルト設定は無効（チェックなし）です。設定状態は、[カテゴリ選択]ダイアログボックスで確認できます（図 1.1 参照）。[カテゴリ選択]ダイアログボックスは、[Navigation]タブウィンドウ内で右クリックし、[カテゴリの選択]を選択すると表示します。

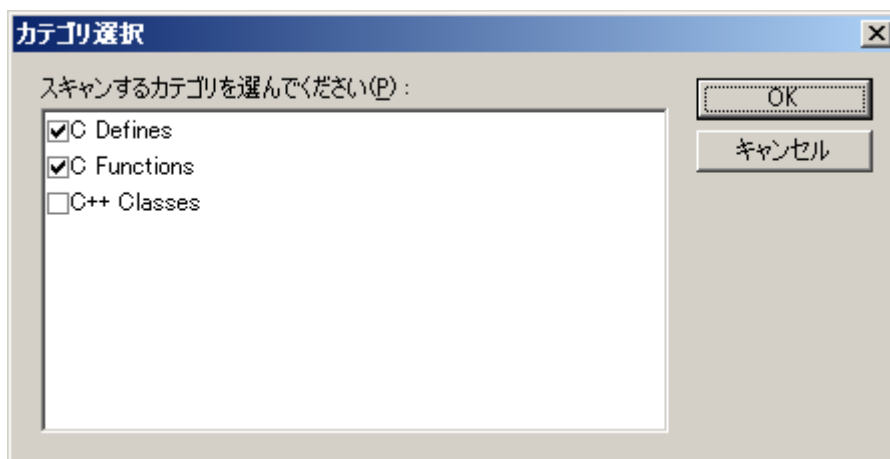


図 1.1 [カテゴリ選択]ダイアログボックス

1.3.3 ナビゲーション機能の"C++ Classes"のウォーニングメッセージ

ナビゲーション機能の"C++ Classes"の設定を有効（チェックあり）に変更すると下記のウォーニングメッセージを表示します（図 1.2 参照）。本機能を有効にするとバックグラウンドでナビゲーション機能が動作するため、PC の CPU 占有率が 100%近くになる場合があります（PC の性能による）。High-performance Embedded Workshop の反応が遅くなる場合があります。



図 1.2 ウォーニングメッセージ

1.4 式の指定に関する注意事項

- (1) 式のシンボルに C++プログラムの関数名は指定できません。
- (2) 関数名として多重定義演算子は指定できません。

1.5 エラーメッセージが出たファイルをエディタで開くときの注意事項

ビルドで C/C++ Compiler、Assembler がアウトプットウィンドウに出力したエラー / ウォーニングメッセージをダブルクリックすると当該ファイルを開き、当該行にカーソルを移動します。しかし、当該ファイルウィンドウがエディタウィンドウエリアで最小化されている場合、当該エラー / ウォーニングメッセージをダブルクリックしても開かれません。このような場合、当該ファイルを元のサイズに戻すか、または、最大化してください。

1.6 オプションダイアログボックス上での相対ディレクトリ指定に関する注意事項

High-performance Embedded Workshop の[ビルド]メニューからコンパイラなど各ツールのオプションダイアログボックスを表示できます。これらのオプションダイアログボックス上でディレクトリパスを指定する時、相対パスを指定しないでください。特に、図 1.3 に示すようなオプションダイアログボックスで"Custom directory"を選択した場合、"Directory"フィールドには必ず絶対パスを指定してください。

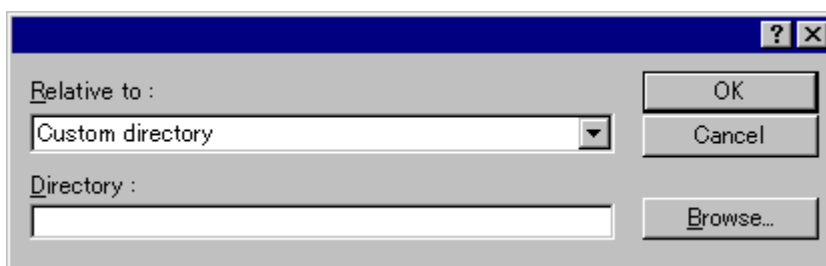


図 1.3 ディレクトリパス指定ダイアログボックス

1.7 ワークスペースの作成言語に関する注意事項

High-performance Embedded Workshop V.4.04 をインストールした時に指定した言語（日本語か英語）と異なる言語の High-performance Embedded Workshop V.3.01 で作成したワークスペースを開くと、ツールバーが表示されません。その場合、[基本設定 -> カスタマイズ]の[ツールバー]タブでツールバーを選択して表示してください。

1.8 日本語表示に関する注意事項

メニューおよびダイアログボックス内の文字列が一部英語で表示される場合があります。

1.9 エディタウィンドウの文字セットに関する注意事項

エディタウィンドウは Unicode をサポートしていません。

1.10 デバッグに関する注意事項（共通）

1.10.1 ソースレベル実行

ステップインを実行すると標準 C ライブラリにも移行します。上位関数に戻るにはステップアウトを使用してください。
また、for および while 文では、1 回のステップで次の行に進まないことがあります。進める場合はもう一度ステップしてください。

1.10.2 ロードモジュール作成後のソースファイル位置移動

ロードモジュール作成後にソースファイルを移動させた場合、作成したロードモジュールのデバッグ中にソースファイルを指定するための[ファイルを開く]ダイアログボックスが表示されることがあります。対応するソースファイルを選択し[開く]ボタンをクリックしてください。

1.10.3 エディタウィンドウ

エディタウィンドウに表示中のプログラムを変更し、ソースファイルとロードモジュールを再ロードしたときは、一旦エディタウィンドウを閉じて、開き直してください。そのまま使用すると、エディタウィンドウの表示が不正になる場合があります。

1.10.4 逆アセンブリウィンドウ

逆アセンブリウィンドウのソースモードを表示し、さらにエディタウィンドウの混合モードを表示している場合に、ステップ実行、またはプログラム実行を行なうと、逆アセンブリウィンドウのソースモードで、プログラムカウンタ(PC)のアイコン(➡)が正しい位置に表示されない場合があります。

逆アセンブリウィンドウのソースモードでデバッグする場合は、エディタウィンドウをソースモードに切り替えてご使用ください。

1.10.5 スタックトレースウィンドウ

割込み関数を実行した状態でスタックトレースウィンドウを表示させると、割込み関数以前の表示が不正となります。

1.10.6 メモリウィンドウのフォント

表示フォントをプロポーショナルフォントに設定している場合、表示が欠ける場合があります。表示フォントを固定幅のフォントに変更してください。

1.10.7 エントリポイント

リンケージエディタの ENTRY オプション等でエントリポイントを指定した場合でも、プログラムのダウンロード時 PC へのエントリポイントアドレスの設定は行ないません。プログラム実行前に PC の設定を行なってください。

1.10.8 レジスタウィンドウ

High-performance Embedded Workshop V.4.04 以降は、レジスタ値をセッションに保存しないようにしました。

1.11 デバッグに関する注意事項 (SuperH/H8 ファミリデバッガ)

1.11.1 ラインアセンブル

ラインアセンブル時の入力基数のデフォルトは基数設定に関係なく、10進数です。16進数で指定する場合は、H'または 0x を指定してください。

1.11.2 SYSROF 形式ファイルのロード

SYSROF 形式のデバッグ対象プログラムはロードできません。デバッグ対象プログラムは ELF/DWARF2 形式で作成してください。

1.11.3 Hitachi Debugging Interface セッションファイル

Hitachi Debugging Interface で保存したセッションファイルは使用できません。High-performance Embedded Workshop のプロジェクトワークスペースとして作成し直してください。なお Hitachi Debugging Interface のコマンドファイルは使用可能です。

1.11.4 プロファイル

プロファイルは、オーバーレイ機能には対応していません。

1.11.5 ウォッチウィンドウおよびローカルウィンドウ

(1) 最適化時の局所変数

最適化オプションでコンパイルされたCソースの局所変数表示は、生成されたオブジェクトコードによっては、正しく表示できないことがあります。逆アセンブリウィンドウを表示し生成されたオブジェクトコードを確認してください。

(2) 変更

変数内容を変更する場合、入力するデータに日本語文字列を指定しないでください。日本語文字列を入力する場合は、メモリウィンドウを使用してください。

1.11.6 オーバーレイの構成ダイアログボックス

オーバーレイセクションに連結させたセクションがその他のオーバーレイセクションにも連結されているかのように表示してしまう場合があります。

下記(1)、(2)、(3)のようにオーバーレイセクションを配置し(例 1)、ビルドの結果、各セクションが例 2 のように配置された場合、[オーバーレイの構成]ダイアログボックスには以下のように表示されます(例 3)。

例 1)

<Section>

(1) "P11, P12"

(2) "P21"

(3) "P31, P32"

例 2)

<Sec> <Start - End >

"P11" 0x1000 - 0x10FF (*1)

"P12" 0x1100 - 0x12FF

"P21" 0x1000 - 0x10FF (*1)

"P31" 0x1000 - 0x105F

"P32" 0x1060 - 0x11FF

*1. 連結セクションを持つ"P11"と、それを持たない"P21"セクションの終了アドレスが同じ。

例 3)

<Section>

(1) "P11, P12"

(2) "P21, P12" (*2)

(3) "P31, P32"

*2. "P12"セクションが"P21"にも連結されているかのように表示。

1.11.7 シミュレータ・デバugg

(1) メモリリソース設定

メモリリソース変更機能の仕様がエミュレータと異なりますので、ご注意ください。

シミュレータ・デバuggの仕様は以下のとおりです。

- (a) 変更前と変更後のメモリ属性(Read、Write、Read/Write)が同一の場合
メモリリソースサイズの変更と判断してメモリリソースサイズを指定値に変更します。
- (b) 変更前と変更後のメモリ属性が異なる場合
メモリリソース属性の変更と判断して、指定範囲のメモリ属性を指定値に変更します。

(2) PC ブレークポイント設定数と条件を指定して実行メニューのテンポラリ PC ブレークポイント設定数

PC ブレークポイント設定数と[条件を指定して実行]メニューのテンポラリ PC ブレークポイント設定数の合計は最大 1,024 個です。したがって PC ブレークポイントを 1,024 個設定した状態では、[条件を指定して実行]メニューのテンポラリ PC ブレークポイントでの指定は無効となります。PC ブレークポイントと[条件を指定して実行]メニューのテンポラリ PC ブレークポイントは設定数の合計が 1,024 個以下で使用してください。

(3) デバuggの設定ダイアログボックス

- ・自動的にターゲットを接続しない
[オプション]タブの[自動的にターゲットを接続しない]チェックボックスの設定にかかわらず[デバuggの設定]完了時にターゲットを接続します。

(4) SH-4 シミュレータ、SH-4 with BSC シミュレータ

- ・SH-4 with BSC シミュレータでは、DMA 転送において、転送元、転送先のアドレスの下位 3 ビットが違ふ場合、転送最後のデータが不正になります。

例) SAR0=2000 DAR0=4004 DMATCR 0=2 CHCR0=5491

2000 番地のメモリ内容 : 0102030405060708

DMA 転送終了後の 4004 番地の内容 : 0106

- ・SH-4 シミュレータ、SH-4 with BSC シミュレータともに、デコード完了したアドレスの命令を変更してもパイプラインリセット実行からの開始になりません。

- ・SH-4 with BSC シミュレータでは、ブレークデータの設定で指定したデータのサイズと異なるサイズでメモリアクセスを行なった場合、ブレークデータの条件が成立しても停止しない場合があります。その場合、メモリアクセスのサイズとブレークデータの設定で指定するデータのサイズを同一にしてください。

- ・SH-4 シミュレータ、SH-4 with BSC シミュレータともに、倍精度 FDIV 命令、倍精度 FSQRT 命令のパイプラインが実機と異なります。F3 ステージのパイプが 1 サイクル多く表示されます。

(5) SH-3DSP シミュレータ

- ・DSP 繰り返し(ループ)制御中の例外コード

DSP 繰り返し(ループ)制御中に例外が発生した場合、EXPEVT(例外事象レジスタ)に設定する例外コードがプログラミングマニュアルとは異なります。

DSP 繰り返し(ループ)制御中の例外コード

一般例外事象	プログラミングマニュアル	シミュレータ
TLB ミス例外/TLB 無効例外(読み出し)	H'070	H'040
TLB ミス例外/TLB 無効例外(書き込み)	H'070	H'060
TLB 保護例外(読み出し)	H'0D0	H'0A0
TLB 保護例外(書き込み)	H'0D0	H'0C0
CPU アドレスエラー(読み出し)	H'070	H'0E0
CPU アドレスエラー(書き込み)	H'070	H'100

・ X/Y メモリアクセスの競合

XRAM メモリ(XROM、YROM、YRAM メモリでも同様)に命令コードとデータを配置した場合、命令コードフェッチによる XRAM メモリアクセスと MOVX、MOVY 命令による XRAM メモリアクセスが同スロットになっても競合によるストールは発生しません。

このため、サイクル数が異なります。

・ 4 の倍数番地以外から実行した場合のパイプライン

4 の倍数番地以外から実行した場合のフェッチのパイプラインがプログラミングマニュアルとは異なります。

このため、4 の倍数番地以外から実行した場合のフェッチで例外が発生した場合の動作が異なります。

例) 4 の倍数番地以外から実行した場合のパイプライン

プログラミングマニュアル	シミュレータ
IF IF ID EX	IF ID EX
IF ID EX	IF ID EX
if ID EX	if ID EX

(6) トレース機能 (SH3、SH3E、SH-3DSP のみ)

・ FPU、MAC、DSP レジスタのアクセス情報

FPU、MAC、DSP レジスタに書き込む命令を実行しても、トレース情報にアクセス情報は表示しません。

・ パイプライン表示

プログラミングマニュアルの表記上ライトバックがない命令でも、命令動作上レジスタへの書き込みが発生する場合、シミュレータ・デバッガのトレース表示では、メモリアクセス、ライトバックのステージとして表示します。サイクル数は正しく表示します。

例) TRAPA 例外のトレース表示

シミュレータ・デバッガでのトレース表示

		IF	DE	EX	MA	SW
05	NOP	06	05			
06	TRAPA #H'10	07	06	05		
--	--	--	--	06		
--	--	--	--	06	06	
--	--	--	--	06	06	
--	--	--	--	06	06	06 (06):SSR<-60000001
--	--	08	--	--	06	06 (06):SPC<-00001006
--	--	09	08	--	--	--

プログラミング
マニュアルの表記

IF	DE	EX	MA	SW
06	05			
--	06			
--	--	06		
--	--	06	06	
--	--	06	06	
--	--	06	06	06 (06):SSR<-60000001
07	--	--		
08	07	--		

- (7) H8SX シミュレータ
H8SX シミュレータは H8SX CPU のミドルモードはサポートしていません。
- (8) シミュレータターゲットの選択
プロジェクト作成時のデバッガターゲット選択では CPU、および動作モードと一致するシミュレータを選択してください。
- (9) SH2A-FPU シミュレータ・デバッガの内蔵 RAM 領域
SH2A-FPU シミュレータ・デバッガのメモリマップでは 0xFFFF80000 番地から 0xFFFFBFFFF 番地を内蔵 RAM 領域として表示しますが、0xFFFFA0000 番地から 0xFFFFBFFFF 番地は予約領域です。このため、SH2A-FPU シミュレータ・デバッガご使用時は以下の点にご注意ください。
- ・ 内蔵 RAM 領域は 0xFFFF80000 番地から 0xFFFF9FFFF 番地以内を使用してください。
 - ・ スタック領域を内蔵 RAM 領域に割り当てる場合も、0xFFFF80000 番地から 0xFFFF9FFFF 番地以内となるようにスタックポインタ初期値を設定してください。
- (10) 旧バージョン用ワークスペースの保存
High-performance Embedded Workshop V.4.04 以降では、旧バージョン形式でワークスペースを保存すると、シミュレータ・デバッガの動作が非常に遅くなる場合があります。その場合、トレース容量を 32,768 以下にしてください。

1.1.2 ウィンドウおよびダイアログボックスに関する注意事項

1.1.2.1 スクロール

インテリジェントマウスで上スクロールができない場合があります。この場合は、ウィンドウ上のスクロールボタンを使用してください。

1.1.2.2 ウィンドウ位置の保存

ウィンドウの表示位置が、セッションのセーブおよびロードまたはバーチャルデスクトップの切り替えにより移動する場合があります。

1.1.3 ヘルプに関する注意事項

英語版 Windows® でヘルプ使用時に以下のメッセージボックスが表示される場合があります。[Download]ボタンをクリックして、"Japanese Text Display Support"をダウンロードするか、[Never download any of these components.]チェックボックスをチェックしてから[Cancel]ボタンをクリックしてください。"Japanese Text Display Support"をダウンロードしなくても、ヘルプの表示には影響ありません。

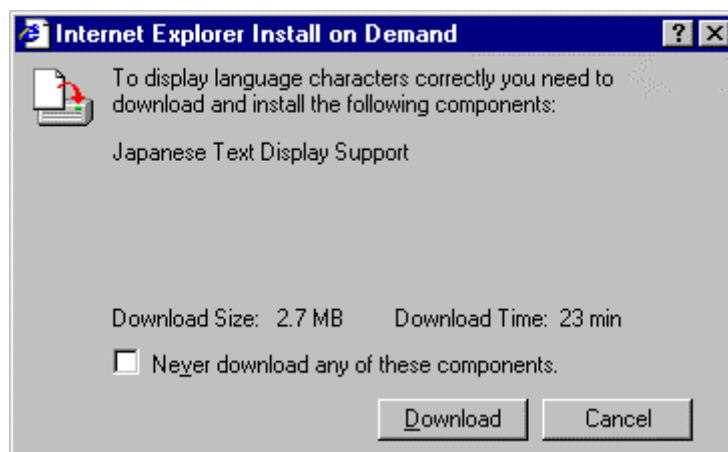


図 1 . 4 Internet Explorer Install on Demand メッセージボックス

1.14 マクロ生成支援機能に関する注意事項

マクロ生成支援機能を使用し、[デバッグ -> 実行]、[デバッグ -> プログラムの停止]の選択を記録したマクロを実行すると、[デバッグ -> 実行]は実行されますが、[デバッグ -> プログラムの停止]は実行されません。

1.15 コマンドに関する注意事項

1.15.1 MEMORY_EDIT コマンド

文字を一重引用符(')で囲っても ASCII 文字列をデータとして入力できません。
文字列をデータとして入力する場合は対応する数値を入力してください。

1.15.2 コマンドバッチファイル実行タイミング (デバッグの設定ダイアログボックス)

- (1) [オプション]タブの[コマンドバッチファイル実行タイミング]で[At target connection]を選択したコマンドファイル中に"INITIALIZE"コマンドを指定した場合、[コマンドライン]ウィンドウで"INITIALIZE"コマンドを実行しないでください。初期化する場合は、[デバッグ -> 初期化]を選択してください。
- (2) [オプション]タブの[コマンドバッチファイル実行タイミング]で[At target connection]を選択したコマンドファイルに以下のコマンドは指定しないでください。
 - (a) [OPEN_WORKSPACE]コマンド
 - (b) [CHANGE_PROJECT]コマンド
 - (c) [CHANGE_CONFIGURATION]コマンド
- (3) [オプション]タブの[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 modules]を選択したコマンドファイルに以下のコマンドは指定しないでください。
 - (a) [OPEN_WORKSPACE]コマンド
 - (b) [CHANGE_PROJECT]コマンド
 - (c) [CHANGE_CONFIGURATION]コマンド
 - (d) [GO]コマンド
 - (e) [GO_RESET]コマンド
 - (f) [GO_TILL]コマンド
 - (g) [STEP]コマンド
 - (h) [STEP_OUT]コマンド
 - (i) [STEP_OVER]コマンド

1.15.3 コマンドバッチファイル実行順序 (デバッグの設定ダイアログボックス)

[オプション]タブの[コマンドバッチファイル実行順序]ではプレースホルダのうち[ファイルディレクトリ]は使用できません。また、相対パス形式でコマンドファイルを指定すると正しくファイルにアクセスできない場合があります。
プレースホルダを適用できないコマンドファイルは絶対パス形式で指定してください。

1.15.4 コマンドラインのファイル指定

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する場合は、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、正しくファイルを参照できなくなるため、ファイル指定をやり直してください。

例) FILE_LOAD ELF/DWARF2 \$(CONFIGDIR)¥¥demo.abs

1.15.5 CACHE コマンド

エミュレータ・デバッガのマニュアルに CACHE コマンドの記載がない場合は CACHE コマンドを指定しないでください。キャッシュのメモリアクセス単位は、0x3FF など固定値のため、指定したアクセス範囲より大きい範囲のキャッシュメモリをアクセスする場合があります。

1.15.6 コマンド短縮形

・ REMOVE_FILE コマンド

High-performance Embedded Workshop V.3.01 から短縮形"RF"を"REM"に変更しました。

1.16 TCL コマンドと High-performance Embedded Workshop コマンドの親和性の拡張に伴う注意事項

TCL コマンドと High-performance Embedded Workshop コマンドの親和性を拡張(*)したことにより、次のような記述の場合、High-performance Embedded Workshop コマンド"memory_display 300 10"の実行結果は出力されません。

```
for {set i 0} {$i < 2} {incr i} {  
    memory_display 300 10  
}
```

High-performance Embedded Workshop コマンドの実行結果を出力する場合は、TCL コマンド"puts"のパラメータとして High-performance Embedded Workshop コマンドを"[]"内に置いてください。

```
for {set i 0} {$i < 2} {incr i} {  
    puts [memory_display 300 10]  
}
```

*. TCL コマンドと High-performance Embedded Workshop コマンドの親和性を拡張したことにより、TCL コマンド"set"のパラメータとして、High-performance Embedded Workshop コマンドの実行結果を変数へ代入できるようになりました。

下記の例では、High-performance Embedded Workshop コマンド"memory_display 300 10"の実行結果が変数"md_300_10"に代入されます。TCL コマンド"set"で変数"md_300_10"を指定すると実行結果を参照できます。

```
set md_300_10 [memory_display 300 10]
```

1.17 TCL/TK コマンド入力の注意事項

(1) 対話モードの中止方法

[TCL ツールキット]で対話モードからコマンド入力モードに戻るには、"/."を入力してください。現在のモードが対話モードかコマンド入力モードかは、"/."を入力して判断してください。

(2) 日本語ディレクトリ

High-performance Embedded Workshop を日本語ディレクトリにインストールした場合、High-performance Embedded Workshop を実行すると TCL/TK のライブラリを仮想ドライブ("u:"ドライブ)に関連付けます。すでに"u:"ドライブを使用している場合、High-performance Embedded Workshop の開始時に"u:"ドライブとの接続を解除し、High-performance Embedded Workshop の TCL/TK ライブラリと接続します。High-performance Embedded Workshop の終了時に、TCL/TK のライブラリは接続を解除します。元のパスとは、自動で再接続しません。

仮想ドライブを使いたくない場合は、High-performance Embedded Workshop を英語ディレクトリにインストールしてください。

(3) ログファイルの内容の消去方法

[TCL ツールキット]を使用している場合、[Console]画面の内容がログファイルに出力されます。ログファイルは以下のディレクトリにテキスト形式ファイルで生成されます。

```
C:\¥Documents and Settings¥<ユーザ名>¥Local Settings¥Temp¥log.txt
```

[TCL ツールキット]を終了するとログファイルの内容は自動的に消去されます。

[TCL ツールキット]の使用中にログファイルの内容を消去したい場合、以下のコマンドを実行してください。

```
set dir $env(TEMP)
set dataFile [open $dir/log.txt {RDWR TRUNC} ]
close $dataFile
```

1.18 TCL ツールキットとコマンドラインのコマンドについての注意事項

(1) "trace" コマンド

(a) [TCL ツールキット]で TCL の"trace"コマンドを実行する場合、コマンド名をすべて小文字で指定してください。

(b) [TCL ツールキット]で High-performance Embedded Workshop の"TRACE"コマンドを実行する場合、コマンド名をすべて大文字で指定してください。

(c) [コマンドライン]ウィンドウで TCL の"trace"コマンドを実行する場合、コマンド名を"tcl_trace"に置き換えて指定してください。

(2) "clock" コマンド

(a) [TCL ツールキット]で TCL の"clock"コマンドを実行する場合、コマンド名をすべて小文字で指定してください。

(b) [TCL ツールキット]でエミュレータの"CLOCK"コマンドを実行する場合、コマンド名をすべて大文字で指定してください。

(c) [コマンドライン]ウィンドウで TCL の"clock"コマンドを実行する場合、コマンド名を"tcl_clock"に置き換えて指定してください。

(3) "event" コマンド

(a) [TCL ツールキット]で TK の"event"コマンドを実行する場合、コマンド名をすべて小文字で指定してください。

(b) [TCL ツールキット]でエミュレータの"EVENT"コマンドを実行する場合、コマンド名をすべて大文字で指定してください。

"CLOCK"コマンドと"EVENT"コマンドは、一部のエミュレータではサポートしていません。

2 ツールチェーン仕様補足

2.1 ファイル拡張子

High-performance Embedded Workshop がツールチェーンの各ツールをビルド実行したとき、コンフィグレーションディレクトリにサブコマンドファイルを残します。C/C++ Compiler と Assembler のサブコマンドファイルの拡張子は入力ファイル名の拡張子を表 2.1 のように変更したファイル名になります。C/C++ Library Generator、OptLinker のサブコマンドファイルはプロジェクト名に表 2.1 の拡張子を付加したファイル名になります。サブコマンドファイルは隠しファイルの属性を持ちます。隠しファイルを表示する場合、ディレクトリを表示するウィンドウのプロパティをすべてのファイルを表示するように設定する必要があります。

表 2.1 ツールチェーンサブコマンドファイル拡張子

拡張子	ファイルグループ
shg	SuperH RISC engine C/C++ Library Generator
shc	SuperH RISC engine C/C++ Compiler
sha	SuperH RISC engine Assembler
H8g	H8S,H8/300 C/C++ Library Generator
H8c	H8S,H8/300 C/C++ Compiler
H8a	H8S,H8/300 Assembler
hik	OptLinker
M16cl	M16C/60、M16C/30、M16C/Tiny、M16C/20、M16C/10、R8C/Tiny Linker
M16ci	M16C/60、M16C/30、M16C/Tiny、M16C/20、M16C/10、R8C/Tiny Librarian
M16cc	M16C/60、M16C/30、M16C/Tiny、M16C/20、M16C/10、R8C/Tiny C Compiler
M16ct	M16C/60、M16C/30、M16C/Tiny、M16C/20、M16C/10、R8C/Tiny mkmrtbl (MR30)
M32cl	M32C/80、M16C/80、M16C/70 Linker
M32ci	M32C/80、M16C/80、M16C/70 Librarian
M32cc	M32C/80、M16C/80、M16C/70 C Compiler
M32ct	M32C/80、M16C/80、M16C/70 mr308tbl (MR308)
M32rl	M32R C Compiler
M32ri	M32R Librarian
M32rm	M32R Load module converter
M32rc	M32R C Compiler
M32ra	M32R Assembler
741k	740 Linker
741b	740 Librarian
100l	R32C/100 Linker
100i	R32C/100 Librarian
100c	R32C/100 C Compiler
100t	R32C/100 mr100tbl (MR100)

3 ツールチェーンアップグレード

以下に示すツールチェーンをプロジェクトで使用している場合、ツールチェーンの新バージョンをインストールすることで、使用しているツールチェーンのアップグレード(変更)を行なうことができます。

- ・ SuperH ファミリ用 C/C++コンパイラパッケージ V.5.1 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.3.0A 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.5.20 Release1 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.5.20 Release1 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 Release00 およびそれ以降
- ・ M32R ファミリ用 C/C++コンパイラパッケージ V.4.20 Release1 およびそれ以降
- ・ 740 ファミリ用 C コンパイラパッケージ V.1.01 Release01 およびそれ以降

(1) SuperH ファミリ用 C/C++コンパイラパッケージ V.5.1x または H8SX,H8S,H8 ファミリ用 C/C++コンパイラパッケージ V.3.0x からのアップグレードの場合

(a) 標準ライブラリ構築ツールの設定

標準ライブラリ構築ツールが追加されました。High-performance Embedded Workshop はアップグレード時、IM OptLinker のオプション情報をもとに、標準ライブラリ入力指定があればライブラリを生成するオプション(Mode : Build a library file)を設定します。標準ライブラリ入力指定がなければ、ライブラリを生成しないオプション(Mode : Do not add a library file)を設定します。

(b) 最適化リンカの設定

IM OptLinker、Librarian、Stype Converter を統合し、OptLinker になりました。アップグレード前の各ツールの登録状態により、アップグレード後の OptLinker のオプション設定が異なります。表 3 . 1 にアップグレード時のオプション継承を示します。

表 3 . 1 アップグレード時のオプション継承

	アップグレード前			アップグレード後			
	IM OptLinker の状態	Librarian の状態	Stype Converter の状態	OptLinker の状態	IM OptLinker オプション情報	Librarian オプション情報	Stype Converter オプション情報
1					x	x	x
2					x	x	
3					x	x	
4					x		x
5					x		x
6					x	x	
7					x		x
8					x		x
9					x		x
10						x	x
11						x	
12					x	x	
13						x	x
14						x	
15					x	x	
16					x		x
17					x		x
18					x		x
19						x	x
20						x	x
21						x	
22						x	x
23						x	x
24						x	
25						x	x
26						x	x
27						x	

: 未登録、 : 登録チェック無し、 : 登録チェック有り、 : 継承、 x : 継承しない

(2) SuperH ファミリ用 C/C++コンパイラパッケージのアップグレードの場合

(a) MAP 最適化用ビルドの自動化

High-performance Embedded Workshop V.2.00 (SH Ver.7.0B / Ver.7.0.01 / Ver.7.0.02) では、C コンパイラで最適化リンケージエディタが出力した外部シンボル割り付け情報を活用した最適化を実行するために、再ビルド用のカスタムフェーズを提供していました。High-performance Embedded Workshop V.2.01 以降では、MAP 最適化実行時に自動的に再ビルド実行が可能になりました。再ビルド用のカスタムフェーズは不要になりますので、図 3 . 1 で示すように、[ビルド -> ビルドフェーズ]で[ビルドフェーズ]ダイアログボックスを開き、"Map optimize"フェーズのチェックをオフにしてください。

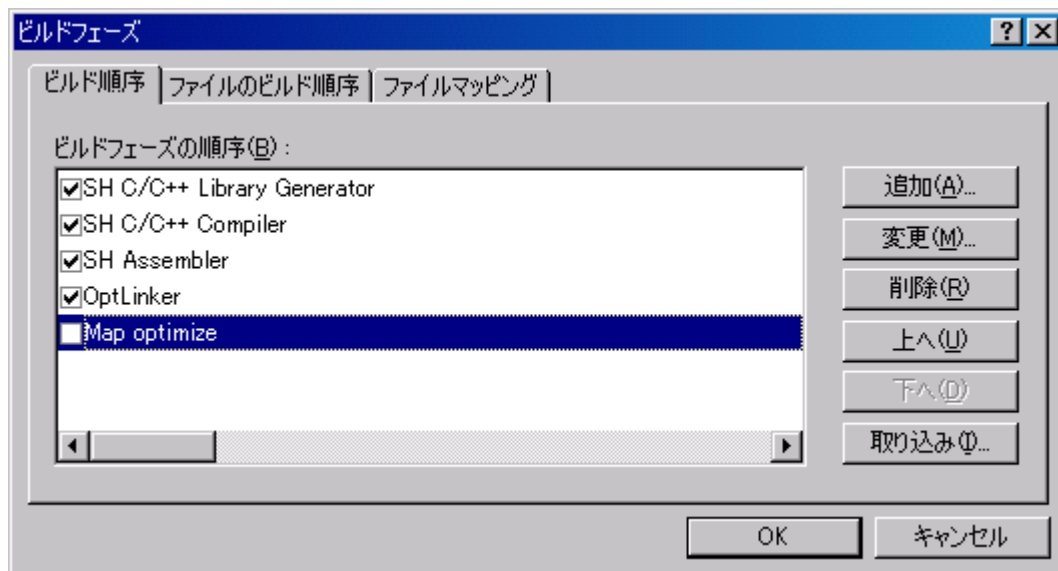


図 3 . 1 [ビルドフェーズ]ダイアログボックス

- 以上 -