

(h) 登録集原文の指定方法

1. 登録集原文の指定方法

複数のCOBOLプログラムに共通の記述を別のソースファイルとしておき、COPY文で取り込むことができます。登録集原文の概念図を下欄に示します。このようにすると、コーディング量を削減でき、記述ミスもなくなるため、開発効率を高めることができます。ここでは、第1章で実習した「reidai1.cbl」というソースファイルの「DATA0」と「YYMMDD」を登録集原文としたものとして説明します。このソースファイルを「reidai2.cbl」とします。

登録集原文はソースファイルとは別のフォルダにまとめて格納するのが一般的です。コンパイルする際には、「CBLLIB」というコンパイル時環境変数で登録集原文が格納されているフォルダを指定します。

<reidai2.cbl>

```
IDENTIFICATION DIVISION.  
PROGRAM-ID.    reidai2.  
:  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
*  
COPY DATA0.  
*  
COPY YYMMDD.  
*  
PROCEDURE DIVISION.  
:
```

登録集原文

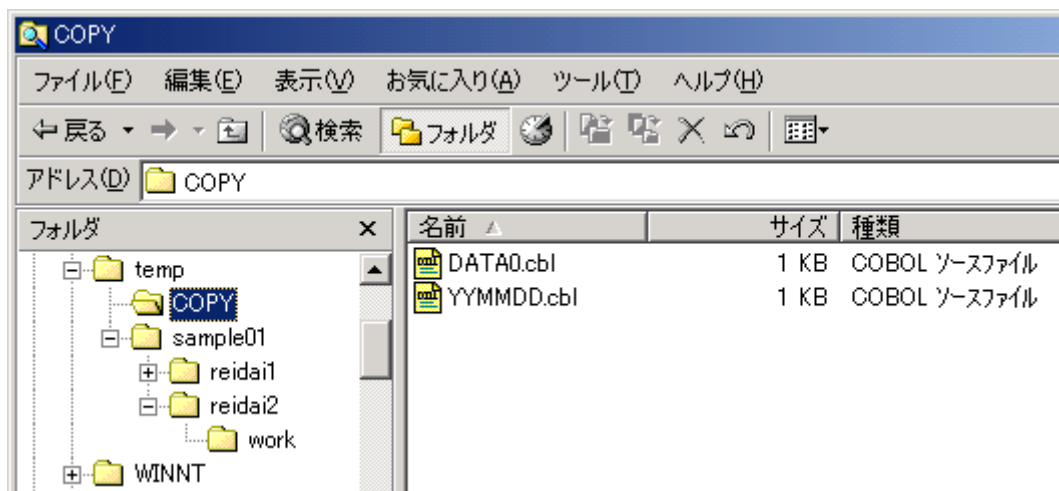
<DATA0.cbl>

```
01 DATA0.  
02 DATA1 PIC X(10) VALUE ALL '*'.  
02 DATA2 PIC X(20) VALUE SPACE.  
02 DATA3 PIC X(10) VALUE ALL '*'.  
:
```

<YYMMDD.cbl>

```
01 YYMMDD.  
02 年 PIC 9(2).  
02 月 PIC 9(2).  
02 日 PIC 9(2).  
:  
:
```

フォルダの構成の例を次に示します。
 ここでは、「COPY」というフォルダに登録集原文を格納しています。



[手順 1] reidai2というプロジェクトをこれまでと同様の手順で作成します。
 ソースファイル「reidai2. cbl」を次に示します。
 「reidai2. cbl」は、「reidai1. cbl」の「DATA0」と「YYMMDD」をCOPY文に変更したものです。

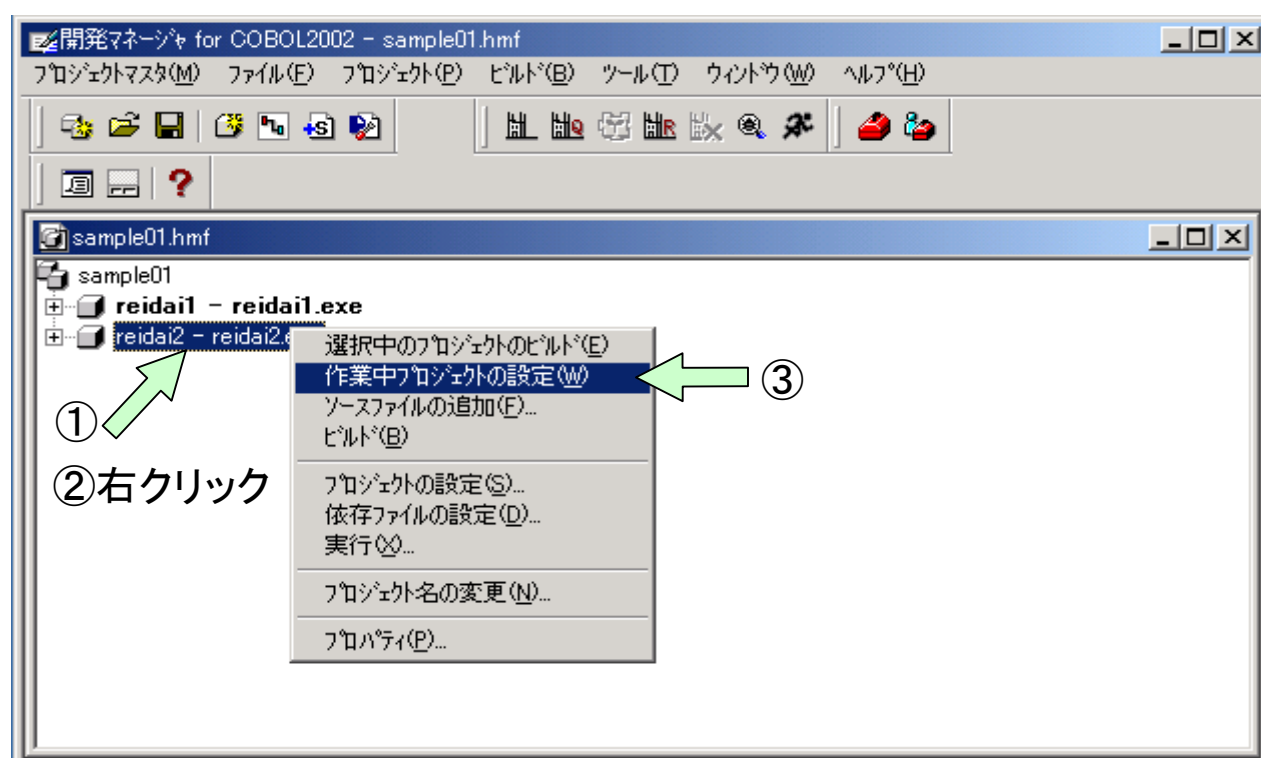
```

000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID.      reidai2.
000300*
000400 ENVIRONMENT  DIVISION.
000500*
000600 DATA    DIVISION.
000700 WORKING-STORAGE  SECTION.
000800*
000900 COPY  DATA0.
001000*
001100 COPY  YYMMDD.
001200*
001300 PROCEDURE  DIVISION.

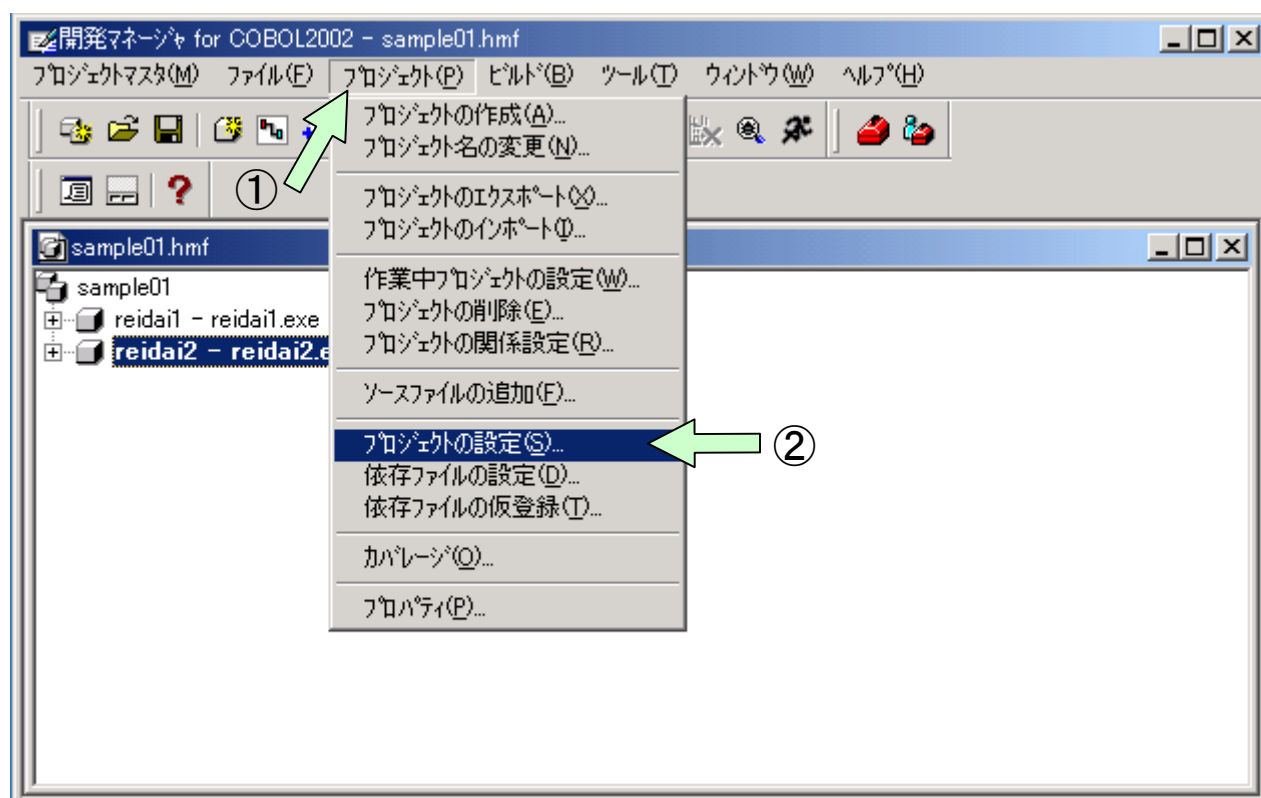
```

以降は「reidai1. cbl」と同じです。

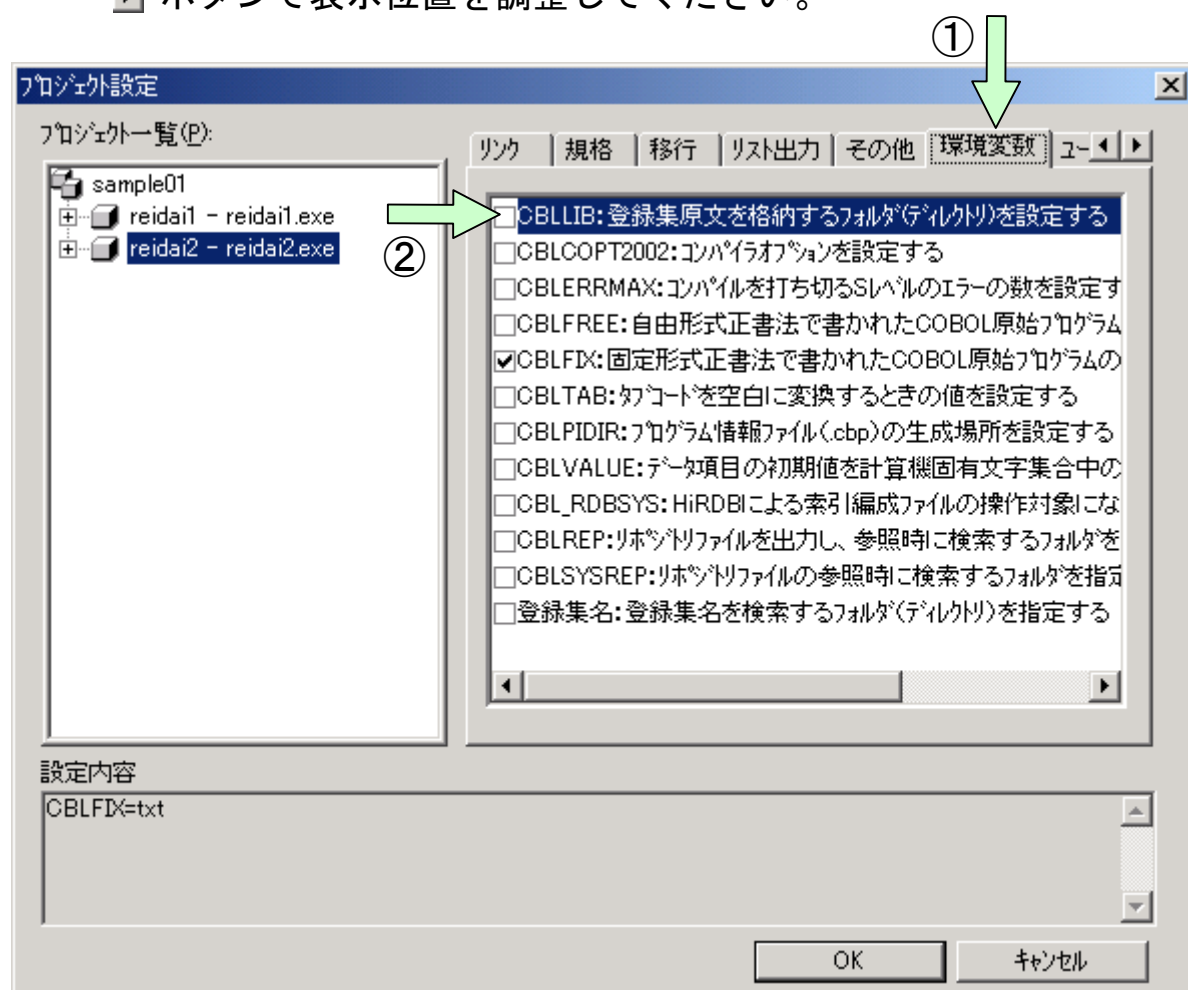
[手順2] プロジェクト「reidai2」を作成したら、作業中のプロジェクトを「reidai2」に設定します。「reidai2」をポイントし右クリックし、プルダウンメニューの中から「作業中プロジェクトの設定(W)」を選択します。



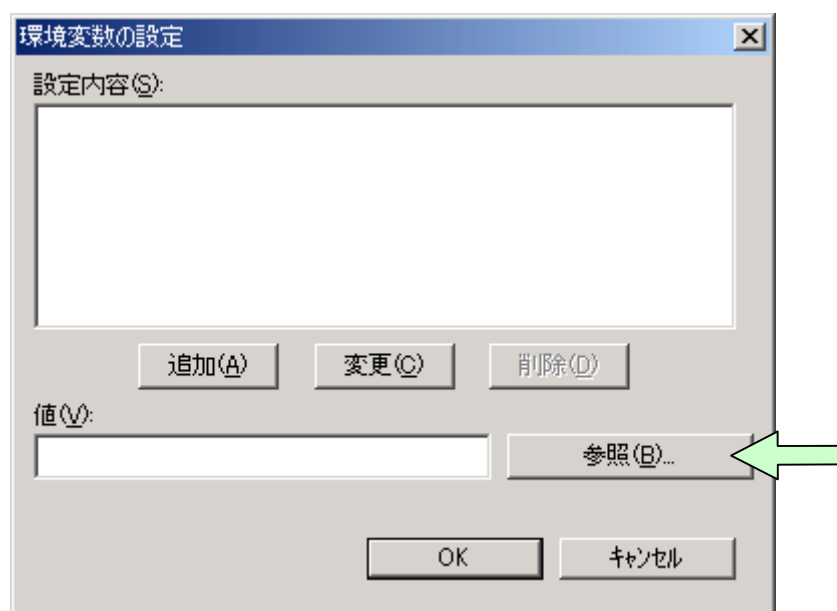
[手順3] 次に環境変数「CBLLIB」で登録集原文のフォルダを指定します。開発マネージャのメニューバーの「プロジェクト(P)」をクリックし、プルダウンメニューの「プロジェクトの設定(S)」を選択します。



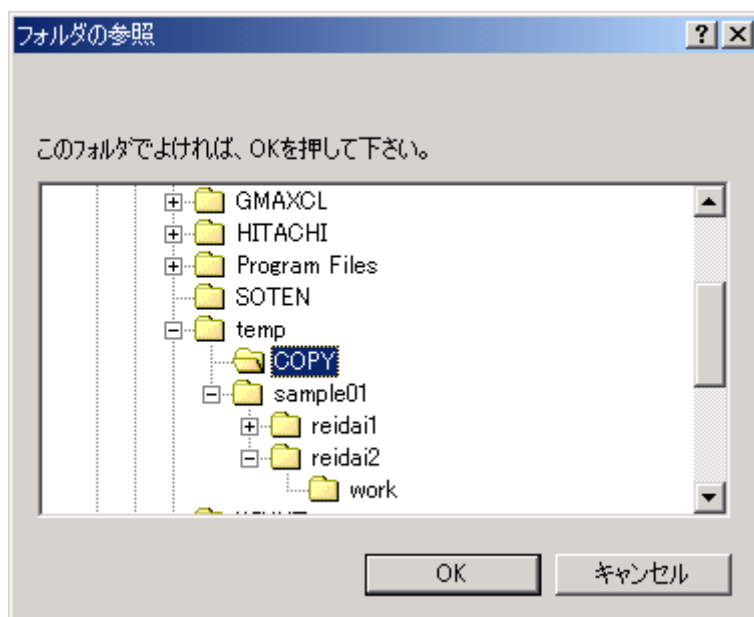
- [手順 4] 「プロジェクトの設定」画面が表示されるので、「環境変数」タブをクリックし、「CBLLIB」をチェックします。「環境変数」タブが見えないときは、 ボタンで表示位置を調整してください。



- [手順 5] 「環境変数の設定」画面が表示されるので、「参照(B)」ボタンをクリックします。

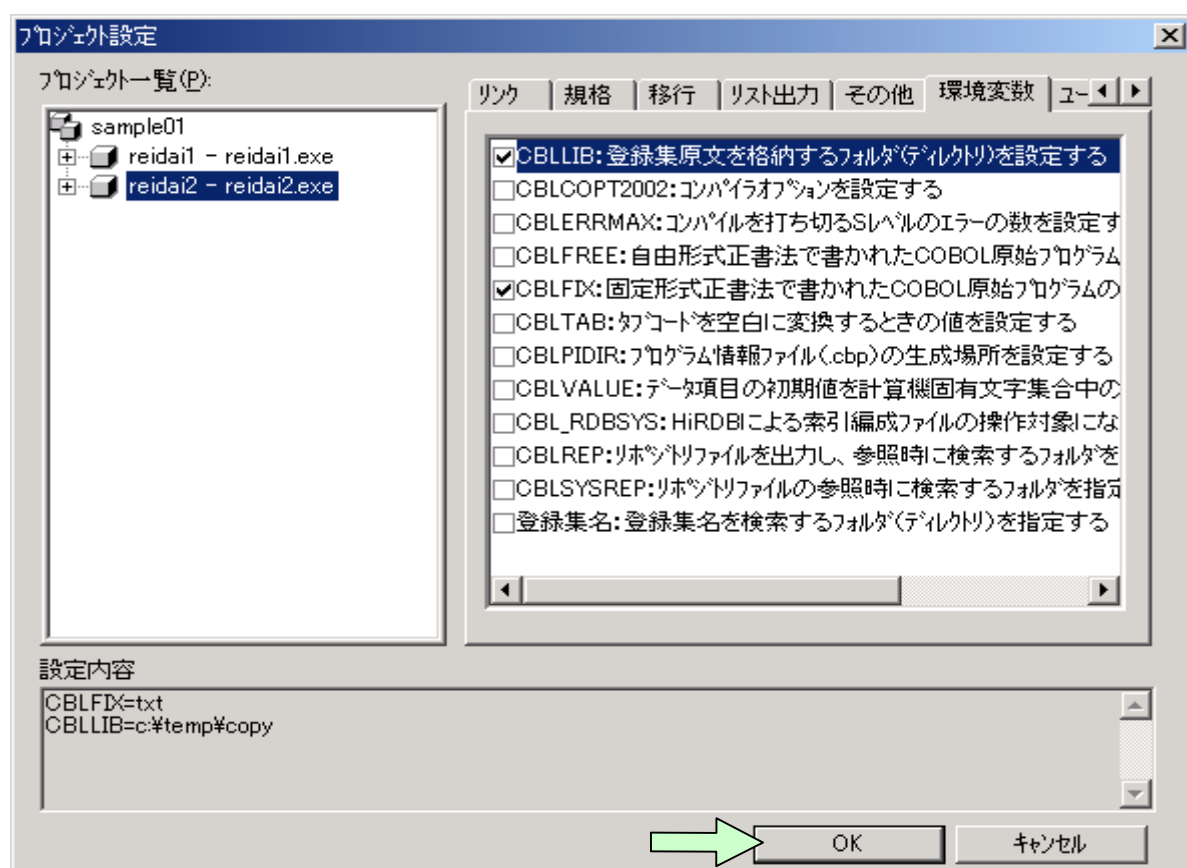


[手順 6] 「フォルダの参照」画面が表示されるので、「COPY」フォルダを指定して「OK」ボタンをクリックします。

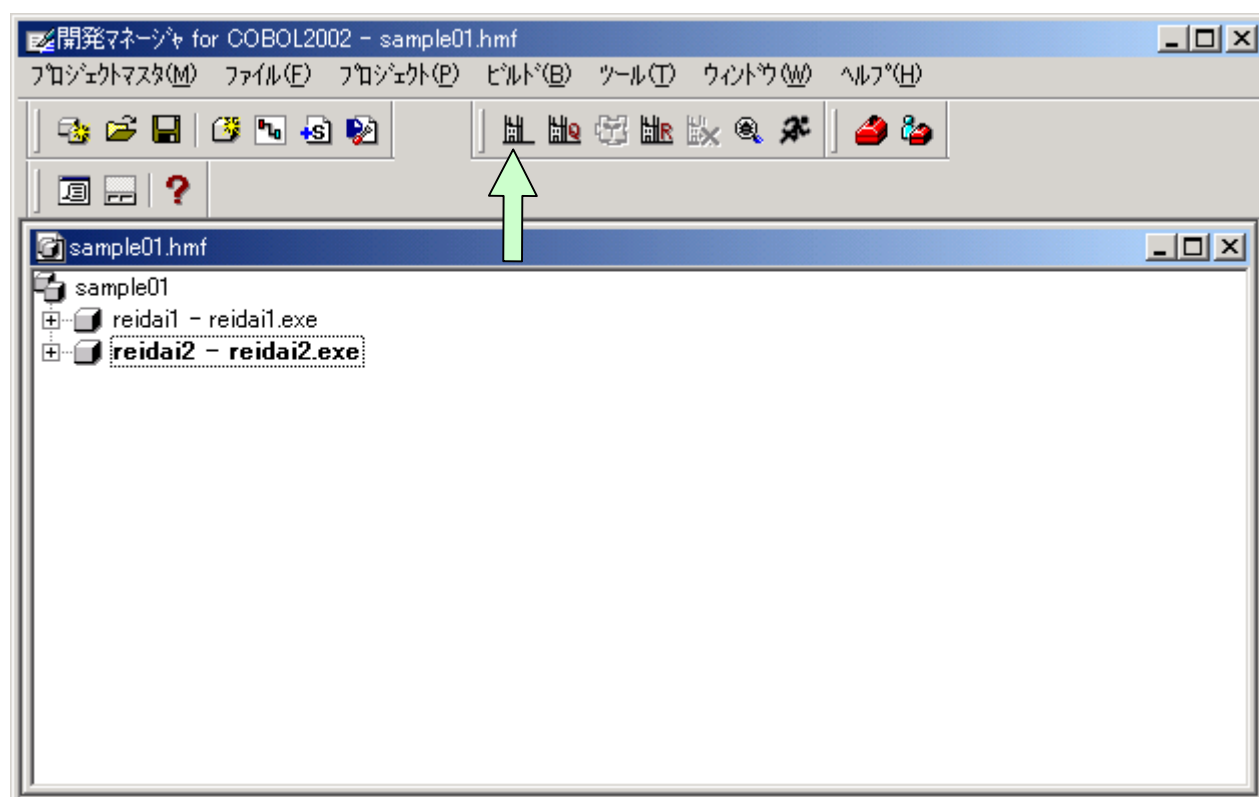


「OK」ボタンをクリックします。

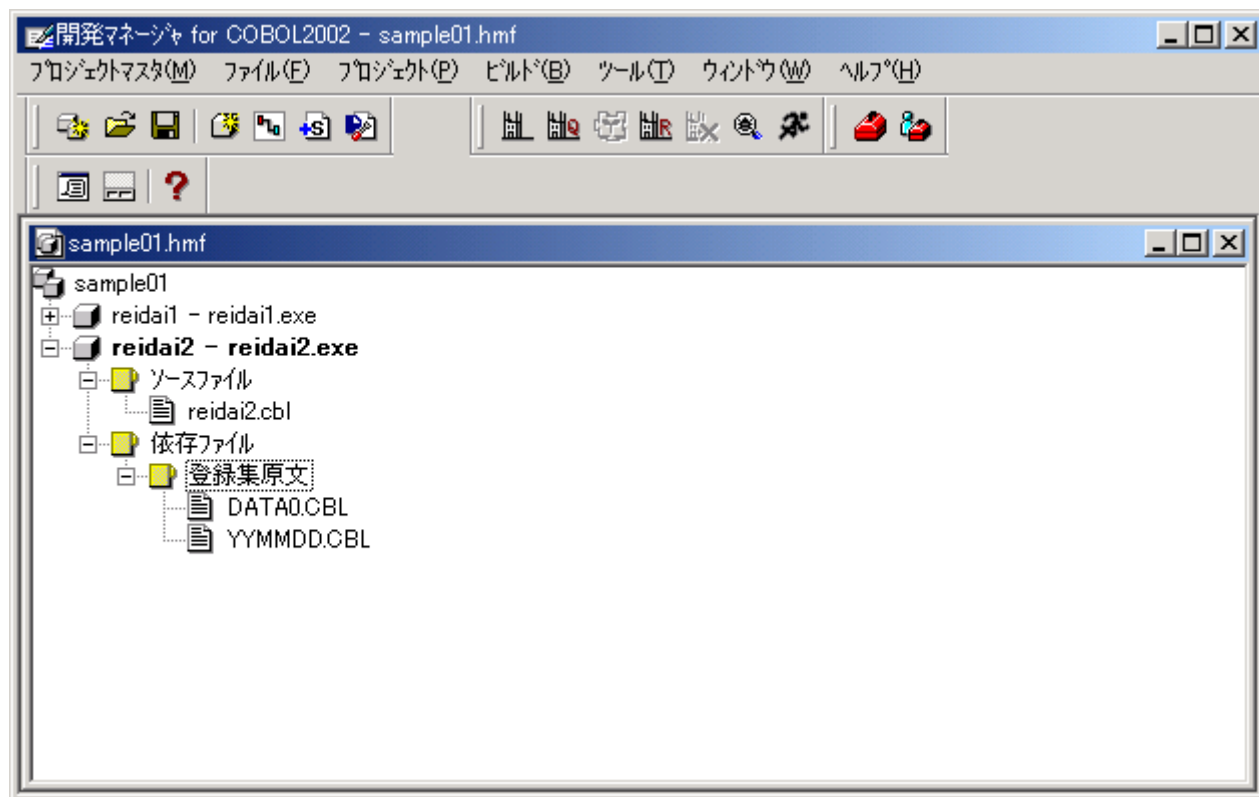
[手順7] 「プロジェクトの設定」画面に戻るので、「OK」ボタンをクリックします。



[手順8] ビルドします。



[手順9] プロジェクト「reidai2」を全て表示すると、依存ファイルとして登録集原文「DATA0.cbl」と「YYMMDD.cbl」が登録されているのがわかります。ビルドすると、開発マネージャがCOPY文を認識して自動的に依存関係を構築するのです。



(i) サブプログラムの追加方法

1つのメインプログラムと複数のサブプログラムから構成される場合のプロジェクトの設定方法

1. プロジェクトの設定方法

ここでは、一つのメインプログラムと複数のサブプログラムから構成される場合のプロジェクトの設定方法について説明します。

(プロジェクト「reidai3」を例題として説明します。)

第1章で実習した「reidai1」のプログラムの「初期処理」、「比較処理」、「出力処理」を各サブプログラムとして、メインプログラムから呼ばれるようにした例で設定方法を示します。

各サブプログラムの名称は次のとおりです。

■メインプログラム : reidai3. cbl

```

:
PROCEDURE DIVISION.
Mein-Sec SECTION.
    CALL 'REIDAI3A' USING YYMMDD.
    CALL 'REIDAI3B' USING DATA0 YYMMDD.
    CALL 'REIDAI3C' USING DATA0.
STOP RUN.

```

■サブプログラム「初期処理」 : reidai3a. cbl

```

IDENTIFICATION DIVISION.
PROGRAM-ID. reidai3a.
ENVIRONMENT DIVISION.
DATA DIVISION.
LINKAGE SECTION.
01 YYMMDD.
    02 年 PIC 9(2).
    02 月 PIC 9(2).
    02 日 PIC 9(2).
PROCEDURE DIVISION USING YYMMDD.
初期処理 SECTION.
    ACCEPT YYMMDD FROM DATE.
EXIT PROGRAM.

```

■サブプログラム「比較処理」 : reidai3b. cbl

```

IDENTIFICATION DIVISION.
PROGRAM-ID. reidai3b.
ENVIRONMENT DIVISION.
DATA DIVISION.
LINKAGE SECTION.
01 DATA0.
    02 DATA1 PIC X(10).
    02 DATA2 PIC X(20).
    02 DATA3 PIC X(10).
01 YYMMDD.
    02 年 PIC 9(2).
    02 月 PIC 9(2).
    02 日 PIC 9(2).
PROCEDURE DIVISION USING DATA0 YYMMDD.
比較処理 SECTION.
    IF 月 = 9
    THEN
        MOVE 'September!!' TO DATA2
    ELSE
        MOVE 'Not September!!' TO DATA2
    END-IF.
EXIT PROGRAM.

```

■サブプログラム「出力処理」 : reidai3c. cbl

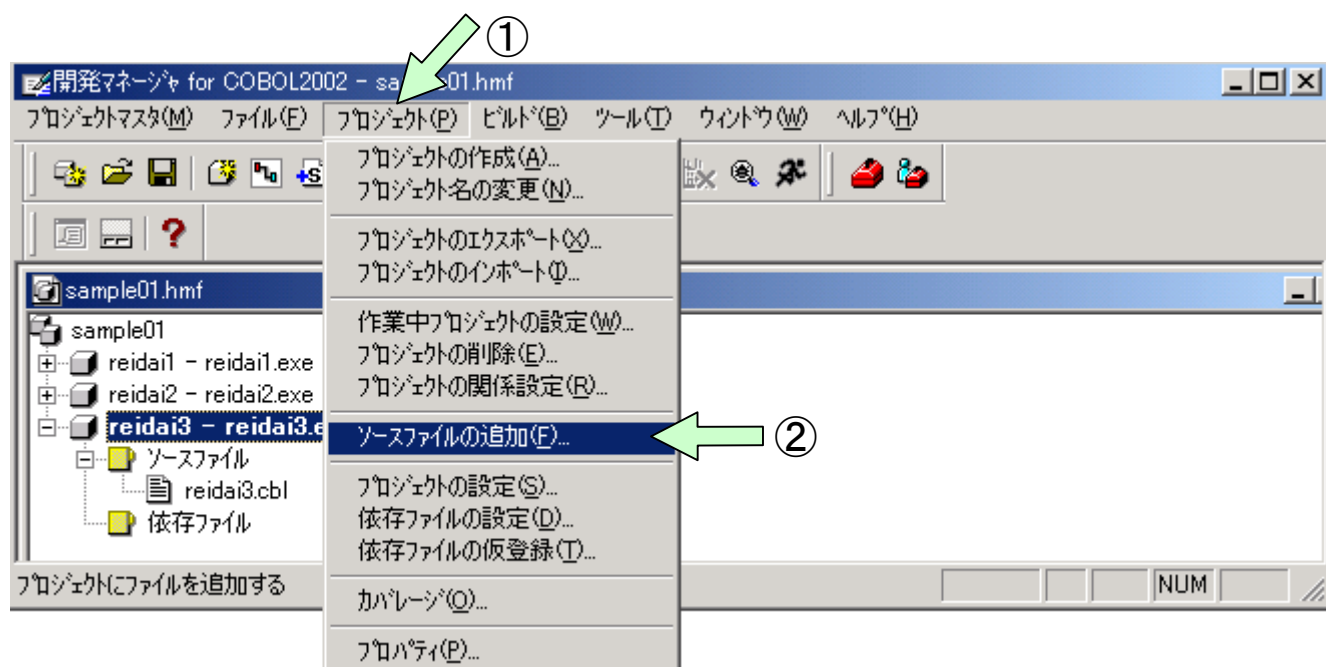
```

IDENTIFICATION DIVISION.
PROGRAM-ID. reidai3a.
ENVIRONMENT DIVISION.
DATA DIVISION.
LINKAGE SECTION.
01 DATA0.
    02 DATA1 PIC X(10).
    02 DATA2 PIC X(20).
    02 DATA3 PIC X(10).
PROCEDURE DIVISION USING DATA0.
出力処理 SECTION.
    DISPLAY DATA0.
EXIT PROGRAM.

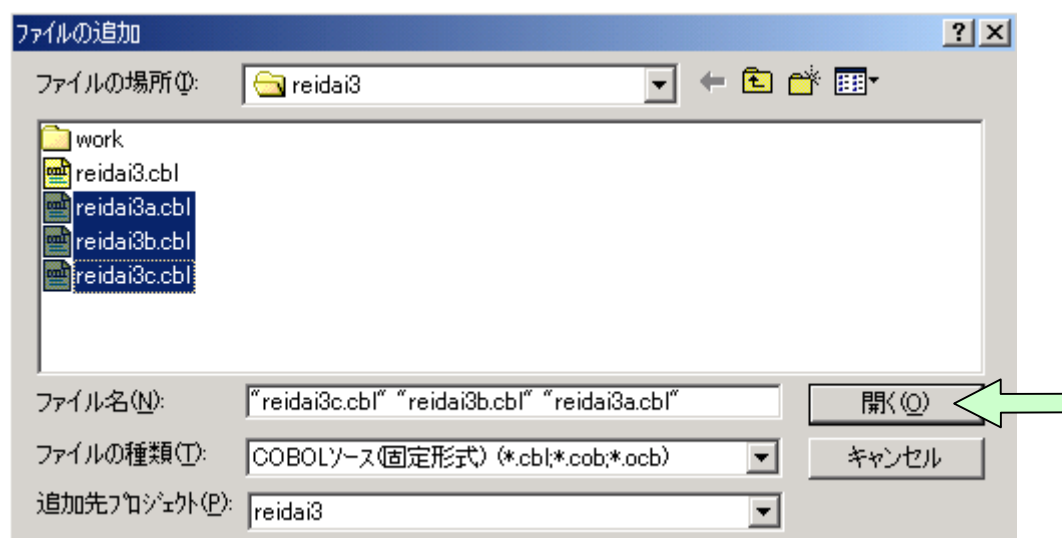
```

メインプログラムのプロジェクト「reidai3」は、これまで示した手順で作成し、各サブプログラムはあらかじめ「reidai3. cbl」と同じフォルダに格納してあるものとして手順を示します。

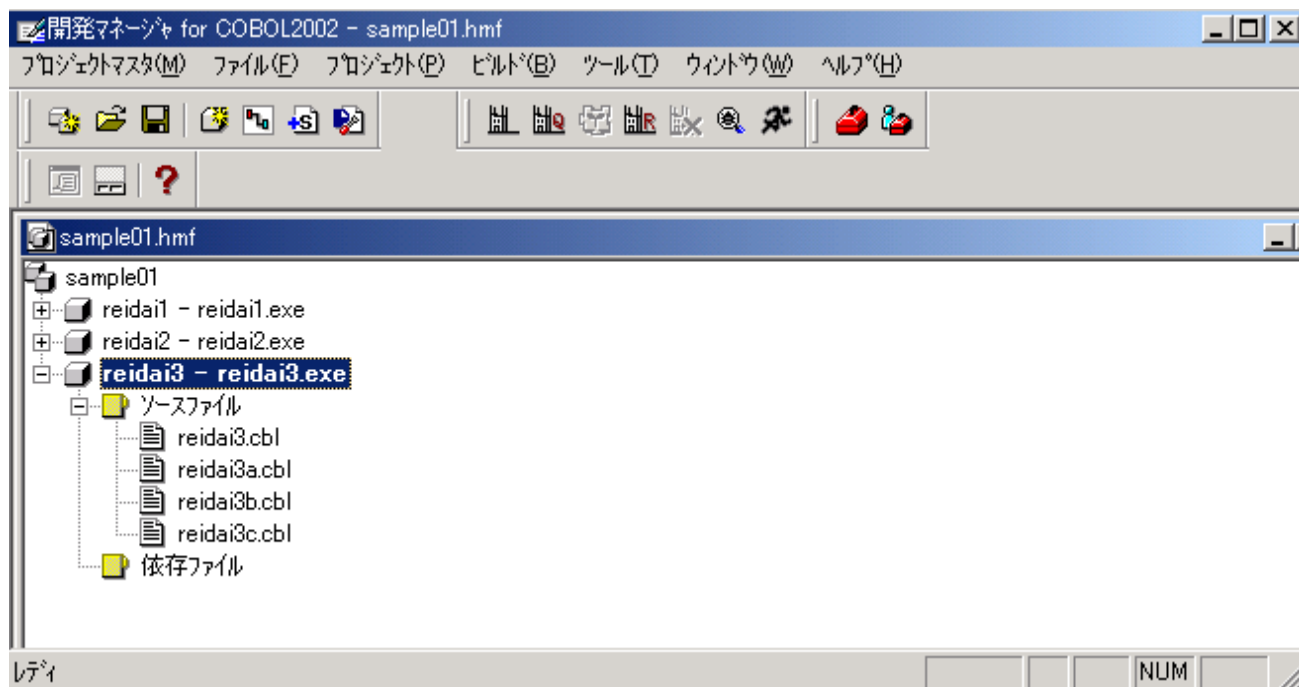
[手順 1] 開発マネージャのメニューバーの「プロジェクト(P)」をクリックし、プルダウンメニューの中の「ソースファイルの追加(F)」をクリックします。
すると、ファイルの追加画面が表示されます。



[手順 2] ここで追加するファイルを選択し、開くボタンを押下します。
複数のサブプログラムを追加できます。



[手順3] reidai3のプロジェクトに3つのサブプログラムが追加されました。
 この後、ビルドを行います。ビルドすると4つのプログラムが結合されて一つの実行可能ファイル(.exe)になります。エラーがなければ続いて実行してください。



[注意事項]

- ①メインプログラムの終了は「STOP RUN」文、サブプログラムの終了は「EXIT PROGRAM」文を使用します。
- ②メイン側のプログラムからサブプログラムを呼び出す「CALL」文では、サブプログラム名をアポストロフィで囲んで指定します。

[例] CALL 'REIDAI3A' USING DATA0 YYMMDD.

プログラム名は基本的に大文字を使用します。これは、サブプログラムの「PROGRAM-ID」段落で定義したプログラム名が大文字と解釈されるからです。

[例] PROGRAM-ID. reidai3a. <-- プログラム名は「REIDAI3A」と解釈

なお、「PROGRAM-ID」段落のプログラム名をアポストロフィで囲んで指定すると、指定したとおりに解釈されます。

[例] PROGRAM-ID. 'reidai3a'. <-- プログラム名は「reidai3a」と解釈

- ③サブプログラムで受け取るデータは「LINKAGE SECTION」として定義します。「LINKAGE SECTION」では、「VALUE」句は使用できません。

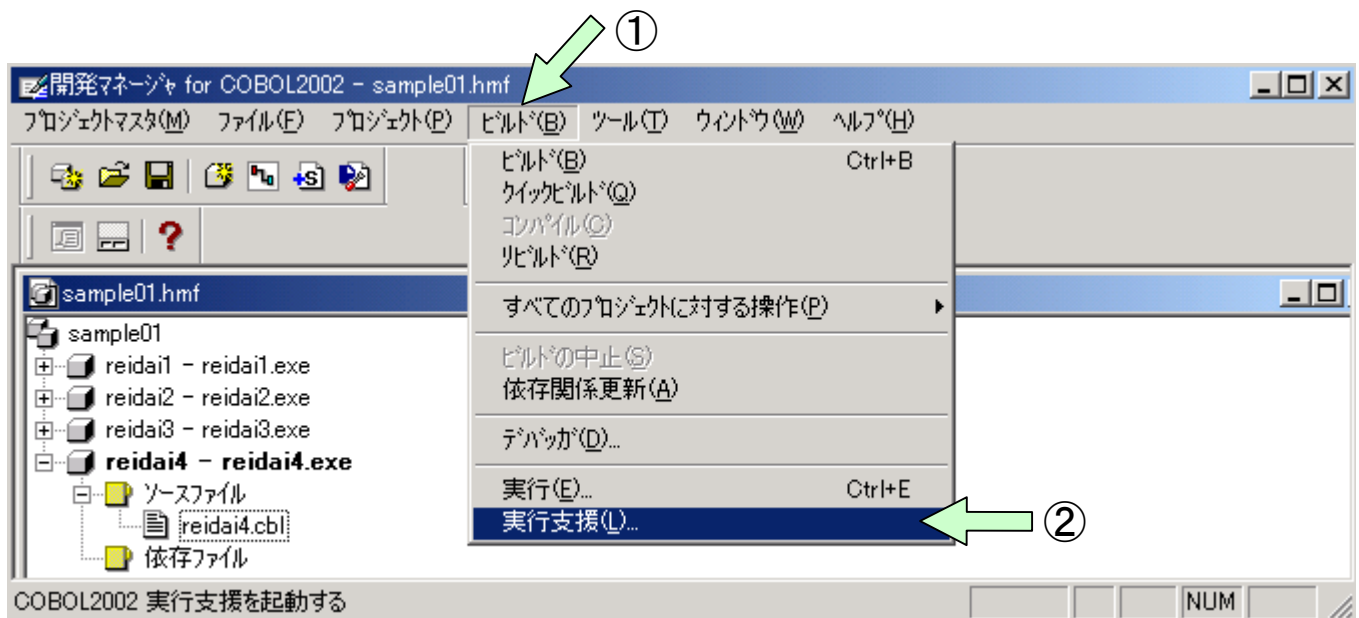
(j) 索引ファイルを新規に 作成する方法

索引ファイルを新規に作成する方法について説明します。

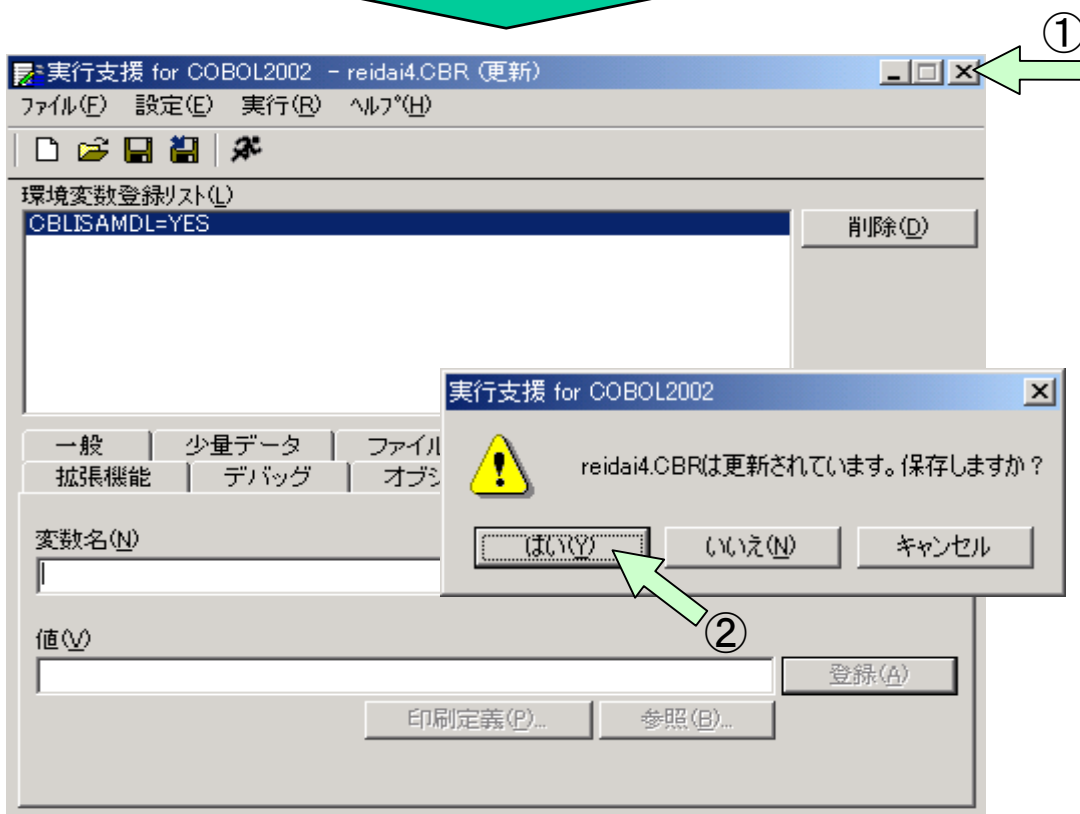
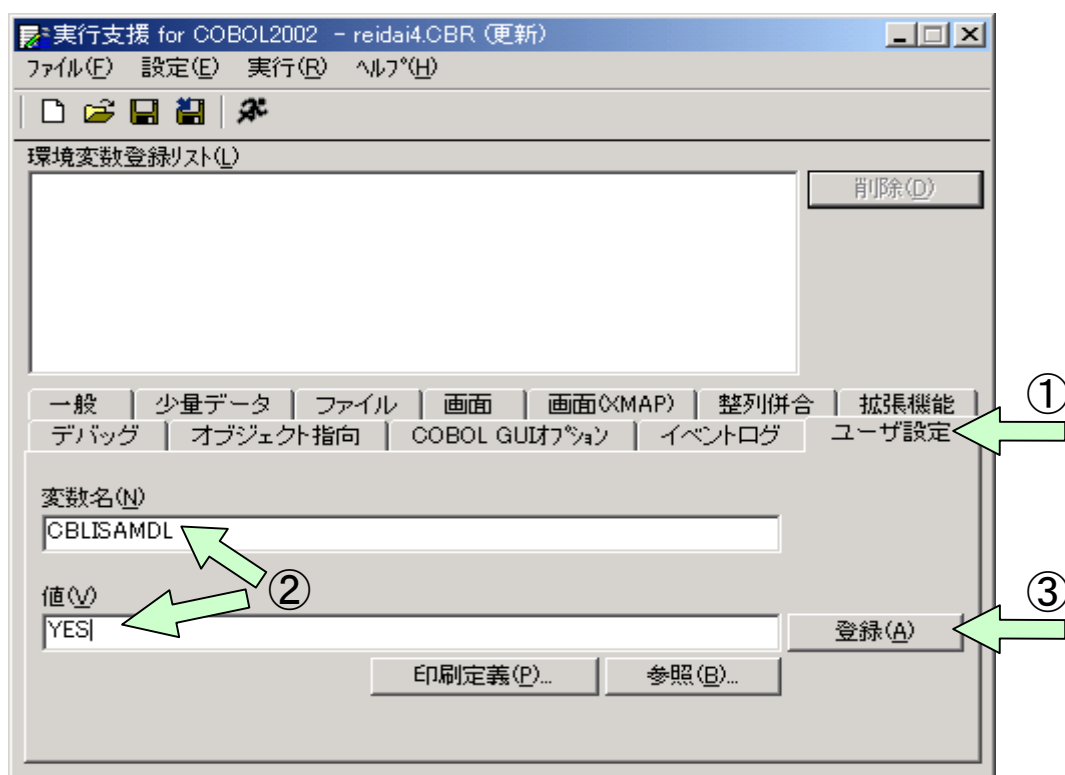
「OPEN OUTPUT」で開いてWRITE文でレコードを書き出すと、通常は新規作成になりますが、索引ファイルの場合は追加モードでレコードが書き出されます。すなわち、一度索引ファイルを作成した後、同じキーで索引ファイルを作成し直そうとしても、「重複キー」エラーになってしまいます。

索引ファイルの新規作成をするときは、実行時環境変数「CBLISAMDLE=YES」を指定します。これにより、既存の索引ファイルが削除され、新規作成することが出来ます。

〔手順 1〕 開発マネージャのメニューバーの「ビルド (B)」をクリックし、プルダウンメニューの中の「実行支援 (L)」をクリックします。すると、実行支援画面が表示されます。



- [手順 2] 実行時環境変数は、「ユーザ設定」タブで設定します。
 変数名に「CBLISAMD L」、値に「YES」と入力して「登録」ボタンを押すと、
 「環境変数登録リスト(L)」に登録されます。閉じる(×)ボタンを押すと
 「保存しますか?」と聞いてきますので、必ず「はい」をクリックして
 ください。



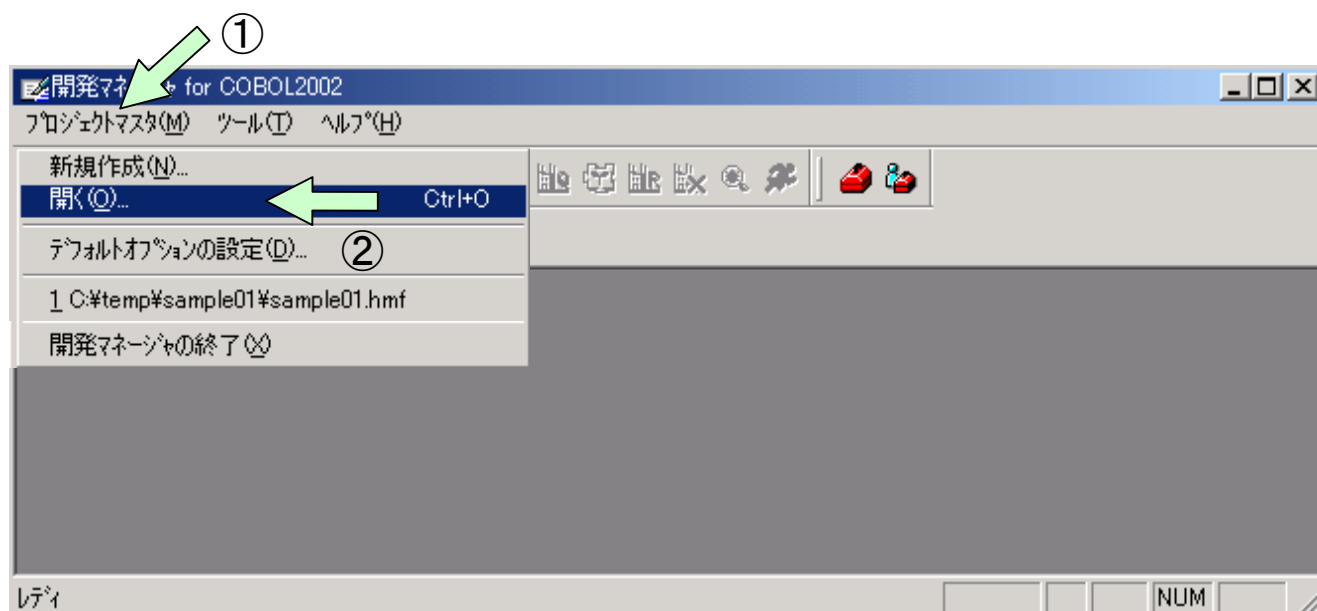
(k) 既存のプロジェクトマスタ ファイルの開き方

— 既に作成済みのプロジェクトマスタファイルを開くには —

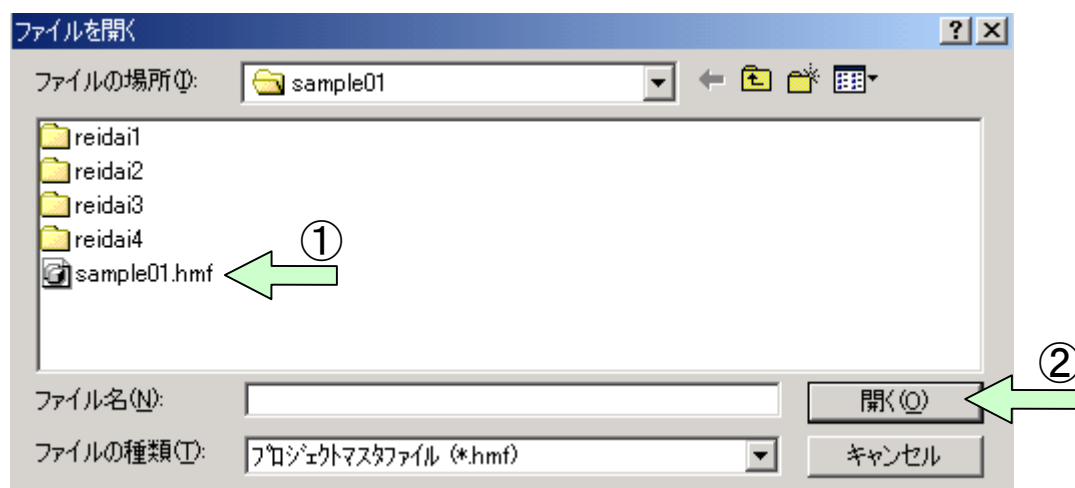
1. 既存のプロジェクトマスタファイルの開き方

ここでは、既に作成済みのプロジェクトマスタファイルの開き方について説明します。

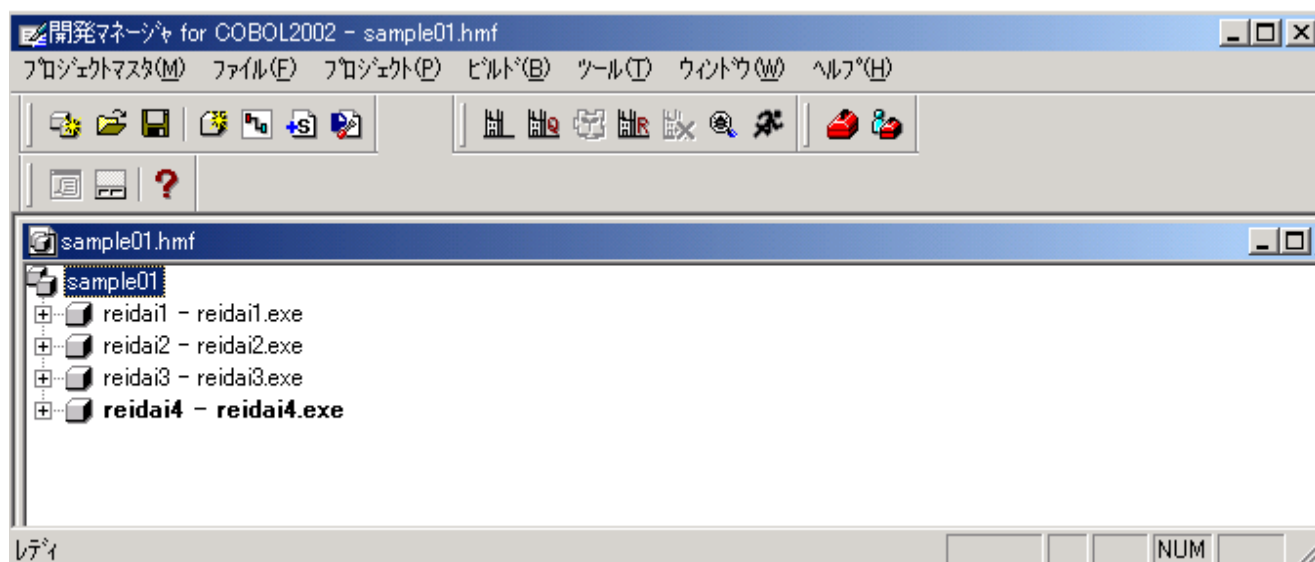
[手順 1] 開発マネージャのメニューバーの「プロジェクトマスタ(M)」、「開く(O)」をクリックします。すると「ファイルを開く」画面が表示されます。



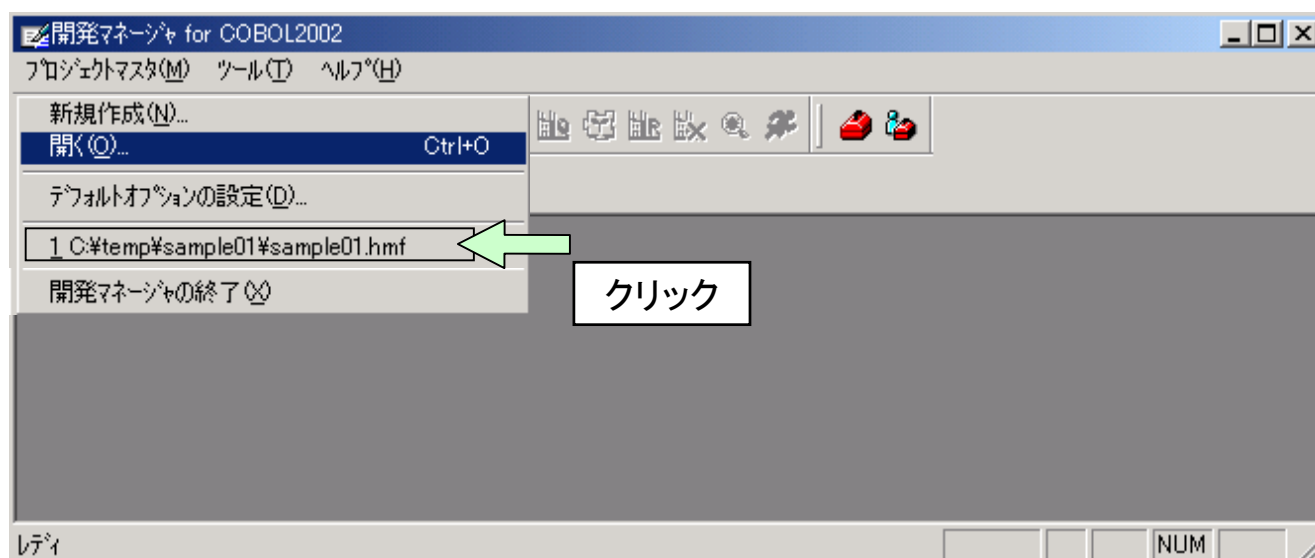
[手順 2] ここで開きたいプロジェクトマスタファイルをクリックし、「開く(O)」ボタンを押します。すると選択したプロジェクトマスタファイルが開きます。下記の例では「SAMPLE01」というプロジェクトマスタファイルを開きます。



[手順3] プロジェクトマスタファイル「sample01」が開けられました。



[補足] 手順1で、「プロジェクトマスタ(M)」をクリックしたとき、最近使用したプロジェクトマスタファイルが表示されます。開きたいファイルが表示されているときは、そのファイルをクリックしてください。
あるいは、エクスプローラで直接プロジェクトマスタファイル(.hmf)をダブルクリックして開くこともできます。



《他社所有名称に対する表示》

- ・ Btrieveは、米国Pervasive Software Inc. の米国およびその他の国における商標または登録商標です。
- ・ ODBCは、米国Microsoft Corp. が提唱するデータベースアクセス機構です。
- ・ OLEは、米国Microsoft Corp. が開発したソフトウェア名称です。OLEIは、Object Linking and Embeddingの略です。
- ・ Windowsは、米国およびその他の国における米国Microsoft Corp. の登録商標です。
- ・ Windows Vistaは、米国Microsoft Corporationの米国及びその他の国における登録商標です。
- ・ その他記載の会社名・製品名は、それぞれの会社の商標もしくは登録商標です。