2. テストデバッガの使用方法

－COBOL2002専用テストデバッグツールの使い方－

― 目　次 ―

1. はじめに
2. コンパイル時のオプション
3. テストデバッガの起動と終了
4. ソースの表示方法と中断点の設定
5. データの内容表示と代入
6. データのトレース
7. エディタとの連携
8. デバッガの色などの変更
9. カバレージ情報の蓄積と表示
10. カウント情報の表示
11. 終わりに

１．はじめに

この章では、COBOL2002専用のテストデバッグツールの使用方法について説明します。

既に「１．COBOL2002の起動から実行まで」を読み終わっているものとして、説明を行います。

例題プログラムは１章で作成したプログラム(reidai1)を使用します。

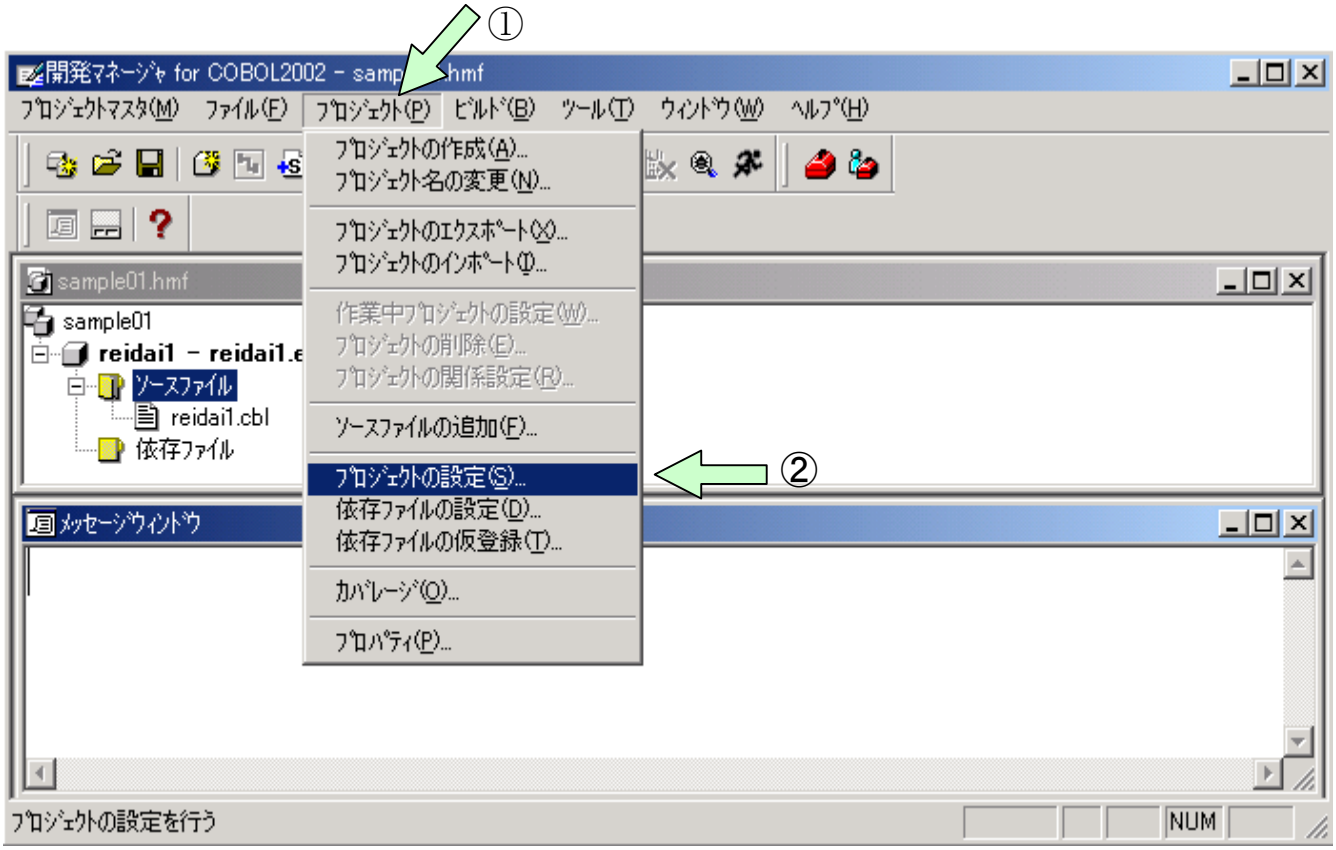
ここでは、COBOL2002専用のテストデバッグツールの機能のうち、特に知っておいて頂きたい基本的な機能について説明します。COBOL教育では、この基本的な操作を理解すれば十分と考えますが、更に詳細を知りたいという場合には、マニュアル「COBOL2002操作ガイド」、「COBOL2002ユーザーズガイド」を参照ください。

2. コンパイル時のオプション

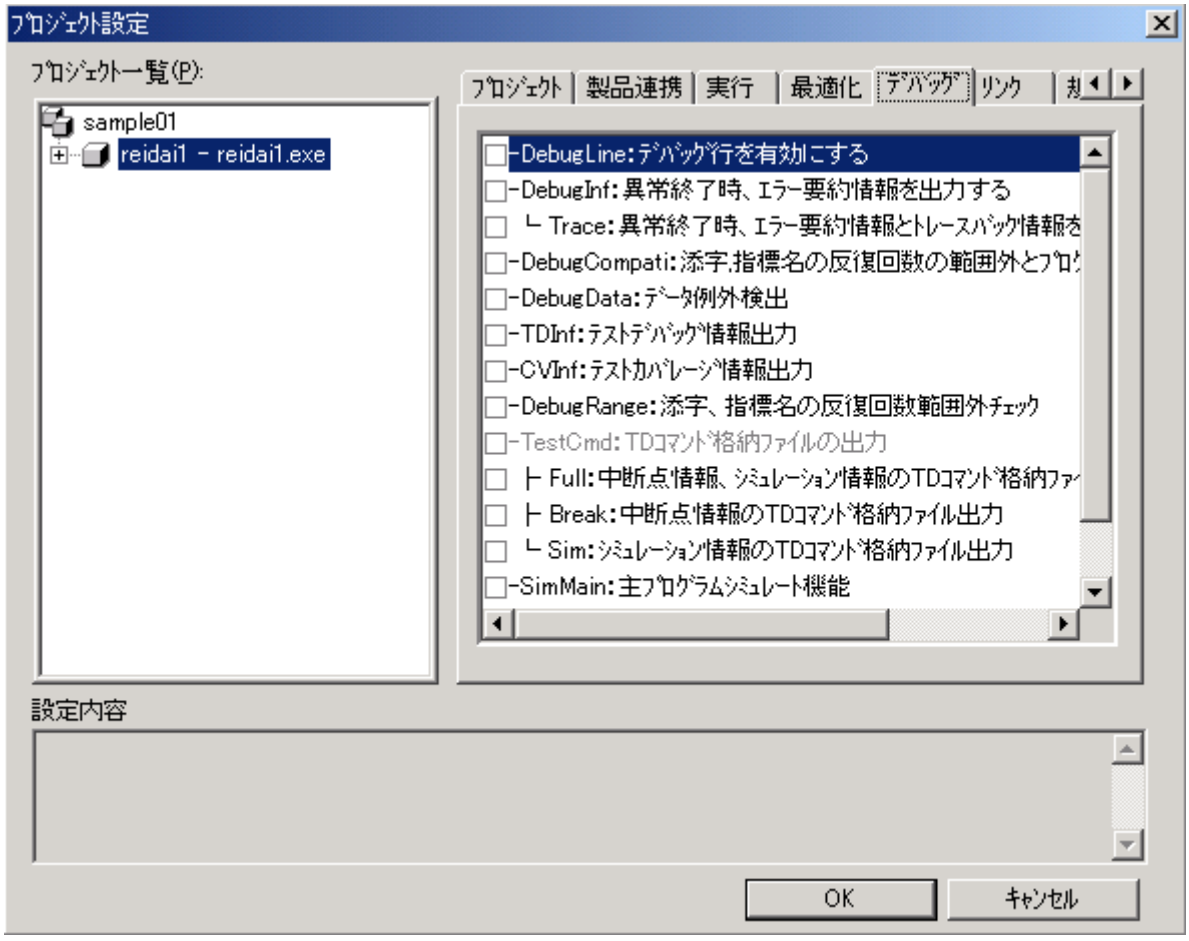
テストデバッグツールを使用するためには、COBOLプログラムのコンパイル時に「コンパイラオプション」を指定する必要があります。

１章で説明したように、デフォルトオプションの設定で「-TDInf」を指定してあれば、ここでオプションを指定する必要はありません。ここでは、デフォルトオプションを設定していない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「ﾌﾟﾛｼﾞｪｸﾄ(P)」をクリックし、プルダウンメニューの「ﾌﾟﾛｼﾞｪｸﾄの設定(S)」をクリックします。
すると、コンパイラオプションの一覧が出力されます。



オプションの一覧が出力されます。
-TDInfをチェックして「OK」ボタンをクリックします。

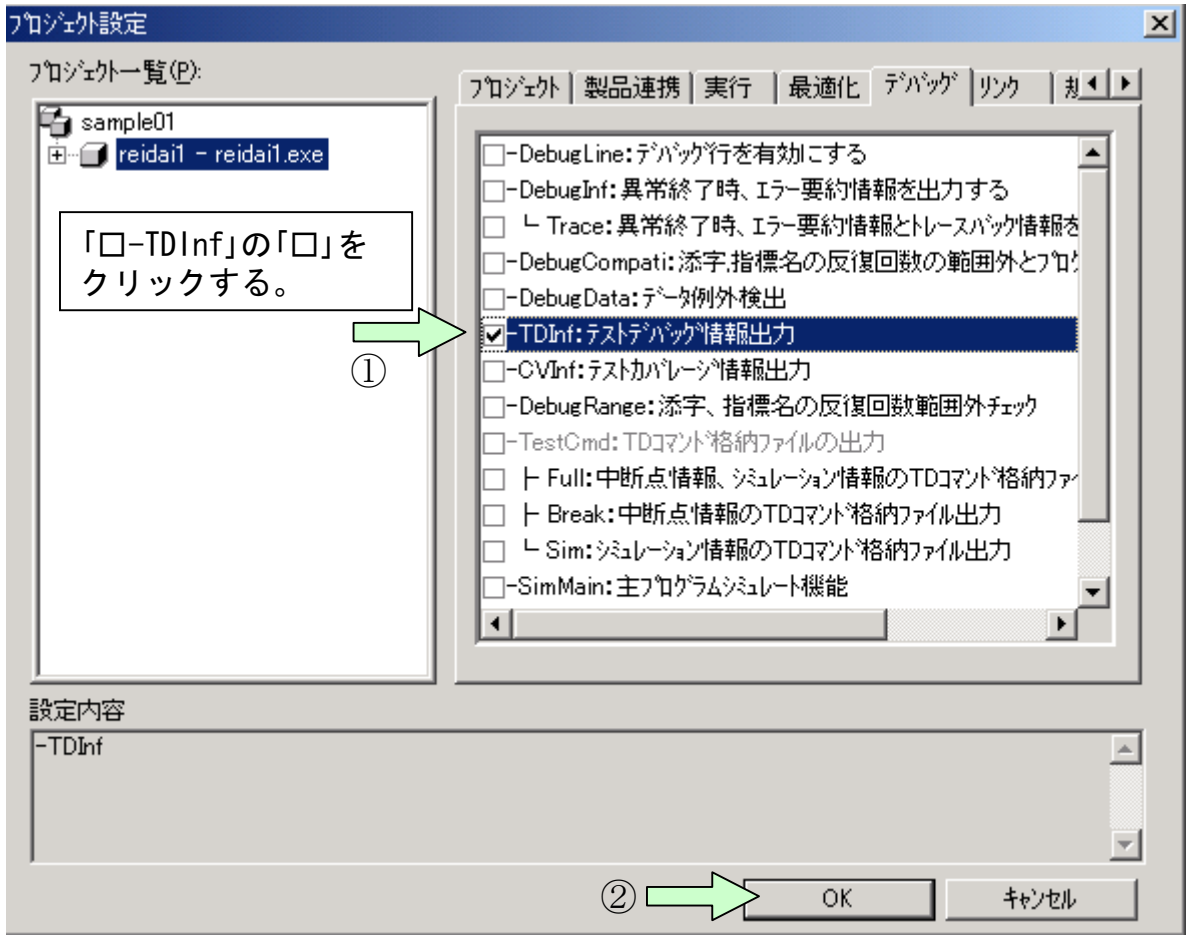


[用語解説] コンパイラオプション

コンパイラオプションは、コンパイラがオプションでサポートしている機能を使うときに指定します。背反する仕様を使い分けるために用意されているオプション等もあります。

デバッグが完了すると、そのプログラムにはデバッグ情報は不要になります。デバッグ情報の出力がオプション機能になっているのは、完成したプログラムに余分な情報を持たなくてもよいよう配慮している意味もあります。

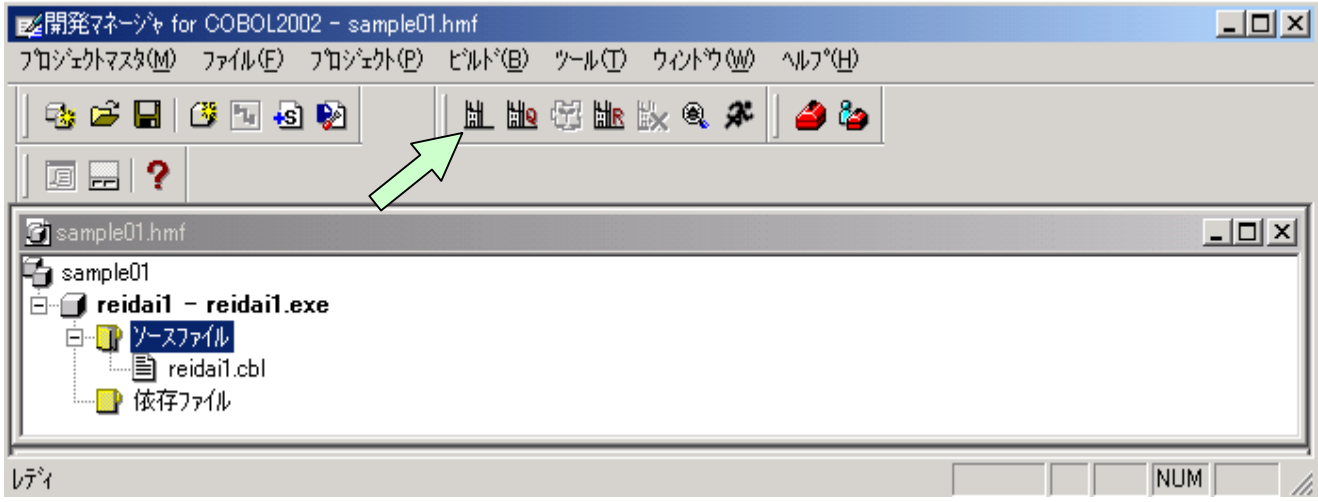
［手順２］コンパイラオプションの一覧の中から「ﾃﾞﾊﾞｯｸﾞ」タブの「-TDInf」の「□」をクリックして「レ」印をつけます。
「レ」印が設定されたことを確認して、「OK」ボタンをクリックします。



［ワンポイントアドバイス］

オプション設定の画面を見てわかるように、多数のコンパイラオプションがあります。
オプションの意味はマニュアルを参照し、必要のないオプションは指定しないようにしましょう。

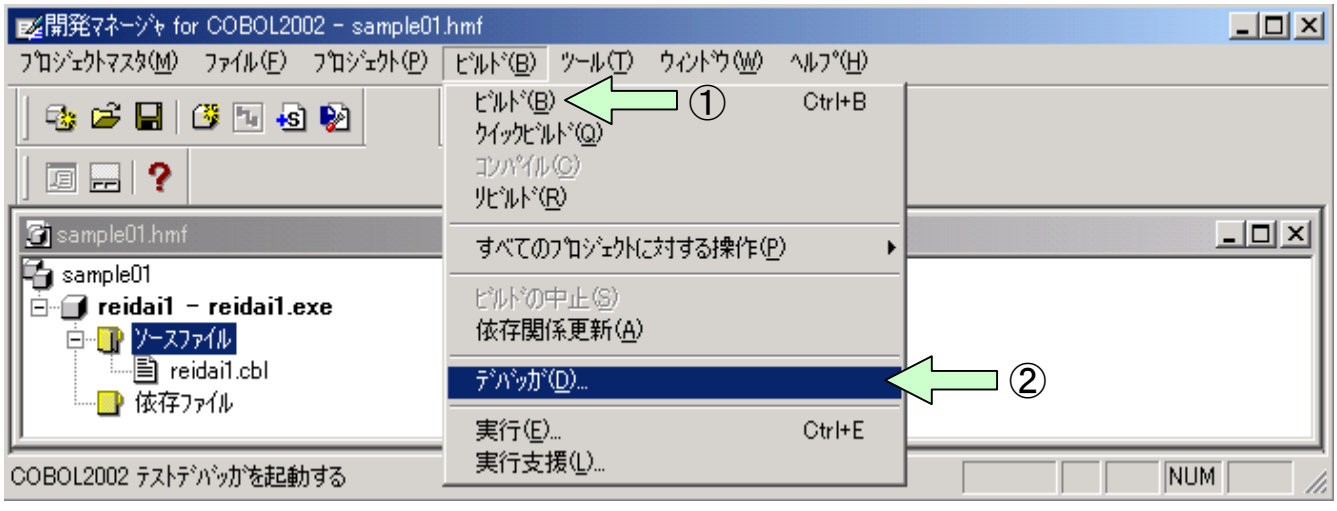
［手順３］開発マネージャ画面に戻りますので、改めてコンパイルを行ってください。

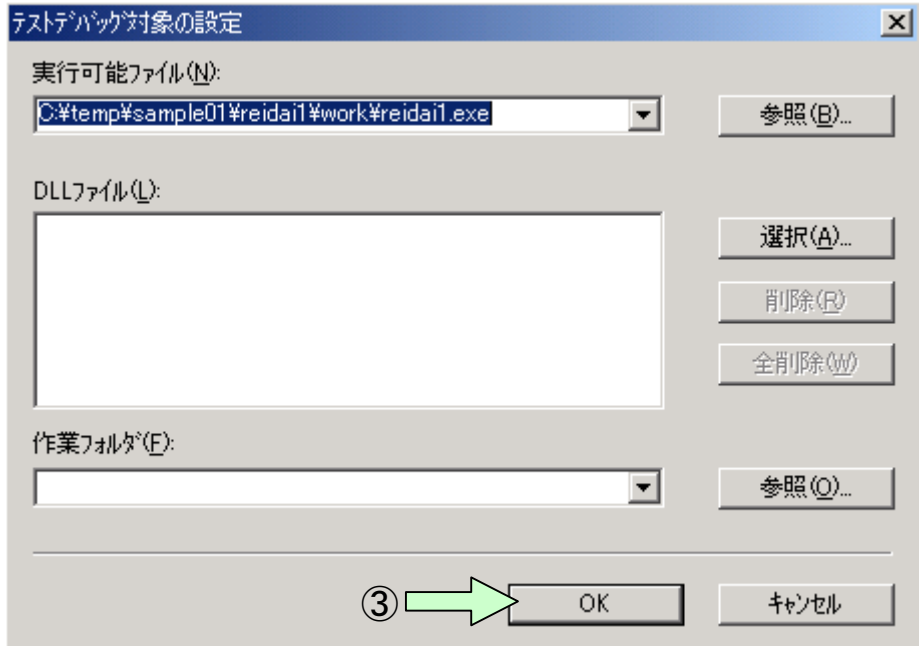


3. テストデバッガの起動と終了

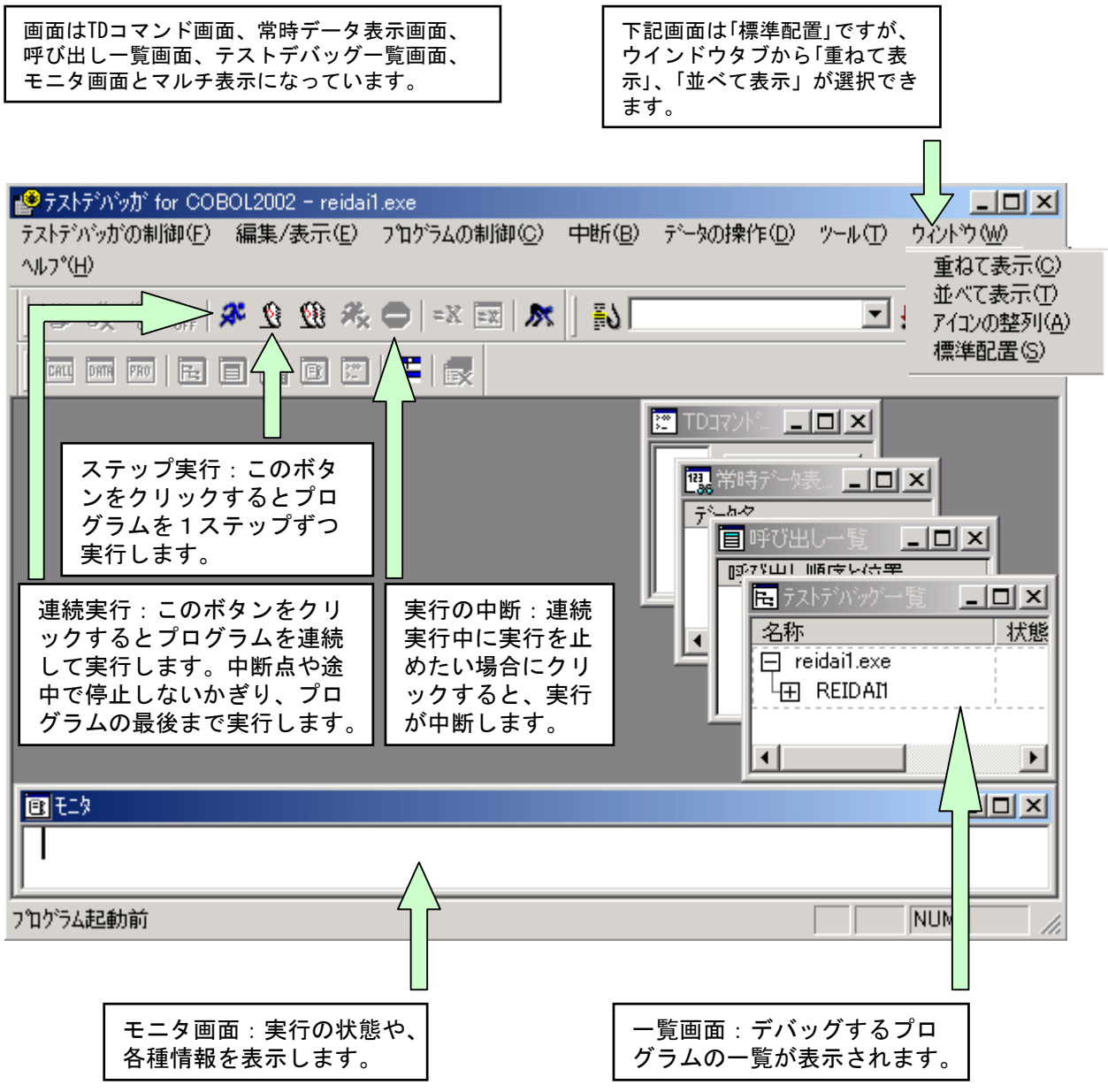
-TDInfオプションを指定して、コンパイル＆リンケージが終わると、テストデバッグツールを使用することができます。（「実行」ボタンをクリックすれば、実行だけすることもできます。）

［手順１］開発マネージャのメニューバーの「ﾋﾞﾙﾄﾞ(B)」をクリックし、プルダウンメニューの中の「ﾃﾞﾊﾞｯｶﾞ(D)」をクリックします。すると、デバッグするプログラムの確認画面が出ますので、「OK」ボタンをクリックします。

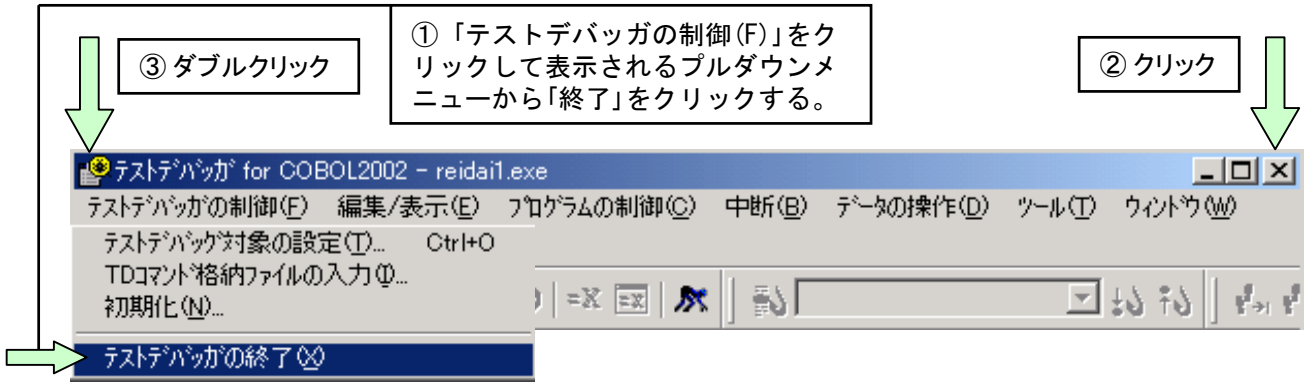




［手順２］以下のような画面が出力されます。主な各部の機能は以下の説明の通りです。基本的には速度調整して実行すれば、デバッグを行っていきます。次の章以降は、デバッグの基本的な操作方法を説明します。



［手順３］デバッグの方法を説明する前に終了方法を説明しておきます。
終了方法は、Windowsの基本操作に沿った３通りの方法があります。

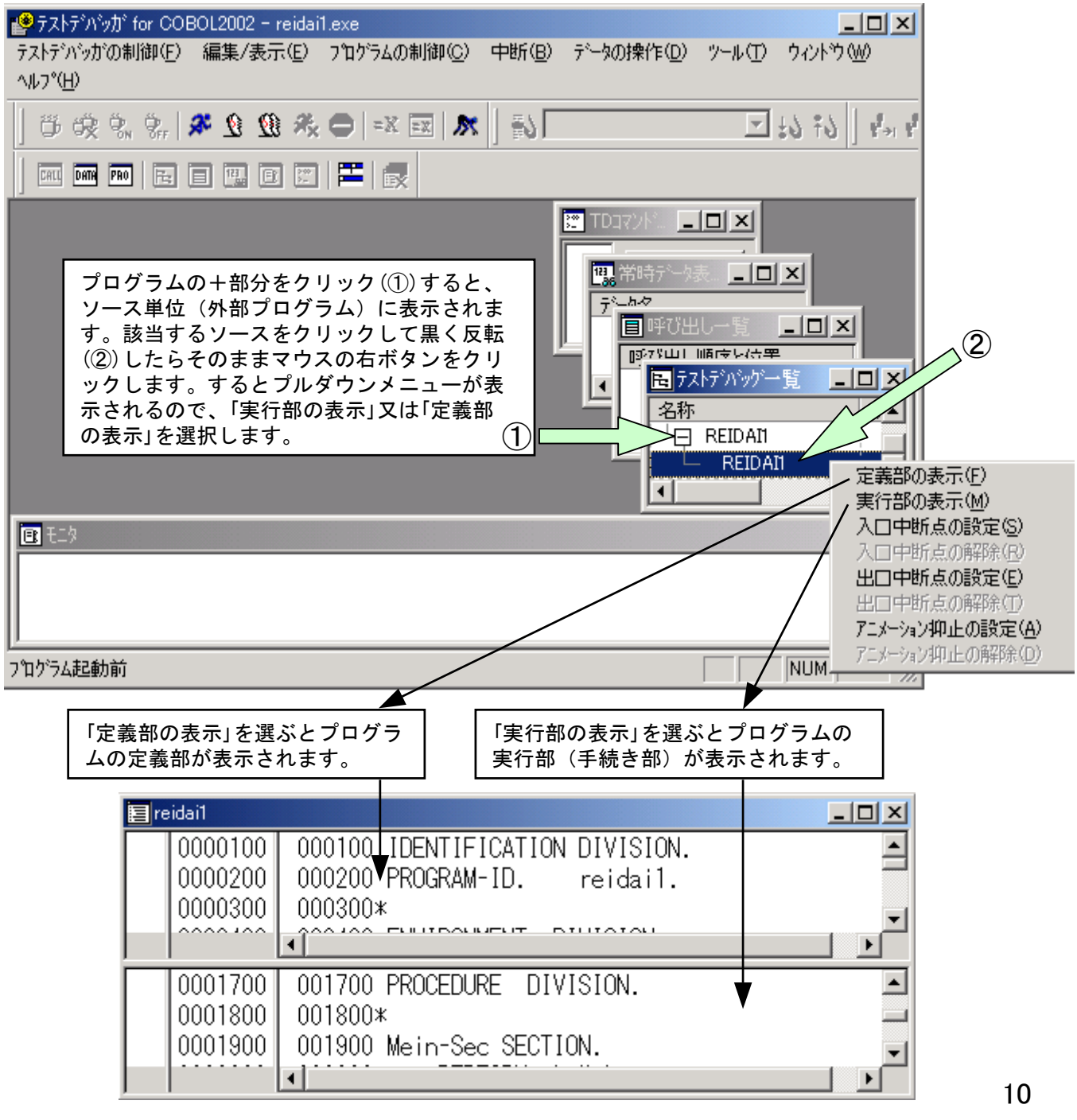


4. ソースの表示方法と中断点の設定

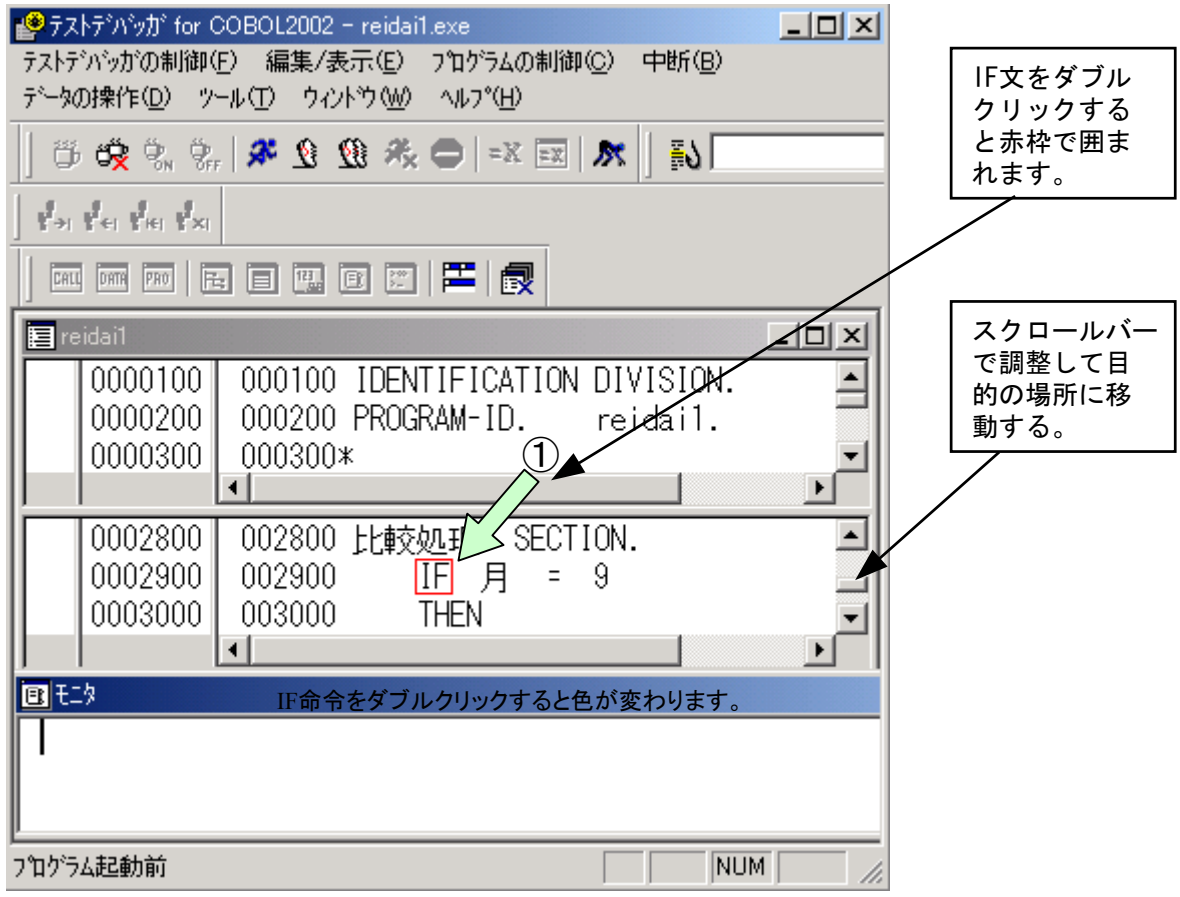
さて、いよいよプログラムのテストデバッグの開始ですが、まず、プログラムの実行部と定義部の表示方法から説明します。

実行部に関しては先に表示しておかなくても、プログラムを実行すれば自動的に表示されます。ただし、事前に中断点等を設定したい場合には、予め表示しておく必要があります。

［手順１］テストデバッグ画面の一覧ウィンドウの中のプログラム名をクリックし、そのままマウスの右ボタンをクリックすると「実行部」と「定義部」のプルダウンメニューが表示されます。これらのどちらかをクリックすると選択したものがそれぞれの画面に表示されます。



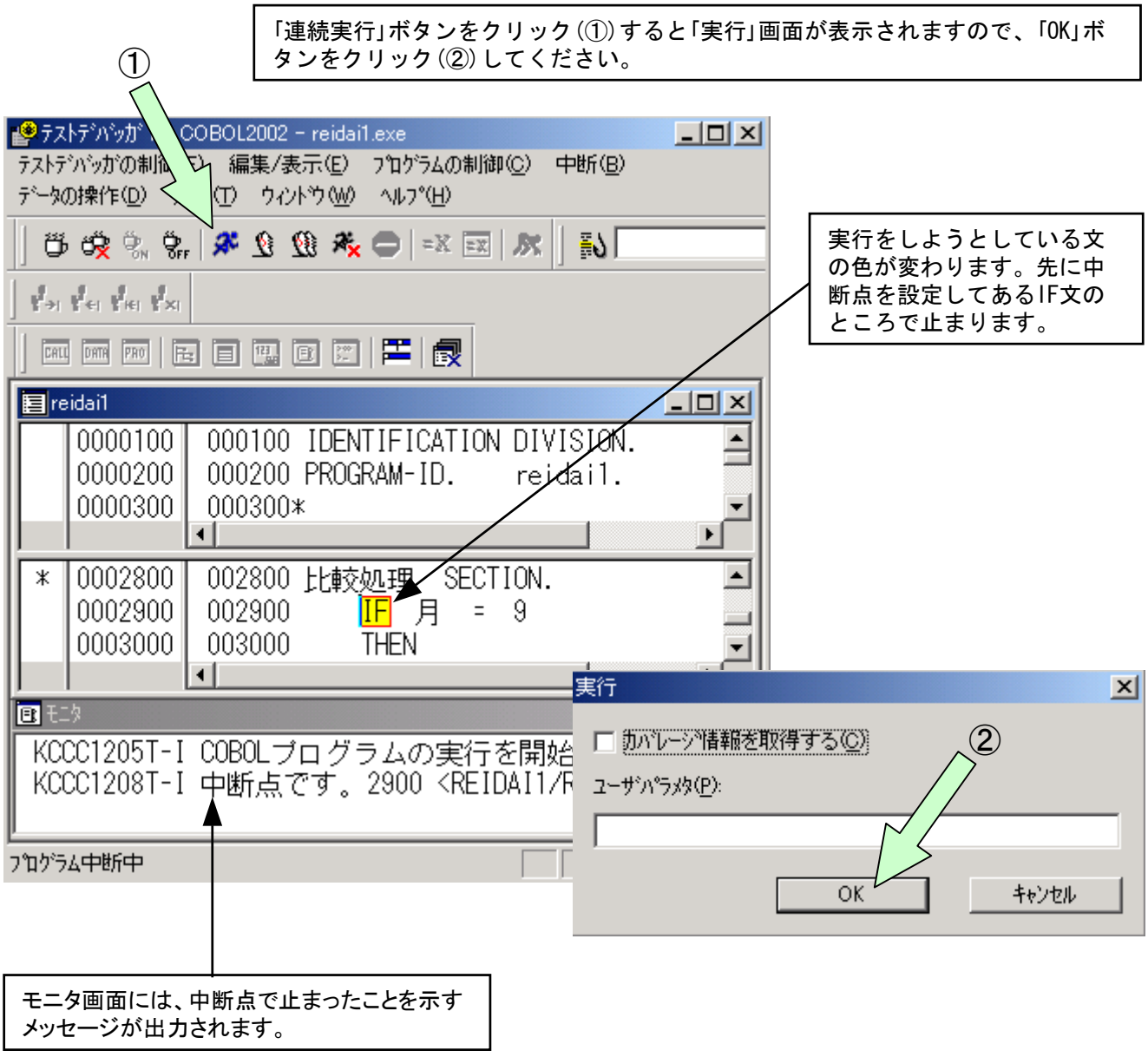
［手順２］中断点を設定します。実行画面のスクロールバーを調整して中断したい文に位置付けます。この例では2900行目のIF文の所まで移動しています。ここで、文字列の「IF」をダブルクリックすると、赤色の枠で囲まれます。これで、中断点を設定できました。この文に制御が渡ってきた時点で（この文を実行する直前で）止まります。



［ワンポイントアドバイス］

このように文やタグ名称をダブルクリックすると、赤色に変わって中断点が設定されます。ここでは、ダブルクリックをして設定しましたが、文等を黒く反転させそのままマウスの右ボタンをクリックするとプルダウンメニューが表示されます。ここで、「中断点の設定/解除」を選んでもできます。なお、設定を解除する場合はもう一度ダブルクリックして、色を戻せば解除されます。

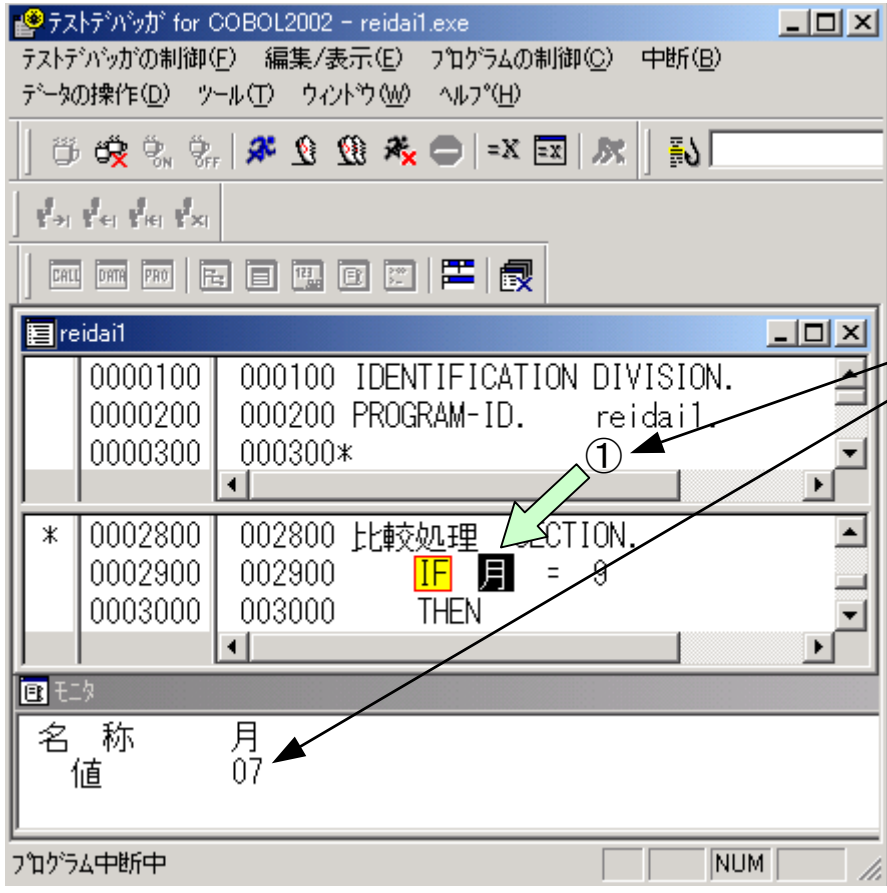
［手順３］中断点はいくつ設けてもかまいませんが、必要最小限にした方がよいと思います。中断点の設定ができたら、「連続実行」ボタンをクリックして、実行してみましょう。



5. データの内容表示と代入

データの内容を表示したり、データに値を設定することができます。

［手順１］データ名称をダブルクリックすると、モニタ画面にデータの内容が表示されます。



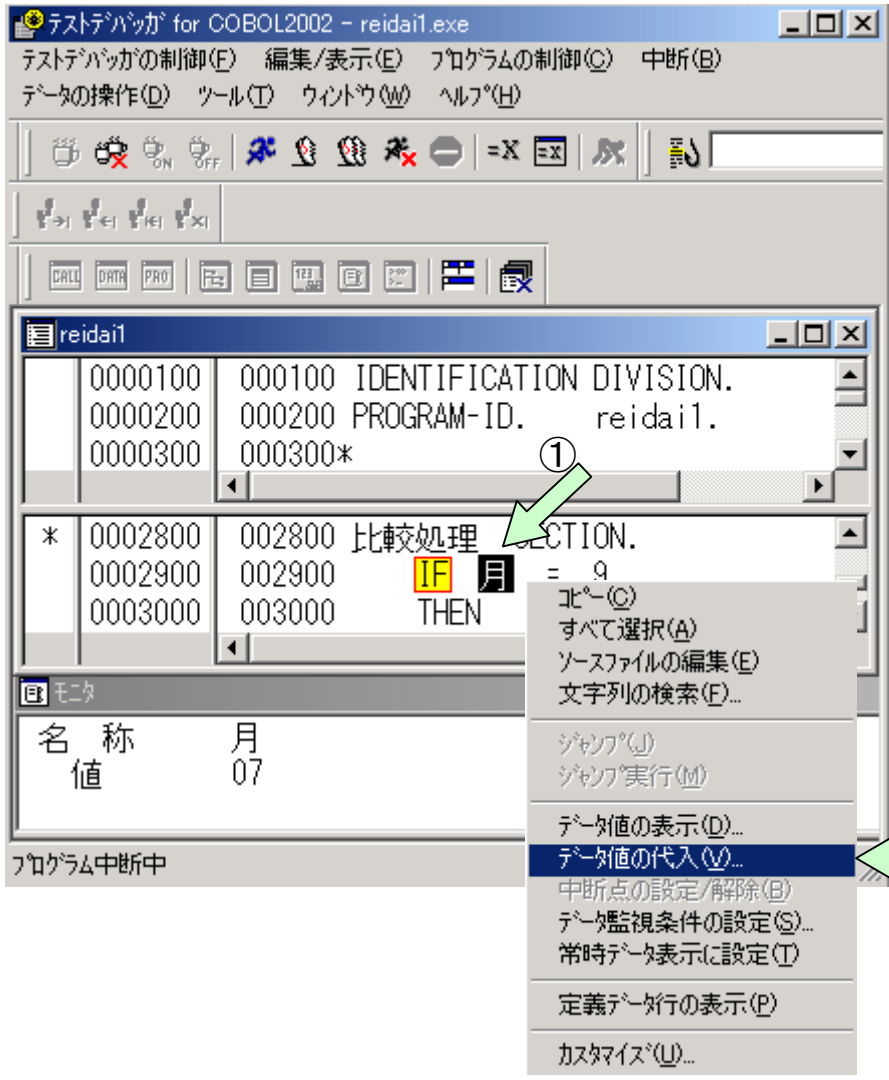
ここでは、データ名「月」をダブルクリックしてみました。

その時のデータの内容がモニタ画面に表示されます。

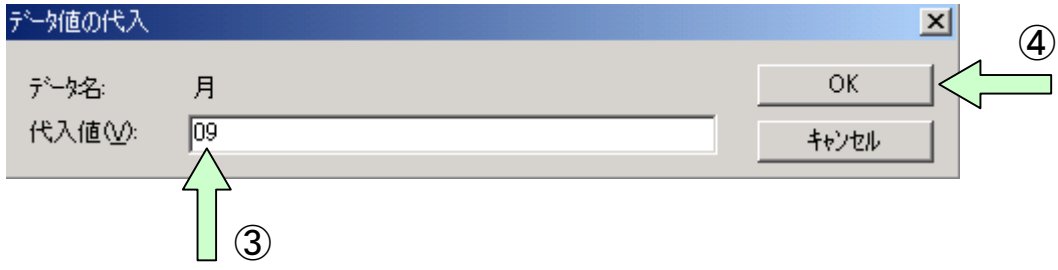
［ワンポイントアドバイス］

このようにデータ名をダブルクリックすると、データの内容が表示されます。ここでは、ダブルクリックをして表示しましたが、データ名をクリックし、マウスの右ボタンをクリックして表示されるプルダウンメニューから「データ値の表示」を選んでもかまいません。

［手順２］データ名に値を設定するには、データ名称をクリックしてそのままマウスの右ボタンをクリックすると、プルダウンメニューが出ますので、その中の「代入」を選択してください。



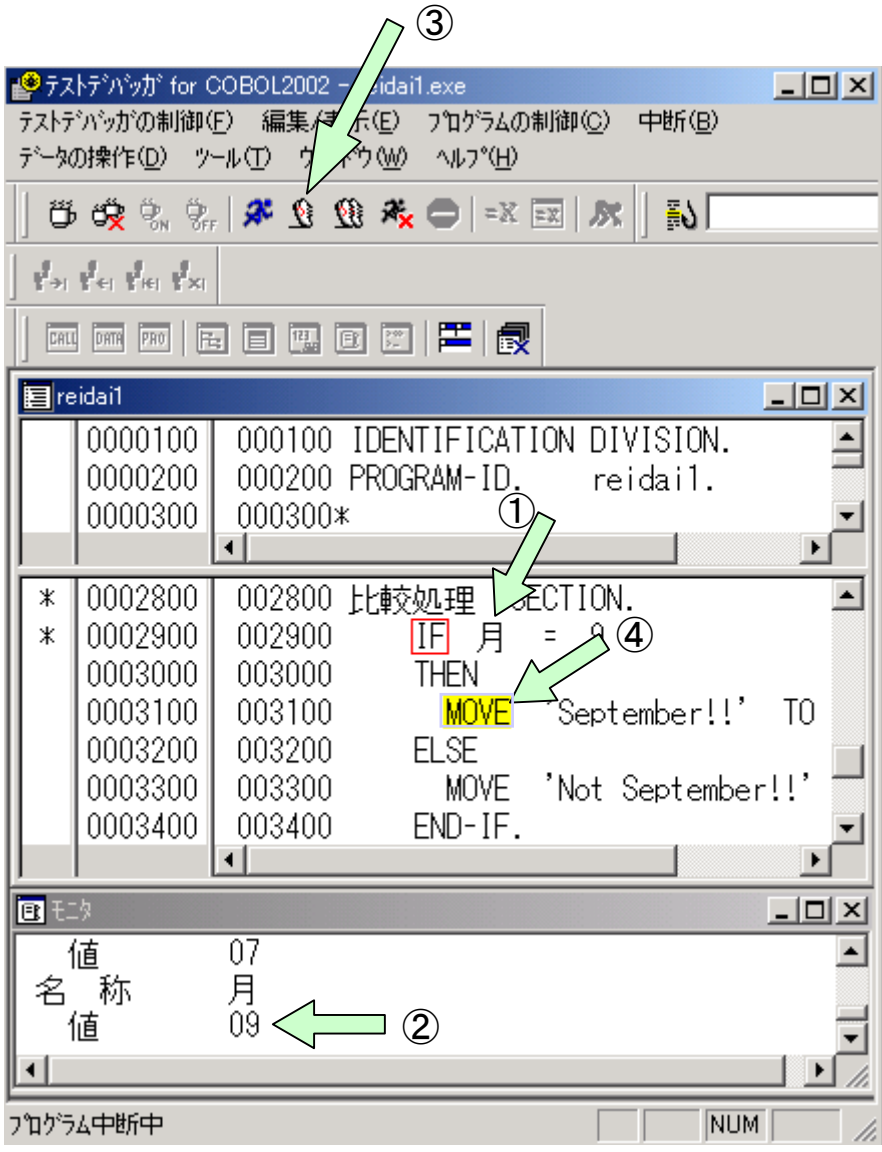
ここでは、データ名「月」をマウスで選択(①)して（位置付けてクリックするだけでもよい）、そのままマウスの右ボタンをクリックするとプルダウンメニューが出力されます。ここで、「データ値の代入」をクリック(②)すると「データ値の代入」画面が出てきます。



代入したい値を入れます。例えば数字項目である「月」に対して値「09」を設定し、「OK」ボタンをクリックします。

データ名が英数字項目の場合は、値を引用符(')で囲んで指定します。

［手順３］データ名に正しく設定されたかチェックして、１ステップだけ実行してみます。（この操作は確認のためにやっているだけです）



データ名「月」をダブルクリック(①)して、表示してみます。
値が「09」である(②)ことを確認します。

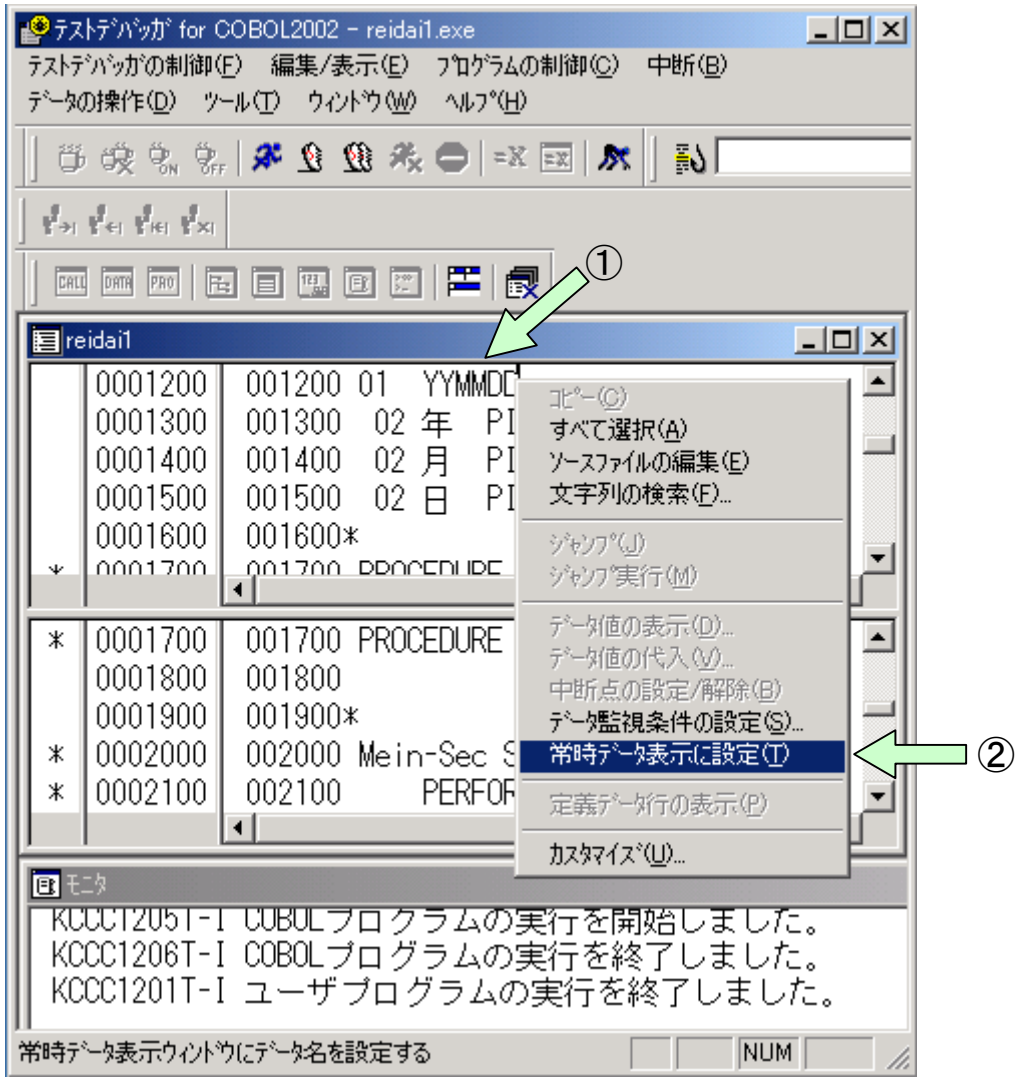
１ステップだけ実行してみます(③)。

実行状態を示す色が１ステップだけ、移動します(④)。

6. データのトレース

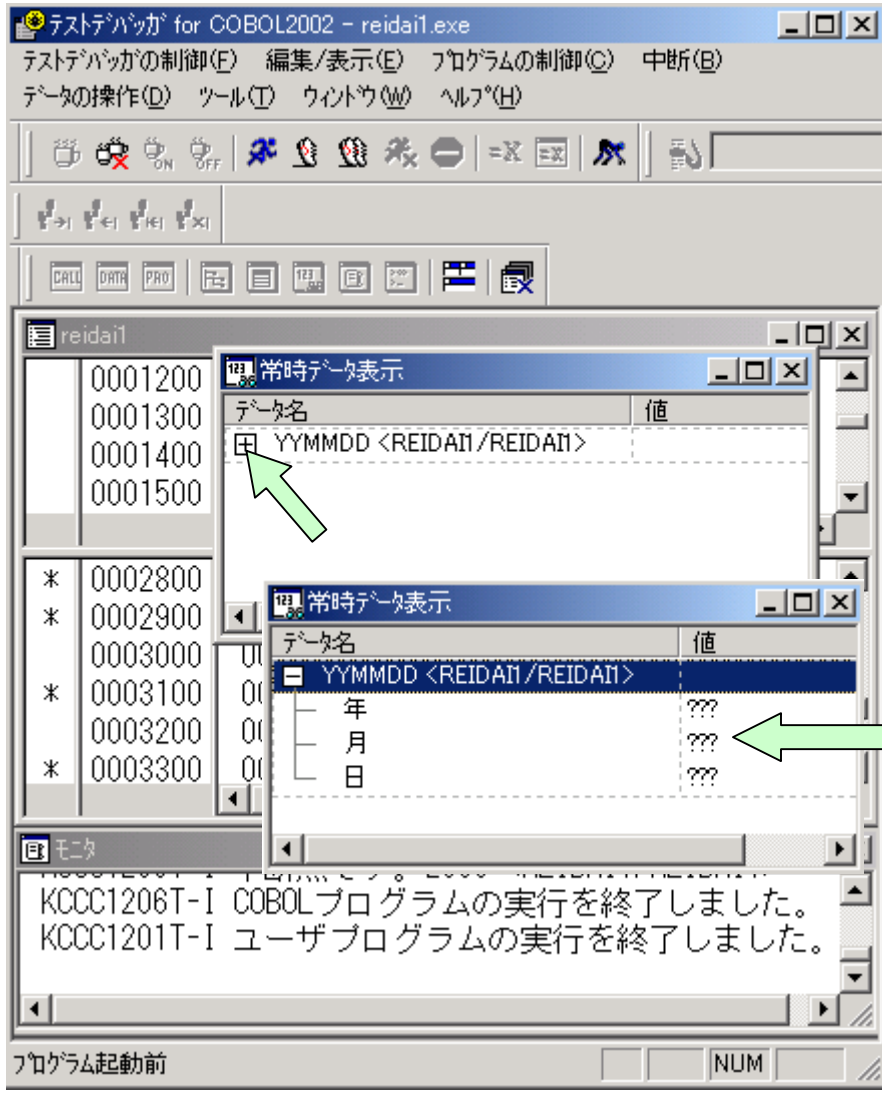
次は、データの状態を常に表示する方法を説明します。

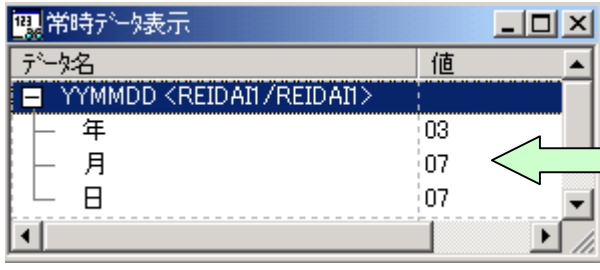
［手順１］定義画面を適当にスクロールして、表示したいデータ名を見つけます。
表示するデータ名をクリックして、そのままマウスの右ボタンをクリックして表示されるプルダウンメニューから「常時データ表示に設定」をクリックします。この例では、データ名「YYMMDD」を常時データ表示します。



［ワンポイントアドバイス］
常時表示を行うデータ名はいくつ指定してもかまいませんが、必要最小限にとどめた方がよいと思います。

［手順２］常時表示画面が表示されてその中に先に指定したデータ名称が表示されます。データ名の＋の箇所をクリックすると、下位項目まで表示されます。この常時表示画面のデータ名の右にデータ名の値が変化した際に値が表示されます。





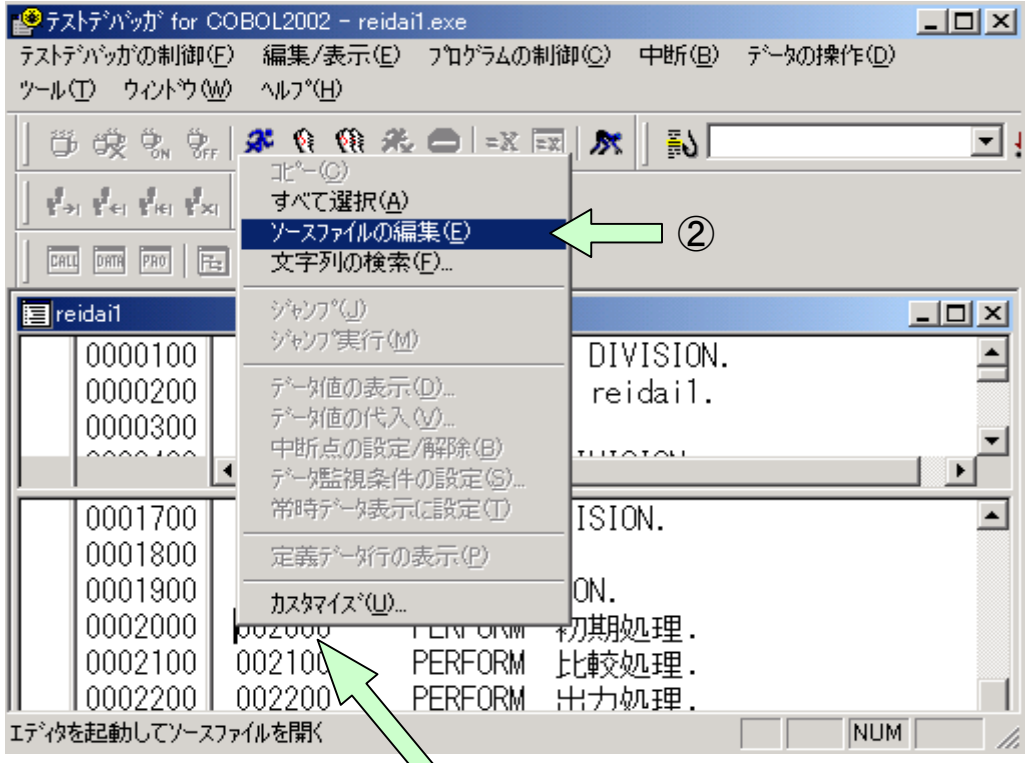
実行ボタンをクリックして実行すると、常時表示画面のデータ名称の値がこのように変化します。

プログラム中で値が変化しないデータ名は何もかわりません。

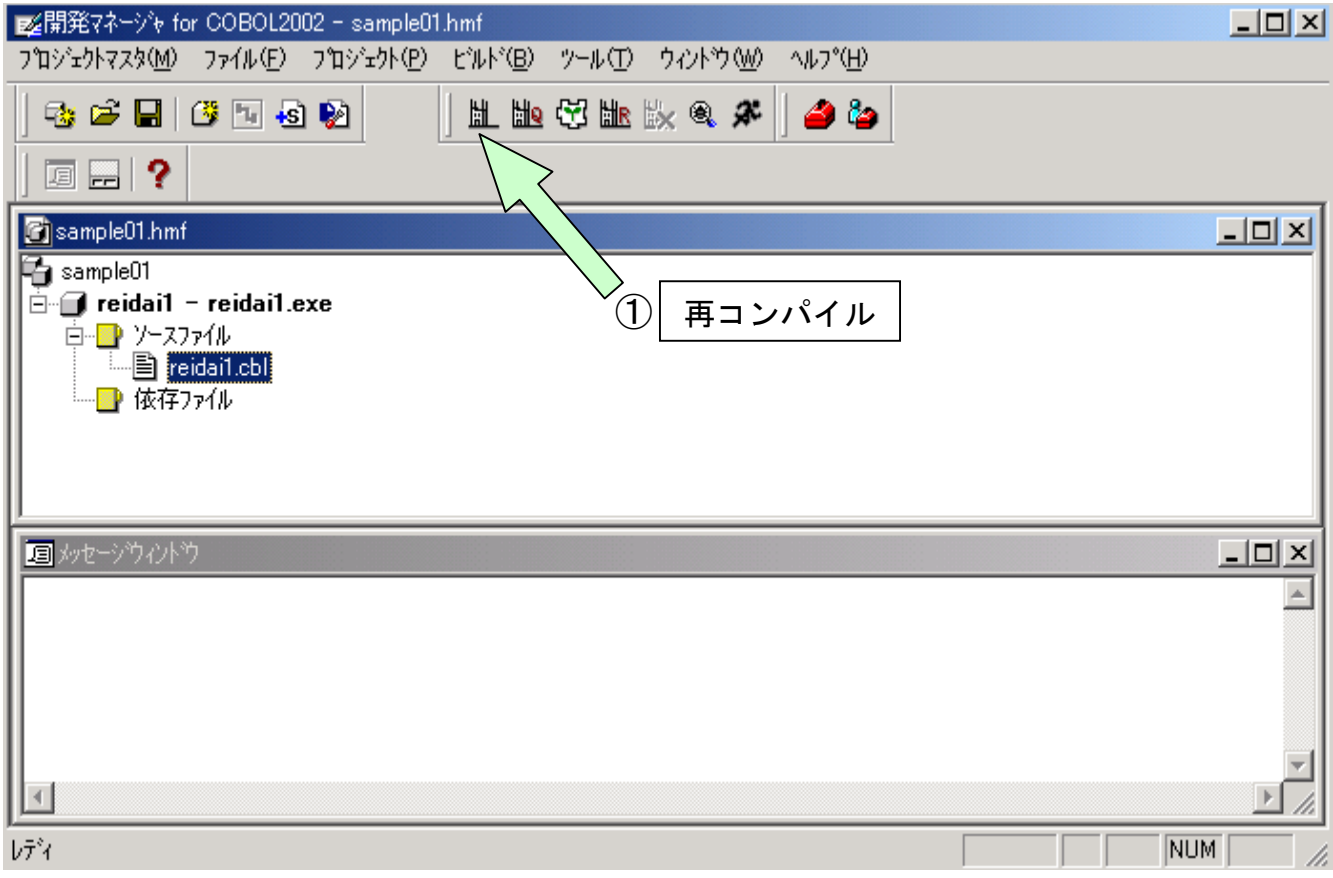
7. エディタとの連携

デバッグ中にプログラムを修正したくなった場合は、以下のやり方で簡単にエディタを自動起動できますので、そこで修正してください。

［手順１］ デバッグ画面中の実行画面・定義画面の該当する行にカーソルを位置付け、右クリックをするとプルダウンメニューが出ます。
そこで、「ソースﾌｧｲﾙの編集(E)」を選択してください。エディタが起動され、該当行にカーソルが位置付きます。



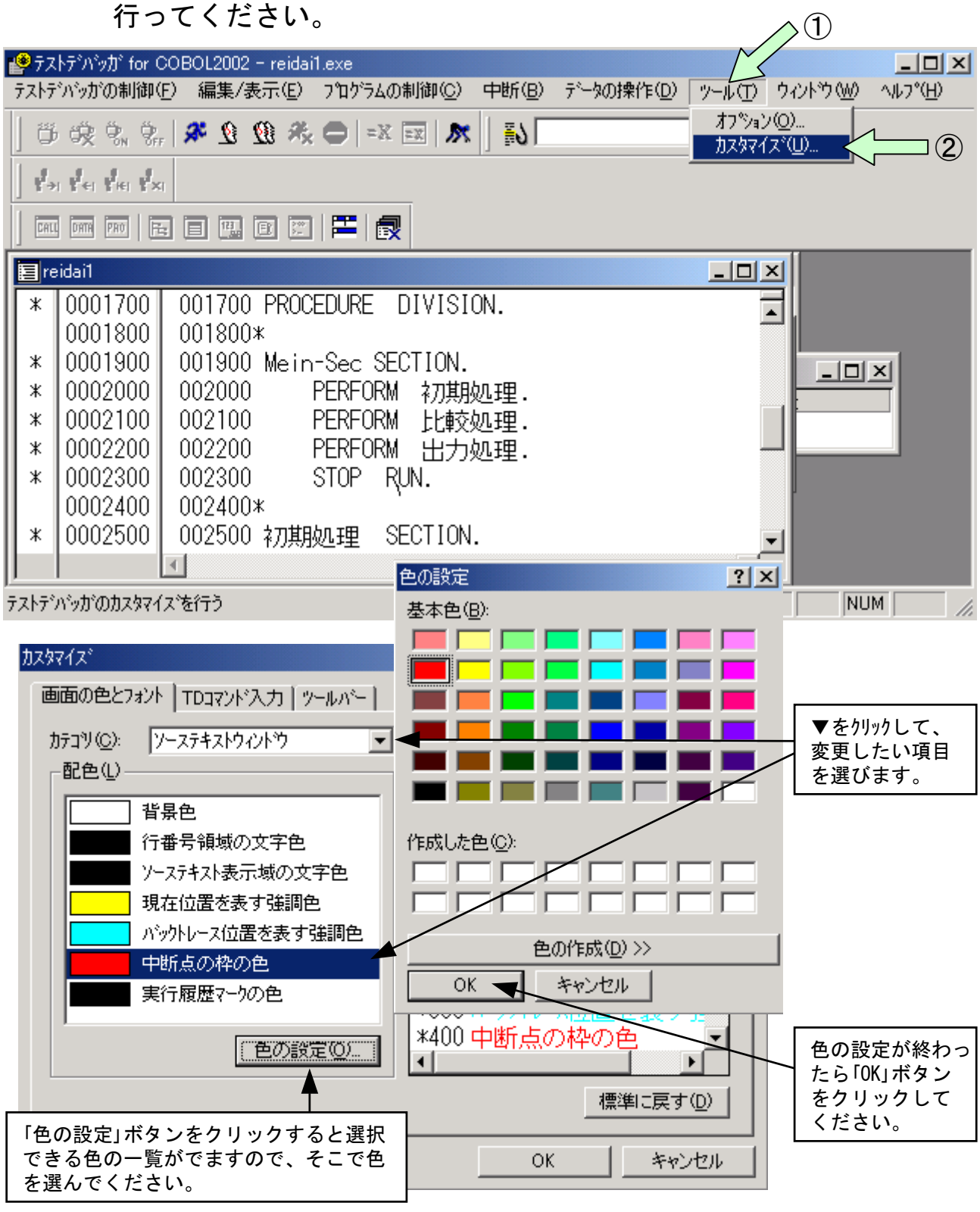
［手順２］修正が完了したら、「上書き保存」してエディタを終了します。
デバッグ画面に戻りますが、デバッグは一旦終了してください。というのは、エディタで修正した部分は現在のデバッグ情報には反映されていませんので、改めてコンパイルし直す必要があるからです。したがって、デバッグを終了して開発マネージャに戻り、再コンパイルしてから再度デバッグをしてください。



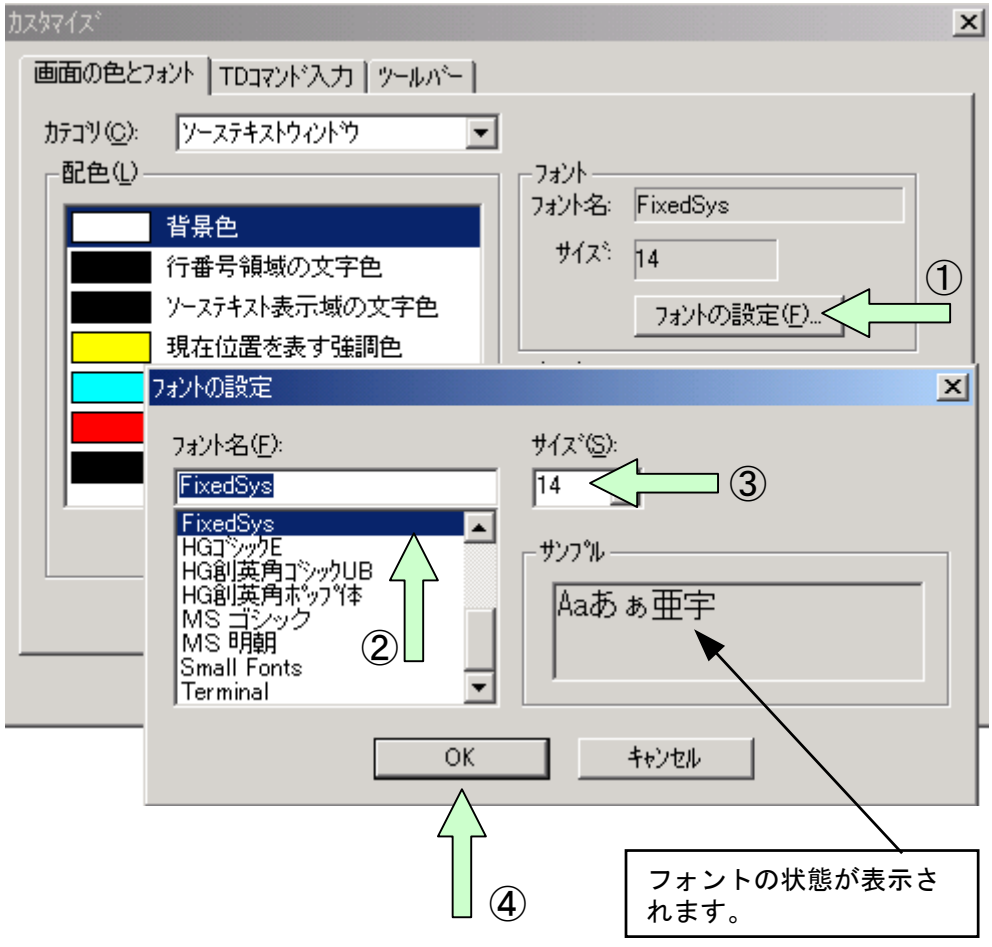
8. デバッガの色などの変更

ここでは、デバッガの画面の配色と、フォントの変更方法を説明します。

［手順１］ デバッグ画面のメニューバーの「ツール(T)」をクリックするとプルダウンメニューが出ます。この中の「カスタマイズ(U)」をクリックするとカスタマイズ画面が出ますので、そこで、色やフォントの設定を行ってください。



フォントの種類やサイズを選んで変更してください。
修正が終わったら「OK」ボタンをクリックしてください。



9. カバレージ情報の蓄積と表示

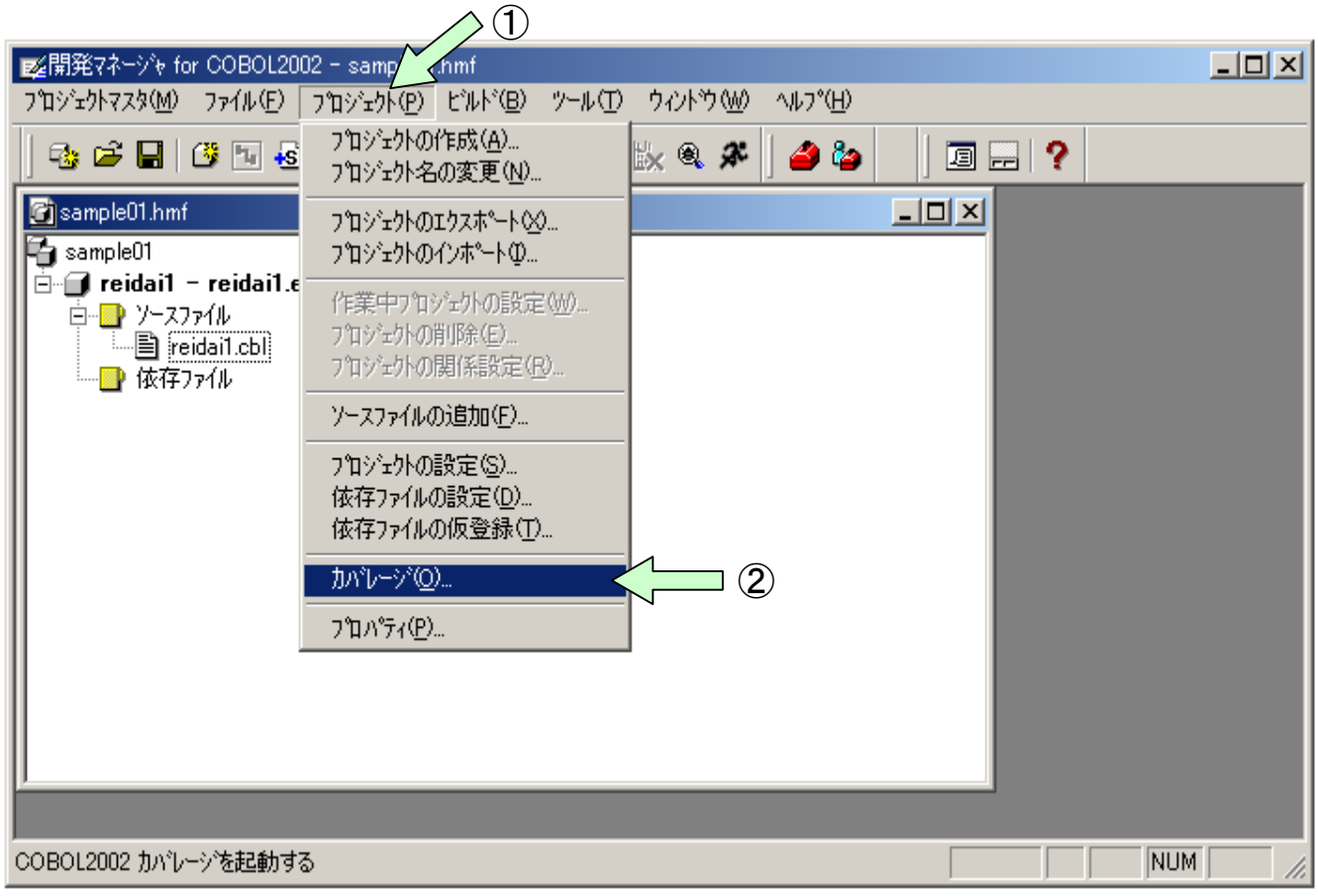
カバレージとは、テスト進捗状況を定量的に把握する機能です。カバレージ情報を採取するには、コンパイラオプション-CVInfを指定してコンパイルします。カバレージ情報には、次の3種類の指標があります。

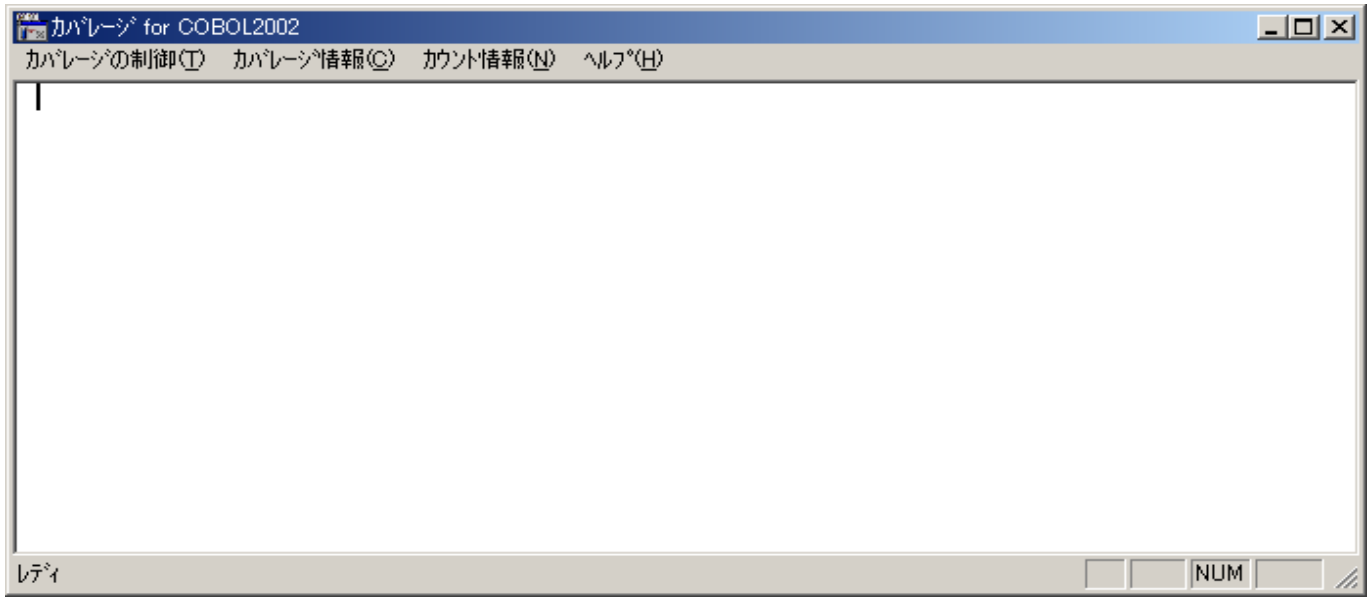
①C0メジャー：実行した文の割合を示します。
C0メジャー＝(実行が済んだ文の数) / (実行文の数)×100 (%)
②C1メジャー：分岐する個所で、実行した分岐先の割合を示します。
C1メジャー＝(実行が済んだ分岐先の数) / (分岐先の数) ×100 (%)
③S1メジャー：実行した呼び出し文(CALL文やINVOKE文)の割合を示します。
S1メジャー＝(実行が済んだ呼び出し文の数) / (呼び出し文の数) ×100 (%)

これらの情報は、プログラム情報ファイルに蓄積されます。プログラム情報ファイルは、実行可能ファイルと同じフォルダに作成されます。

［ワンポイントアドバイス］
プログラム情報ファイルを実行可能ファイルとは別のフォルダに作成したい場合は、コンパイル時の環境変数CBLPIDIRでフォルダを指定します。コンパイル時の環境変数は、コンパイラオプションを指定するときと同様に「プロジェクトの設定(S)」をクリックし、コンパイラオプション一覧の中から「環境変数」タブをクリックして設定します。

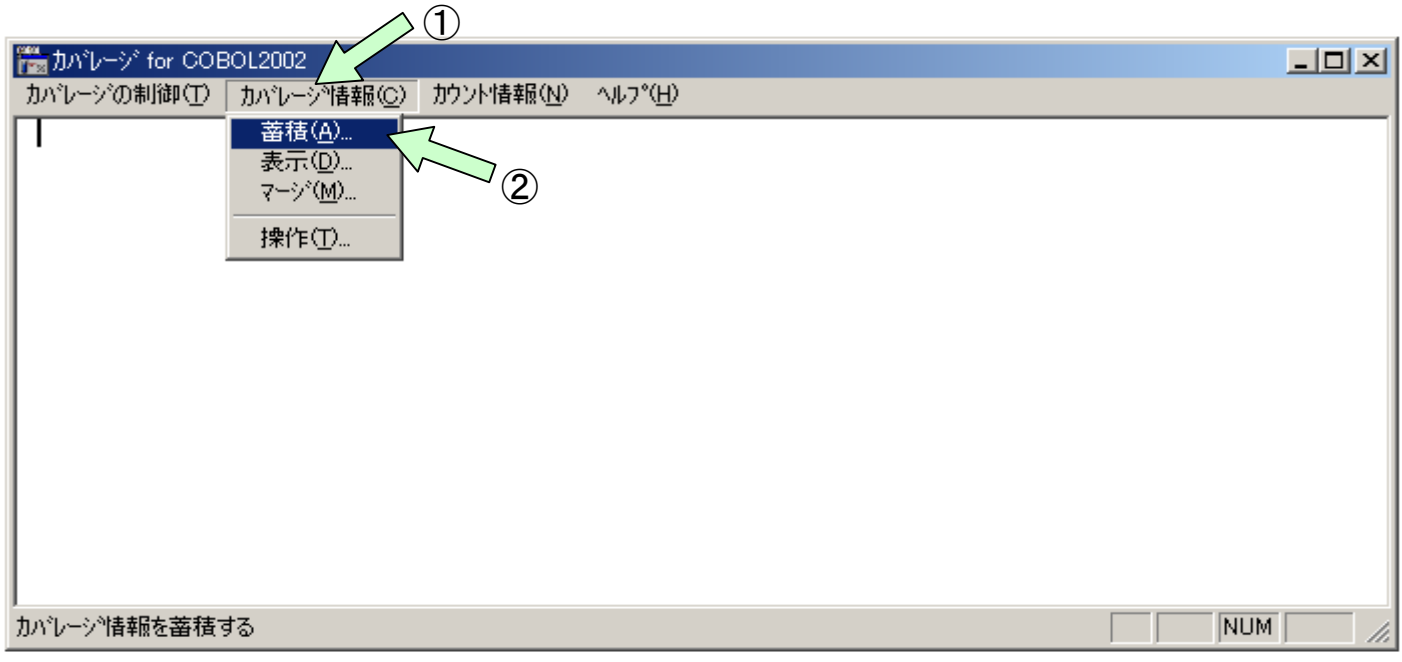
［手順1］ カバレージ情報の蓄積
開発マネージャのメニューバーの「プロジェクト(P)」をクリックし、プルダウンメニューの中の「カバレージ(O)」をクリックします。すると、カバレージのウインドウが開かれます。

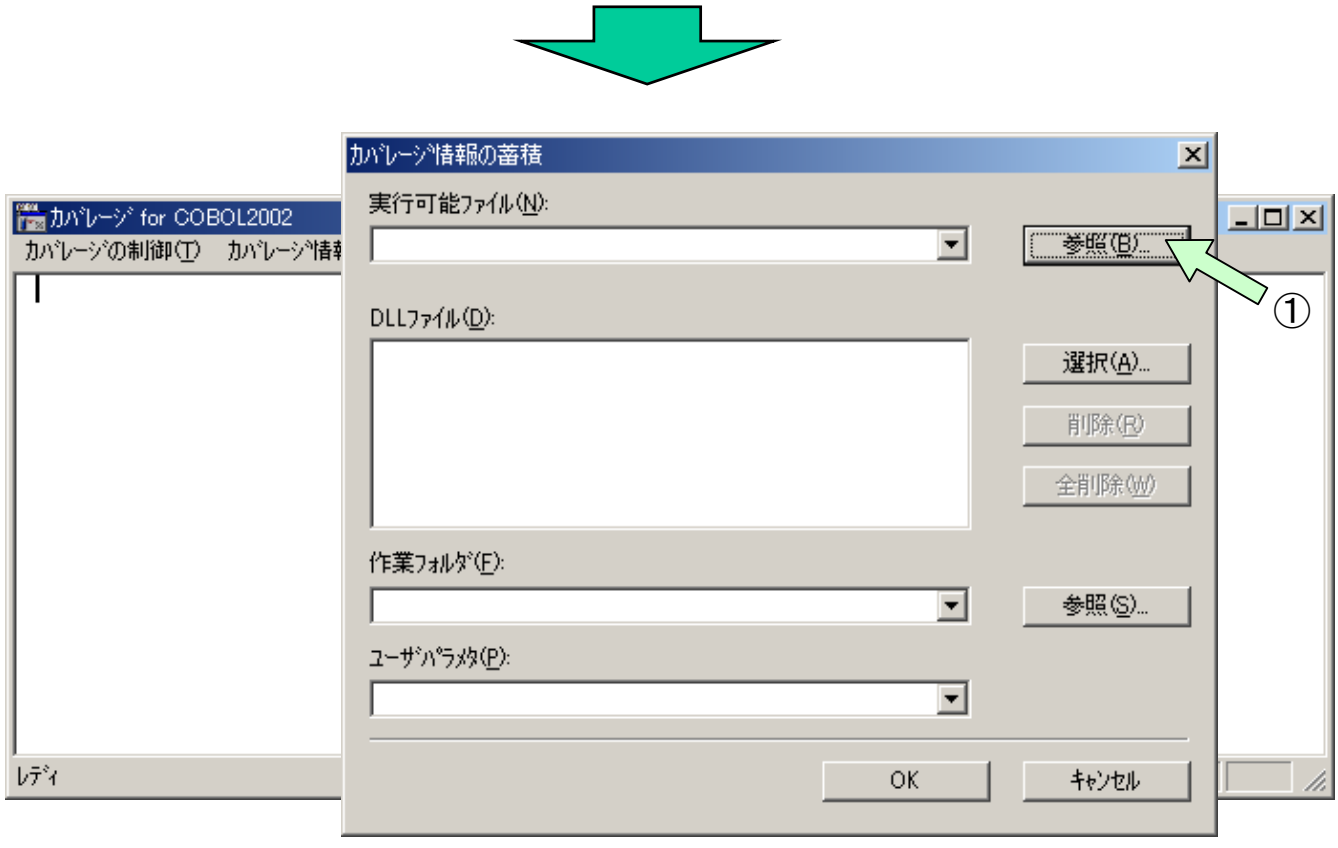


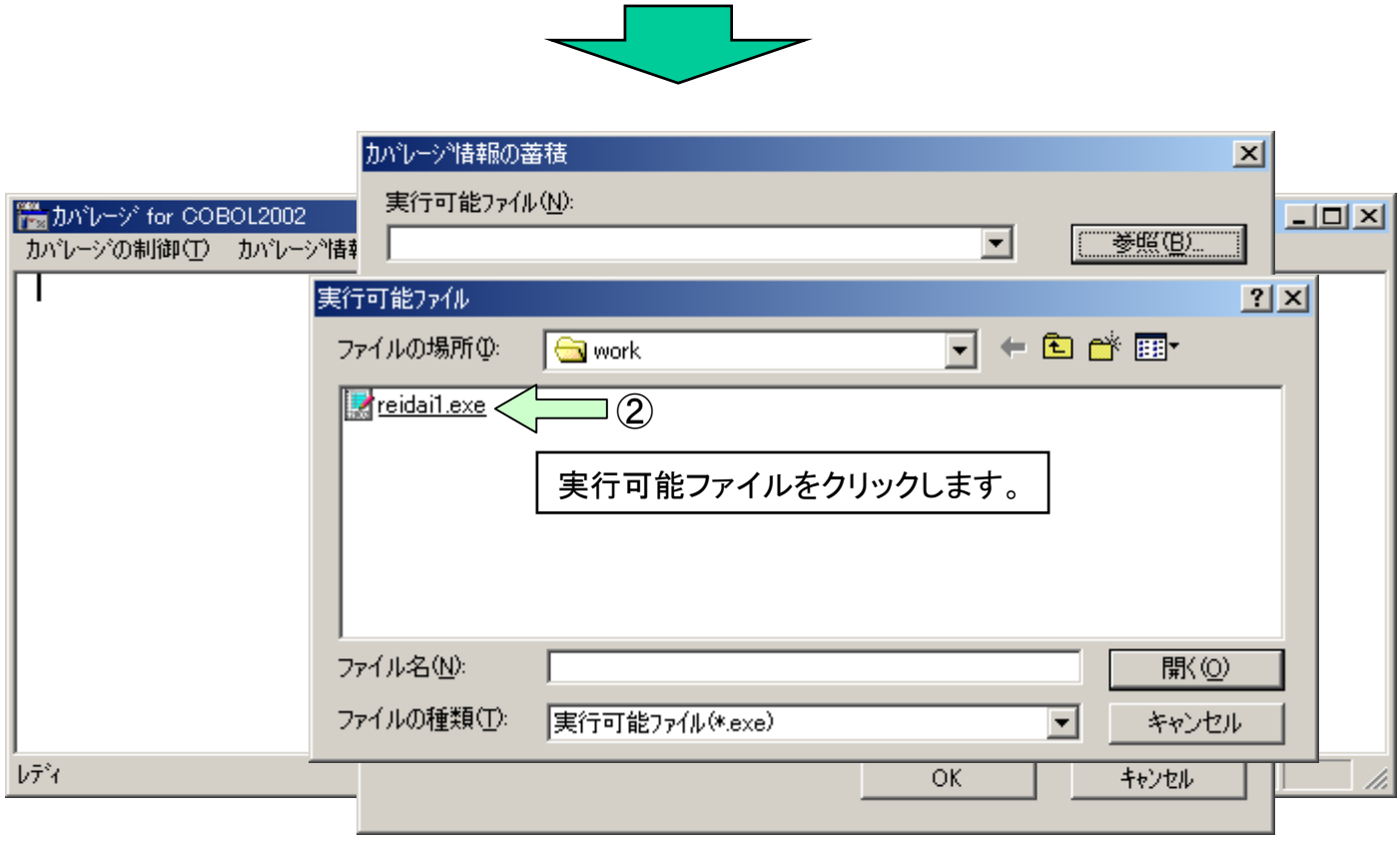


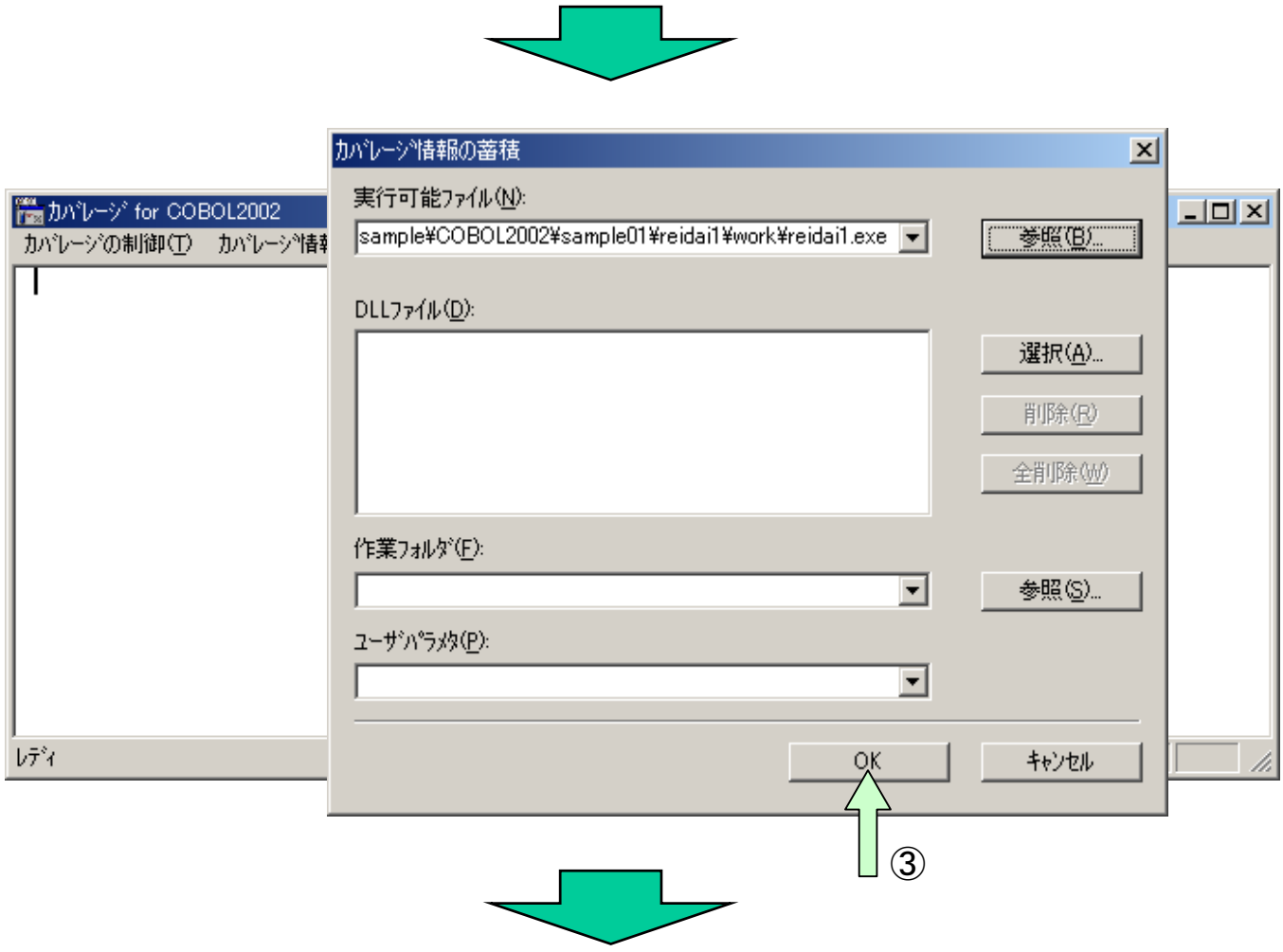
［手順２］ カバレージ情報の蓄積
カバレージウインドウのメニューバーの「カバレージ情報(C)」をクリックし、プルダウンメニューの「蓄積(A)」をクリックすると、「カバレージ情報の蓄積」画面が表示されます。「実行可能ファイル(N)」の参照ボタンで実行可能ファイルを指定します。「OK」ボタンをクリックすると、プログラムが実行されカバレージ情報が蓄積されます。

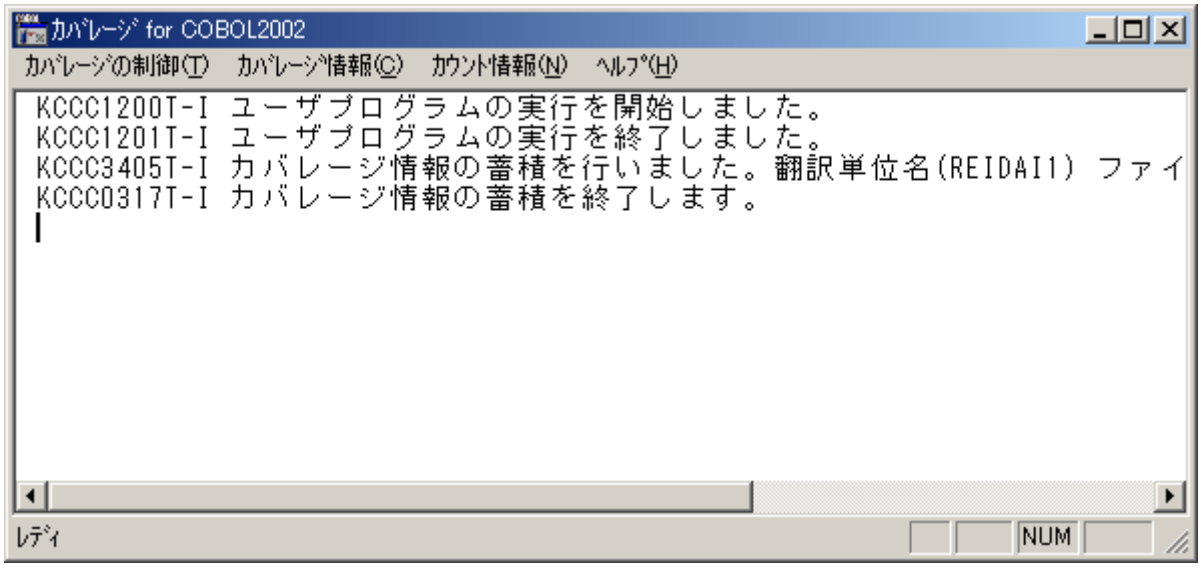
［ワンポイントアドバイス］
デバッガからの実行でもカバレージ情報を蓄積することができます。





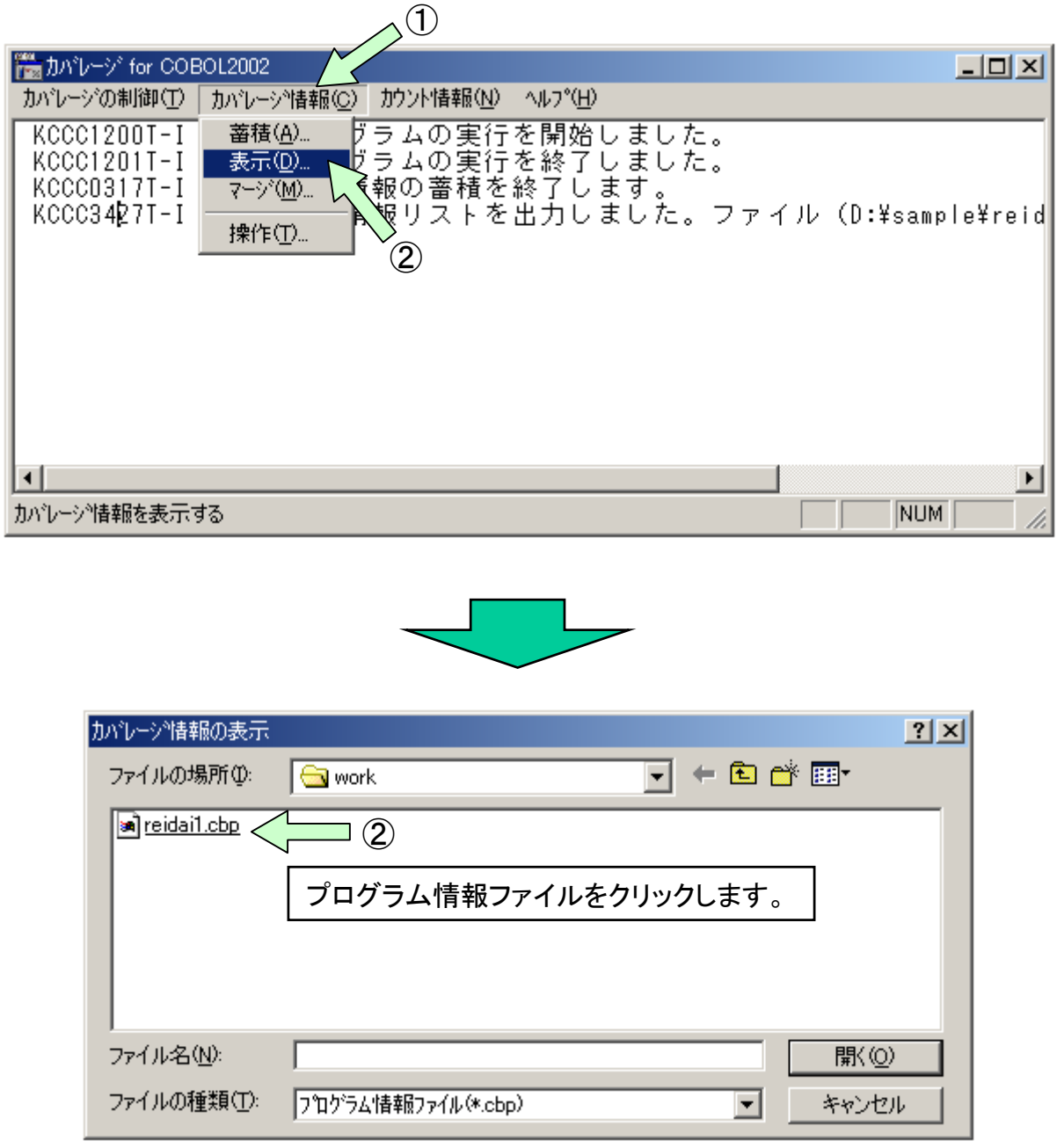






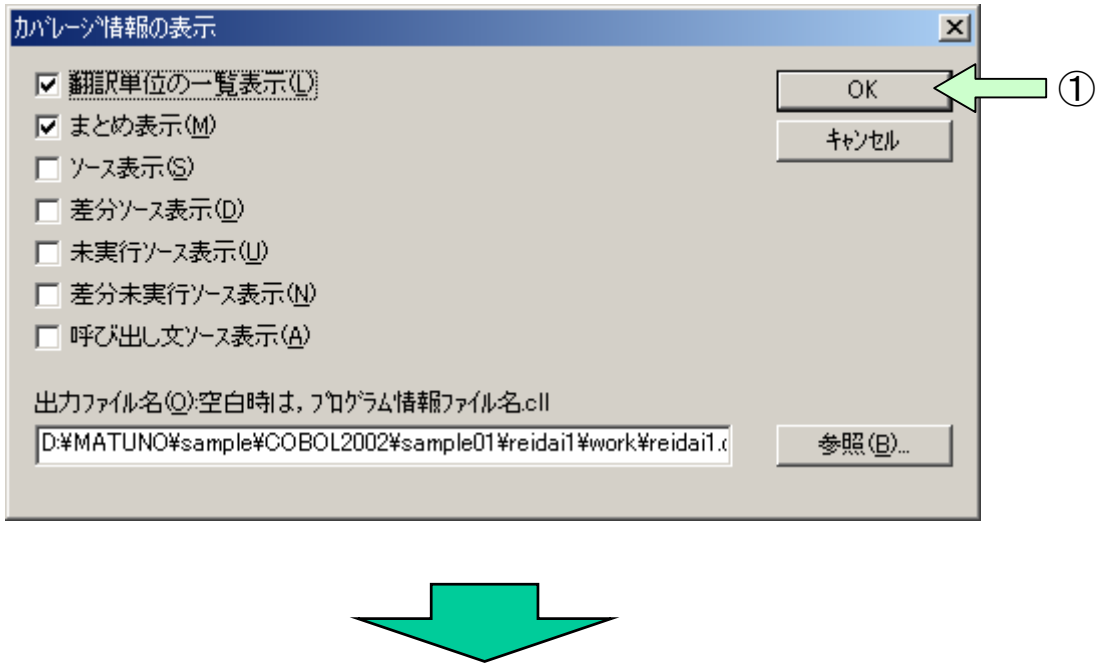
［手順３］ カバレージ情報の表示

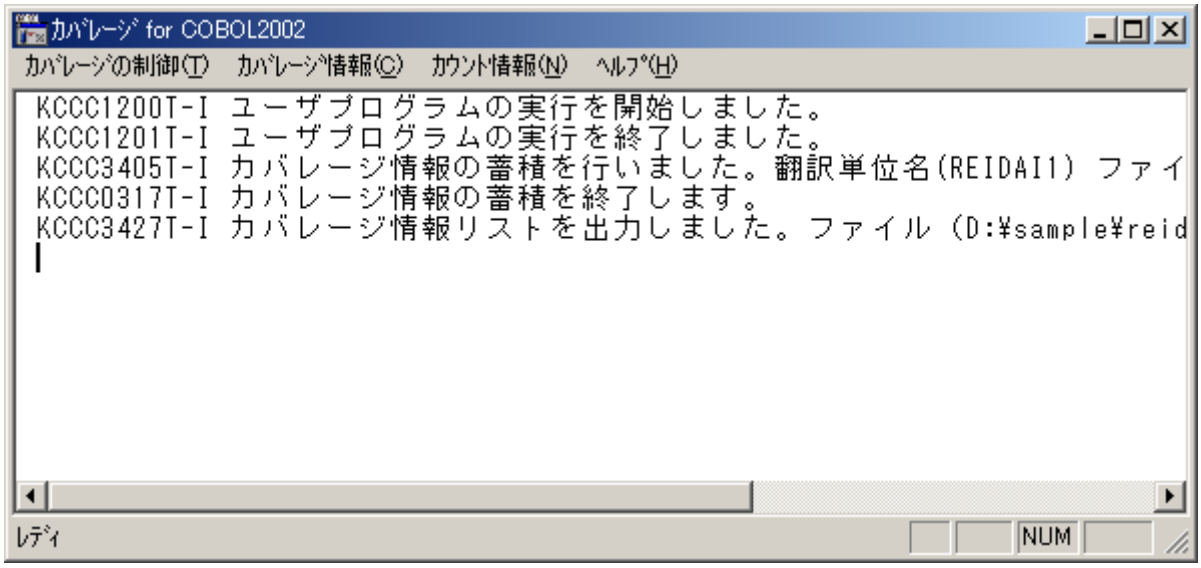
カバレージウインドウのメニューバーの「カバレージ情報(C)」をクリックし、プルダウンメニューの「表示(D)」をクリックします。すると、「カバレージ情報の表示」画面が表示されるので、プログラム情報ファイル(.cbp)を指定します。



［手順４］ カバレージ情報の表示

「カバレージ情報の表示」画面の中の表示したい項目をクリックします。ここでは、「翻訳単位の一覧表示」と「まとめ表示」をクリックし、「OK」ボタンをクリックします。



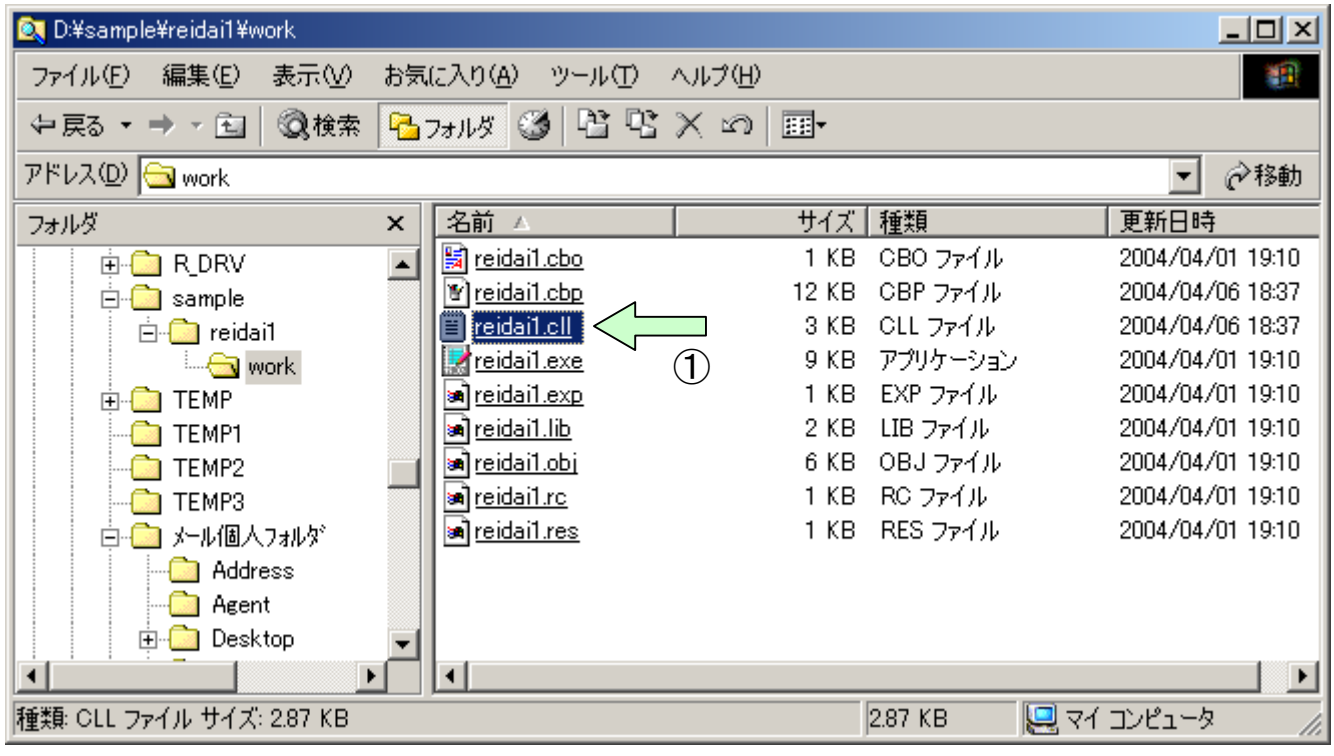


［手順５］カバレージ情報の表示

実行可能ファイルと同じフォルダに.cllという拡張子のファイルが生成されています。このファイルをCOBOLエディタやメモ帳で開いて、カバレージ情報を見ることができます。カバレージ情報の表示例を次ページに示します。

［ワンポイントアドバイス］

同じ条件で複数回実行してもテスト回数は１回として扱われます。実行ルートが異なるテストをする度にカバレージ情報は蓄積されます。



[カバレージ情報の表示例]

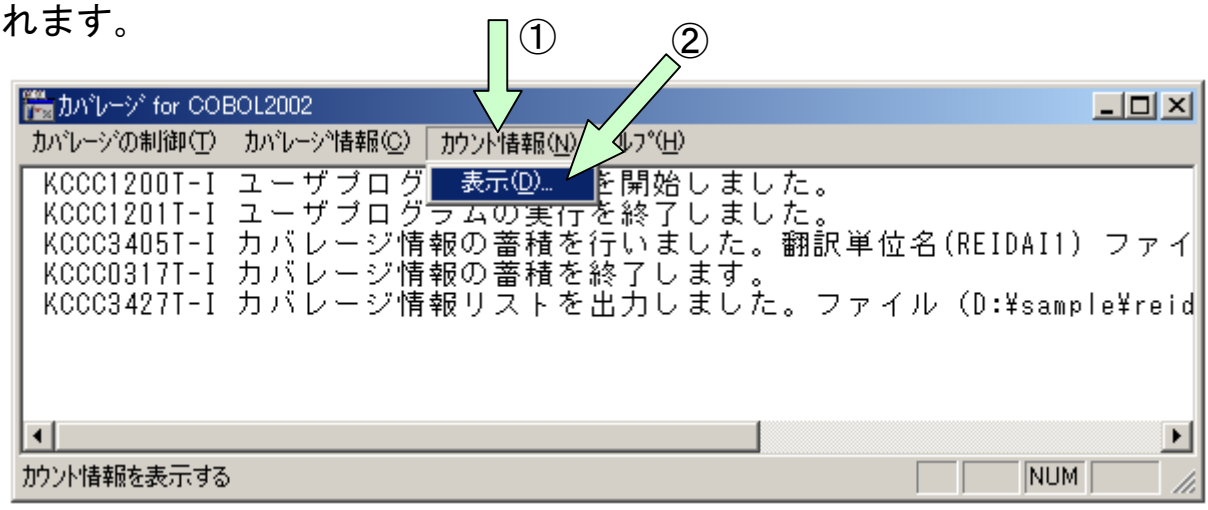
「まとめ情報」の例を次に示します。

```
                          ****************************
                          *      カバレージ情報      *
    COBOL2002 (X)  01-01  ****************************        2004-04-06 18:37:15
------------------------------------------------------------------------------------------

  プログラム名   : REIDAI1
  コンパイル日時: 2004-04-01 19:10:39                          変更回数     :     0
  テスト日時     : 2004-04-06 18:37:03                          テスト回数   :     1

*****************************************************************************************
*                    <C0>    <差分C0>    <C1>    <差分C1>    <S1>    <差分S1>     *
*     対象総数          9           0         2           0         0           0     *
*     実行済数          8           0         1           0         0           0     *
*     未実行数          1           0         1           0         0           0     *
*     カバレージ率   88.8%        0.0%      50.0%       0.0%      0.0%        0.0%     *
*****************************************************************************************

                                  ********************************************
                                  *    カ バ レ ー ジ 情 報 一 覧    *
    COBOL2002 (X)  01-01          ********************************************        2004-04-06 18:37:15

                  *----------C0----------* *--------差分C0--------* *----------C1----------* *--------差分C1--------*
番号  名  称   種別 対象総数 実行済数   C0   対象総数 実行済数 差分C0 対象総数 実行済数   C1   対象総数 実行済数 差分C1
   1 REIDAI1     P        9        8  88.8%        0        0   0.0%        2        1  50.0%        0        0   0.0%

*****************************************************************************************************************************
*      合計              9        8  88.8%        0        0   0.0%        2        1  50.0%        0        0   0.0%*
*****************************************************************************************************************************
```

10. カウント情報の表示

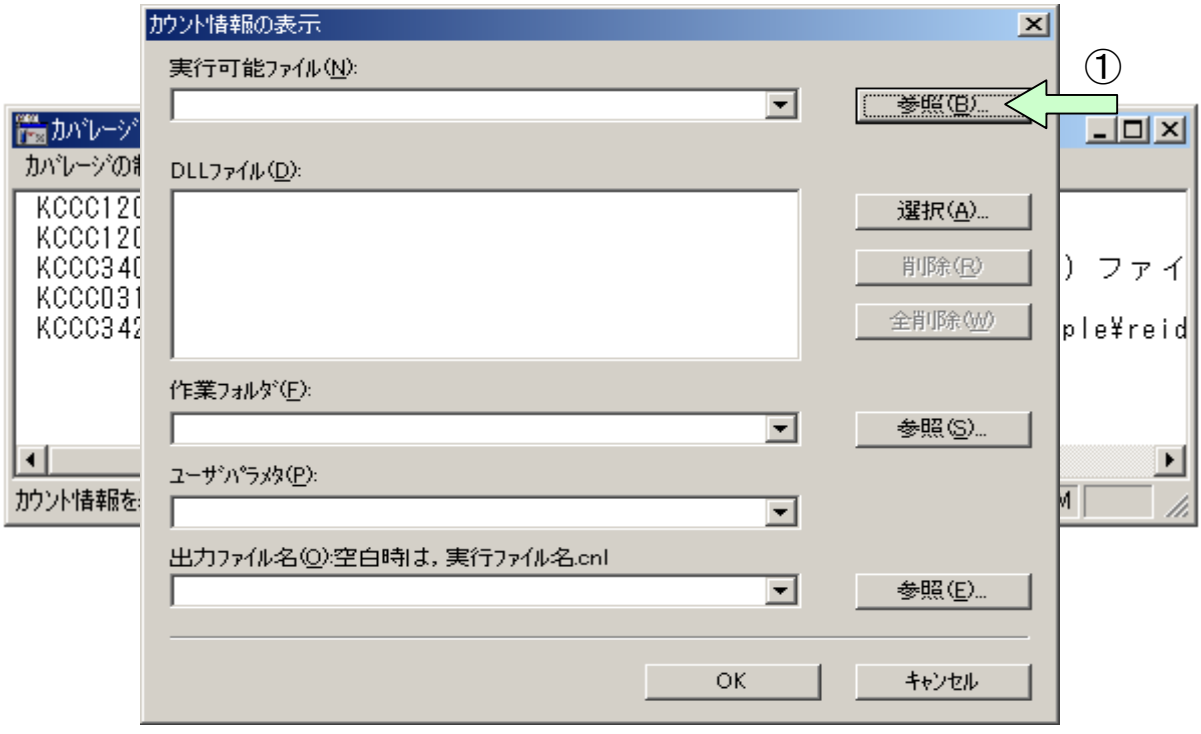
カウント情報は、プログラム中の文の実行回数を示します。

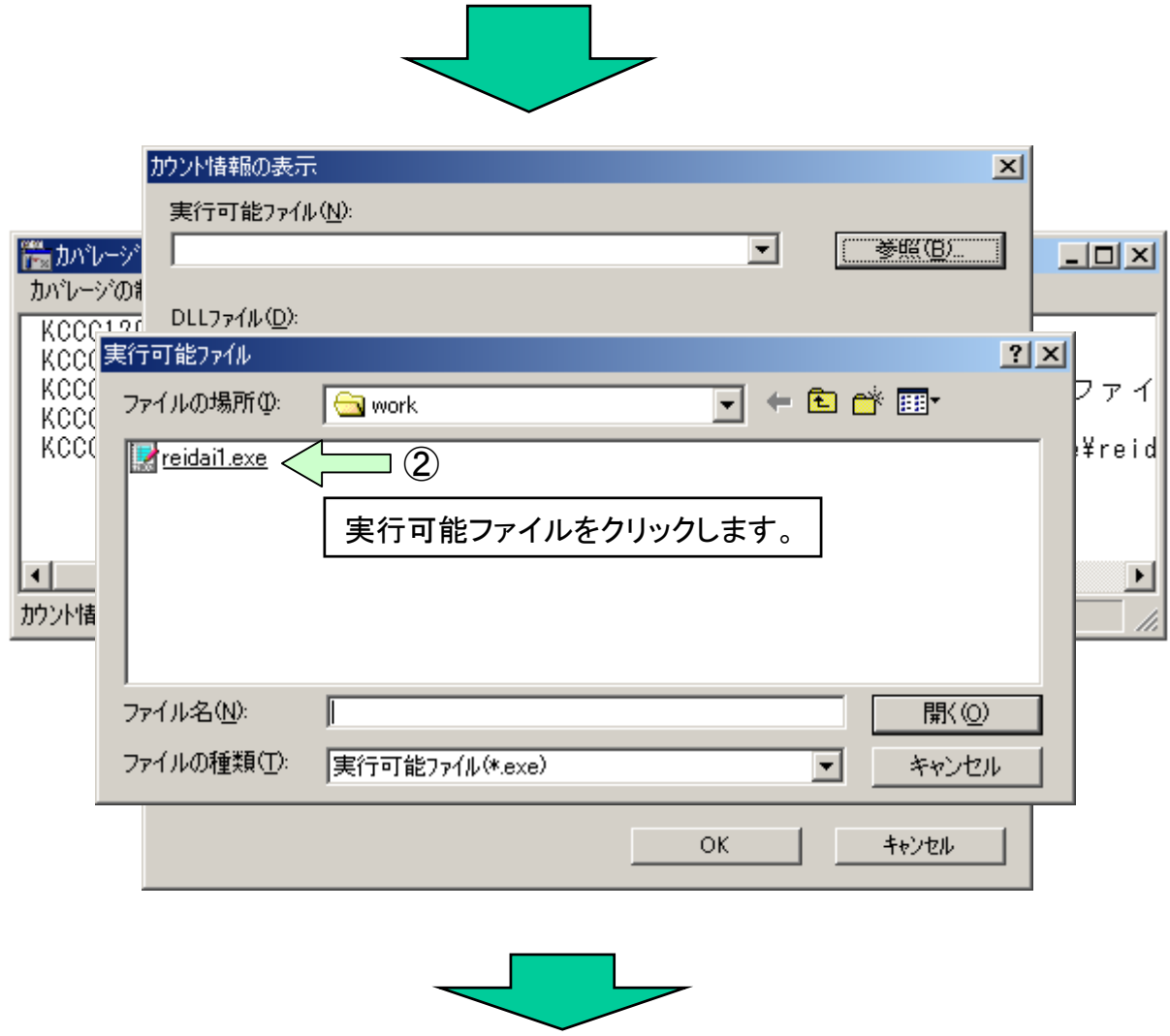
［手順１］ カウント情報の表示
カバレージウインドウのメニューバーの「カウント情報(N)」をクリックし、プルダウンメニューの「表示(D)」をクリックすると、「カウント情報の表示」画面が表示されます。

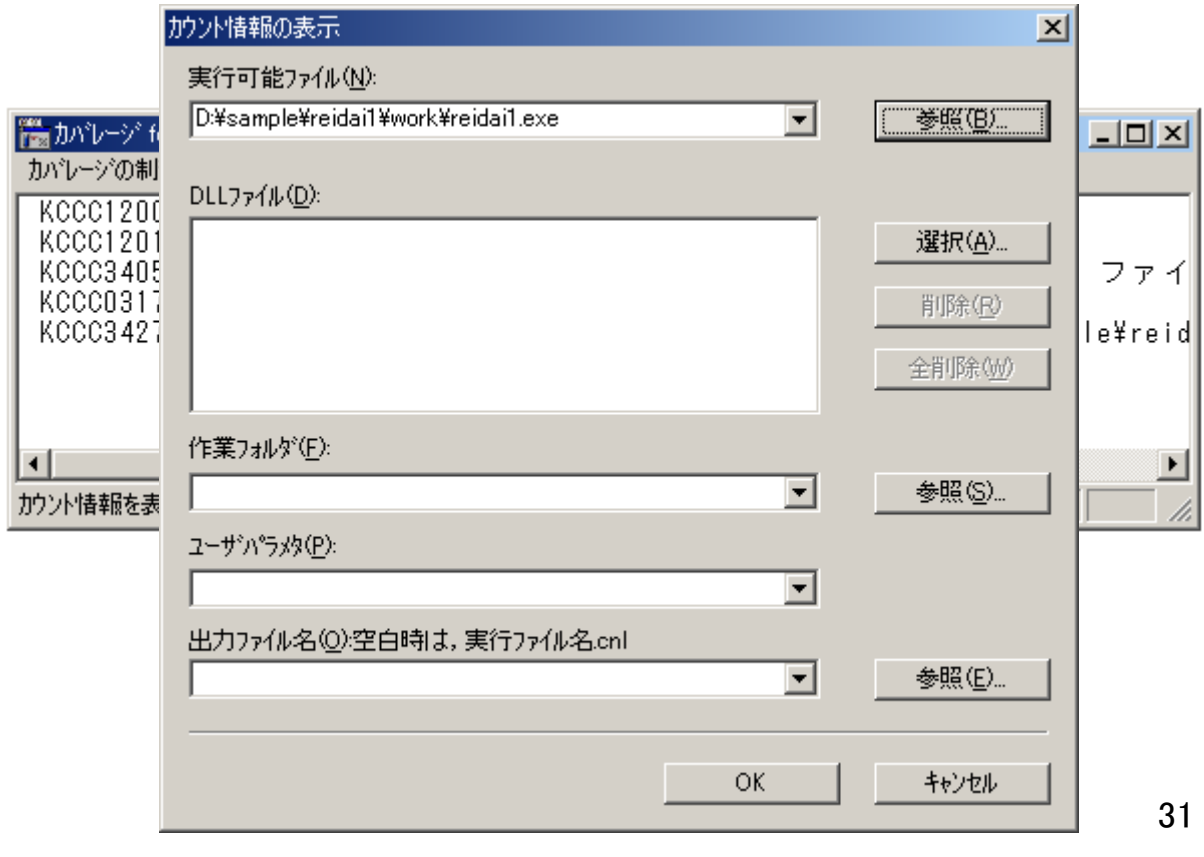


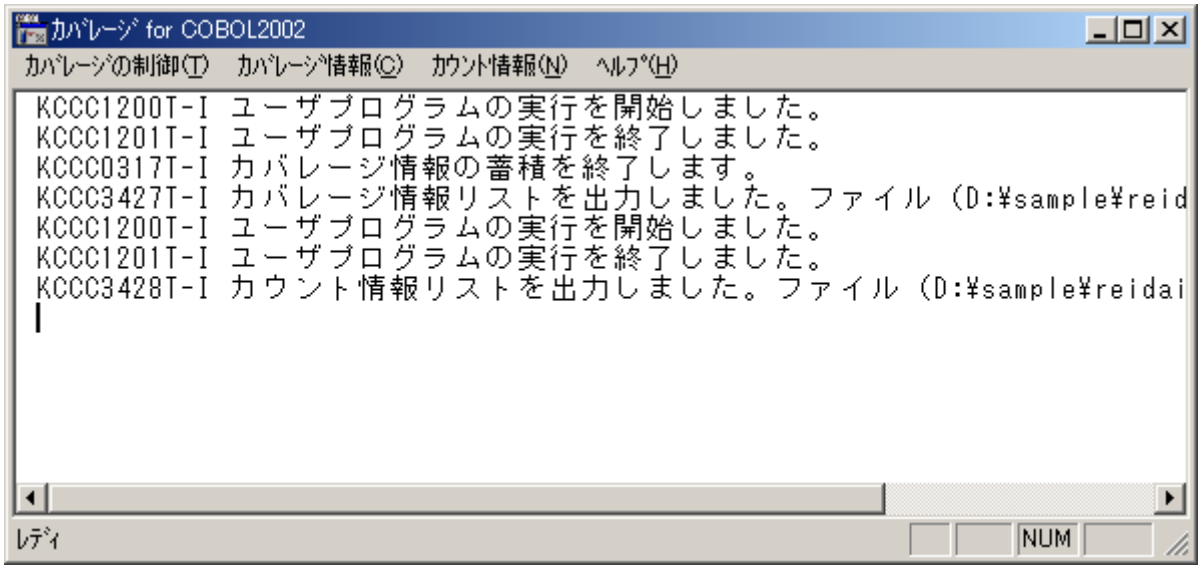
［手順２］ カウント情報の表示

「カウント情報の表示」画面で、「実行可能ファイル(N)」の参照ボタンをクリックし実行可能ファイルを指定します。



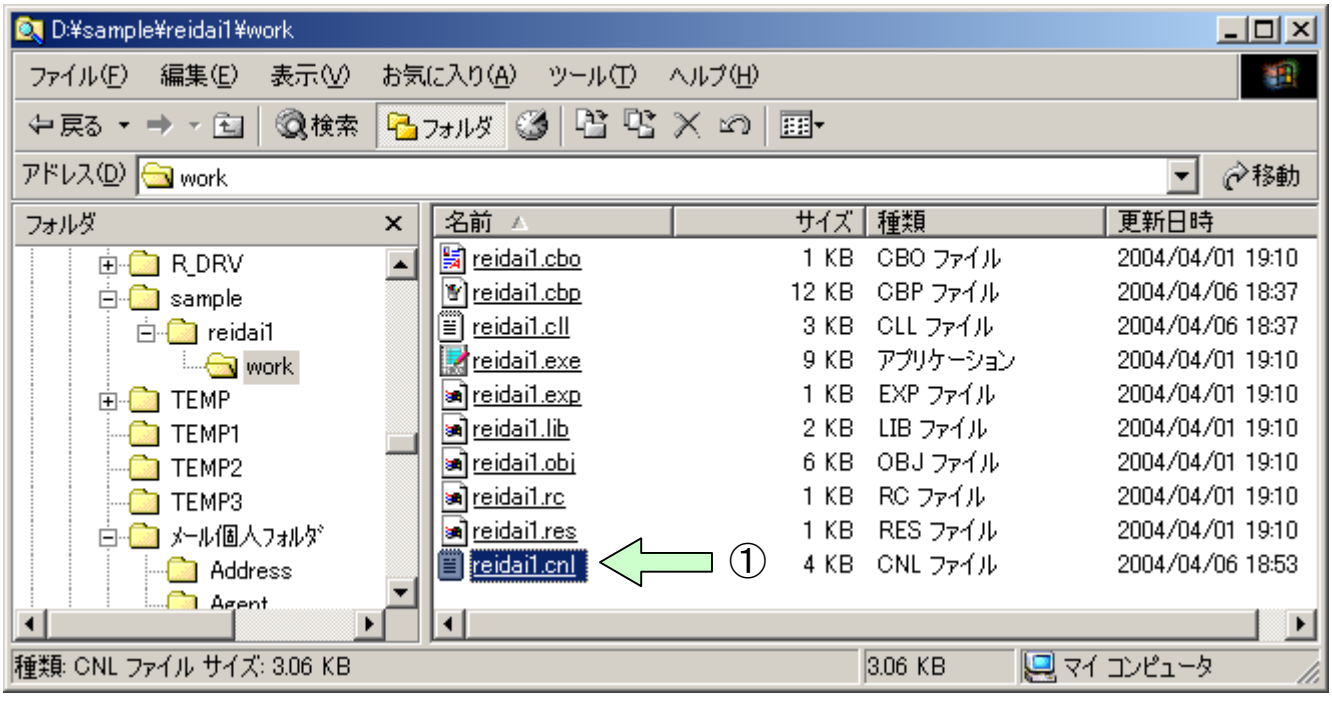






［手順３］カウント情報の表示

実行可能ファイルと同じフォルダに.cnlという拡張子のファイルが生成されています。このファイルをCOBOLエディタやメモ帳で開いて、カウント情報を見ることができます。



[カウント情報の表示例]

```
                          ****************************
                          *       カウント情報       *
    COBOL2002 (X)  01-01  ****************************          2004-04-06 18:53:00
-------------------------------------------------------------------------------------------

  プログラム名   : REIDAI1
  コンパイル日時: 2004-04-01 19:10:39
  実行日時       : 2004-04-06 18:53:00


-------------------------------------------------------------------------------------------
プログラム名 : REIDAI1
  実行回数     -------------------------------------------------------------------------------
               0001700 PROCEDURE  DIVISION.       aa
               0001701
               0001900 Mein-Sec SECTION.
         1     0002000     PERFORM  初期処理.
         1     0002100     PERFORM  比較処理.
         1     0002200     PERFORM  出力処理.
         1     0002300     STOP  RUN.
               0002500 初期処理  SECTION.
         1     0002600     ACCEPT  YYMMDD  FROM  DATE.
               0002800 比較処理  SECTION.
         1     0002900     IF  月  =  9
               0003000     THEN
         0     0003100       MOVE  'September!!'  TO  DATA2
               0003200     ELSE
         1     0003300       MOVE  'Not September!!'  TO  DATA2
               0003400     END-IF.
               0003600 出力処理  SECTION.
         1     0003700     DISPLAY  DATA0.
               0003800
               0003900
```

11. 終わりに

テストデバッグツールにおける基本的な使用方法は以上の説明で終わりです。一通りのデバッグを行う場合には、今までの説明の機能だけで十分であると思います。

しかし、テストデバッグツール自身には、その他の機能も備わっていますので、それらをお知りになりたい方は、マニュアル「COBOL2002操作ガイド」を参照ください。