

# ユーザ定義拡張ツールでの メイク・ファイルのインポート

(CSIDE Ver.4 用資料)

第3版 2004年12月27日

## ◎資料内容◎

本資料は CSIDE の IDE 環境へ移行しようと考えていらっしゃる方や CSIDE の IDE 環境を使いたいが今までのワークスペースが使えなくなるのは困るという方に向けて、ルネサステクノロジ製コンパイラ・パッケージ(Ver.5.1 以前は除く)以外で CSIDE が対応しているコンパイラの出力するメイク・ファイルをインポートする手順について説明します。

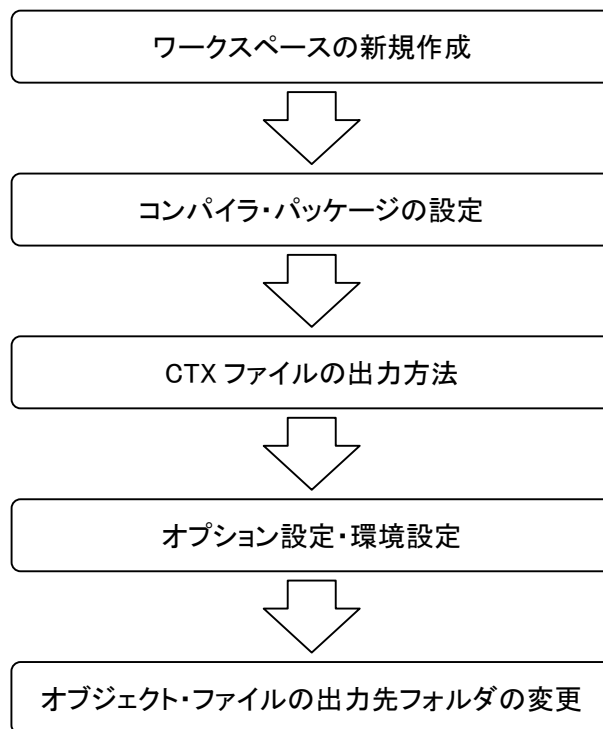
CSIDE のインポート機能を使うことでメイク・ファイルに示されているビルド環境を容易に CSIDE へ移植できます。ワークスペースを CSIDE に移植することで、ソース編集、ビルドからデバッグまで CSIDE 上で行なうことが可能となります。デバッグ中のソース編集 - ビルド - ロードがシームレスになりデバッグ作業を効率良く行なうことができます。

この資料の内容は、弊社 CSIDE for PLAMiGE SH を使い、ルネサス製 C コンパイラ ver.5.1 で作成したメイク・ファイルを例にして実際の画面と共に説明していきます。この説明で使用しますルネサス製 C コンパイラ ver.5.1 以外のコンパイラでも設定する個所は同じですのでこの資料を参考に設定ください。

※HEW(Ver.1.2)/HEW2/HEW3の出力するメイク・ファイルのインポートに関する資料は別資料になりますのでご注意ください。

※CSIDE でサポートしておりますコンパイラにつきましては、オンラインマニュアルの「対応言語について」に記載されていますのでそちらをご確認ください。

本資料の流れは以下の通りとなります。



## ◎ワークスペースの新規作成

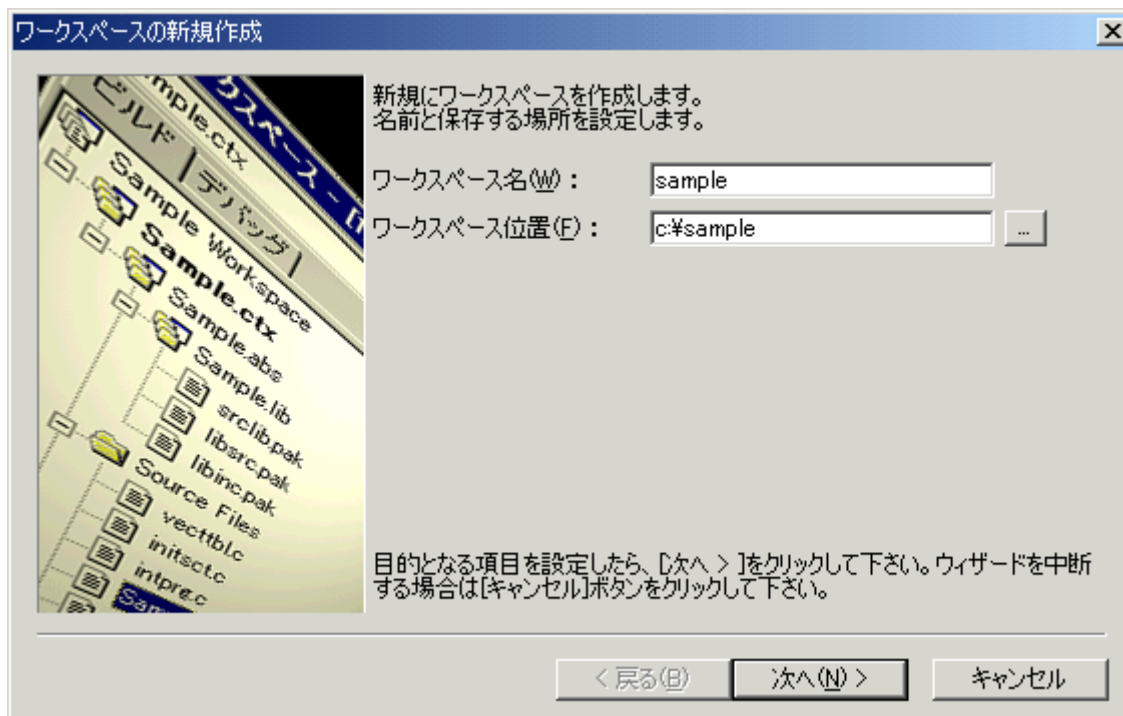
1. [ワークスペースの新規作成]を押してください。



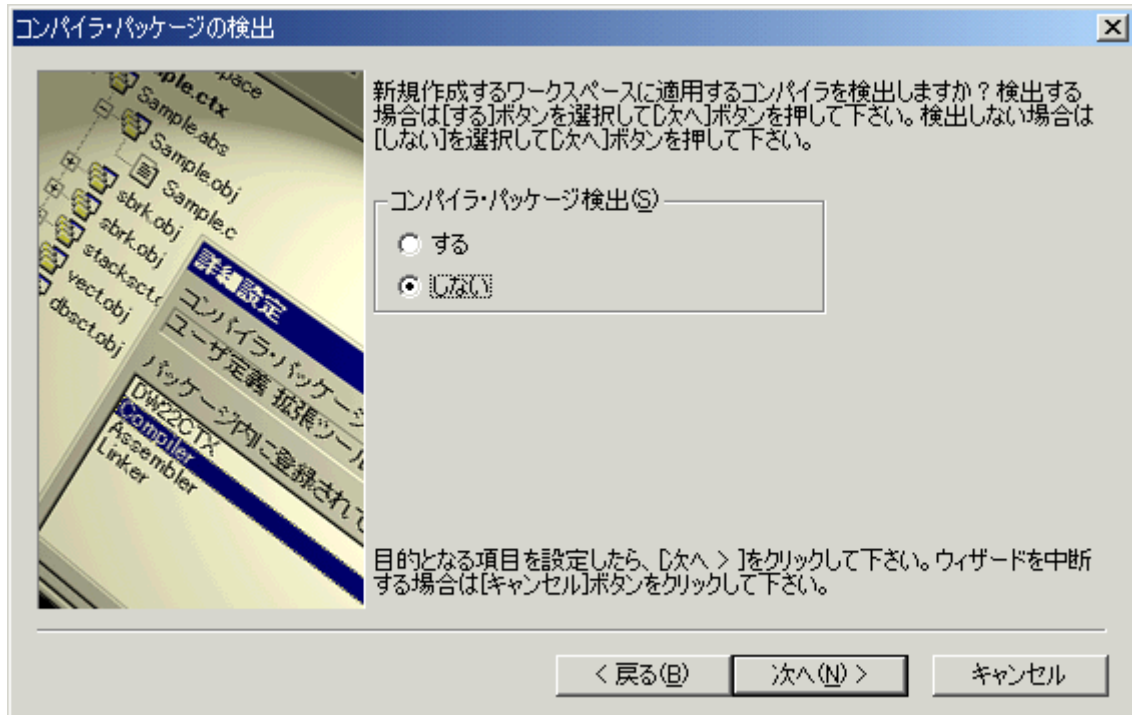
2. [ワークスペースの新規作成]ウィザードが始まります。

ここでは、ワークスペース名とダミーのワークスペース位置を指定してください。

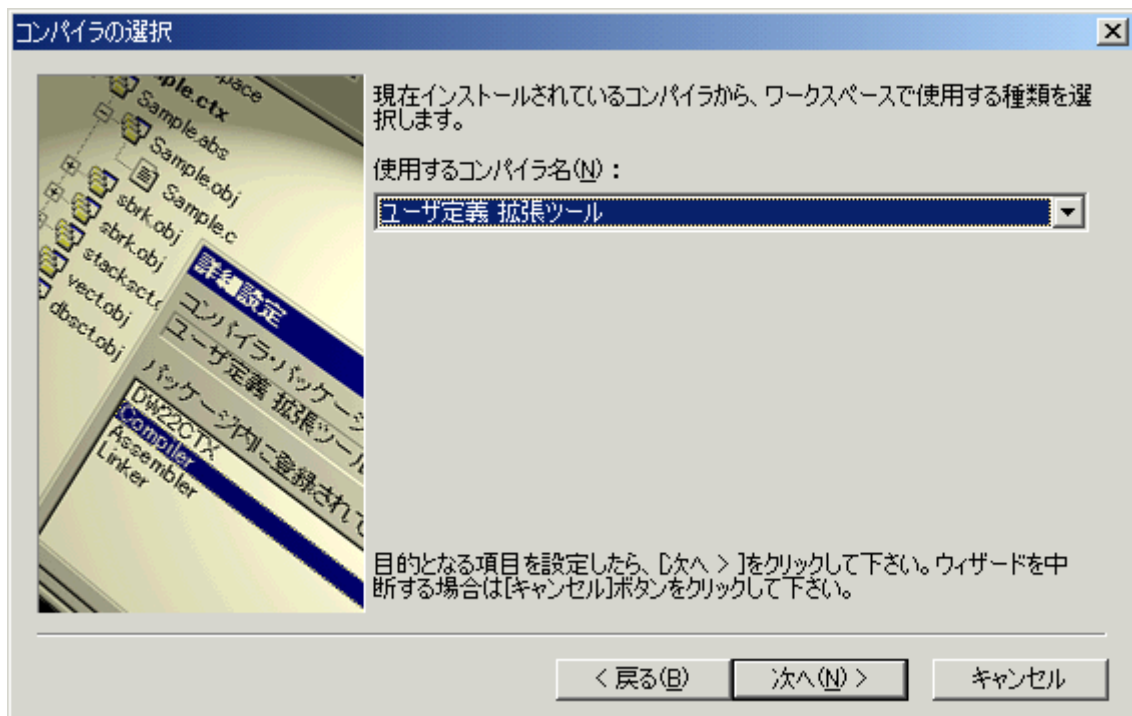
(ダミーというのは実際には使用しません。詳しくは6を御覧ください。)



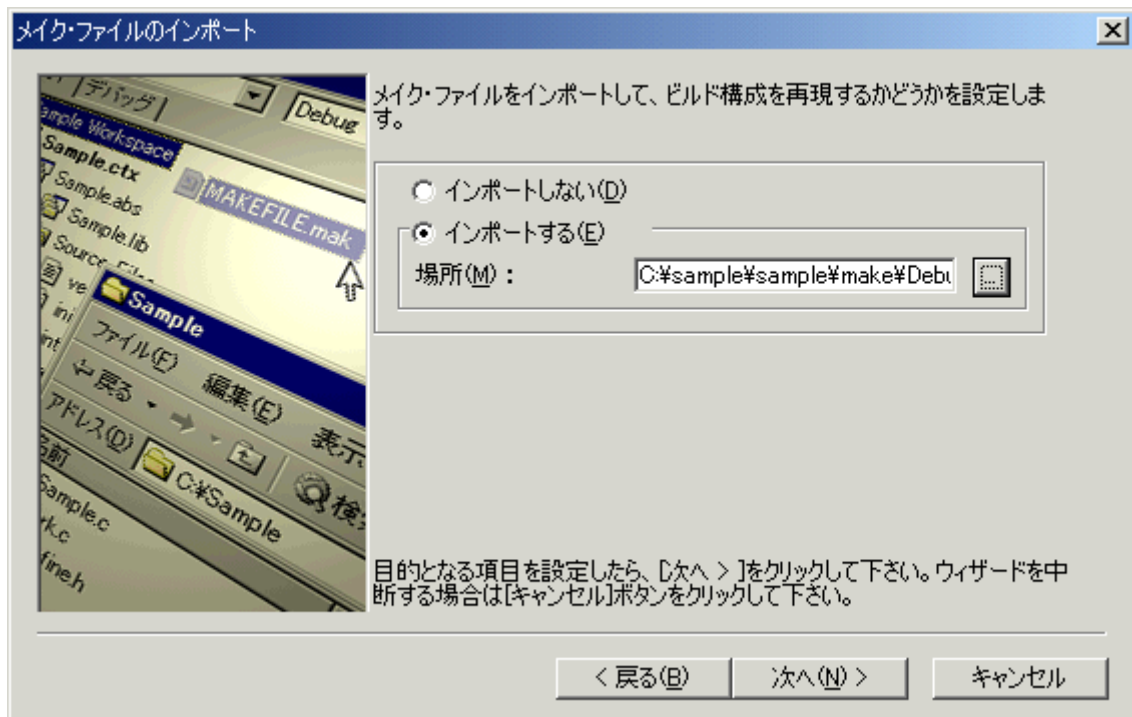
3. コンパイラ・パッケージの検出は行ないませんので[しない]を選択し[次へ]で進んでください。



4. [ユーザ定義 拡張ツール]が選択されているのを確認しそのまま[次へ]で進んでください。

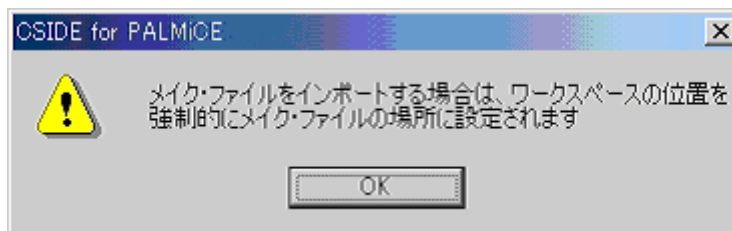


5. メイク・ファイルをインポートします。[インポートする]を選択し、メイク・ファイルのパスを指定してください。

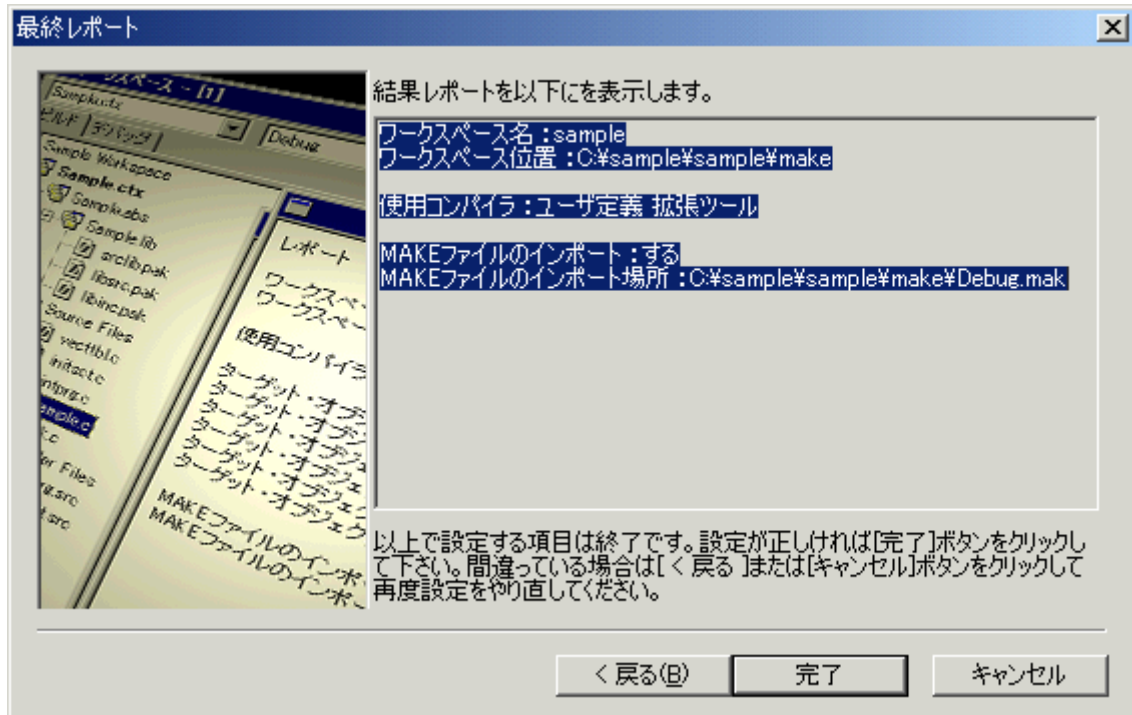


※インポートを行なうメイク・ファイルは、同じ環境で動作するメイク・ファイルでなければインポートを行っても、メイク・ファイルを元に CSIDE の設定を行ないますので動作しませんのでご注意ください。

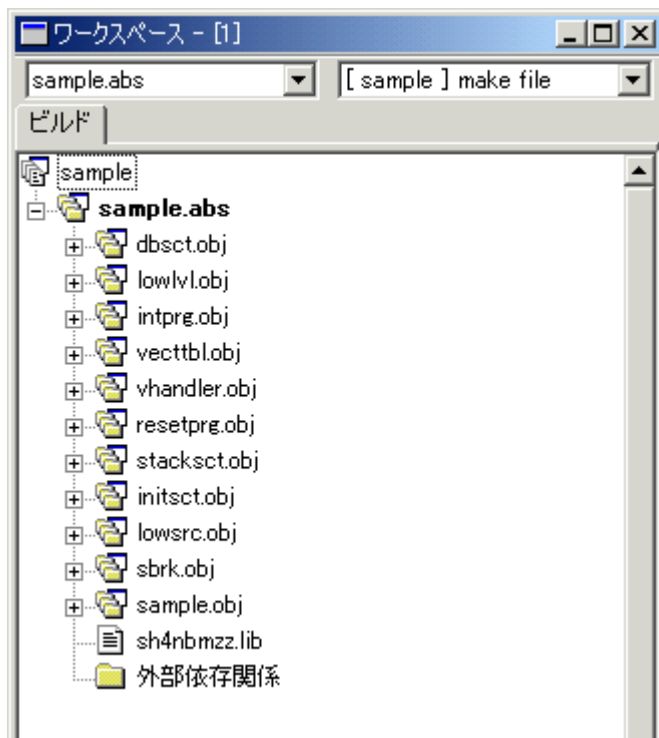
6. 上の画面で[次へ]を押すと以下のメッセージが表示されます。  
 メイク・ファイルをインポートした場合は、強制的にワークスペースの位置がメイク・ファイルの場所へと設定されます。よって2で指定したワークスペース位置は使用なくなります。



7. 最終レポートが表示されます。内容を確認して[完了]を押してください。



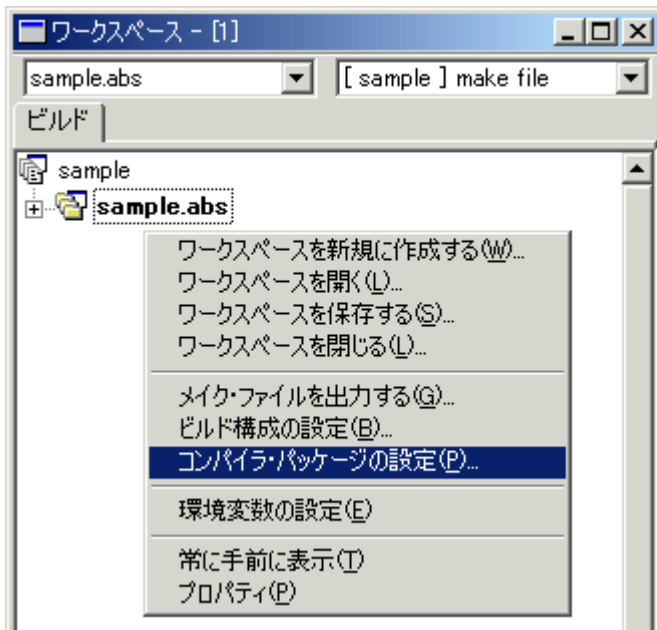
8. ワークスペースウィンドウが開きインポートされたワークスペースが開かれました。



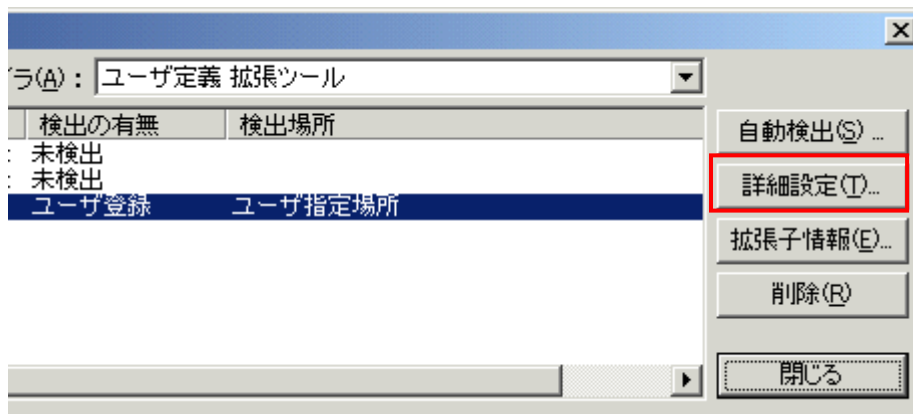
## ◎コンパイラ・パッケージの設定

コンパイラを選択でユーザ定義 拡張ツールを選択した場合、自動的にツールの名前が「拡張ツール No.XX」  
として登録されます。また、メイク・ファイルをインポートした場合は、メイク・ファイル内の拡張子情報を自動的に登  
録します。ワークスペースの拡張子情報に設定されていない拡張子は CSIDE のワークスペース上に追加するこ  
とができませんので、ここではコンパイラの設定と拡張子(ctx)の追加をおこないます。

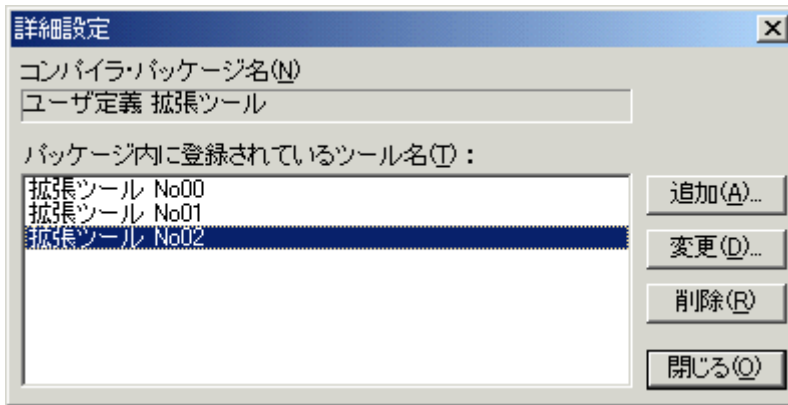
1. ワークスペースウィンドウ上でポップアップメニュー内の[コンパイラ・パッケージの設定]を選択してくだ  
さい。



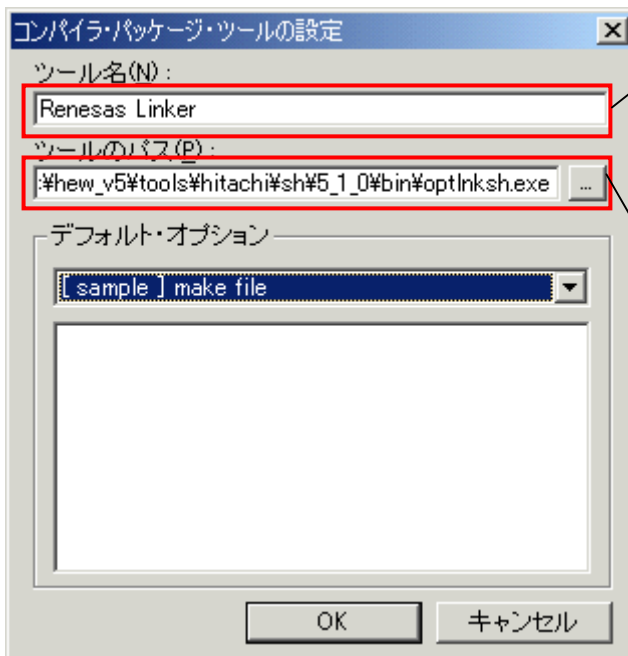
2. [コンパイラ・パッケージの設定]ダイアログボックス内の[詳細設定]を押してください。



3. 設定されているツールの一覧が表示されています。  
 どれかを選び[変更]を押してください。



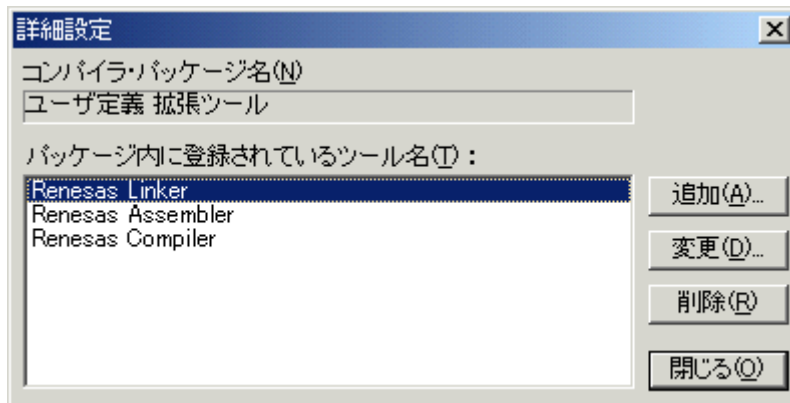
4. [コンパイラ・パッケージ・ツールの設定]ダイアログボックス内のツール名、ツールのパスを確認してください。



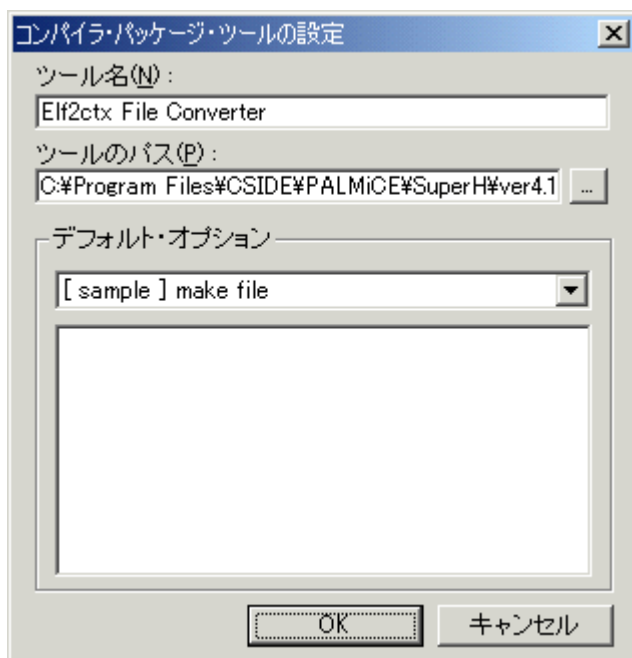
ツールの実行ファイル名から判断し、ご自身でわかりやすい名前ツール名を変更してください。

表示されています実行ファイルへのパスが、お使いのツールの実行ファイルへのパスで間違いないか確認ください。  
 パスが違っていれば、正しいパスを指定してください。

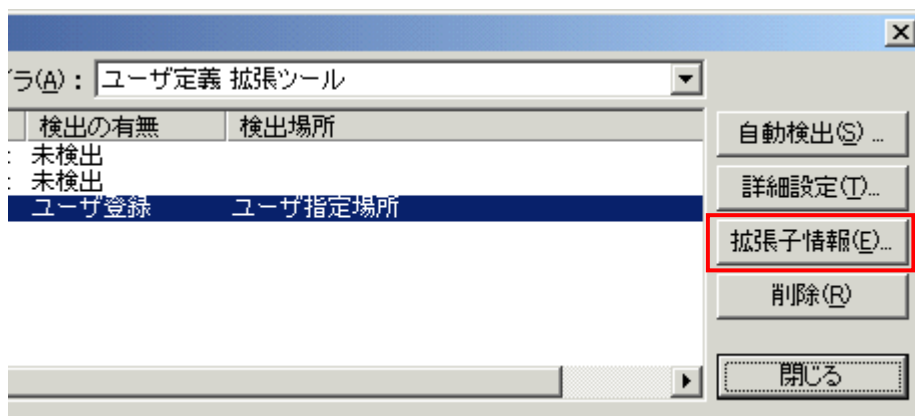
5. 全てのツール名を変更し終わったら、コンバータを追加します。[追加]を押してください。



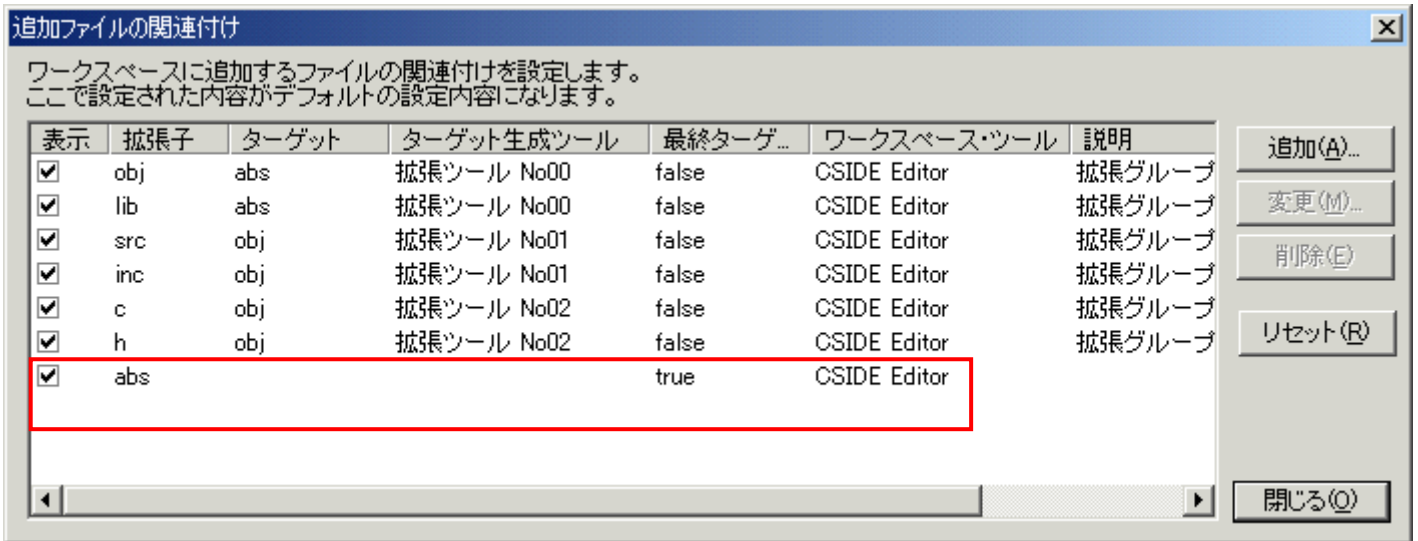
6. ツール名はわかりやすい名前をつけてください。ツールのパスには必要なコンバータを指定してください。



7. コンバータの追加ができれば[詳細設定]は閉じて[コンパイラ・パッケージの設定]に戻ります。次に[拡張子情報]を押してください。



8. 拡張子 ctx を追加します。[追加]を押してください。



①CTX とは

CTX フォーマットは、弊社独自のデバッグ情報付きオブジェクト・ファイル・フォーマットです。弊社製品でデバッグを行なう場合には、コンバータを使用し CTX 形式に変換する必要があります。ただし、例外的に CTX に変換する必要がない場合もあります。例えばルネサス C ver.5.1 以前の SYSROF 形式の ABS ファイルはそのままロードすることが可能です。対応言語、コンバートに関してはオンラインマニュアルの[対応言語について]欄に詳しい記載がございますのでそちらをごらんください。

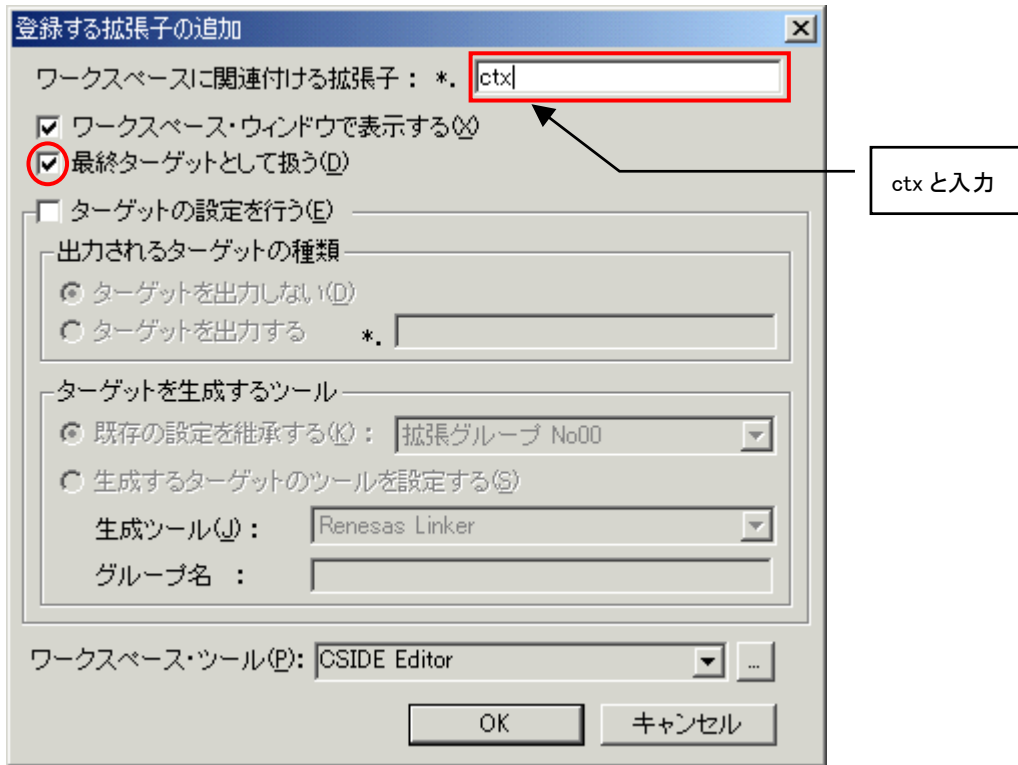
②拡張子情報について

ワークスペースのファイル関連付け設定を示します。ファイルの依存関係の設定や表示/非表示の設定を行ないます。この拡張子情報はワークスペースごとに管理されており、ここに追加されていない拡張子のファイルはワークスペース内に追加することができませんので拡張子を登録する必要があります。

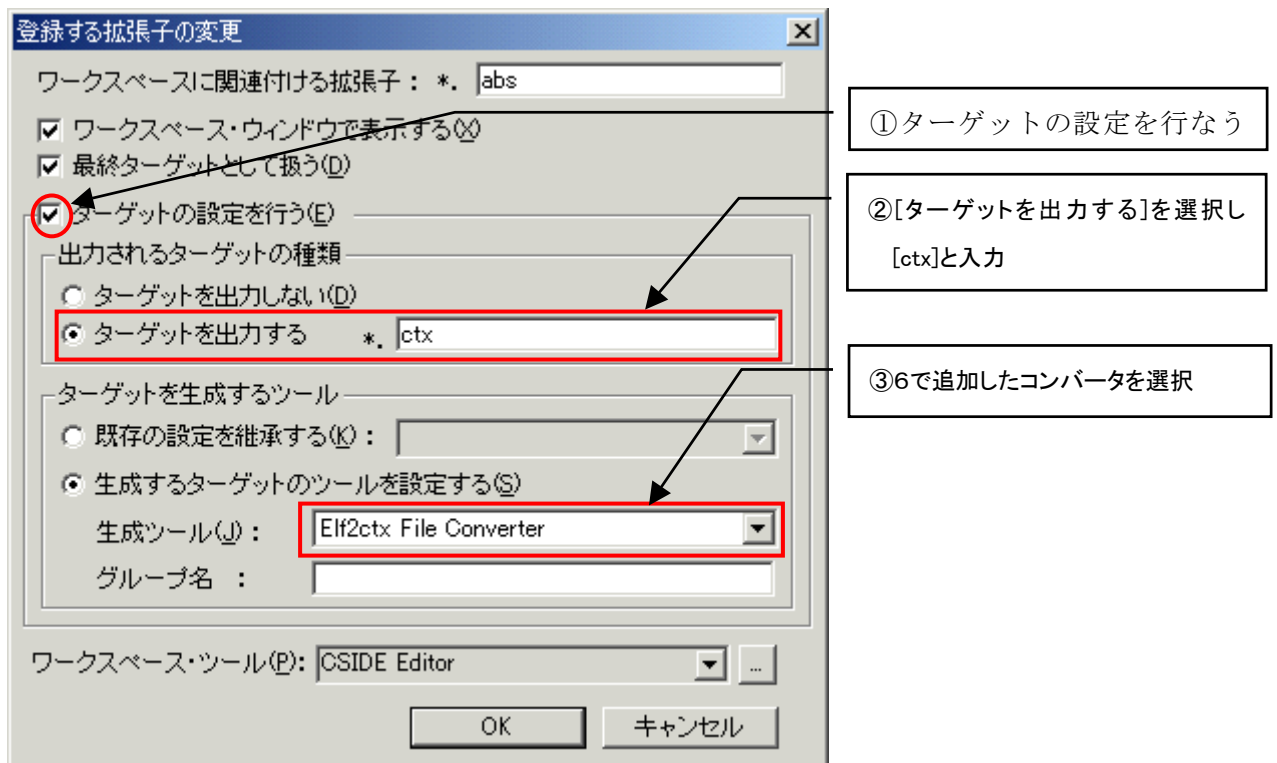
③なぜ拡張子情報に CTX を追加するのか

ユーザ定義 拡張ツールを使ったメイク・ファイルのインポートでは、メイク・ファイル内に定義されています。拡張子は拡張子情報に自動的に追加されます。インポート直後はメイク・ファイル内に ctx 形式が定義されておらずワークスペース内に ctx ファイルを追加することができませんので、自身で ctx 形式を定義する必要があります。登録する際には ctx 形式と abs 形式の依存関係、生成ツールなどの登録を行います。

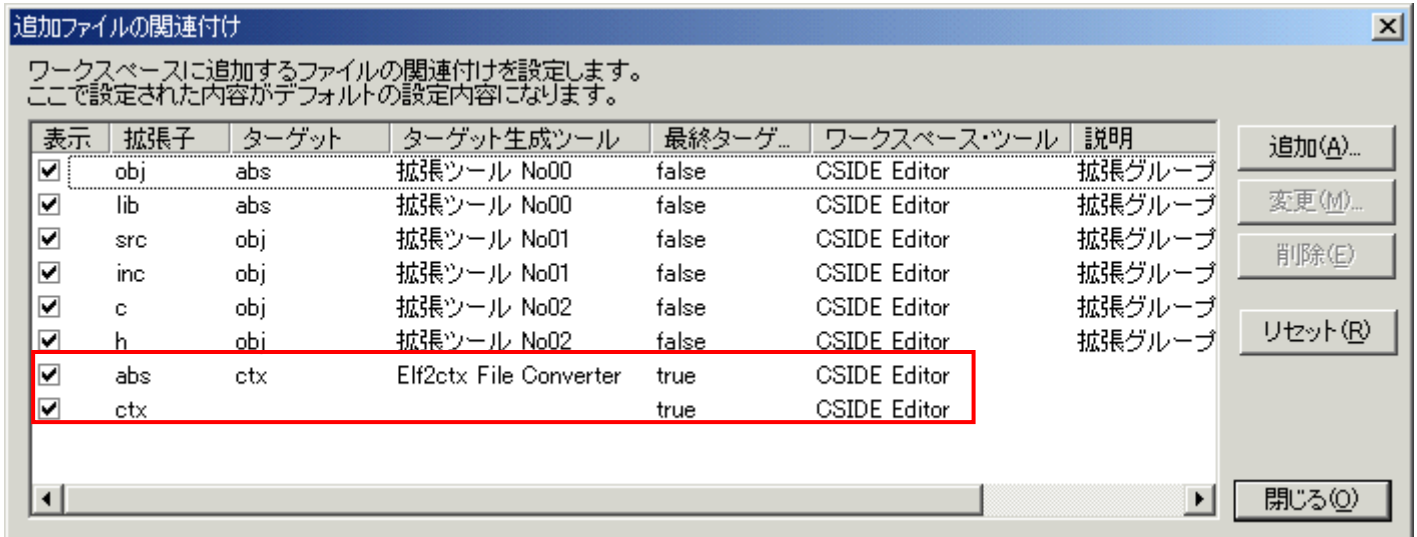
9. [ワークスペースに関連付ける拡張子]に ctx と入力し[最終ターゲットとして扱う]にチェックを入れてください。[最終ターゲットとして扱う]にチェックが入っていないとctxファイルを出力することができません。チェックを入れた後[OK]を押してください。



10. abs ファイルから ctx を出力する設定を行ないます。拡張子の abs を選択し[変更]を押し下の画面に従って設定してください。設定後[OK]を押してください。



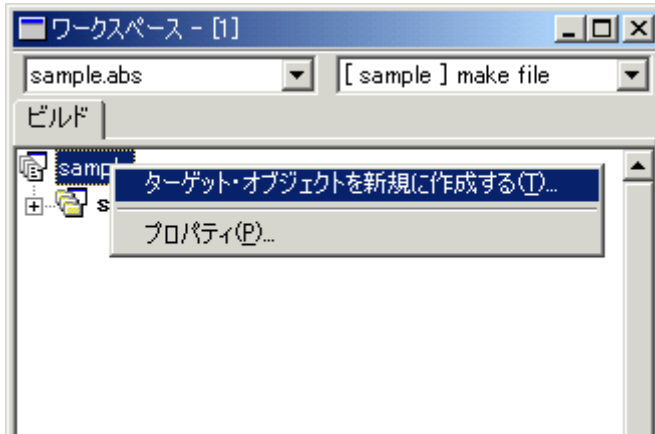
11. 最終的に[追加ファイルの関連付け]ダイアログボックスは以下の通りになります。



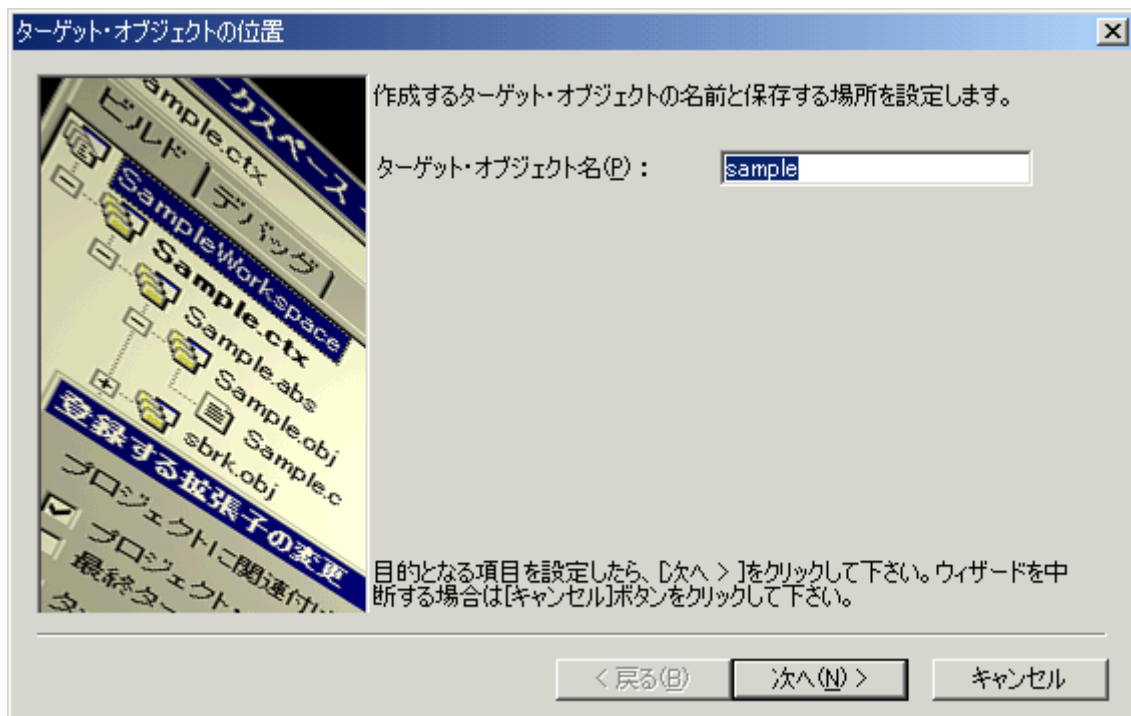
## ◎CTX ファイルの出力方法

メイク・ファイルをインポートする場合は、出力されるターゲット・オブジェクトが CTX 以外になっています。弊社デバッガを使用してデバッグを行なうには CTX 形式が出力される必要があるため下記の手順で CTX ファイルを出力するようにしてください。

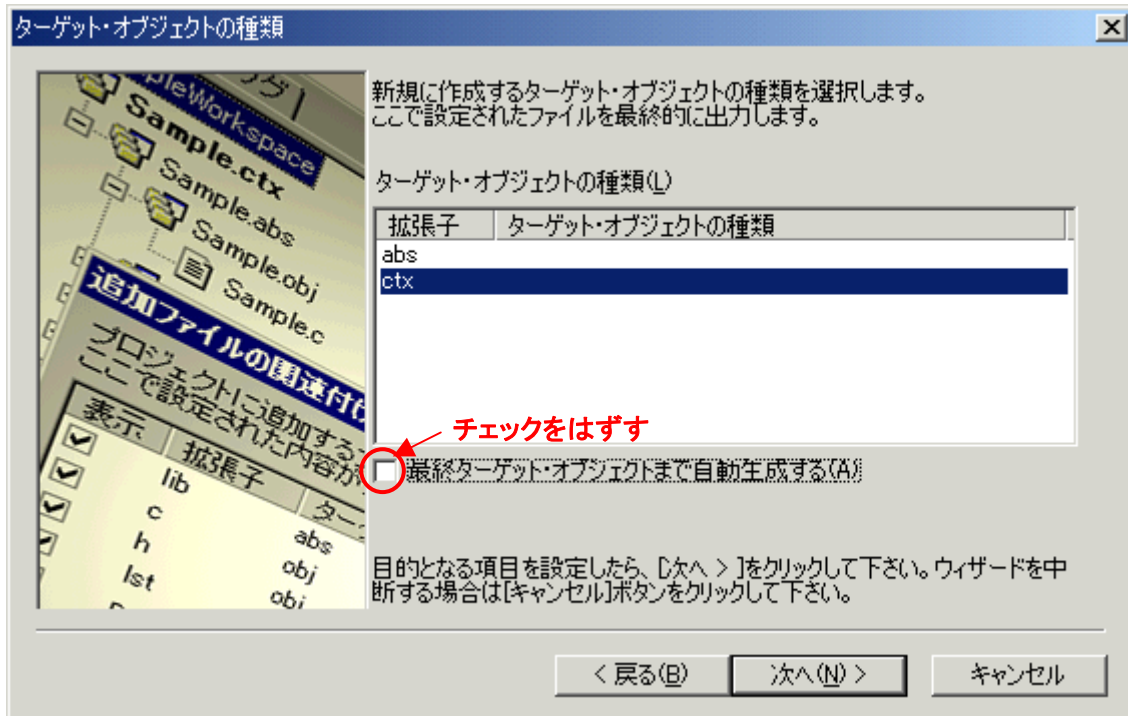
1. ワークスペースウィンドウ内のワークスペース名を選択しポップアップメニューの[ターゲット・オブジェクトを新規に作成する]を選択してください。



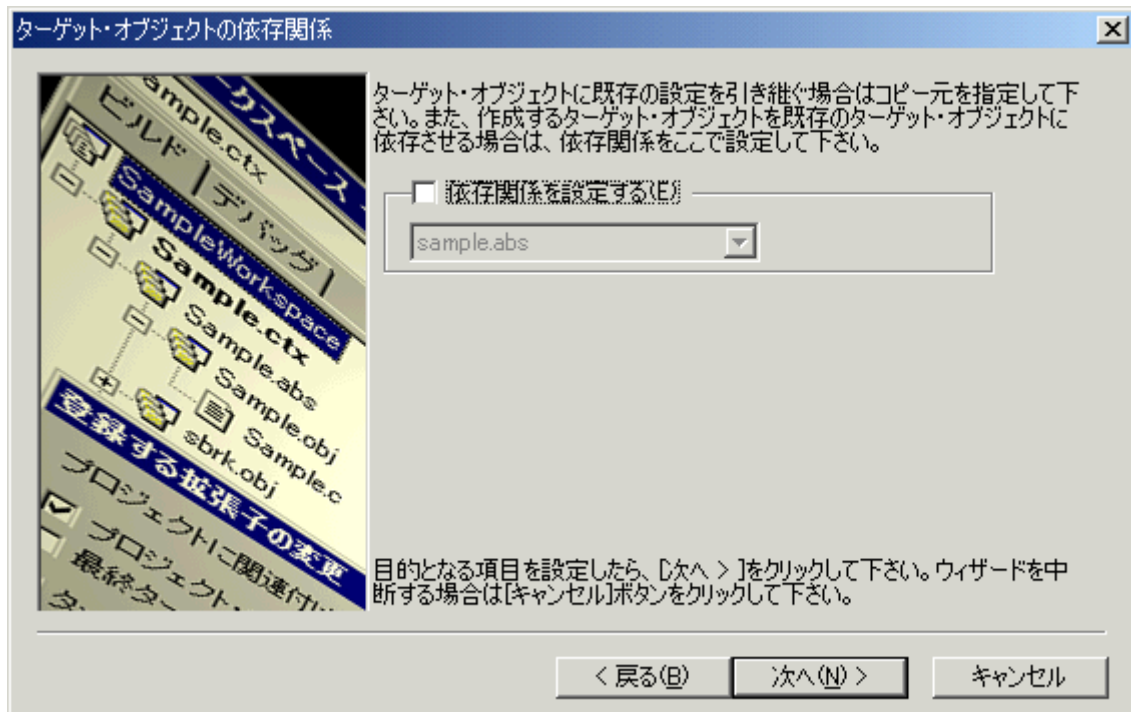
2. 最終出力するオブジェクト名(ここでは CTX ファイル名)を指定して[次へ]で進んでください。



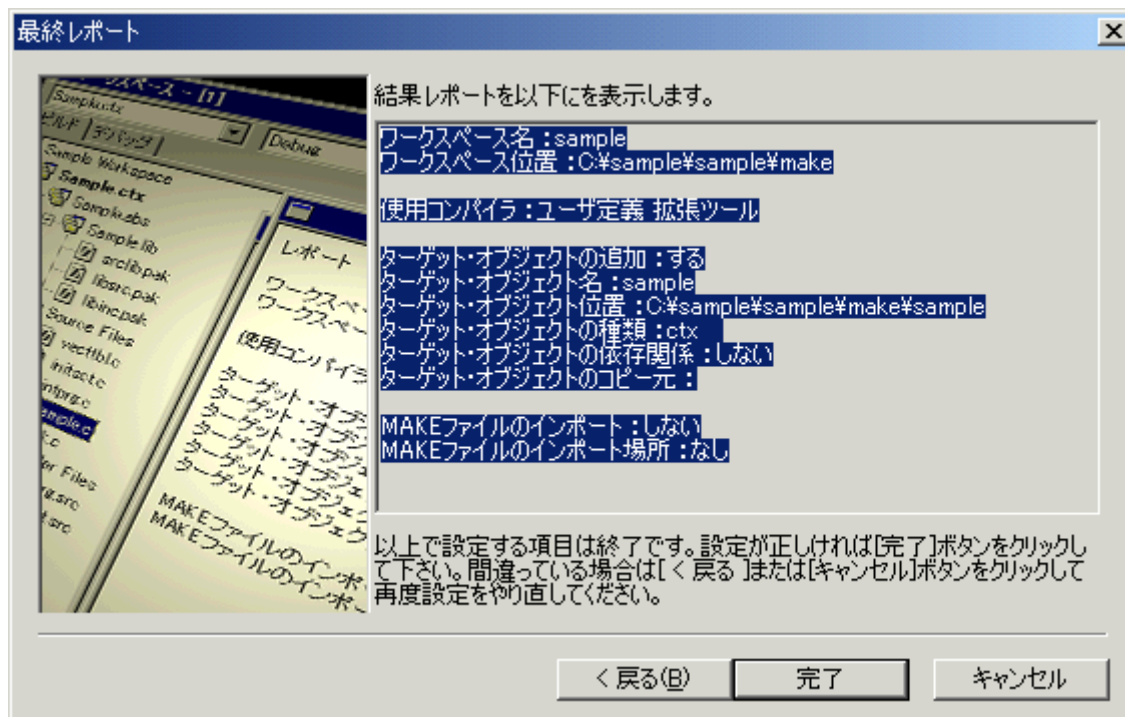
3. [最終ターゲット・オブジェクトまで自動生成する]のチェックボックスをはずしてください。  
 はずしておかないと追加する ctx に abs と lib が自動的に関連付けされます。  
 チェックが外れている、ctx が選択されているのを確認して[次へ]で進んでください。



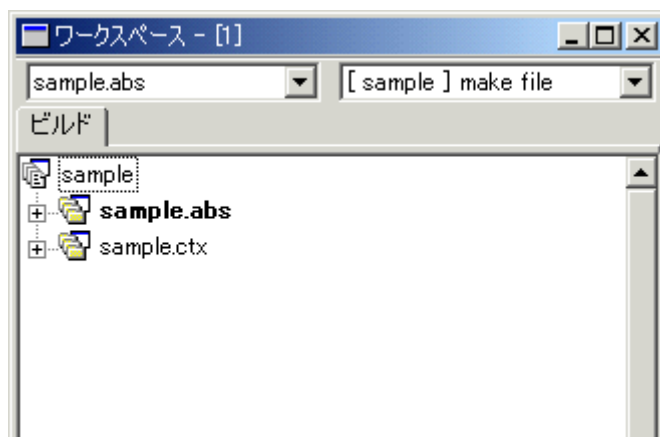
4. そのまま[次へ]で進んでください。



5. 追加に対するレポートが表示されます。内容確認後[完了]で終了してください。

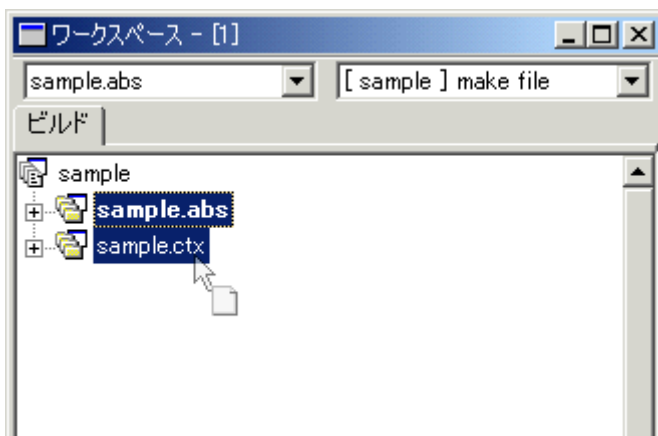


6. ウィザードの始めで指定した[オブジェクト名].ctx ファイルがワークスペース内に追加されます。

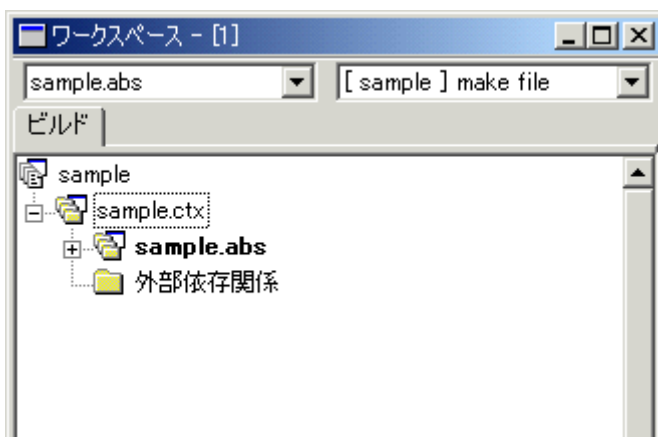


7. ctxファイルと abs ファイルの依存関係を設定します。

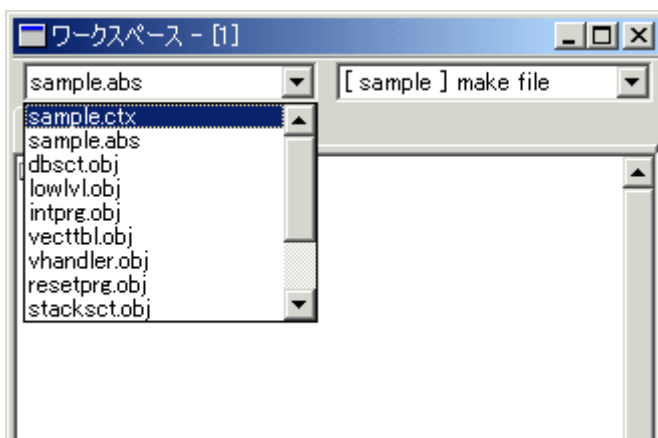
インポートされた abs ファイルを追加された ctx ファイルの上にドラッグ & ドロップします。



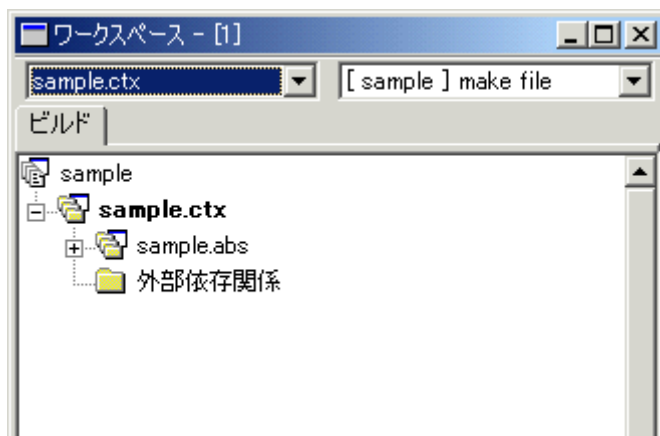
8. ctx ファイルの下に abs ファイルが移動します。これで ctx ファイルは abs ファイルを元に生成するということになります。



9. 最終出力オブジェクトを変更します。太字で表示されているオブジェクトが最終出力オブジェクト(アクティブ・ターゲット・オブジェクト)になります。右上のコンボボックスを xxxx.abs から xxxx.ctx に変更してください。

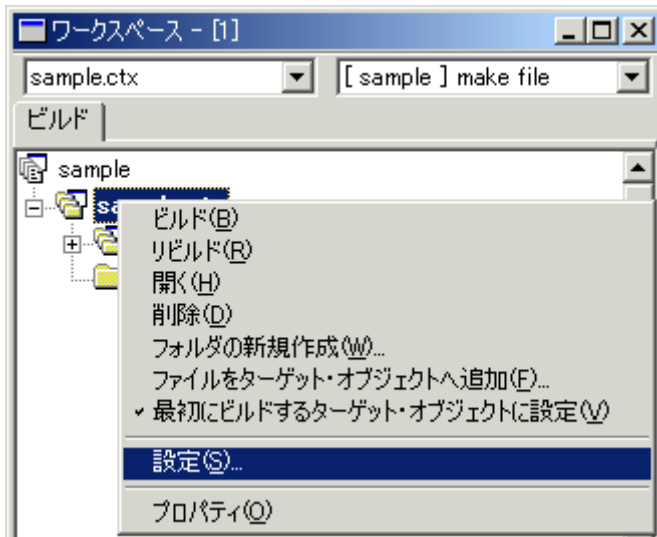


10. ctx ファイルが太字になったことで最終出力ファイルが abs ファイルから ctx ファイルへと変更されたことが確認できます。これでビルドをすれば ctx ファイルが出力されるようになります。



## ◎オプション設定・環境設定

1. オブジェクト・ファイル上でポップアップメニュー内の[設定]を選択してください。



2. オプション設定ダイアログボックスが開かれます。

メイク・ファイルの指示に従いコマンド・オプションは取り込まれていますが、更にワークスペース上での操作を効率的に行なうために、リンク・オプションの INPUT・OUTPUT はシステム変数に置き換えてください。また、必要に応じてオプションの変更を行なってください。

### ※システム変数について

CSIDE にはオプション設定ダイアログで特定のパスや名前に変換されるシステム変数というものを用意しています。例えば、入力ファイル名を指定する場合、特定のパスを登録することも可能ですが、全オプションで共通の設定にしたい時に、システム変数を入力することで、ビルド時に各ファイルのパス名に自動的に変換されます。

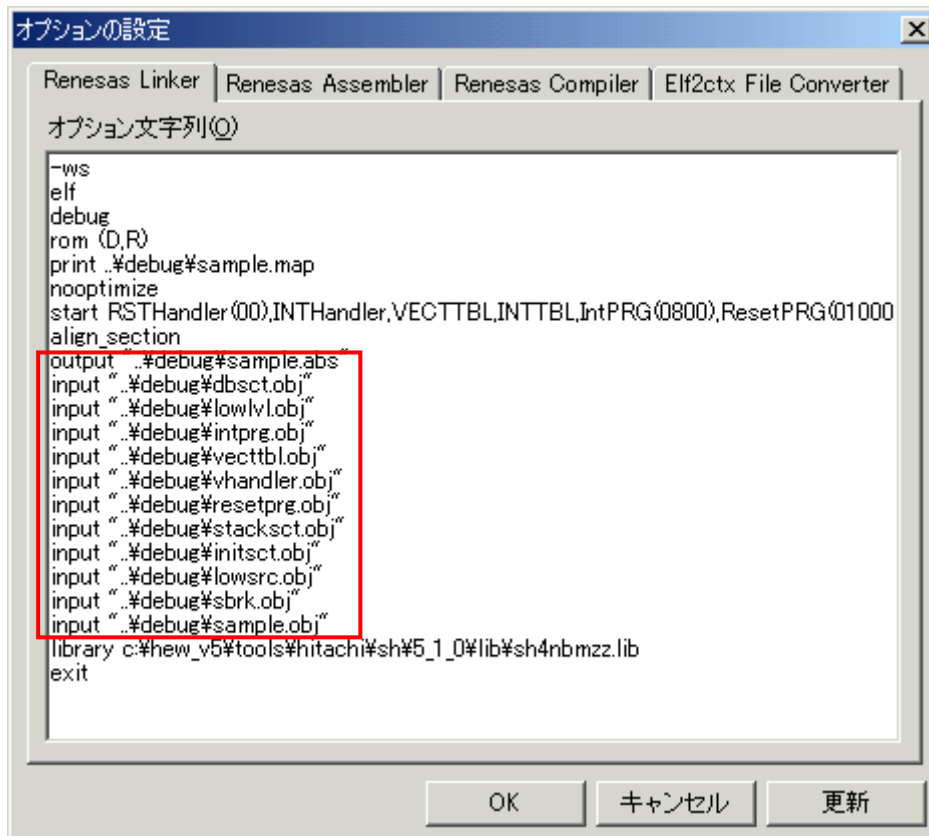
### ◎システム変数一覧

システム変数名	説明
\$(WSP)	現在開かれているワークスペースのパスに変換されます。
\$(WSP_TITLE)	現在開かれているワークスペースの名前に変換されます。
\$(DCP)	現在アクティブなデバッグ構成に変換されます。(Debug/Release)
\$(TRG_PATH)	指定したファイルに変換されます。 アセンブラ/コンパイラなどの出力ファイル・オプションとして使用されます。
\$(SRC_PATH)	指定したファイルに依存する先頭のファイル名(フルパス)に変換されます。 アセンブラ/コンパイラなどの入力ファイル・オプションなどに使用されます。
\$(SRC_TITLE)	指定したファイルに依存する先頭のファイルのタイトルに変換されます
\$(DEPEND_FILE)	指定したファイルに依存する全ての依存関係に変換されます。 例えば、リンク・オプションの入力ファイルなどに使用されます。

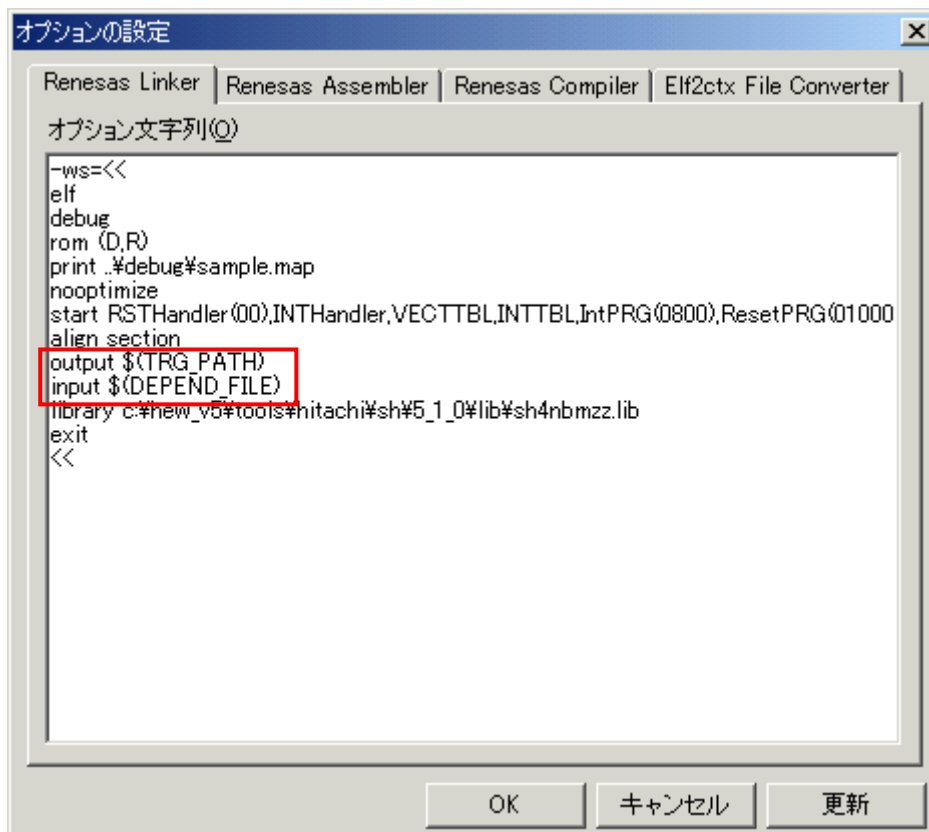
※システム変数はオプション設定ダイアログでしか有効ではありません。

※システム変数を組み合わせて使用する場合はエンマーク"¥"を区切り文字として使用して下さい。

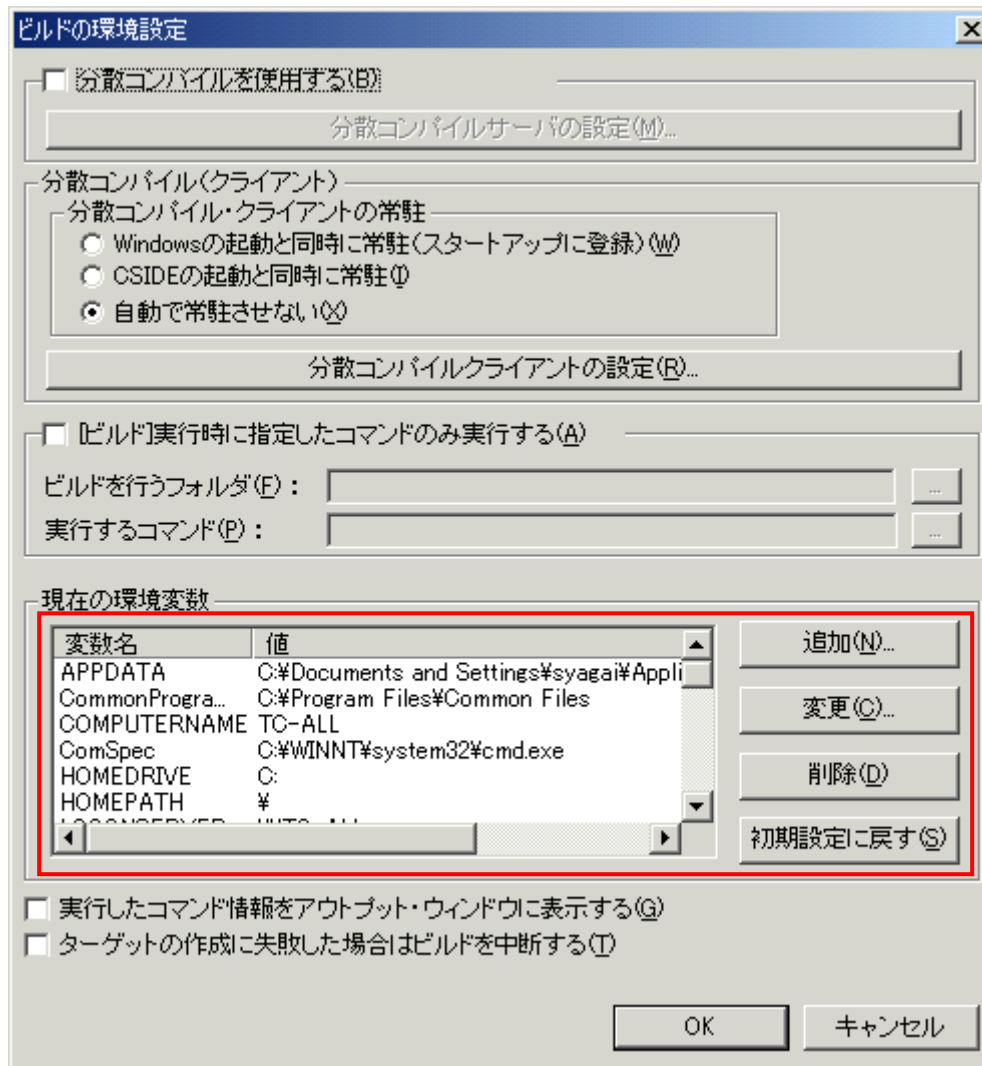
3. オプション設定の一例を示すとこのようになります。



※ユーザ定義 拡張ツールではGUIでのオプション設定は行なえません。



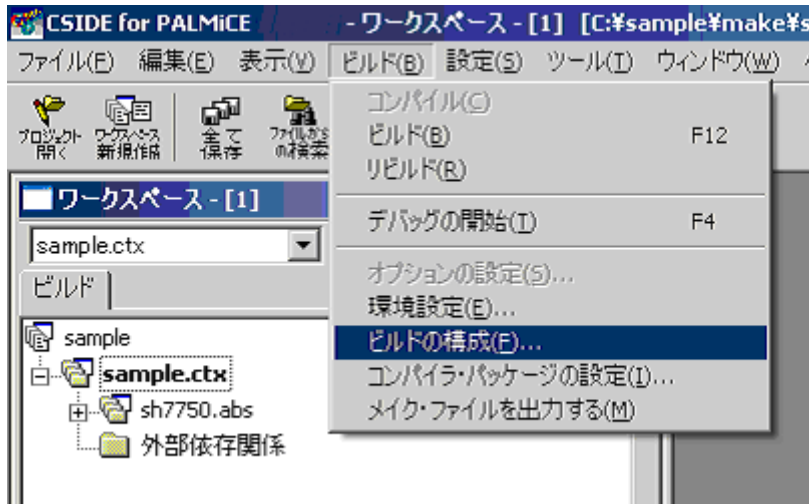
4. メニューバー[ビルド]-[環境設定]を開き、[現在の環境変数]に必要な環境変数を追加してください。



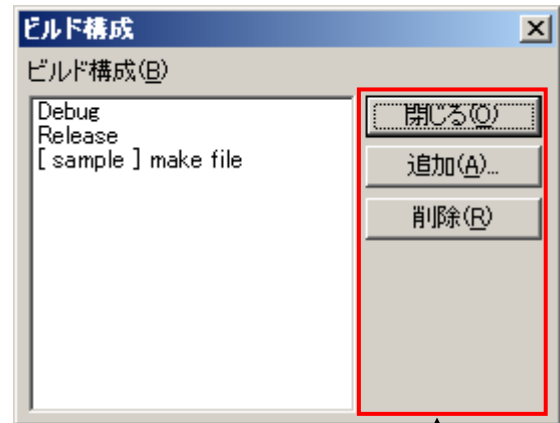
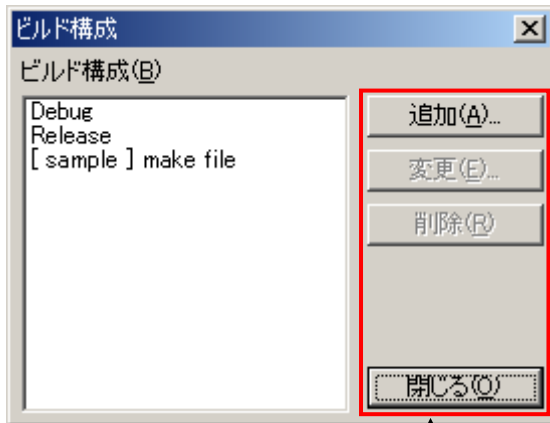
## ◎オブジェクト・ファイルの出力先フォルダの変更

メイク・ファイルをインポートするとオブジェクト・ファイルの出力先が”[ワークスペース名] make file”に設定されます。これはメイク・ファイルが示す出力先のままですと作成したオブジェクト・ファイルに上書きされてしまうことを避けるために別のフォルダに出力する仕様になっています。以下で出力先を Debug、Release に戻して使用する方を説明いたします。

○メニューバー[ビルド]-[ビルドの構成]を選択してください。



※CSIDE のバージョンによって変更を行うための仕様が異なります。(各 CSIDE のリリース時期により異なります。) これ以降は表示されます[ビルド構成]ダイアログが、下図で示す内の該当するどちらか一方に進んでください。



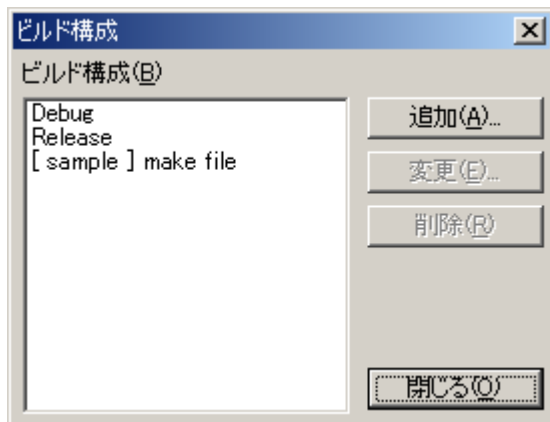
↑ ボタンの表示が異なります

→①2004年9月以降をご覧ください。

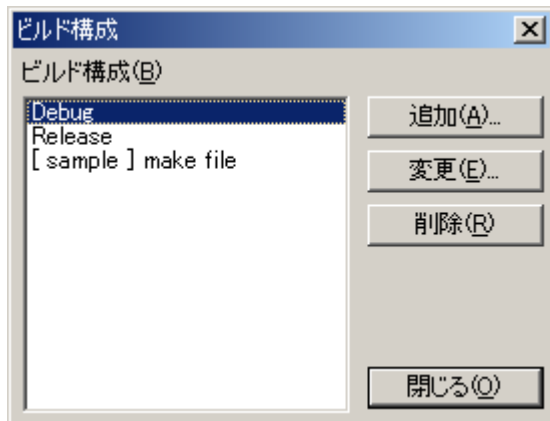
→②2004年9月以前をご覧ください。

①2004年9月以降

1. [ビルド構成]ダイアログには現在設定されているビルド構成が表示されます。メイク・ファイルをインポートした場合、オプション設定は [ワークスペース名] make file フォルダに受け継がれます。このときの Debug と Release は、デフォルトのコンパイルオプションの設定となります。

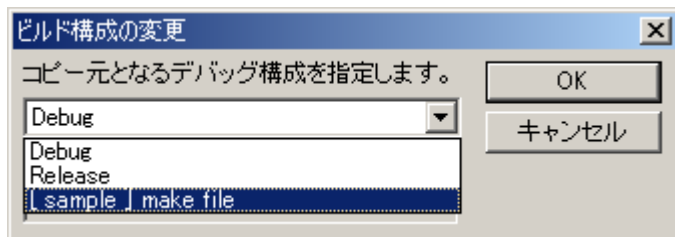


2. ビルド構成にある Debug を選択し、[変更]ボタンを押してください。



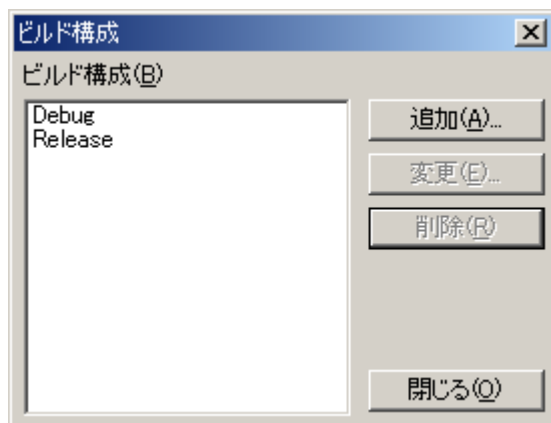
3. [ビルド構成の変更]ダイアログが開かれます。

[コピー元となる...] のプルダウンメニューから [ワークスペース名] make file を選択し、[OK] ボタンを押してください。



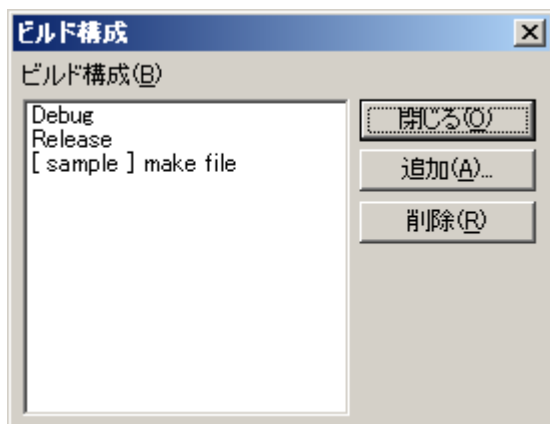
4. [ビルド構成]ダイアログに戻ります。

必要に応じて[ワークスペース名] make file を削除してください。

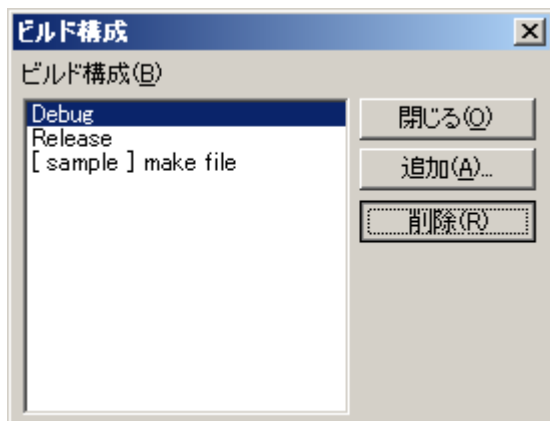


②2004年9月以前の仕様

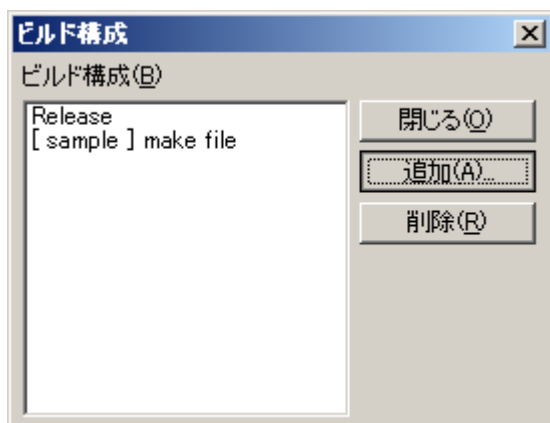
1. [ビルド構成]ダイアログには現在設定されているビルド構成が表示されます。メイク・ファイルをインポートした場合、オプション設定は [ワークスペース名] make file フォルダに受け継がれます。このときの Debug と Release は、デフォルトのコンパイルオプションの設定となります。



2. ビルド構成にある Debug を一度削除します。

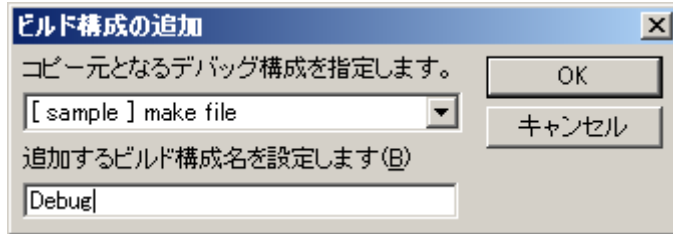


3. 削除後に追加ボタンを押してください。



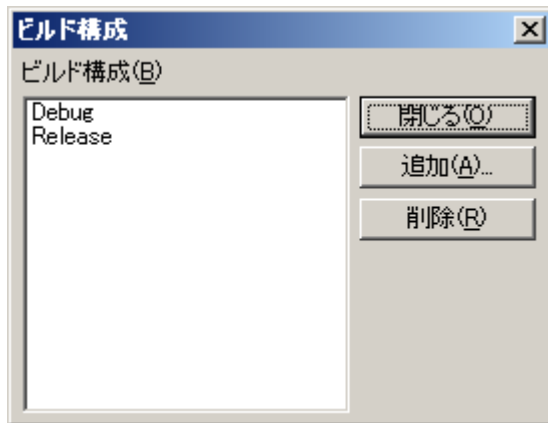
4. [ビルド構成の追加]ダイアログボックスが表示されます。

追加するビルド構成名に Debug と入力しコピー元となるデバッグ構成に[ワークスペース名] make file を選択します。[ワークスペース名] make file に設定されているコンパイルオプションをコピーした Debug フォルダが作成するという意味になります。入力後[OK]ボタンを押してください。また、必要であれば同様の手順で Release を作成してください。



5. 最終的に[ワークスペース名] make file が不要であれば削除してください。

これでメイク・ファイルの設定を受け継いだ Debug が作成されました。Release も作成されました場合、双方が同じオプション設定になっていますので、必要に応じて設定を変更してください。

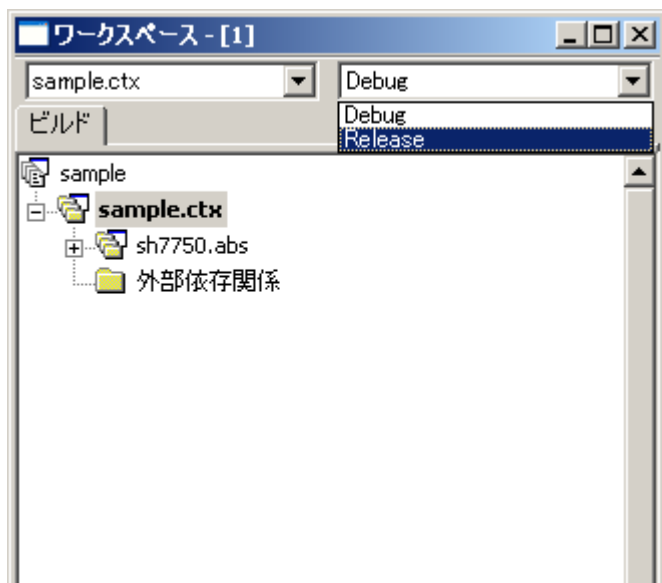


## ※ビルド構成の設定例

上記の通りに操作を行なうと同一のオプション設定の Debug と Release が作成されてしまいます。

ここではオプション設定を分けた使い方の一例を示します。

1. ワークスペースウィンドウ内の右上のコンボボックスから Release を選択します。



2. オプション設定を変更します。

例えば、Release では最適化を行なうように設定をします。

そうすることで Debug は最適化しない、Release は最適化するという設定になります。この様にオプション設定を別にすることで、ビルド構成を変更するだけで出力されるファイルに対して最適化を行なうか行なわないかを変えることができます。

この様にオプション設定の異なるビルド構成を複数用意すれば、簡単に用途に合ったオブジェクト・ファイルを出力することができます。

---

ユーザ定義 拡張ツールでのメイク・ファイルのインポート  
(CSIDE Ver.4 用)

---

2004年12月 第3版

**Computex**<sup>®</sup>  
株式会社コンピューテックス

テクニカルセンタ

〒605-0846 京都市東山区五条橋東四丁目 432-13 對嵐坊ビル

TEL. 075(551)0373 FAX. 075(551)2585

Web Site : <http://www.computex.co.jp/>

E-mail : [support@computex.co.jp](mailto:support@computex.co.jp)