

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。

コンパイラパッケージ

Call Walker

本ドキュメントでは、ルネサス製コンパイラパッケージ(SuperH, H8, M16C, R8C, およびRXファミリ)に同梱されているスタック情報解析ツール「Call Walker」の使い方について説明します。

目次

1. スタック情報ファイルの作成方法	2
2. Call Walkerの起動	4
3. ファイルのオープンとCall Walkerの画面	4
4. スタック情報の編集	8
5. アセンブラプログラムのスタック使用量	11
6. スタック情報のマージ(連結).....	12
7. その他の機能.....	14

Call Walker は、リンカージェネータが出力したスタック情報ファイル(*.sni)、またはシミュレータデバugg(SuperH ファミリ, H8 ファミリ, またはRX ファミリのみ)が出力したプロファイル情報ファイル(*.pro)を読み込んで、静的なスタック使用量を表示します。

スタック情報ファイルに出力できないアセンブリプログラムのスタック使用量は、編集機能を用いて情報を追加することが可能であり、システム全体のスタック使用量を求めることもできます。

編集したスタック使用量に関する情報は、呼び出し情報ファイル(*.cal)として保存または読み込みが可能です。

また、呼び出し情報ファイルをマージ(連結)することも可能です。

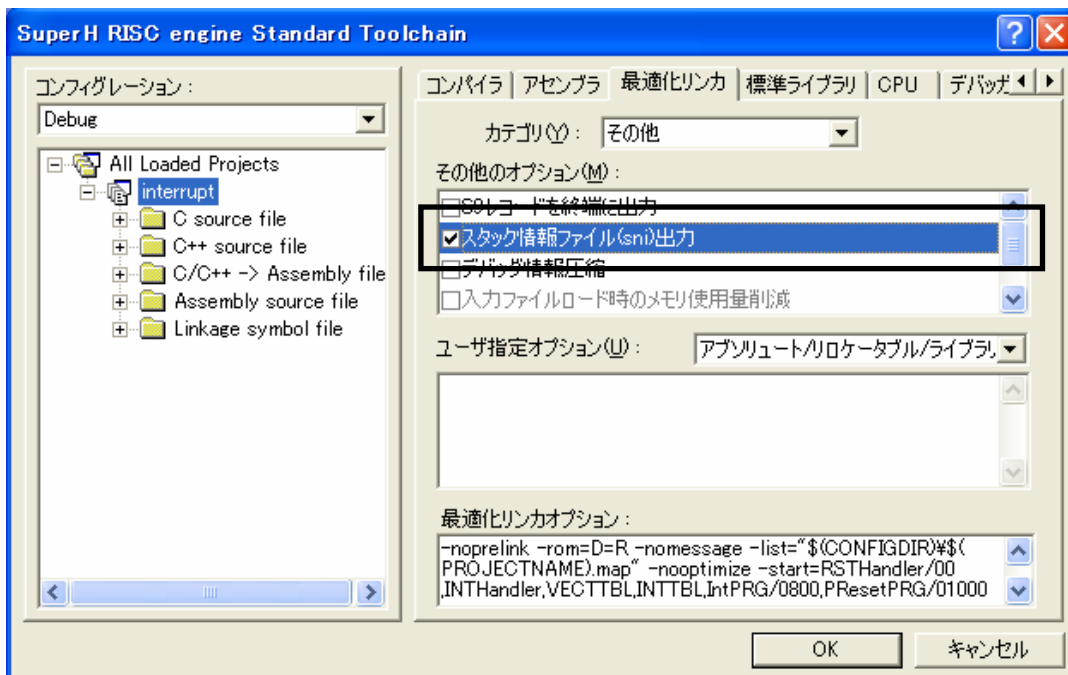
1. スタック情報ファイルの作成方法

1.1 SuperHファミリ, H8 ファミリ, またはRXファミリ用コンパイラの場合

以下の手順に従って、スタック情報ファイル、またはプロファイル情報ファイルのいずれかを作成します。

スタック情報ファイル(*.sni)の作成方法

スタック情報ファイルは、「最適化リンカ」の以下のオプションにより作成することができます。



ダイアログボックスメニュー： [最適化リンカ] - [カテゴリ] - [その他] - [スタック情報ファイル(sni)出力]

コマンドライン : *STACK*

プロファイル情報ファイル(*.pro)の作成方法

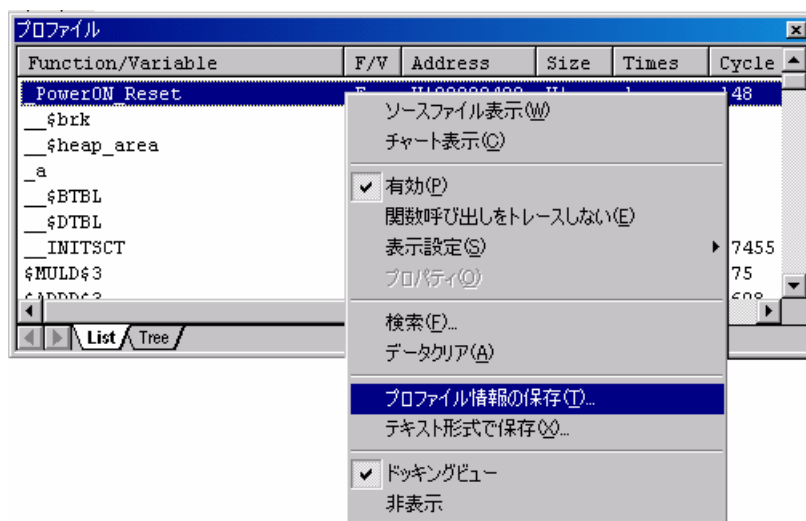
以下のプロファイル機能により、ユーザプログラムを実行させます。

実行終了後プロファイルウィンドウで右クリックし、プロファイル情報の保存によりプロファイル情報ファイル(*.pro)が作成されます。

プロファイル情報作成方法の詳細は、シミュレータ/デバッガ ユーザーズマニュアル「プロファイル情報を見る」を参照してください。

【プロファイルウィンドウ】

[表示] -> [パフォーマンス] -> [プロファイル]



1.2 M16Cファミリ, またはR8Cファミリ用コンパイラの場合

High-performance Embedded Workshop でビルドしリンクした場合、何の設定もすることなく、常にスタック情報ファイル(*.sni)は作成されます。ただし、コンパイル時に"-finfo"オプションを指定してください。"-finfo"オプションを指定しないと、スタック情報ファイルに必要な情報が入りません。

High-performance Embedded Workshop でビルドしていない場合は、スタック情報ファイル作成ツール "gensni"を使用して生成します。"gensni"は、Windowsのコマンドプロンプトから起動します。"gensni"はアブソリュートモジュールファイル(*.x30)を入力とし、スタック情報ファイルを出力します。

【例】アブソリュートモジュールファイル test.x30 からスタック情報ファイル test.sni を作成する場合

```
>gensni test.x30
```

- 【注】
1. M16Cファミリ, またはR8Cファミリ用コンパイラの場合、プロファイル情報ファイルを使用できません。
 2. Call Walker の使用できる各製品のバージョンは次のとおりです。
 - M16Cシリーズ, R8Cファミリ用Cコンパイラパッケージ[M3T-NC30WA]: V.5.42 Release 00以降
 - M32Cシリーズ用Cコンパイラパッケージ[M3T-NC308WA]: V.5.41 Release 00以降
 - R32Cシリーズ用Cコンパイラパッケージ: 全てのバージョンで使用可能

2. Call Walkerの起動

以下のいずれかの方法で起動します。

スタートメニューから起動

プログラム -> [Renesas] -> [High-performance Embedded Workshop] -> [Call Walker]をクリック

High-performance Embedded Workshop から起動

[ツール] -> [Renesas Call Walker]をクリック

【注】 M16C ファミリ, または R8C ファミリ用コンパイラの場合、スタートメニューは以下のように表示します。

プログラム -> [Renesas] -> パッケージ名 -> [Call Walker]

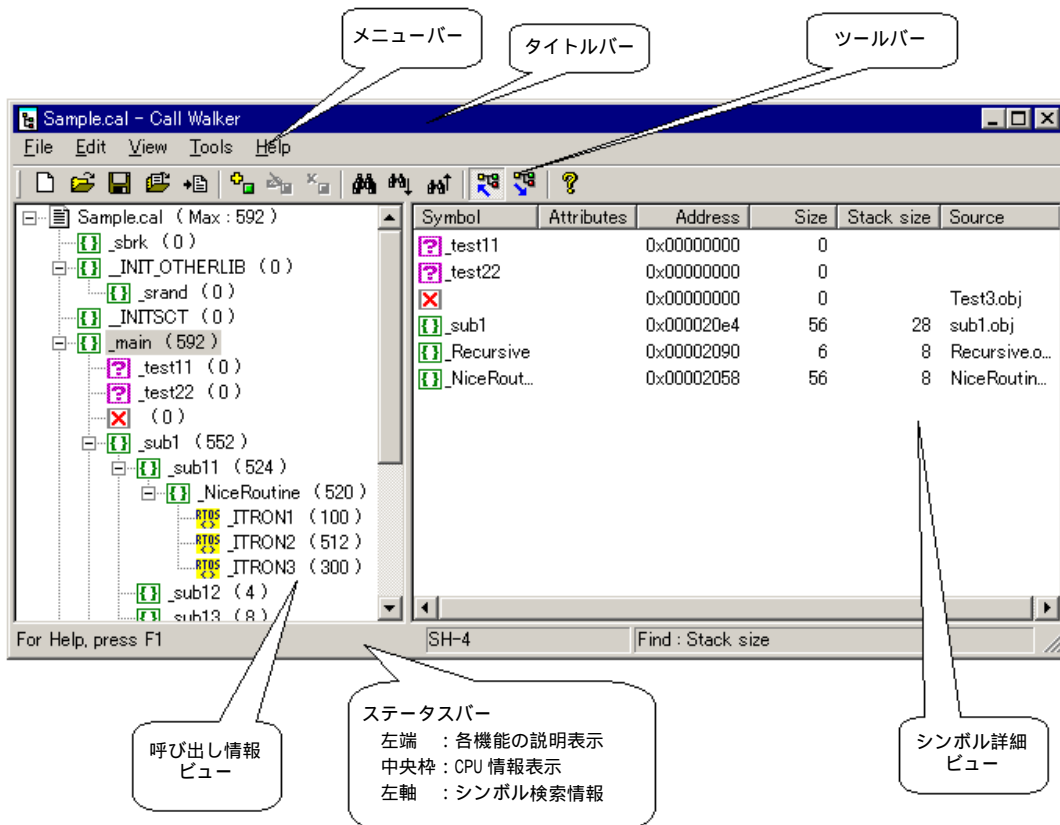
3. ファイルのオープンとCall Walkerの画面

Call Walker を起動後、スタック情報ファイル(*.sni)またはプロファイル情報ファイル(*.pro)を[File] -> [Import Stack File...]でオープンします。

[File] -> [Open...]は既存の編集ファイル(*.cal)をオープンする際に、使用します。

オープンすると以下の画面が出力されます。

【注】 SuperH ファミリ, または H8 ファミリ用コンパイラの場合、標準ライブラリ以外のアセンブラ関数はスタック使用量が0と表示されます。「4. スタック情報の編集」をご参照の上、スタック使用量を設定してください。



呼び出し情報ビュー

シンボル間のリンク階層構造を表示します。


各シンボル名の右側に使用しているスタック使用量を表示しています。


(1) シンボルの区分表示

シンボル名の左側に、シンボルの種類をアイコンで表示しています。

次の種類があります。

 編集中のファイル

 アセンブラのシンボル

 C/C++の関数

 再帰呼び出し(リカーシブコール)関数または循環関数

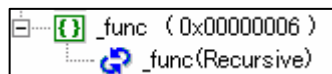
(a) 再帰呼び出し(リカーシブコール)

関数内で自関数を呼び出す場合に表示されます。

【例】

```
void func(int x)
{
    x++;
    if(x != OFF)
        func(x);

    if(x == MAX)
        return;
}
```

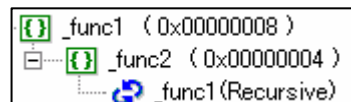


(b) 循環関数

間接的に自関数を呼び出す場合に表示されます。

【例】

```
void func1(int a)
{
    func2(10);
}
void func2(int b)
{
    func1(9);
}
```



 RTOS 関数 (ITRON などのシンボル)

参照元シンボル不明関数

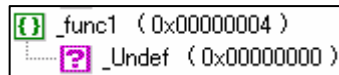
下記例の場合、関数 func1()で関数 Undef()を呼び出していますが、関数 Undef()の実体がない場合、関数 Undef()に本アイコンが表示されます。

実際には実体のない関数呼び出しは、リンク時にエラーになりますが、リンクオプション change_message を使用することにより、エラーをウォーニングに変更できます。

ウォーニングにすると、ロードモジュールが作成できるので、スタック情報ファイルも作成されます。

【例】

```
void func1(void)
{
    Undef();
}
```

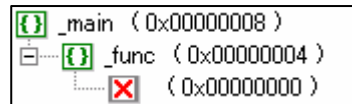


アドレス参照未解決関数

下記のように、関数をテーブル呼び出ししている場合に表示されます。

【例】

```
static int (*key[3])()=
{nop, stop, play};
void func(int x)
{
    (*key[a])();
}
```



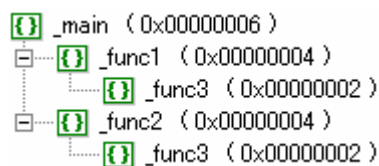
省略表示シンボル

本ツールでは、リンク階層を全て表示するため、ユーザアプリケーションが大きい場合、表示量が膨大になります。そこで、各シンボルの階層表示について最初の1つだけを表示し、省略表示シンボルで他の同様な部分は表示を省略します。

この表示は、[View] -> [Show All Symbols / Show Simple Symbols]で切り替えることができます。

【例】

Show All Symbols



Show Simple Symbols



RTOS ハンドラ

RTOS タスク

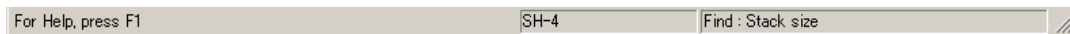
可変個引数を持つ関数

シンボル詳細ビュー

Symbol	Attri...	Address	Size	Stack size	Source
_INT_TxI1...	I	0x000004...	0x00000002	0x00000004	intprg.obj
_abort		0x000008...	0x00000002	0x00000004	CallWalker2...
_sbrk		0x000008...	0x0000002c	0x00000008	sbrk.obj
_sub		0x000008...	0x00000002	0x00000004	CallWalker2...
_nop		0x000008...	0x00000002	0x00000004	CallWalker2...
_PowerON...		0x000004...	0x00000016	0x00000004	resetprg.obj
_play		0x000008...	0x00000002	0x00000004	CallWalker2...
_stop		0x000008...	0x00000002	0x00000004	CallWalker2...
_INT_TG11...	I	0x000004...	0x00000002	0x00000004	intprg.obj
_INT_TG10...	I	0x000004...	0x00000002	0x00000004	intprg.obj
INT_TG11	I	0x00000004	0x00000002	0x00000004	intprg.obj

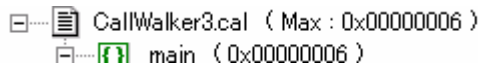
各シンボルごとのアドレス、属性、スタック使用量を表示します。
シンボルをクリック後、右クリックすると各編集コマンドを実行できます。

ステータスバー



現在オープンしているスタック情報ファイルを作成した際の、CPU 種別などを表示します。

最大スタック使用量



現在オープンしているスタック情報ファイルの、静的な最大スタック使用量を表示します。

標準ライブラリバージョン選択(SuperH ファミリ, または H8 ファミリ用コンパイラ向け)



現在オープンしているスタック情報ファイルを作成した際の、標準ライブラリバージョンを選択します。
これにより、標準ライブラリ内アセンブラ関数のスタック使用量を表示します。

【注】 M16C ファミリ, R8C ファミリ, または RX ファミリ用コンパイラの場合、標準ライブラリバージョンを選択する必要はありません。何が選択されていても影響ありません。

4. スタック情報の編集

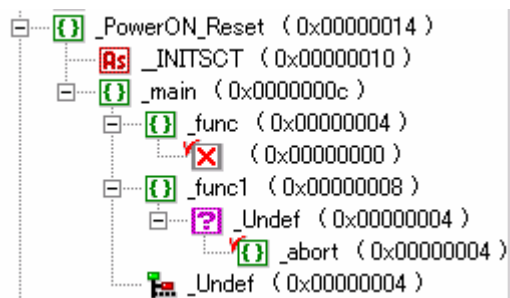
ファイルをオープン中、右側のシンボル詳細ビューでシンボル名を選択し、EditメニューのAdd...、Modify...、またはDelete...コマンドで、シンボルの追加、変更、または削除ができます。

シンボル詳細ビューで右クリックしても、同様の操作が行えます。

本ツールは静的なスタック最大使用量を算出できますが、多重割り込みなどによる、動的な最大使用量を調査するためには、ユーザ側で情報ファイルを編集する必要があります。

また、左側の呼び出し情報ビューでシンボルをドラッグ&ドロップすると、シンボルの位置を変更できます。

シンボルの移動や編集をすると、以下のように左側の呼び出し情報ビューの該当シンボルに変更を示すチェックが付きます。

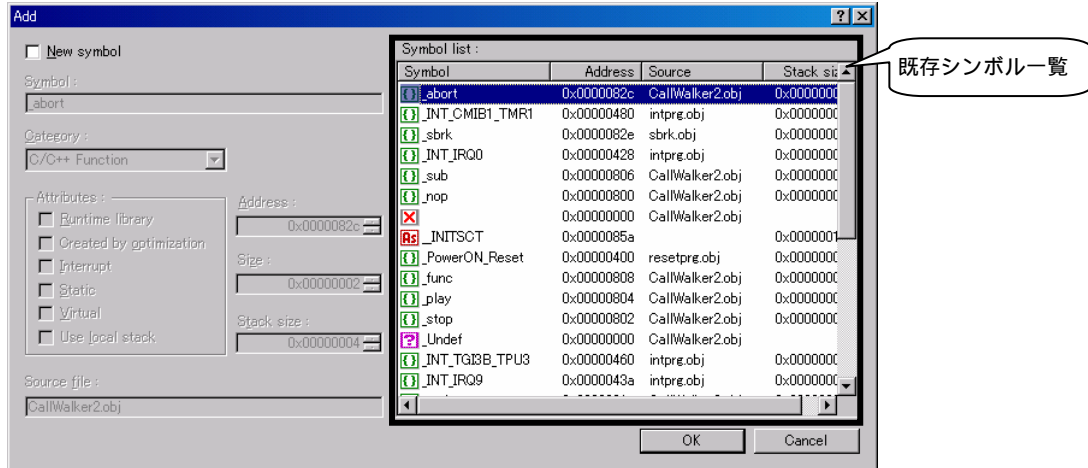


以降に、各コマンドの説明をします。

Add...コマンド

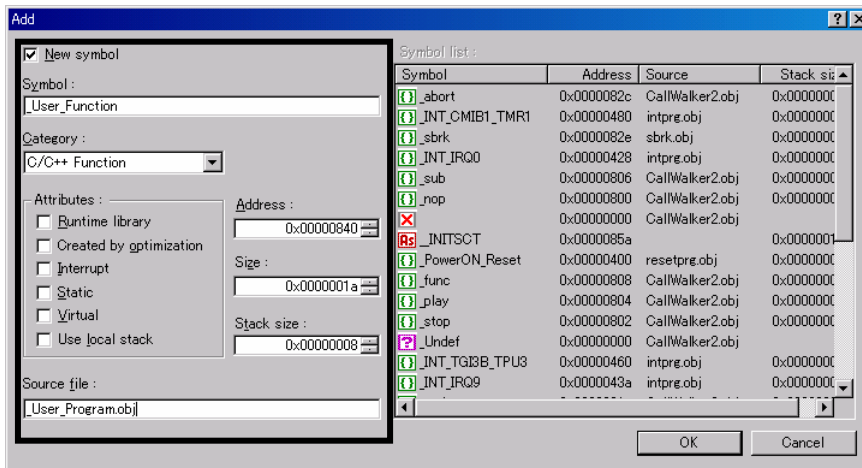
(1) 既存シンボルの追加

Add...コマンドをクリックすると、以下のダイアログボックスが表示されます。右側の一覧は現在のファイル内のシンボルです。既存のシンボルを追加する場合は、このリストからシンボルを選択しOKボタンをクリックします。



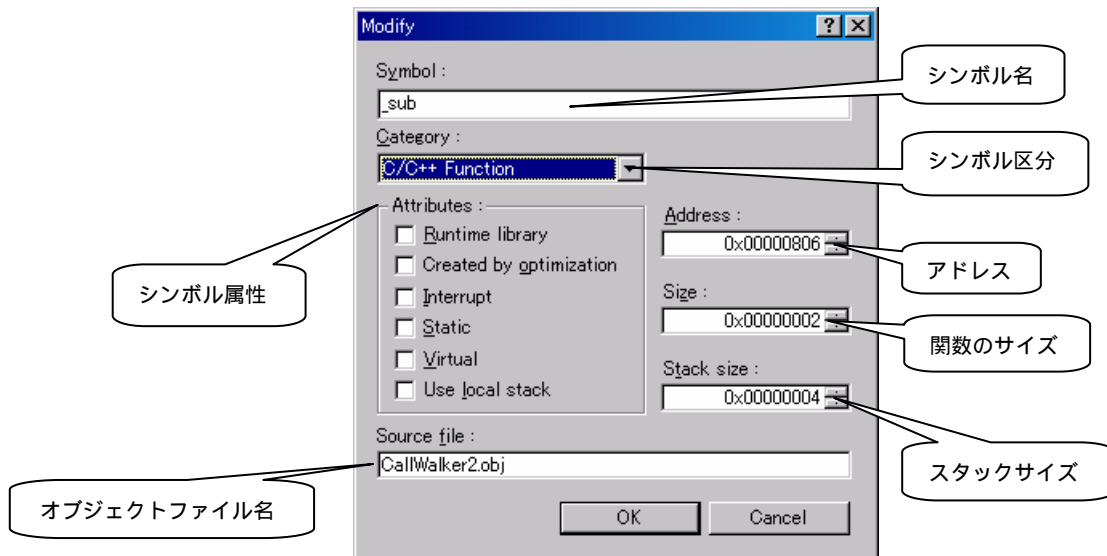
(2) 新規シンボルの追加

左側のチェックボックスをチェックすると、シンボルを新規作成できます。その際に、シンボル名、シンボル区分、属性、アドレス、スタック使用量などを決定できます。



Modify...コマンド

情報を変更したいシンボルを選択し、Modify...コマンドをクリックすると、以下のダイアログボックスが表示されます。ここで各種情報を変更することができます。



Delete...コマンド

スタック使用量調査に不要なシンボルを選択し(左側、右側どちらでも良い)、Delete...コマンドをクリックするとシンボルを削除できます。

5. アセンブラプログラムのスタック使用量

本章は、SuperH ファミリ, H8 ファミリ, および RX ファミリ用コンパイラ向けの内容です。M16C ファミリ, または R8C ファミリ用コンパイラには該当しません。

C/C++ プログラムと違い、アセンブラプログラムは、アセンブルをしてもスタック使用量を自動で算出することができません。

そのため、Call Walker 上でアセンブラプログラム関数のスタック使用量を編集する必要がありました。

しかし、.STACK 制御命令を使用すると、アセンブラプログラム関数内にスタック使用量を記述することができます。Call Walker は.STACK 制御の数値を画面に表示します。

.STACK 制御命令説明

シンボルに対して、Call Walker で参照するスタック使用量を定義します。1つのシンボルに対して定義できるスタック値は1度のみ有効です。2度以上指定した場合は、その定義を無効とします。また、指定できるスタック値は、H'00000000 ~ H'FFFFFFFE の範囲の2の倍数のみとし、それ以外を指定した場合はその定義を無効とします。

- ・ 定数値を指定する。
 かつ
- ・ 前方参照シンボル外部参照シンボル、相対アドレスを使わずに指定する。

.STACK アセンブラ制御命令記述方法

```
.STACK <シンボル>=<スタック値>
```

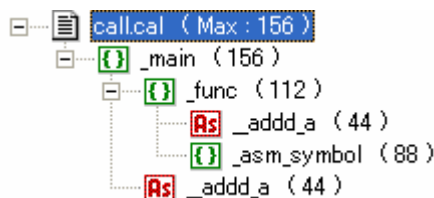
アセンブラプログラム例

```
.CPU          SH1
.EXPORT      _asm_symbol
.SECTION     P, CODE, ALIGN=4
_asm_symbol:
    .STACK    _asm_symbol=88
    :
RTS
.END
```

_asm_symbol 関数のスタックサイズを「88」に設定

Call Walker の表示例

下記のように、Call Walker 上で _asm_symbol 関数のスタック使用量が、「88」と表示されるようになります。



備考

- (1) .STACK 制御命令は Call Walker にスタックサイズを表示させる機能です。プログラムの動作に影響を与えないでください。
- (2) 本制御命令は、SuperH RISC engine アセンブラでは Ver.7 から、H8S,H8/300 シリーズアセンブラでは Ver.6.01 から、RX ファミリアセンブラでは V.1.00 Release 00 からサポートしています。

6. スタック情報のマージ(連結)

保存後、または編集時のスタック情報ファイルと、他のスタック情報ファイルをマージ(連結)することができます。

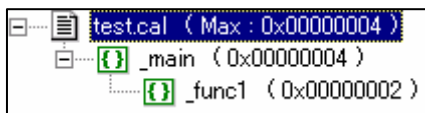
これにより、編集したスタック情報が、再ビルド後のスタック情報に上書きされません。

マージの例

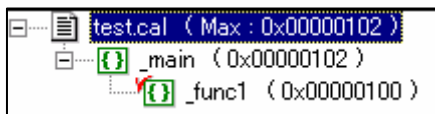
- (1) test.c の内容

```
void main(void)
{
    func1();
}
```

- (2) Call Walker でスタック情報ファイルをオープン



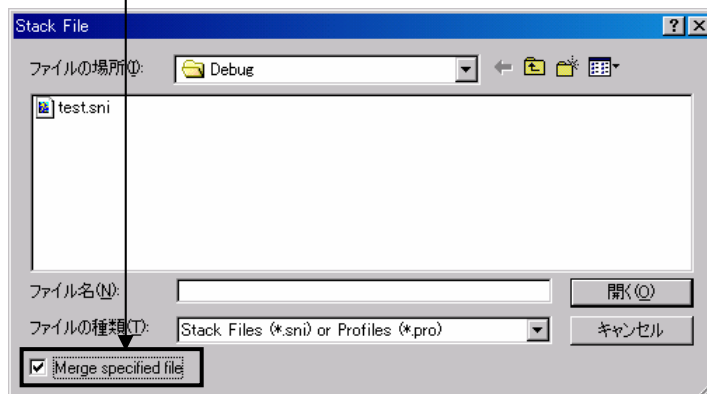
- (3) 内容を変更(func1 のスタック使用量を 100 に変更)



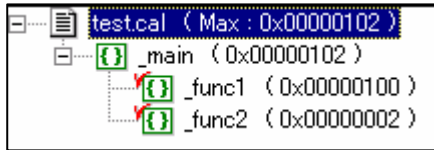
- (4) test.c の内容を変更後にビルド(func2 の呼び出しを追加)

```
void main(void)
{
    func1();
    func2();
}
```

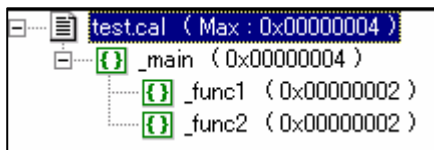
- (5) Call Walker で test.cal を開いたまま test.sni をオープンします。
このとき、**ここ**をチェックし、開くボタンを押します。



- (6) すると、(3)で変更した func1 のスタック使用量を残しつつ、func2 の情報が追加になります。
これがスタック情報のマージ(連結)です。



もし、(5)でチェックをしない場合は、以下のように(3)で変更した func1 のスタック使用量は変更前の値に戻ってしまいます。

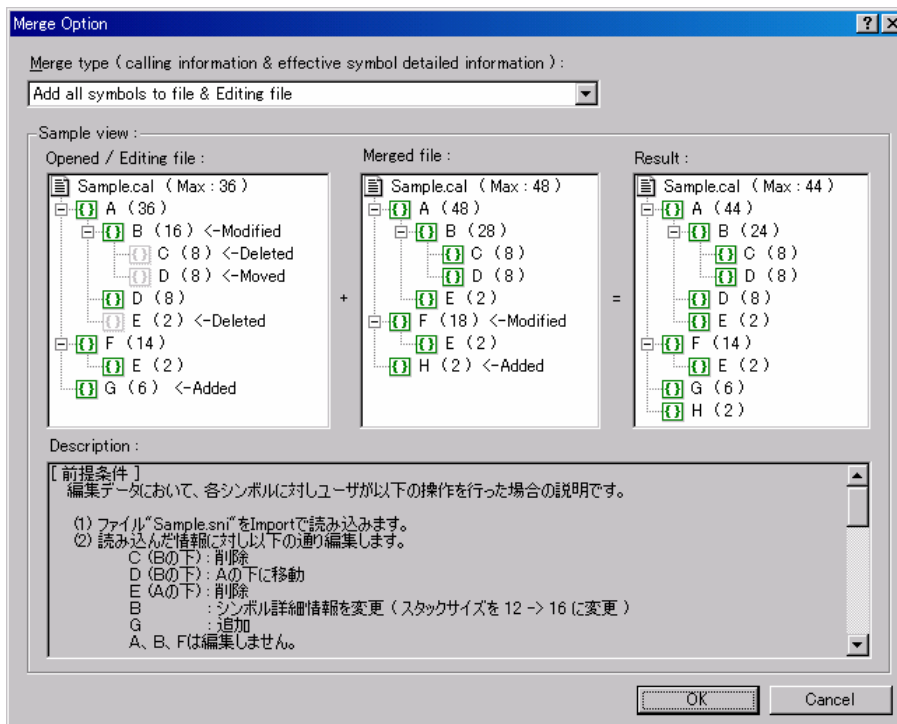


マージの詳細オプション

マージをする際の方法を変更することもできます。5 通りの方法があります。
詳細なマージ方法は、本ダイアログボックスの Description を参照してください。

【指定方法】

[Tools] -> [Merge Option...]



備考

マージ機能は Call Walker のバージョン 1.3 以上で使用できます。

7. その他の機能

リスト出力

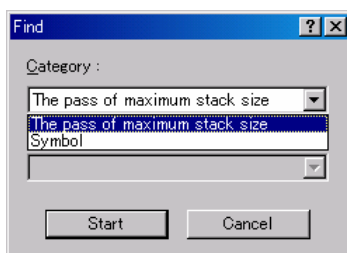
スタック情報をテキスト形式で、ファイルに出力することができます。

【指定方法】

[File] -> [Output List...]

検索機能

以下のダイアログボックスで、2通りの呼び出し情報ビューからの検索ができます。



- (1) スタック使用量が最大のパスを検索
- (2) シンボル名の検索

【指定方法】

[Edit] -> [Find...]

[Edit] -> [Find Next...] (次を検索)

[Edit] -> [Find Previous...] (前を検索)

呼び出し情報ビューの表示形式設定

以下のコマンドで、スタック使用量の表示について2通りの方法を選択できます。

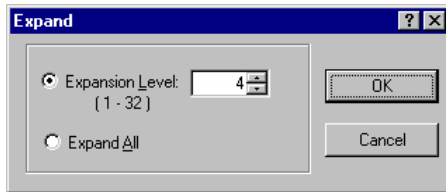
- (1) Show Required Stack
下から上にスタック使用量が積み重なっていきます。
- (2) Show Used Stack
上から下にスタック使用量が積み重なっていきます。

【指定方法】

[View] -> [Show Required Stack] または [Show Used Stack]

呼び出し情報ビューのツリー表示を展開するレベルの設定機能

以下のダイアログボックスで、選択したシンボルのツリー表示を展開するレベルについて 2 通りの方法を選択できます。



(1) Expansion Level

選択したシンボルのツリー表示を展開するレベルを指定します。

(2) Expand All

選択したシンボルのツリー表示をすべて展開します。

【指定方法】

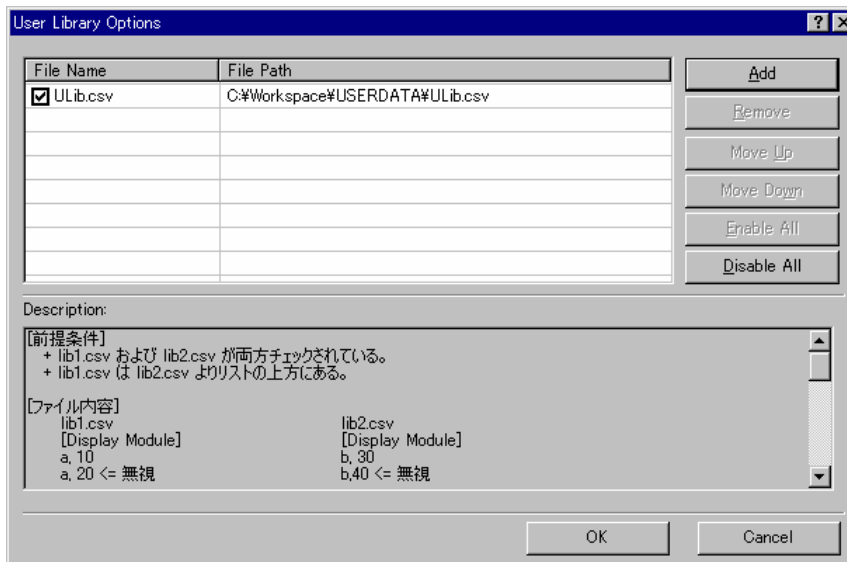
[View] -> [Expand...]

【備考】

呼び出し情報ビューのツリー表示を展開するレベルの設定機能は、Call Walker のバージョン 2.00 以上で使用できます。

ユーザライブラリのスタック使用量の設定機能

表示しているシンボルのスタックサイズを指定したスタック情報ファイルから読み込んだスタックサイズに置き換え、表示内容を更新します。以下のダイアログボックスで、ユーザライブラリのスタック情報ファイルを指定し、スタック使用量を設定できます。



【指定方法】

[Tool] -> [User Library Options...]

【備考】

ユーザライブラリのスタック使用量の設定機能は、Call Walker のバージョン 2.00 以上で使用できます。

ホームページとサポート窓口

ルネサステクノロジホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2009.01.23	-	初版発行
2.00	2009.10.01	1, 2, 6, 7, 11, 15	RX ファミリの追加
			Call Walker バージョン 2.00 の機能追加

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事務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。
 - 1 1. 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。
 - 1 2. 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。
 - 1 3. 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。

D039444