

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。

# SuperH RISC engine ファミリ C/C++コンパイラ

## 最適化について

### 1. 最適化レベル

SHC/C++コンパイラ V.7 より、スーパーコンピュータ向けの最新最適化導入を適用しています。

これにより、ポインタや外部変数の別名解析、制御文を含めたデータフロー解析を行い、より強力な最適化を実現しています。

さらに、High-performance Embedded Workshop には最適化レベルを簡単に調節する機能が付いており、旧バージョンのコンパイラとの互換性を保てるようにしています。

High-performance Embedded Workshop での新規プロジェクト作成時には、この最適化レベルは強く設定されています。

下記方法により、最適化レベルを調節することが出来ます。

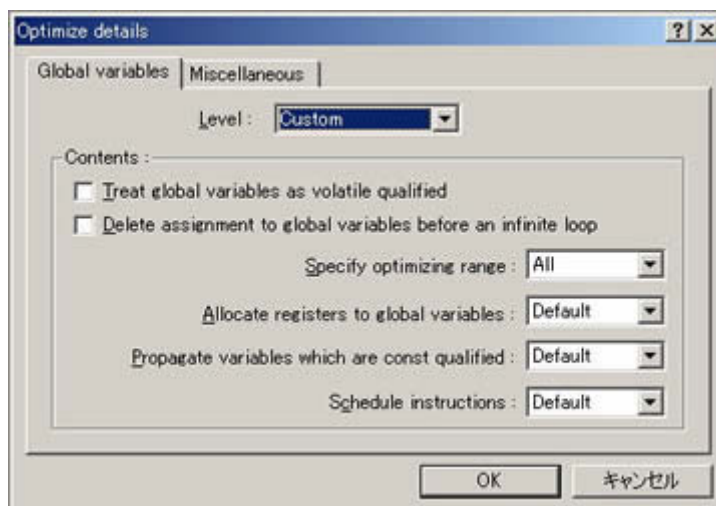
### 最適化レベル設定手順 - High-performance Embedded Workshop

(1) High-performance Embedded Workshop メニュー : [Options]-[SuperH RISC engine Standard Toolchain] を選択し、表示されるダイアログで[C/C++]タブを選択します。

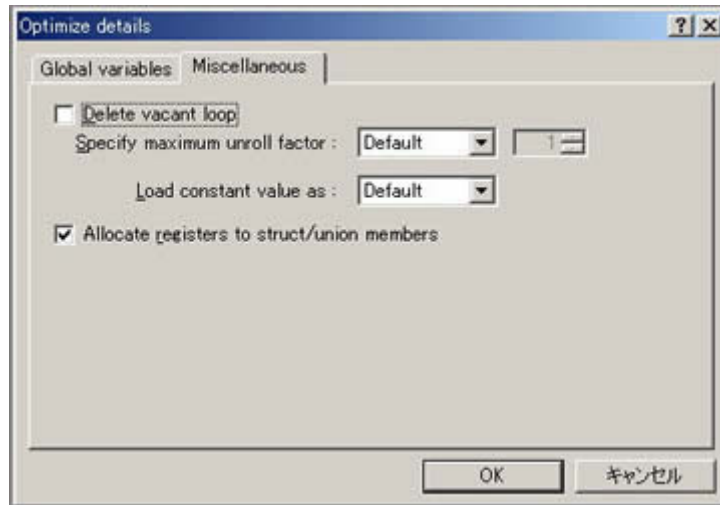
(2) [Category]で[Optimize]を選択し、[Details]ボタンを押してください。

(3) 表示されるダイアログで[Global variables]タブを選択します。

[Level]で最適化レベルの変更が可能です。 [Level]の選択では、"Level1"、"Level2"、"Level3"、"Custom" が選択可能で、Level3 が最も最適化が強くなります。



(7) なお、同ダイアログの[Miscellaneous]タブは、最適化レベルとは無関係に設定可能です



### 最適化レベル設定手順 - コマンドライン

コマンドラインには、High-performance Embedded Workshop のようなレベル設定を持っていません。オンラインマニュアル「SuperH RISC engine C/C++コンパイラ 添付資料」をご参照の上、適宜追加オプションを指定して下さい。

### 最適化によるアプリケーション動作への影響例

下記 (1) のようなソースプログラムを-GLOBAL\_Volatile=0 オプション (High-performance Embedded Workshop では[Treat global variables as volatile qualified]に未チェック)でコンパイルすると、外部変数 flag に対しても最適化が行われ、下記 (2) のようなコード生成になります。これは実際、想定している動作とは異なります。

(1)

ソースプログラム	解説
<pre>extern int flag; void test() {     flag = 1;     if( flag == 0 ) foo(); }</pre>	<p>flag のセット 割込みなどで flag が変わっていたら foo()を呼ぶ</p>

(2)

アセンブリ出力	解説
<pre>_test:     MOV     #1, R2     MOV.L   L12+2, R6     MOV.L   R2, @R6     RTS     MOV     #0, R0  L12:     .DATA. L   _flag</pre>	<pre>flag=1; return 0;</pre>

-GLOBAL\_Volatile=1 オプション (High-performance Embedded Workshop では[Treat global variables as volatile qualified]にチェック) でコンパイルすると、下記のようなコード生成になり、想定している動作となります (-GLOBAL\_Volatile=1 は、volatile 指定の無い外部変数を最適化対象外にするオプションです)。

アセンブリ出力	解説
<pre>_test:     STS.L   PR, @-R15     MOV     #1, R2     MOV.L   L13+2, R5     MOV.L   R2, @R5     MOV.L   @R5, R6     TST     R6, R6     BF      L12     MOV.L   L13+6, R2     JSR     @R2     NOP  L12:     LDS.L   @R15+, PR     RTS     MOV     #0, R0  L13:     .DATA. L   _flag     .DATA. L   _foo</pre>	<pre>flag=1;  flag を再度ロードして 0 かどうか判定  foo();  return 0;</pre>

しかし、-GLOBAL\_Volatile=1 オプション (High-performance Embedded Workshop では[Treat global variables as volatile qualified]にチェック) でコンパイルすると、ファイル内の外部変数 (あるいはプロジェクト全体の外部変数) に対して、最適化対象外となり、コンパイラの最適化効果が得られ難くなります。

割込みなどで変更される可能性のある外部変数には `volatile` を指定するなどし、`-GLOBAL_Volatile=0` を指定することで、よりコンパイラの最適化を有効に使用できるようになります。

ソースプログラム	アセンブリ出力
<code>extern volatile int flag;</code>	<code>_test:</code>
<code>int test()</code>	<code>STC.L PR, @-R15</code>
<code>{</code>	<code>MOV #1, R2</code>
<code>  flag = 1;</code>	<code>MOV.L L13+2, R5</code>
<code>  if( flag == 0 ) foo();</code>	<code>MOV.L R2, @R5</code>
<code>    return 0;</code>	<code>MOV.L @R5, R6</code>
<code>}</code>	<code>TST R6, R6</code>
	<code>BF L12</code>
	<code>MOV.L L13+6, R2</code>
	<code>JSR @R2</code>
	<code>NOP</code>
	<code>L12:</code>
	<code>LDS.L @R15+, PR</code>
	<code>RTS</code>
	<code>MOV #0, R0</code>
	<code>L13:</code>
	<code>.DATA.L _flag</code>
	<code>.DATA.L _foo</code>

## 2. MAP 最適化

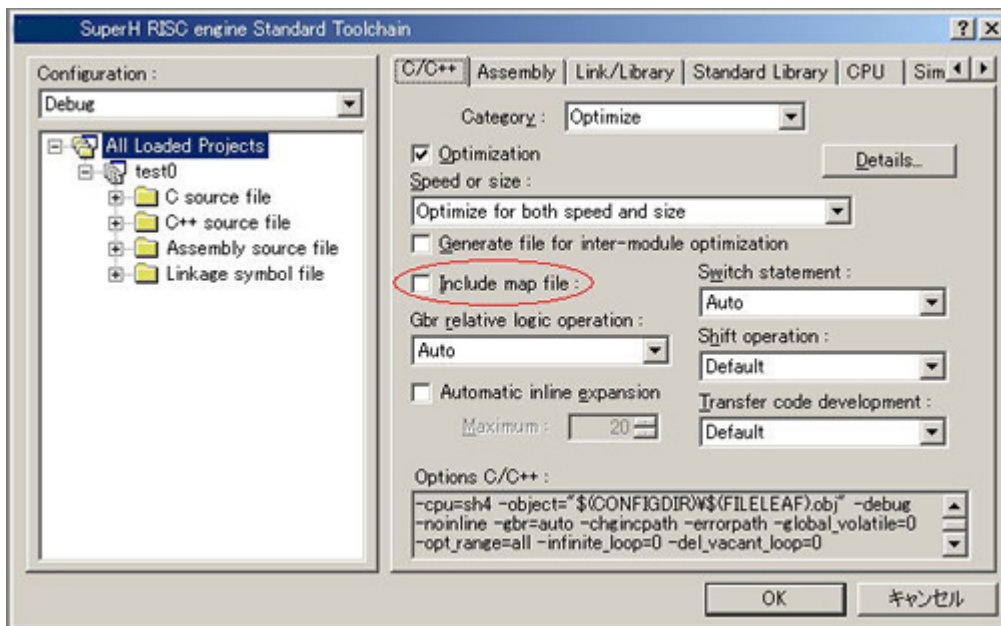
SHC/C++ コンパイラ Ver7 には、外部変数アクセス最適化（通称、MAP 最適化）と呼ばれる新しい最適化技術を導入しています。

プロジェクト作成時には、この最適化は無効になっています。

下記方法により、本最適化を有効にすることで、さらに最適化を強化することが可能です。

### MAP 最適化有効化手順 - High-performance Embedded Workshop

- (1) High-performance Embedded Workshop メニュー : [Options]-[SuperH RISC engine Standard Toolchain] を選択し、表示されるダイアログで [C/C++] タブを選択します。
- (2) [Category] で [Optimize] を選択します。
- (3) [Include map file] にチェックを付けます。



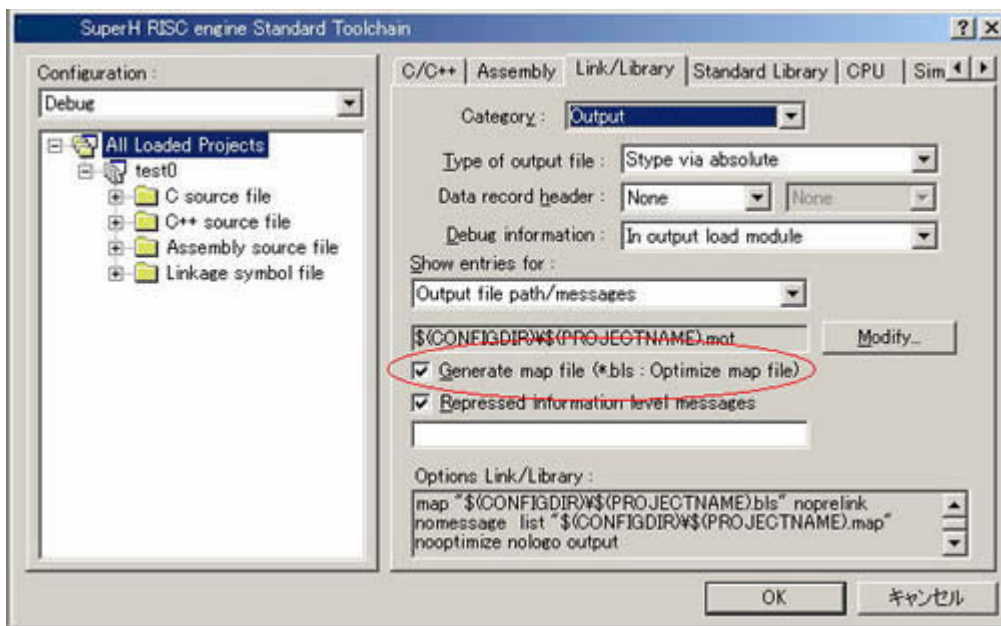
(4) [Include map file]のチェックを付けると、下記のようなメッセージが出力される場合があります。



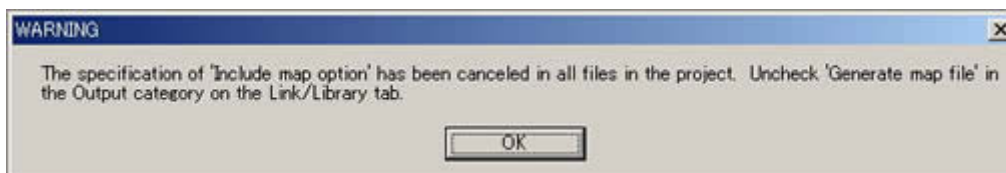
MAP 最適化を使用するためには、(1)から(3)までのコンパイラの設定の他に、下記リンケージエディタの設定が必要になります。

上記メッセージは、下記リンケージエディタの設定を自動的に変更する旨のメッセージです。

同ダイアログ内、[Link/Library]タブ、[Category]-"Output"にて、[Generate map file]に自動的にチェックが付きます。



(5) [Include map file]のチェックを外すと、下記のようなメッセージが出力される場合があります。



MAP 最適化を使用しない場合には、(4)[Generate map file]のチェックが不要になります。

上記メッセージは、本チェックボックスのチェックを外す旨のメッセージです。MAP 最適化が有効になっている状態でビルドを行うと、下記のような手順でビルドが行われます。

順番	フェーズ名	説明
(1)	SH C/C++ Library Generator	標準ライブラリ構築
(2)	SH C/C++ Compiler	1 度目のコンパイル (全ソースが対象)
(3)	SH Assembler	アセンブル
(4)	OptLinker	1 度目のリンク
(5)	SH C/C++ Compiler second pass	2 度目のコンパイル ([Include map file]にチェックしたファイルが対象)
(6)	OptLinker second pass	2 度目のリンク

## MAP 最適化有効化手順 - コマンドライン

### 1 度目 :

コンパイル - 通常のコンパイルオプション

リンク - 通常のリンクオプション + map オプション

### 2 度目 :

コンパイル - 通常のコンパイルオプション + map オプション

リンク - 通常のリンクオプション 例

```
shc.exe -sub=cc.sub <C/C++ source>.c
```

```
asmsh.exe -sub=asm.sub <asm source>.src
```

```
optlnk.exe -map=<bls file>.bls -sub=lnk.sub
```

```
shc.exe -map=<bls file>.bls -sub=cc.sub <C/C++ source>.c
```

```
asmsh.exe -sub=asm.sub <asm source>.src
```

… アセンブラには map オプションが指定できません。

```
optlnk.exe -sub=lnk.sub
```

## MAP 最適化使用時に出力される可能性があるエラーメッセージ

MAP 最適化使用時、下記エラーメッセージが出力される可能性があります。

*L2410 (E)Address value specified by map file differs from one after linkage as to "シンボル"*

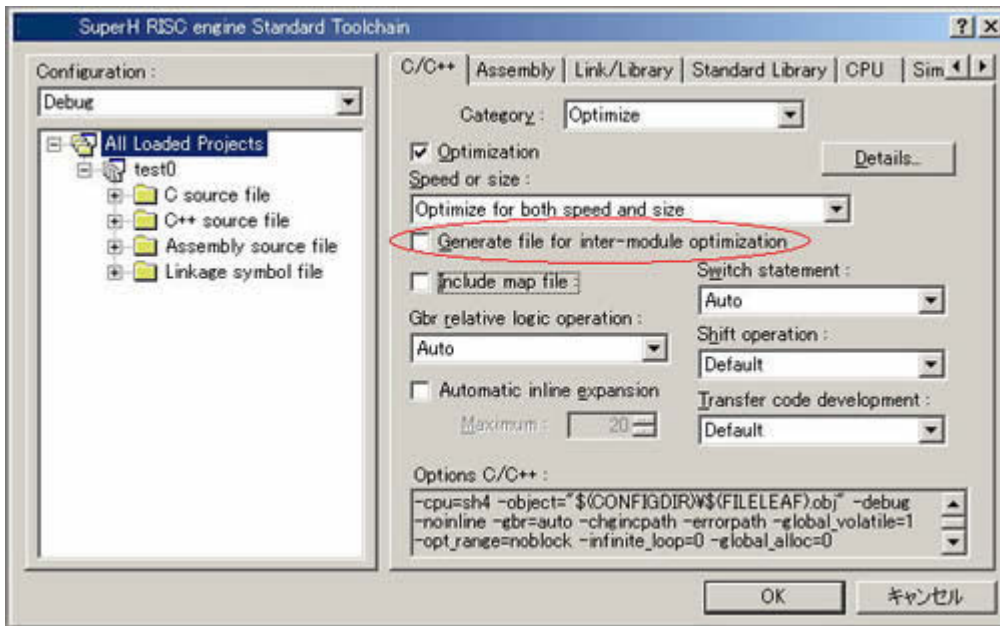
"シンボル"のアドレス値がコンパイル時に使用した外部シンボル割り付け情報ファイル (\*.bls) 内のアドレスとリンク後のアドレスで異なっています。

コンパイル時の map オプション指定前後でプログラムを変更していないか確認、あるいは、Build All して下さい。または、リンケージエディタの最適化によって、コンパイル時の map オプション指定前後のシンボル並び順が変わることがあります。コンパイル時 map オプションを無効にするか、リンケージエディタの最適化を無効にしてください。

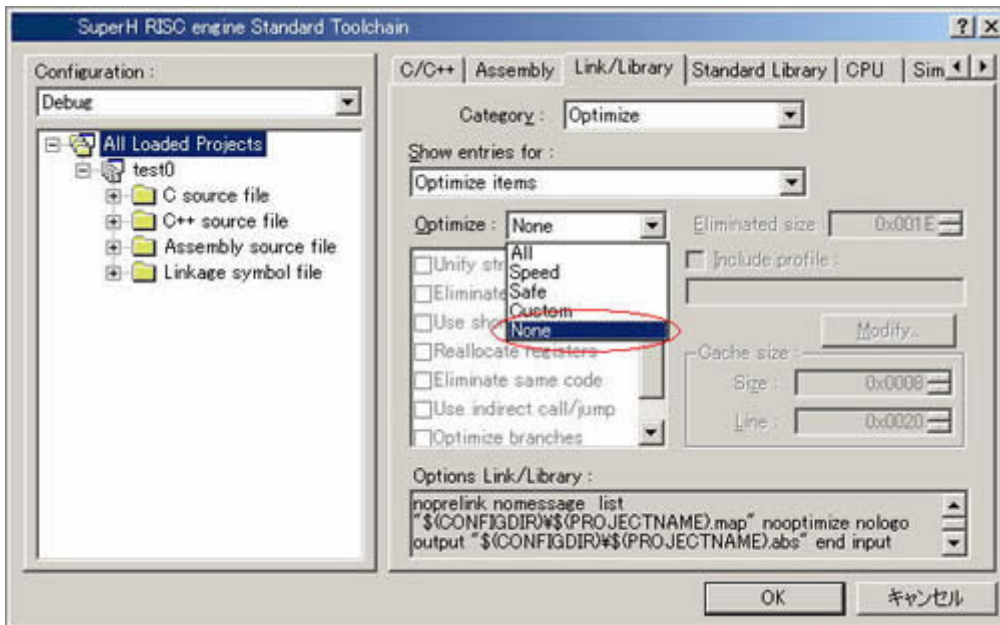
## リンケージエディタ最適化の無効化 - High-performance Embedded Workshop

下記(1)もしくは(2)のいずれかの方法で最適化リンケージエディタの最適化を無効化出来ます。

(1) [Standard Toolchain]ダイアログで、下記チェックボックスを外します。



(2) 同ダイアログで、下記リストボックスから"None"を選択します。



## リンケージエディタ最適化の無効化 - コマンドライン

下記(1)もしくは(2)のいずれかの方法で最適化リンケージエディタの最適化を無効化出来ます。

- (1) 1度目および2度目のコンパイルオプションで`-goptimize`を外します。
- (2) 1度目および2度目のリンクオプションで`-nooptimize`を付けます。

MAP 最適化の内容につきましては、「SuperH RISC engine C/C++コンパイラ、アセンブラ、最適化リンケージエディタユーザーズマニュアル」の「2.2.4 Optimize オプション」の MAP オプションの項、および「4.2.2 Output オプション」の MAP オプションの項をご参照下さい。

### 3. Dhrystone Ver2.1 を使用した最適化比較

MAP 最適化	最適化レベル	プログラムサイズ(*)
なし	Level1	1,532 byte
	Level2	1,488 byte
	Level3	1,480 byte
あり	Level1	1,448 byte
	Level2	1,400 byte
	LevelL3	1,392 byte

(\*)CPU は SH-1 です。標準ライブラリのプログラムサイズは含んでいません。