

スクール COBOL2002 GUI 構築パック

操作入門書

- (株)日立製作所 -

(2005年2月作成)

目 次

1 . スクールCOBOL2002 GUI構築パックの概要	3
2 . GUI構築パックの操作手順	4
2 . 1 GUIプロジェクトを作成	5
2 . 2 GUI構築パックの起動	10
2 . 3 フォームプロパティ（属性）の設定	12
2 . 4 画面デザイン（フォームに部品を配置）.....	16
2 . 5 部品プロパティの設定	19
2 . 6 イベント手続きの登録	23
2 . 7 イベント手続きの記述	25
2 . 8 GUI構築パックの終了	27
2 . 9 コンパイル方法	28
2 . 10 デバッグ方法	29
2 . 11 実行	33
3 . 補足資料	34
3 . 1 部品のデータ型とCOBOLのデータ型の対応	34
3 . 2 部品のデータ型の検索方法	35
3 . 3 トラブルシューティング	40
3 . 3 . 1 GUIアプリケーション作成時の注意事項	40
3 . 3 . 2 GUIアプリケーション実行時の注意事項	41
3 . 4 コーディングする際の注意事項	42
3 . 5 GUI構築パックの用語説明	43

1 . スクール COBOL2002 GUI 構築パックの概要

スクール COBOL2002 GUI 構築パック（以降「GUI 構築パック」と略します。）は、COBOL の知識を活かして Windows アプリケーション特有のイベント駆動型の GUI（Graphical User Interface）アプリケーションを作成できます。

「GUI 構築パック」のプログラミング方法の特長を次に示します。

■ ビジュアルな画面デザイン

マウス操作で、「部品パレット」から部品を選び「フォームエディタ」に貼り付けて、GUI アプリケーション画面を作成します。

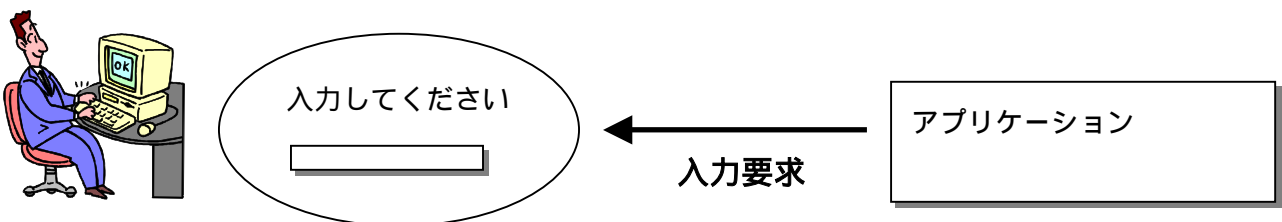
■ 手続きの記述

イベント（例えばボタンのクリック）に対応する手続きを COBOL で記述します。

1 . 1 イベント駆動型アプリケーションとは

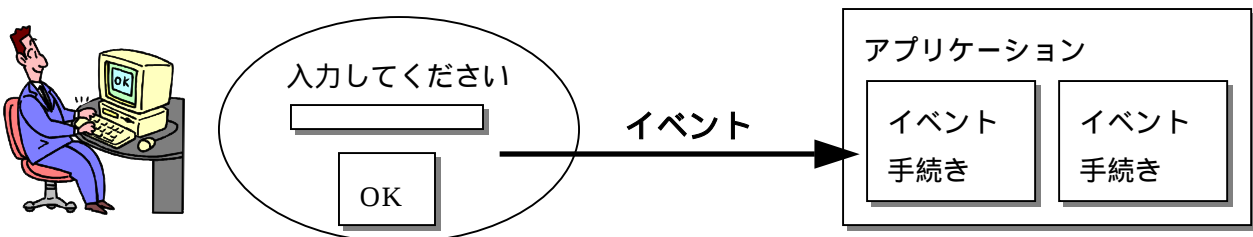
イベント駆動型アプリケーションは、利用者からの操作によるイベントの発生をきっかけにして、手続きが実行されるアプリケーションです。これに対し、従来の COBOL アプリケーションのように、処理手続きの流れに沿って画面表示するアプリケーションを、手続き型アプリケーションといいます。

■ 手続き型アプリケーションの入力処理



手続き型アプリケーションの場合、アプリケーションが処理する過程で入力が必要になったときに、入力を要求します。利用者はアプリケーションからの要求を待つ形態になります。

■ イベント駆動型アプリケーションの入力処理



イベント駆動型アプリケーションの場合、利用者が「ボタンをクリックする」、「テキストボックスに文字を入力する」、などの操作をすることによって処理が呼び出されます。アプリケーションは、利用者からの操作（イベント）に対応した処理をします。

2 . GUI 構築パック」操作手順

「GUI 構築パック」を使ったアプリケーションの作成手順を説明します。
簡単な GUI アプリケーションを作ってみましょう。次の仕様で作成します。

仕様（GUI1.exe）

画面 1 の構成： 入力要求

フォーム（form0）、メッセージラベル（Label1）、反転ボタン（Button1）

反転ボタンをクリックすると、メッセージラベルの文字が反転する。
再度クリックするとメッセージラベルの文字が再度反転し、元の表示状態になる。



図 2 . 1 画面 1 反転前



図 2 . 2 画面 1 反転後

2 . 1 GUI プロジェクトの作成

COBOL で GUI アプリケーションを作成するときは、まず、ソースファイル等の資源を管理するためのプロジェクトファイルを作成します。ここでは、「GUI プロジェクト」作成方法について説明します。

注意：プロジェクトの作成を始める前に「D:¥TEMP¥GUI¥」フォルダを作成してください。

COBOL2002 開発マネージャ（以降、「開発マネージャ」と呼びます）を起動し、「ファイル(E)」の「新規作成(N)」を、をクリックします。新規作成画面が表示されます。

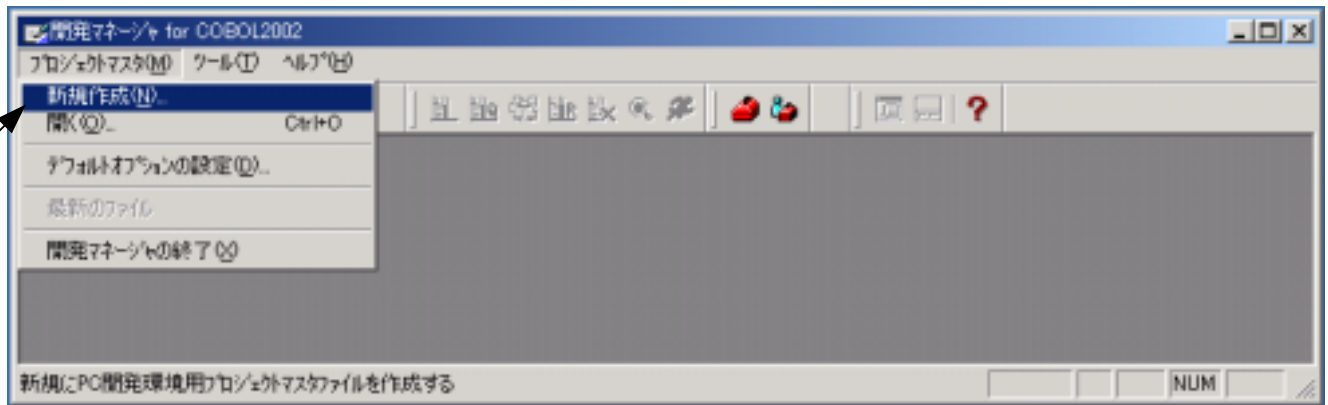


図 2 . 3 「開発マネージャ」画面、上記の選択状態

「プロジェクトマスタ名」に「GUI1」を設定します。

「参照」ボタンをクリックして D ドライブの「TEMP」の下の「GUI1」フォルダに位置付け

「OK」ボタンをクリックします。

「完了」ボタンをクリックします。

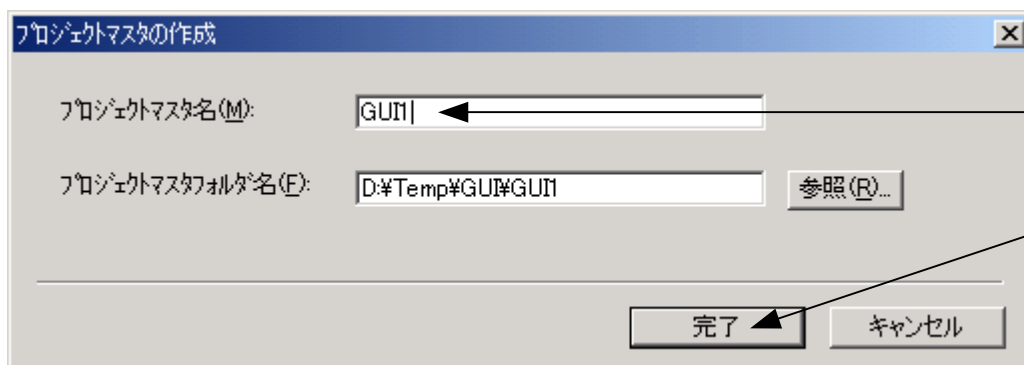


図 2 . 4 新規作成画面

GUI1 フォルダを作成しますかというダイアログが出ますので、「OK」ボタンをクリックします。

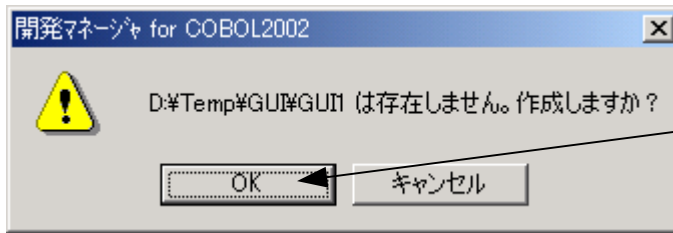


図 2.5 確認のインフォメーションダイアログ

プロジェクト名に「SAMPLE11」を指定します。

「プロジェクトフォルダと作業フォルダを同一にする。」をチェックします。

「OK」ボタンをクリックします。

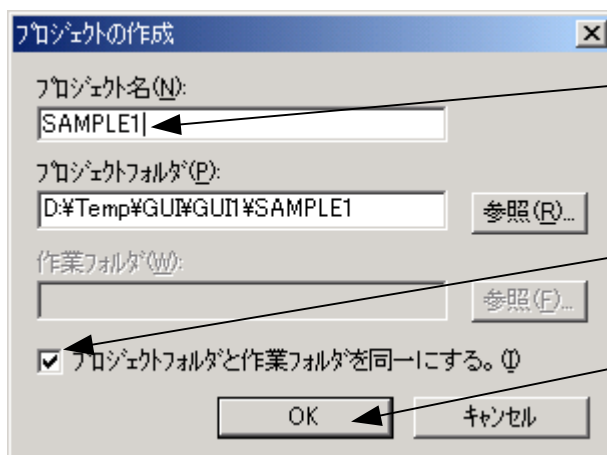


図 2.6 プロジェクトの作成画面

SAMPLE1 フォルダを作成しますかというダイアログが出ますので、「OK」ボタンをクリックします。

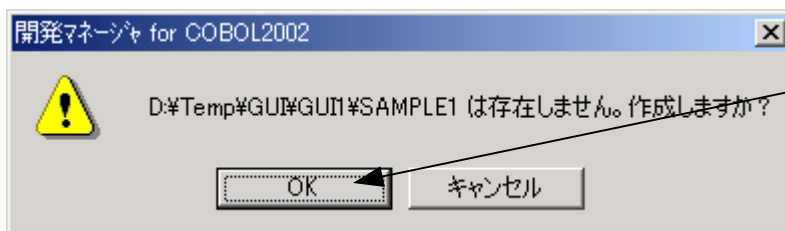


図 2.7 フォルダ作成ダイアログ

プロジェクトの追加画面が出ますので、「実行形式プログラム」を選択し「次へ(N)」ボタンを押します。

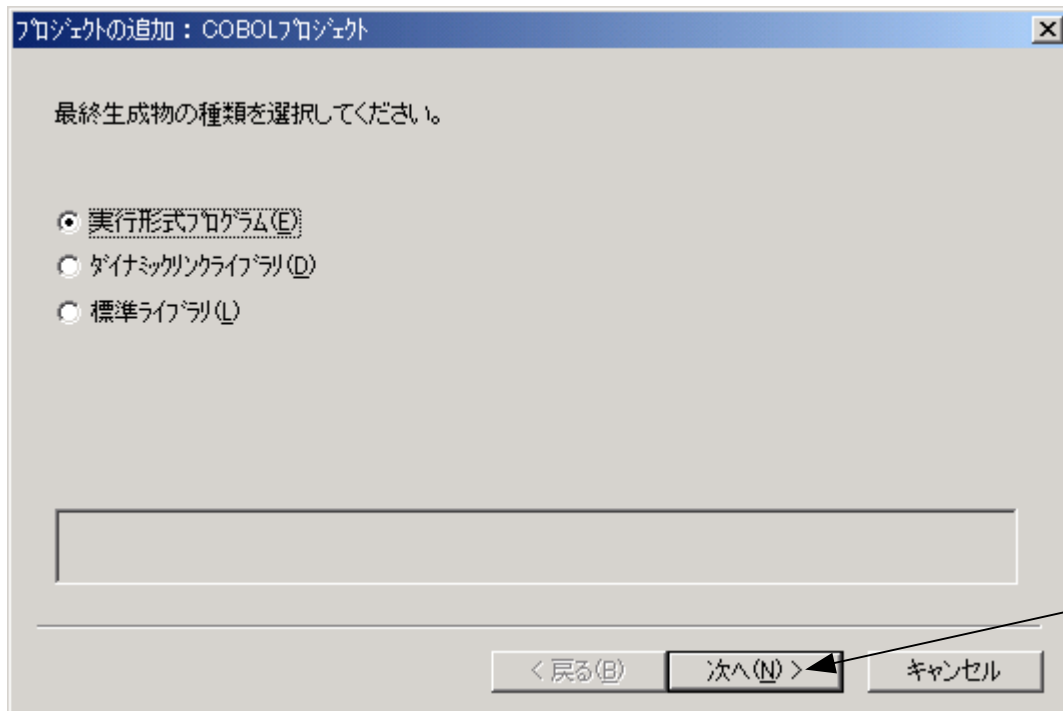


図 2 . 8 プロジェクトの追加画面 1

プロジェクトの種類を選択し、「次へ(N)」ボタンを押します。

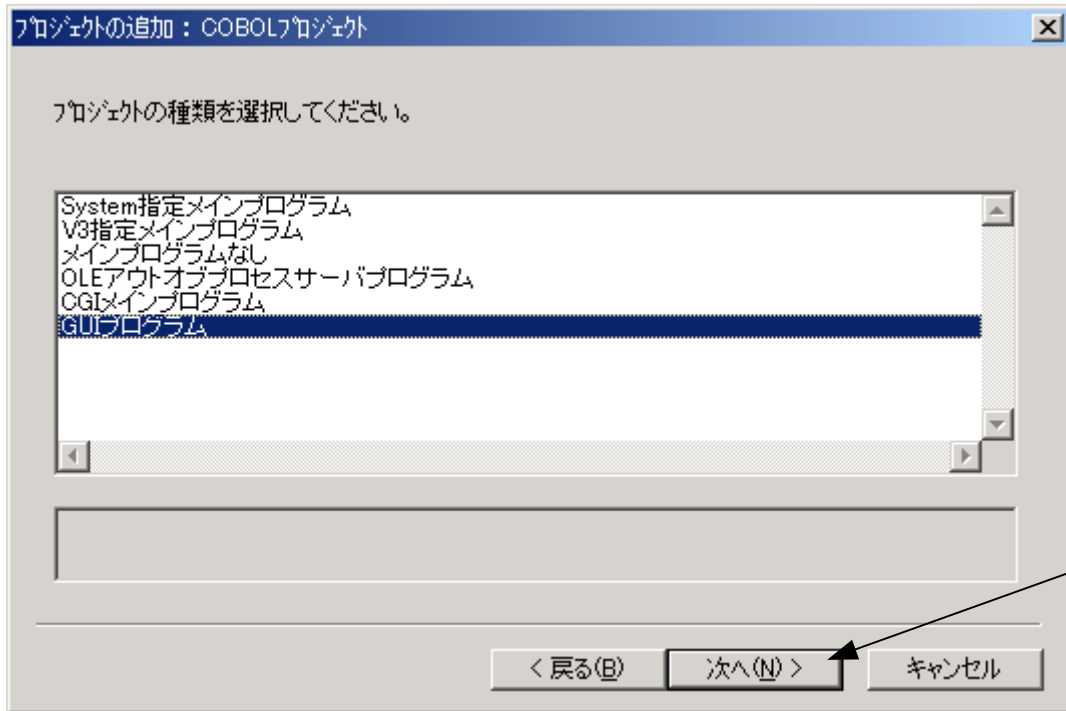


図 2 . 9 プロジェクトの追加画面 2

出力ファイルを入力し、「次へ(N)」ボタンを押します。

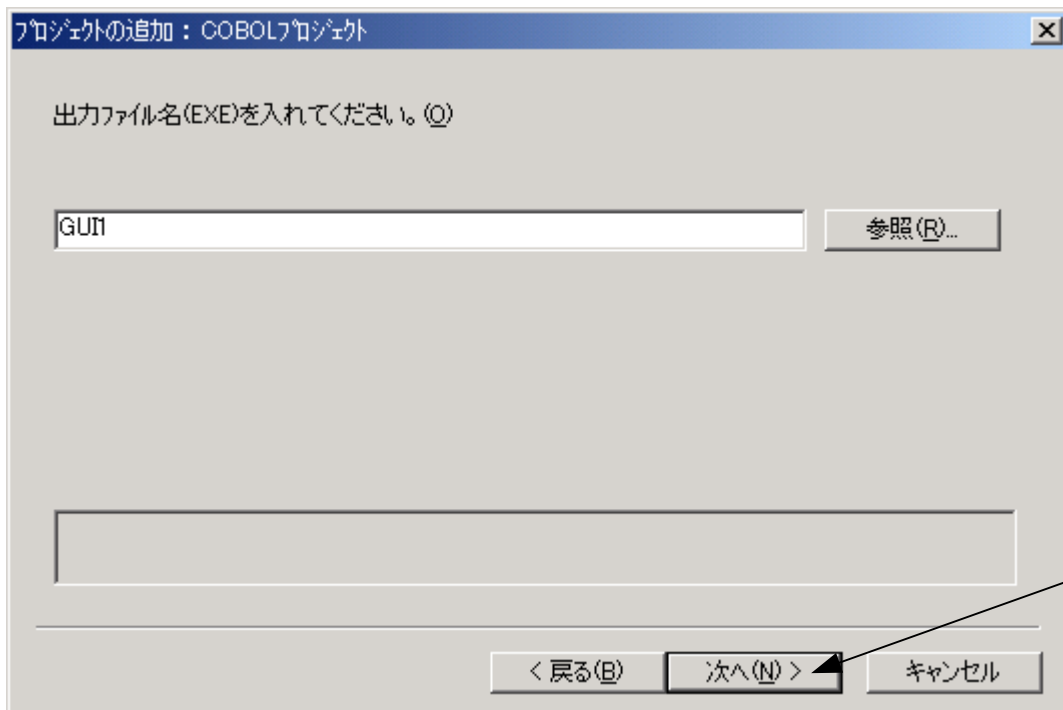


図 2 . 1 0 プロジェクトの追加画面 3

GUI 定義ファイルを入力し、「完了」ボタンを押します。

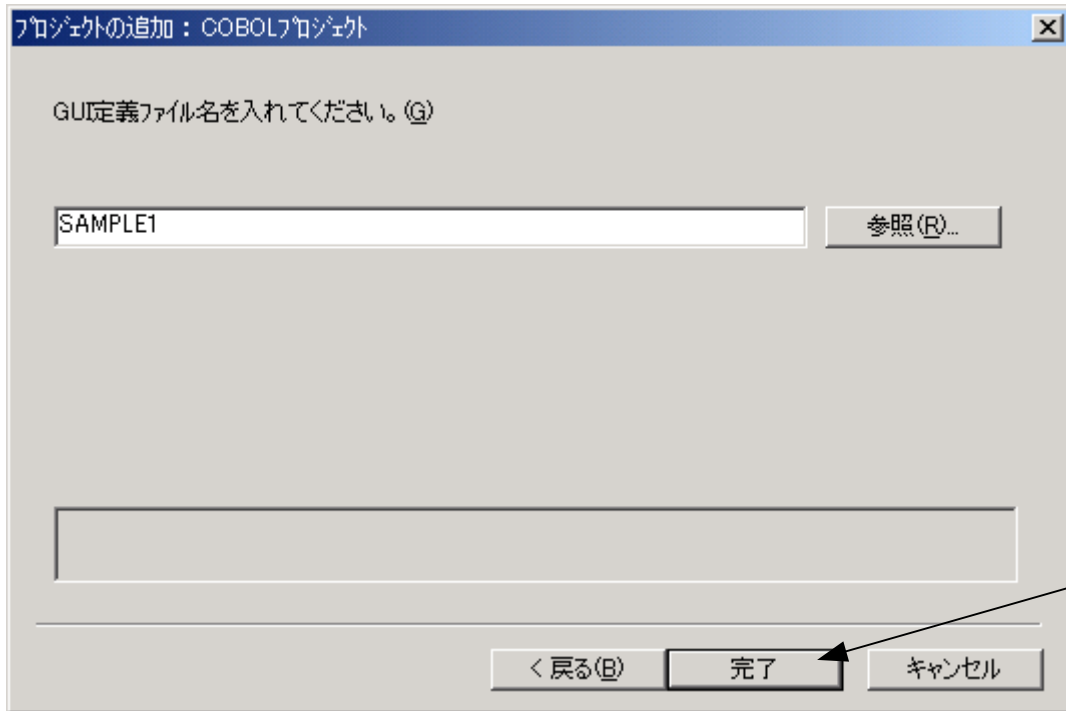


図 2 . 1 1 プロジェクトの追加画面 4

「開発マネージャ」に「GUI1.hmf」の構成が表示されます。

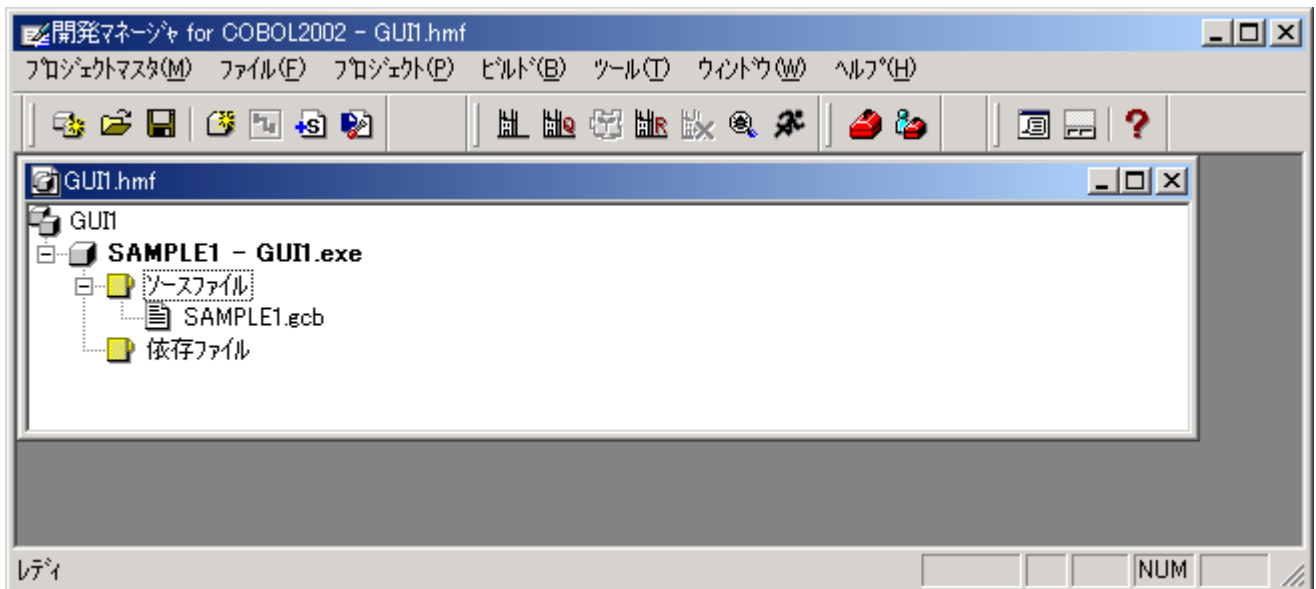


図 2 . 1 2 「開発マネージャ」の構成画面

2 . 2 GUI 構築パックの起動

(a)GUI 構成パックの画面構成

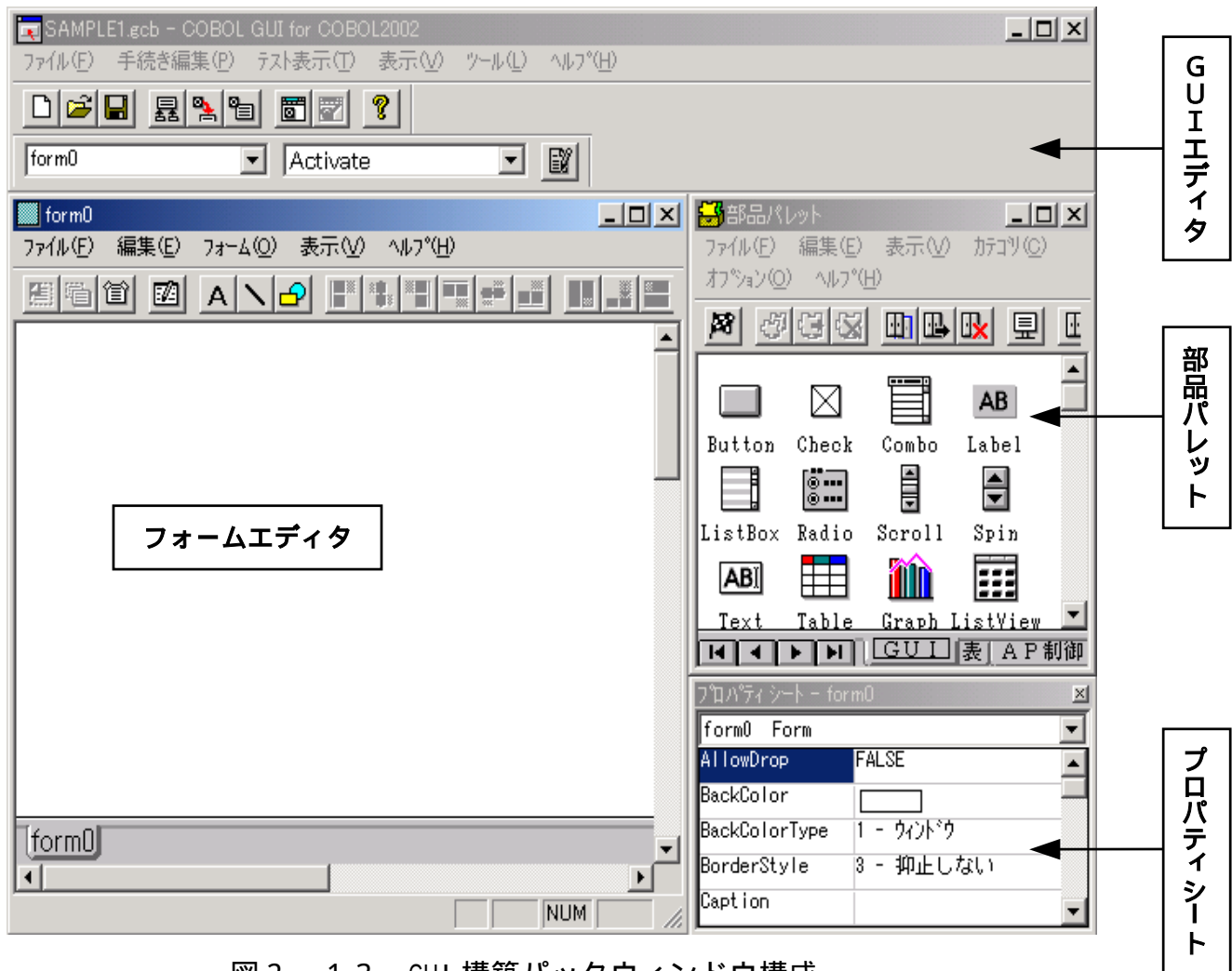


図 2 . 1 3 GUI 構築パックウィンドウ構成

ワンポイントレッスン

GUI エディタ：GUI 定義や手続きの記述で使用するエディタを起動したりファイルの保存を行ったりするメニュー。

フォーム：作成したプログラムを実行すると、このフォームでデザインした画面が表示される。

部品パレット：ここに表示されている部品を使って、フォームに貼り付け画面をデザインする。

プロパティシート：ここに表示されている設定値を変更することで、部品の色や外観をデザインできる。

(b) 起動方法

GUI1.hmf を「開発マネージャ」で開き、表示されている「SAMPLE.gcb」をダブルクリックします。

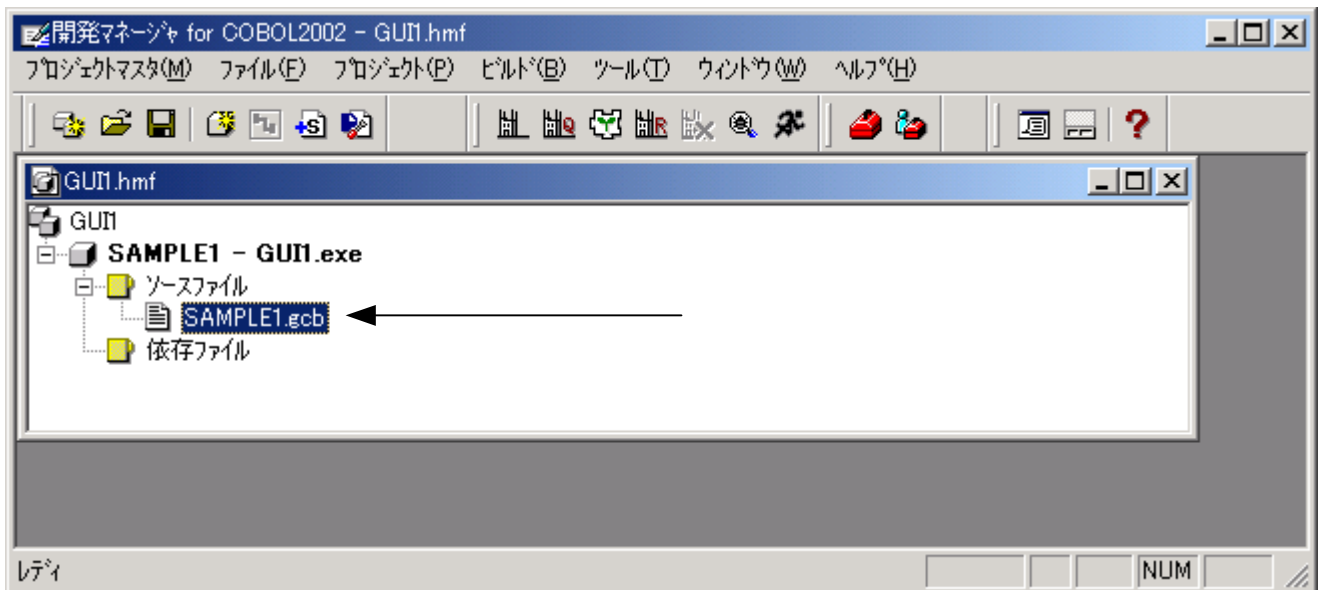


図 2 . 1 4 「開発マネージャ」からダブルクリック

GUI エディタが起動されます。



図 2 . 1 5 GUI エディタウィンドウ

2 . 3 フォームプロパティ(属性)の設定

GUI エディタの「ファイル(F)」から「フォームエディタの起動(E)」を選択します。
(新規作成時にはフォームエディタは自動的に起動しますので の処理は不要です。)

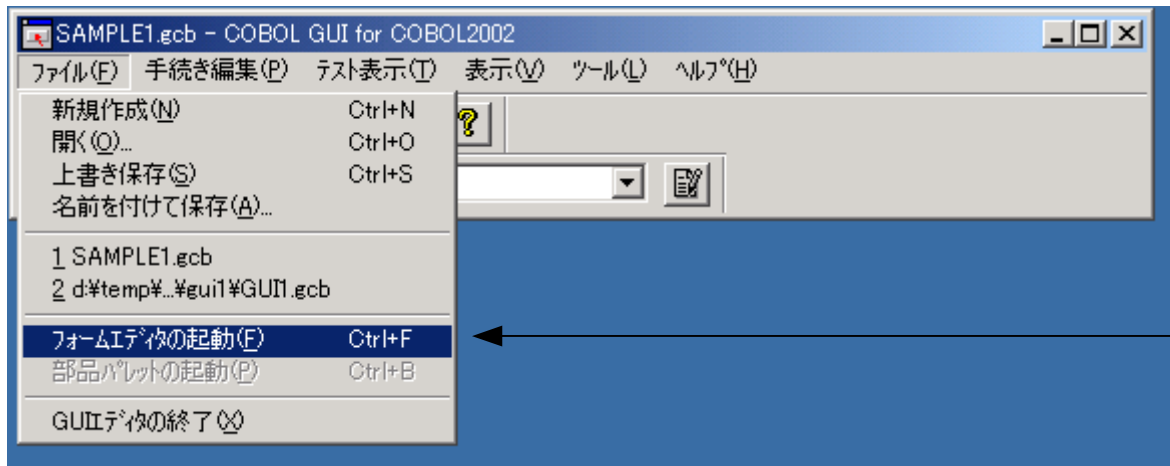


図 2 . 1 6 GUI エディタからフォームエディタ起動メニュー選択

フォームエディタが起動されます。

フォームの上でマウス左ボタンをダブルクリックするとフォームプロパティダイアログが表示されます。

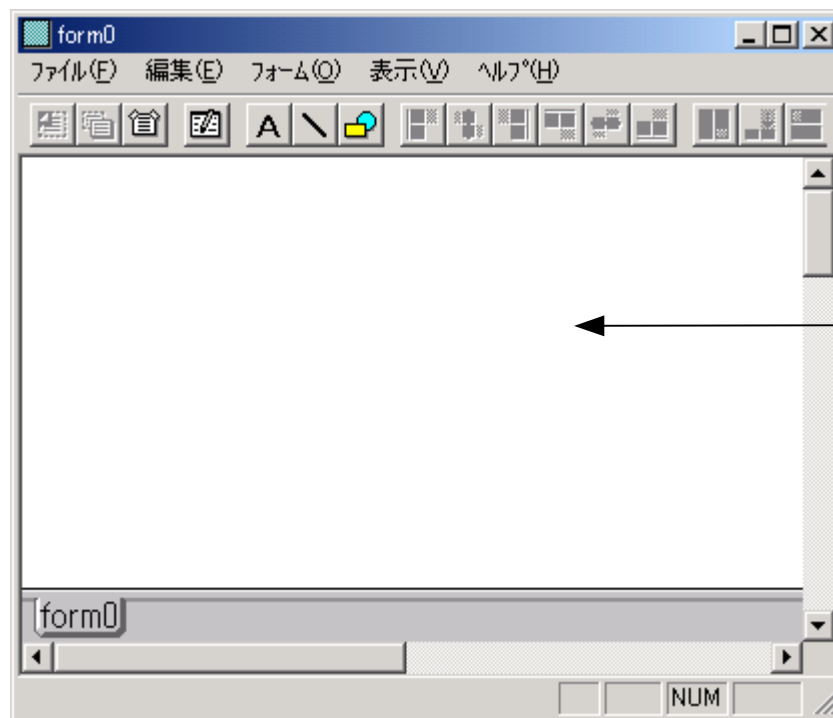


図 2 . 1 7 フォームエディタ

フォームサイズを変更します。ここではフォームサイズの「ウィンドウサイズ」を選択して、現在表示されているフォームエディタのサイズに設定します。
この設定では、フォームエディタのウィンドウ枠をドラッグすることによって、フォームサイズの変更ができます。

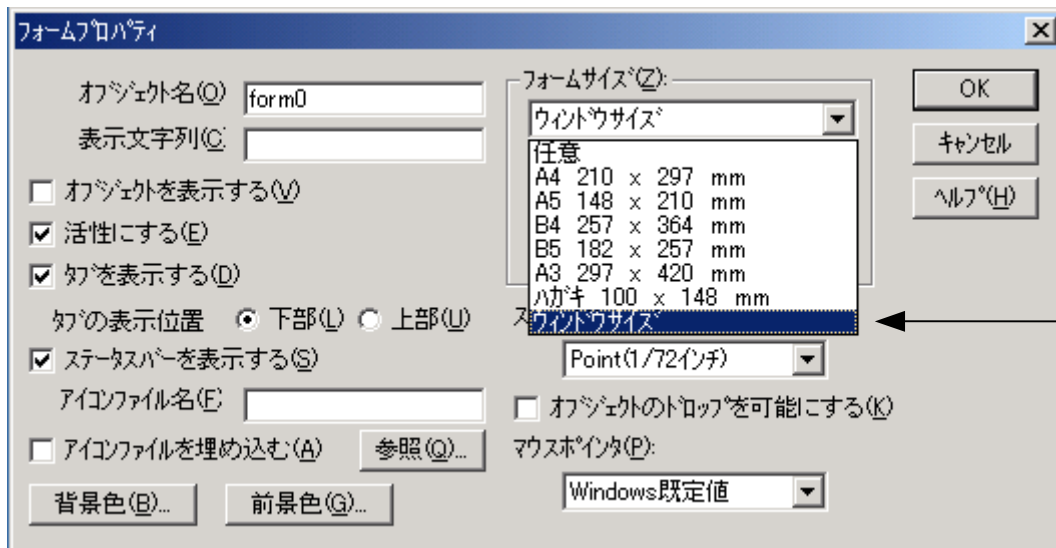


図 2 . 1 8 フォームプロパティダイアログ

フォームの色、フォーム上で表示される文字の色は、それぞれ「背景色(B)...」、「前景色(G)...」で変更します。ここでは「背景色(B)...」について説明しますが「前景色(G)...」の設定も「背景色(B)...」と同様に設定します。
「背景色(B)...」をクリックします。

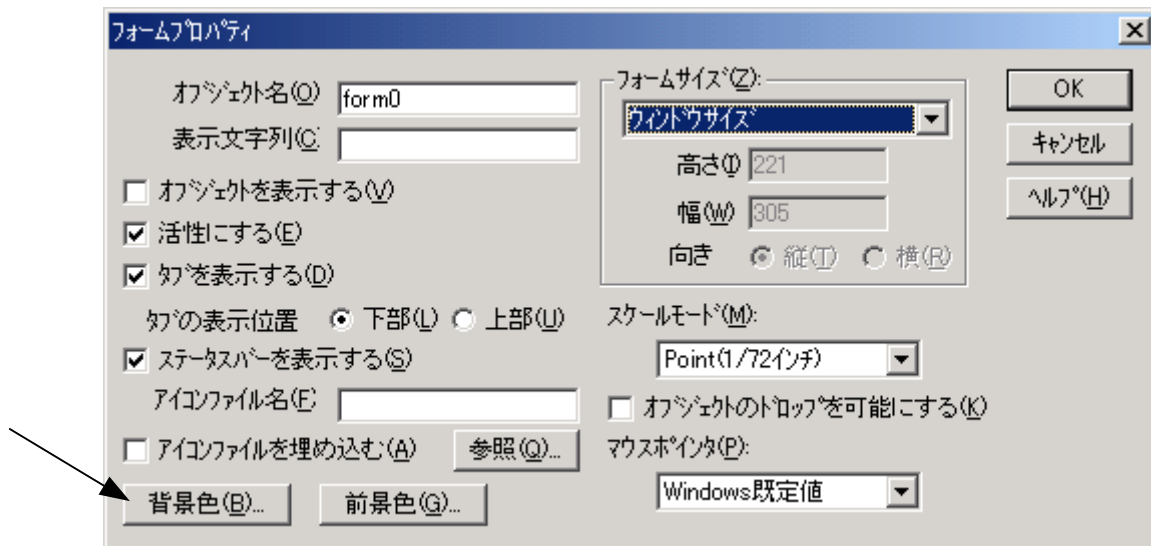


図 2 . 1 9 背景色をクリック

色の設定ダイアログが表示されます。基本色を使用したい場合は表示されている色をクリックします。（ここではグレーを指定します。）
その他の色を使用したい場合は「作成した色」を選択して、「色の作成」をクリックします。

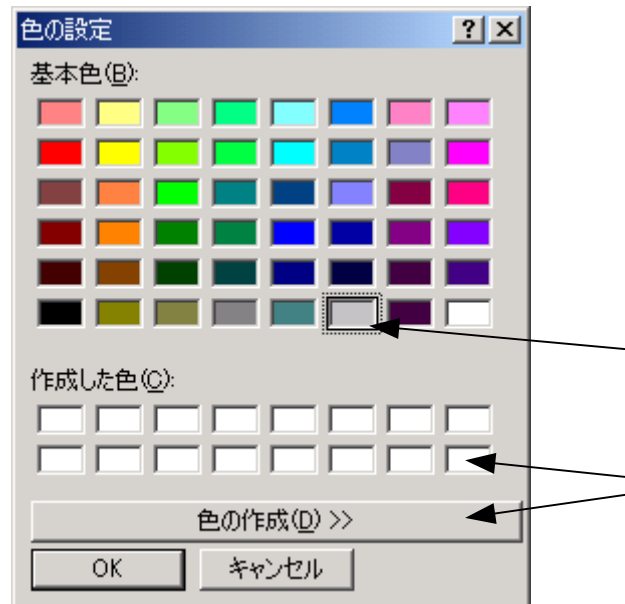


図 2 . 2 0 色の設定ダイアログ（その 1）

色を作成し、「色の追加」をクリックすると、選択した「作成した色」に新しい色が追加されます。

色の設定ダイアログの「OK」ボタンをクリックします。

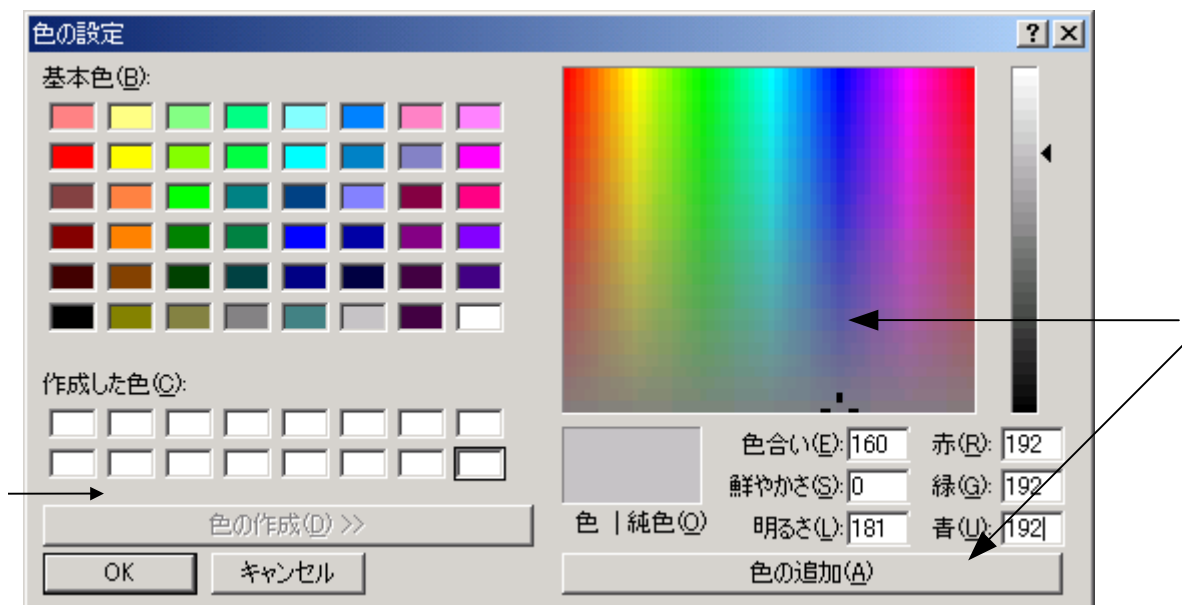


図 2 . 2 1 色の設定ダイアログ（その 2）

フォームプロパティダイアログの「OK」ボタンをクリックします。

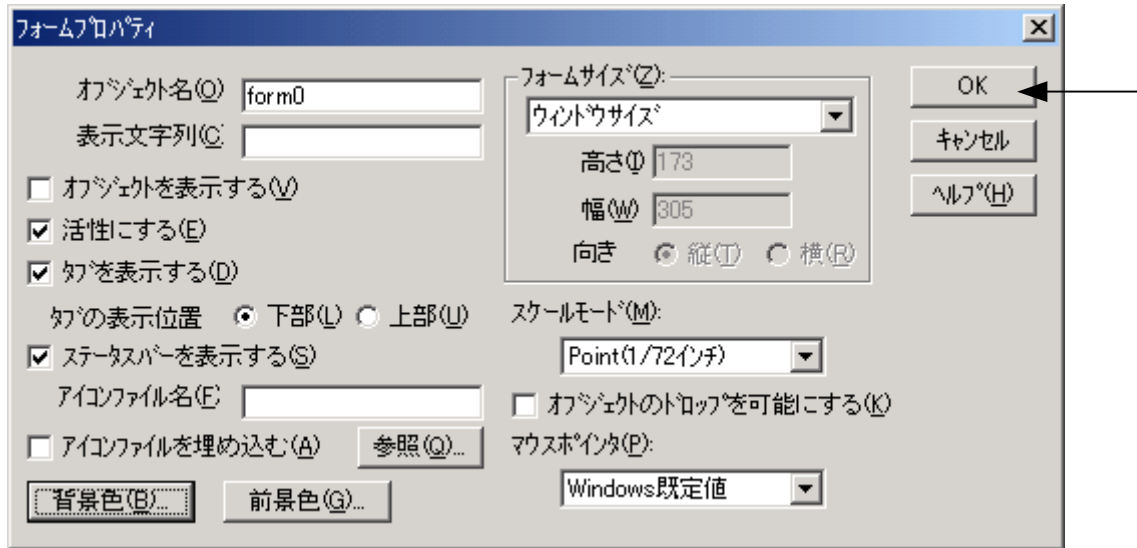


図 2 . 2 2 フォームのプロパティ変更完了

フォームの色が変更され、フォームサイズがウィンドウサイズに設定されました。

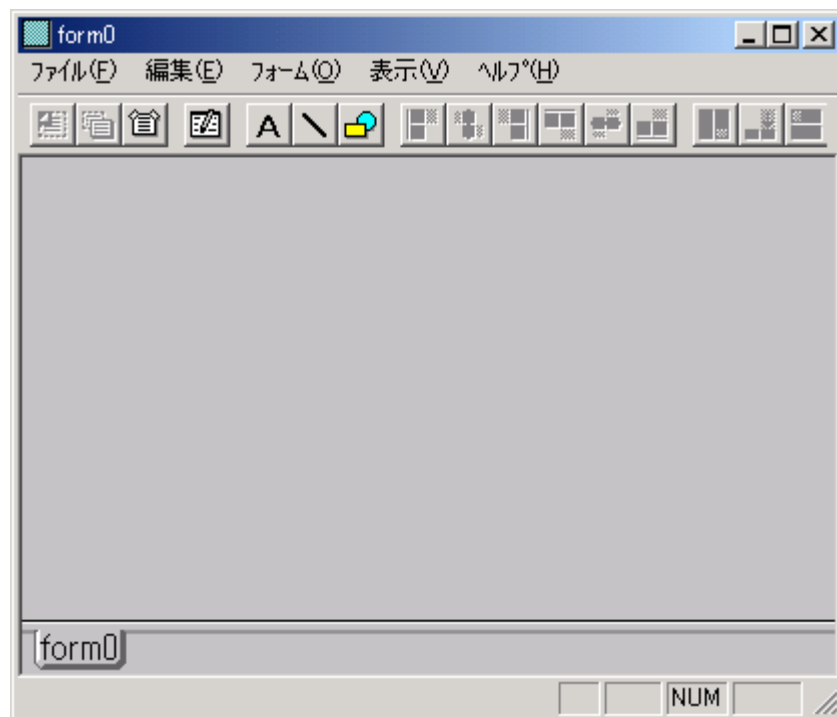


図 2 . 2 3 色が変わったフォームエディタ

2.4 画面デザイン（フォームに部品を配置）

GUI エディタの「ファイル(F)」から「部品パレットの起動(P)」を選択します。
 （新規作成時には部品パレットは自動的に起動しますので の処理は不要です。）

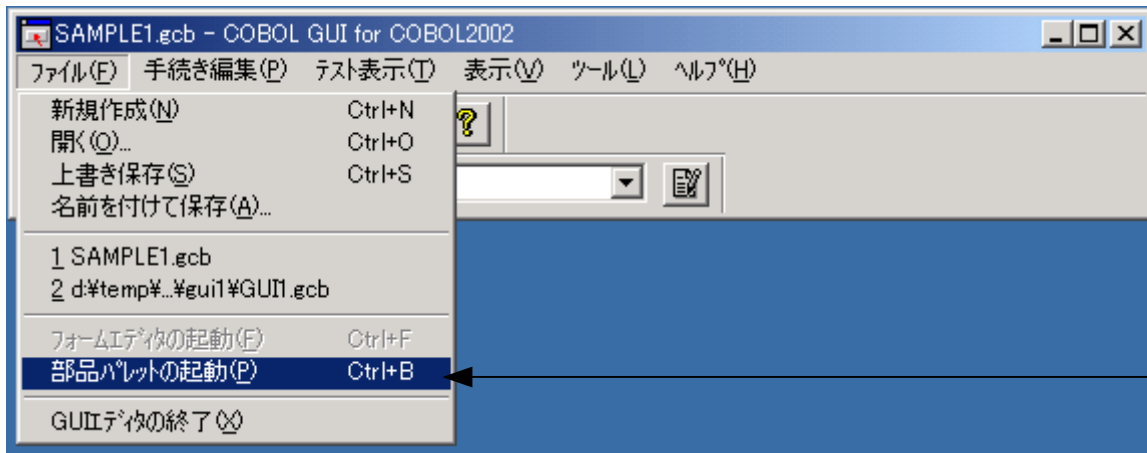


図 2.24 GUIエディタから部品パレット起動

部品パレットが起動されます。



図 2.25 部品パレット

ラベルを配置しましょう。部品パレットの「Label」をドラッグアンドドロップでフォームエディタに貼り付けます。

ボタンを配置しましょう。部品パレットの「Button」をドラッグアンドドロップでフォームエディタに貼り付けます。

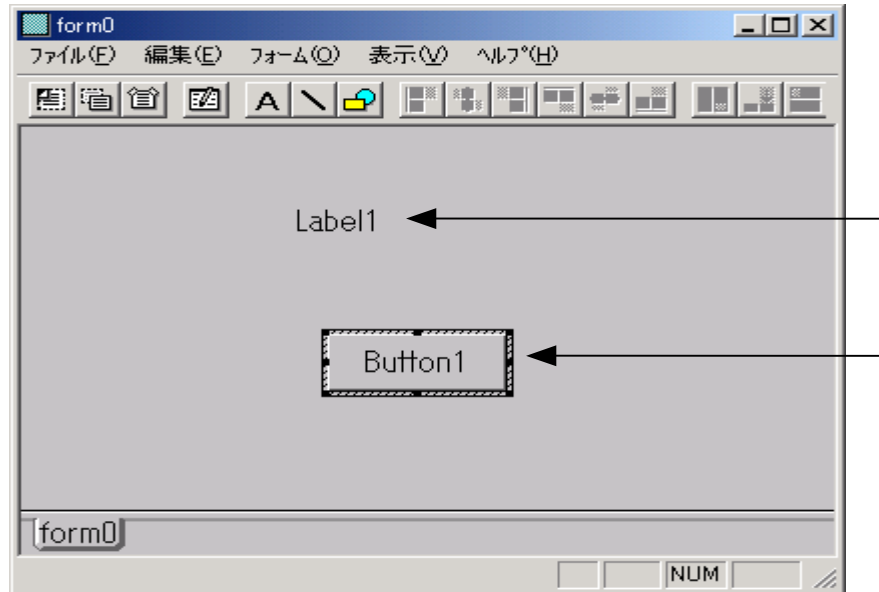


図 2 . 2 6 部品パレットからドラッグアンドドロップ

ワンポイントレッスン

この図ではラベルに「Label1」、ボタンに「Button1」という初期値の名前が設定されています。この名前は利用者が付け直すことができ、部品に与えられた機能と名前の関係を分かりやすくできます。また、データ名を日本語で記述することにより、更に分かりやすくすることができます。

配置した部品は選択してドラッグすることにより、配置やサイズの変更が自由にできます。

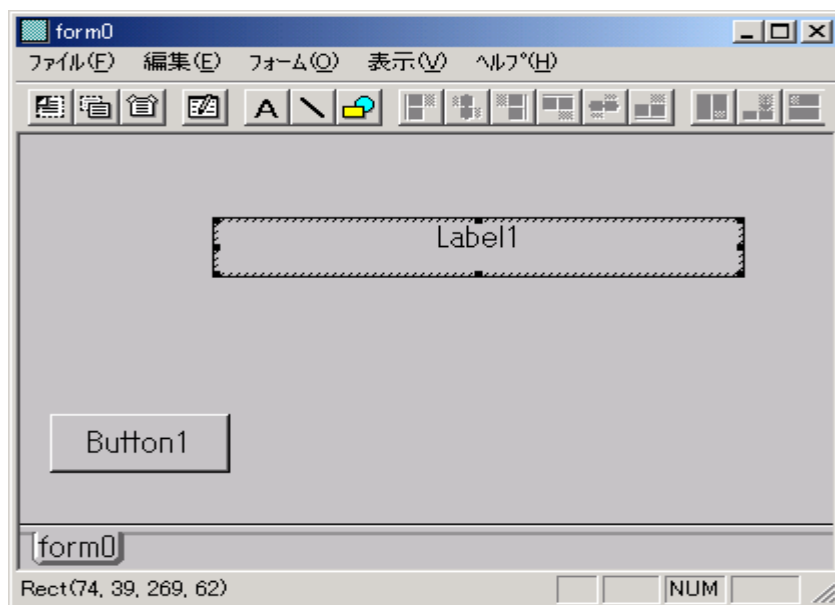


図 2 . 2 7 サイズ変更と配置変更

部品の位置をきちんと合わせるには、整えたいときはグリッドを表示させると便利です。フォームエディタの「表示(V)」から「グリッド(G)...」を選びます。

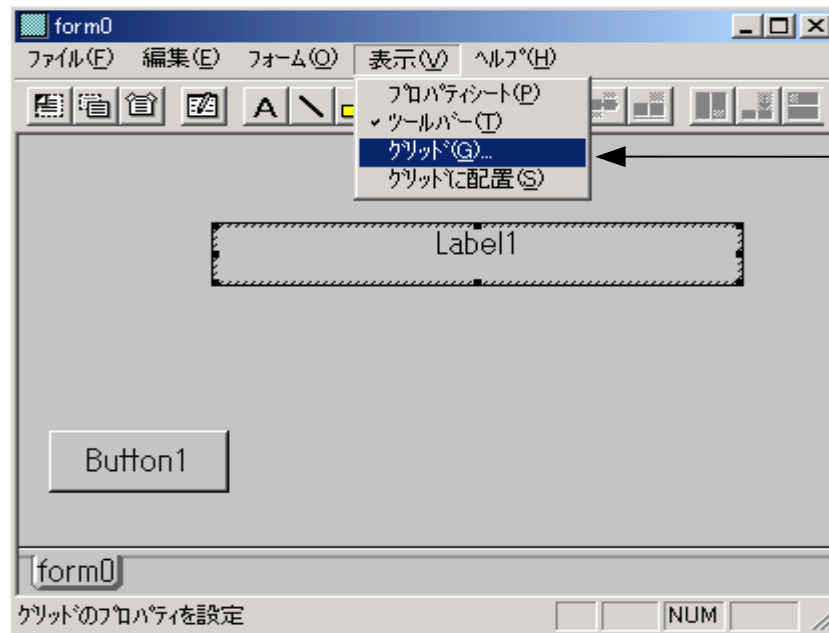


図 2 . 2 8 フォームエディタの表示メニューからグリッドを選ぶ

ダイアログの「グリッドを表示する(V)」をチェックすると GUI 部品のレイアウトがしやすくなります。また、フォームエディタの「表示(V)」から「グリッドに配置(S)」を選択すると、グリッドに合わせた部品の配置が簡単にできます。

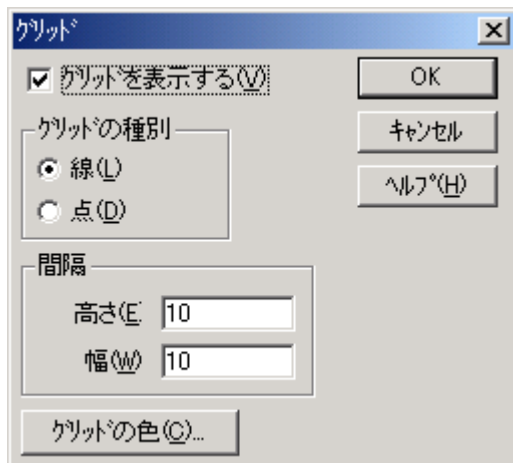


図 2 . 2 9 グリッドダイアログ

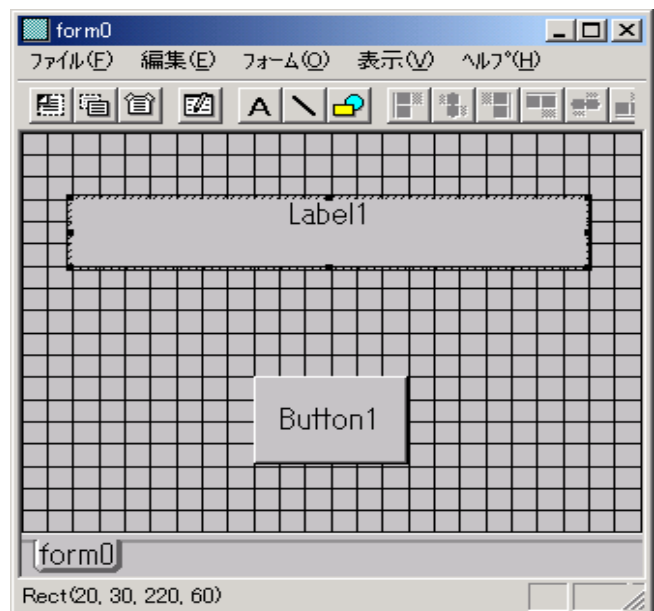


図 2 . 3 0 グリッドが表示された画面

2 . 5 部品プロパティの設定

ボタンの色や文字など、部品の設定をしたいときはプロパティシートを使います。フォーム上に配置した部品をクリックして、フォームエディタの「表示(V)」から「プロパティシート(P)」を選択します。

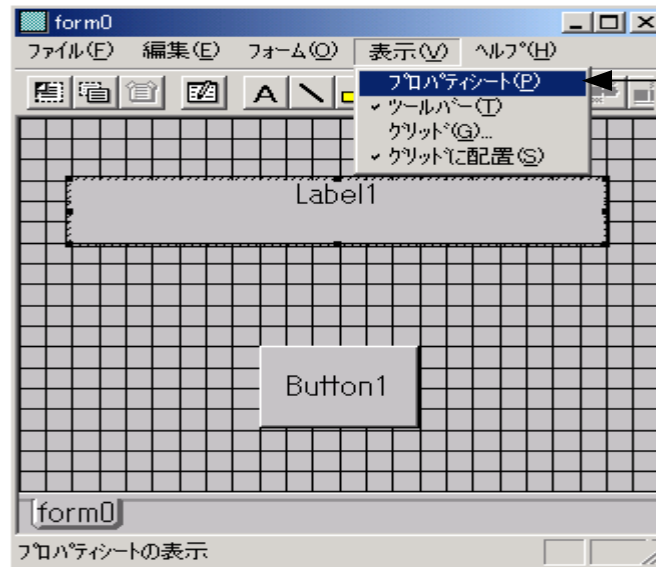


図 2 . 3 1 フォームエディタのプロパティシートメニュー選択

プロパティシートでは、部品がそれぞれ持っている属性（プロパティ）の設定をすることができます。プロパティシートに表示されているプロパティが、その部品が持つプロパティです。プロパティは部品ごとに異なります。部品ごとのプロパティの詳細は「2 . 7 イベント手続きを記述しましょう」で説明するヘルプを参照してください。



図 2 . 3 2 プロパティシート

プロパティシートを使ってラベルの文字列を設定してみましょう。
 プロパティシートの「Caption」をダブルクリックし、表示したい文字列を入力します。
 (ここでは「Hello!!」を記述します。) 記述後に ENTER キーを押します。



図 2 . 3 3 Caption 選択状態になったプロパティシートに文字列設定
 ラベルの文字列が変更されました。

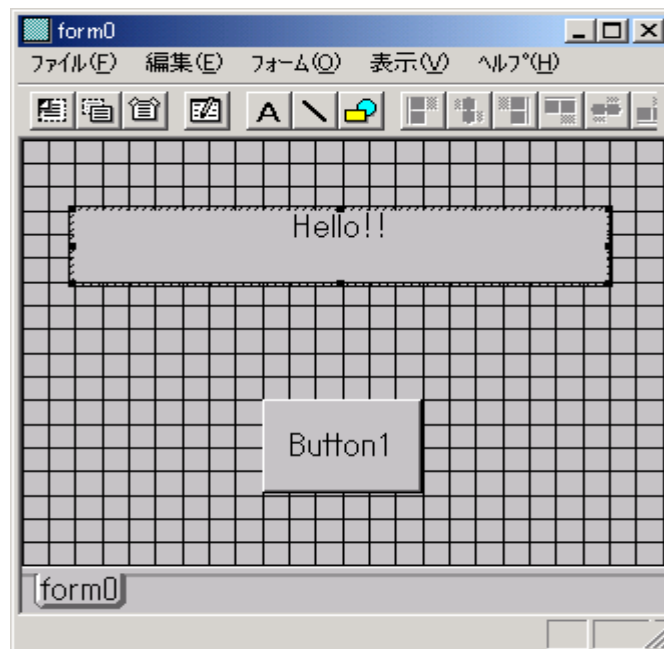


図 2 . 3 4 文字列変更後

プロパティシートを使ってラベルのフォントを変更してみましょう。

プロパティシートの「Font」をダブルクリックするとフォントダイアログが表示されます。

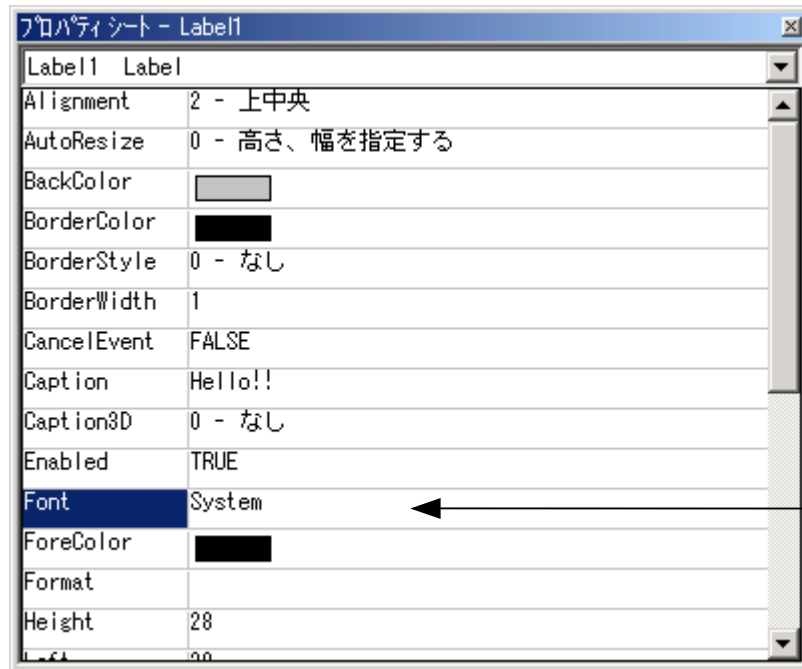


図 2 . 3 5 Font プロパティの変更

反転表示が分かりやすいようなフォントを設定します。

(ここでは MS ゴシック/太字/18 ポイントを選択します。)

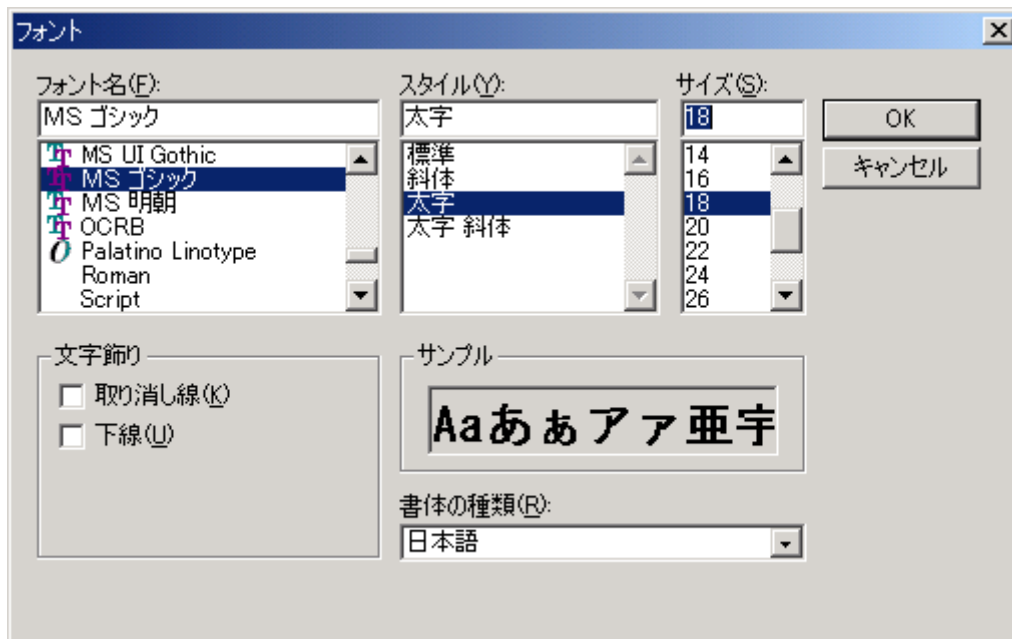


図 2 . 3 6 フォントダイアログと文字列の設定

ラベルのフォントが変更されました。

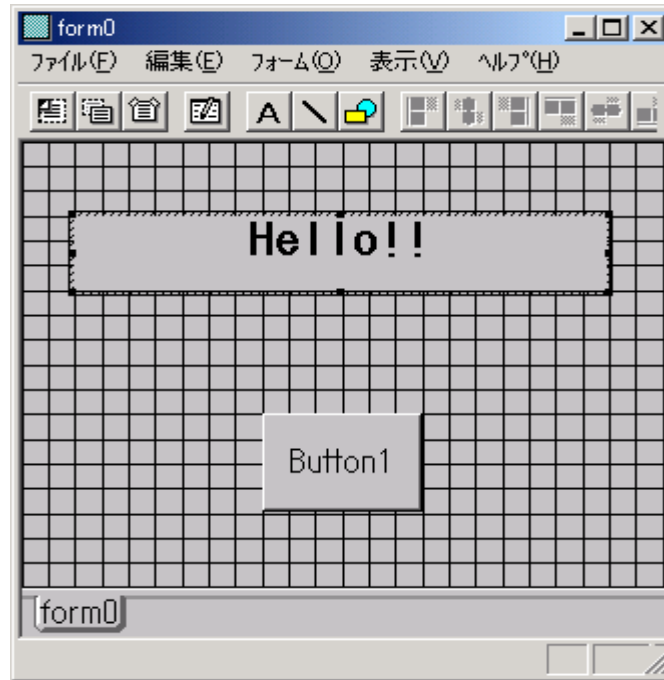


図 2 . 3 7 フォント変更後

同じようにボタンの文字列も設定してみましょう。

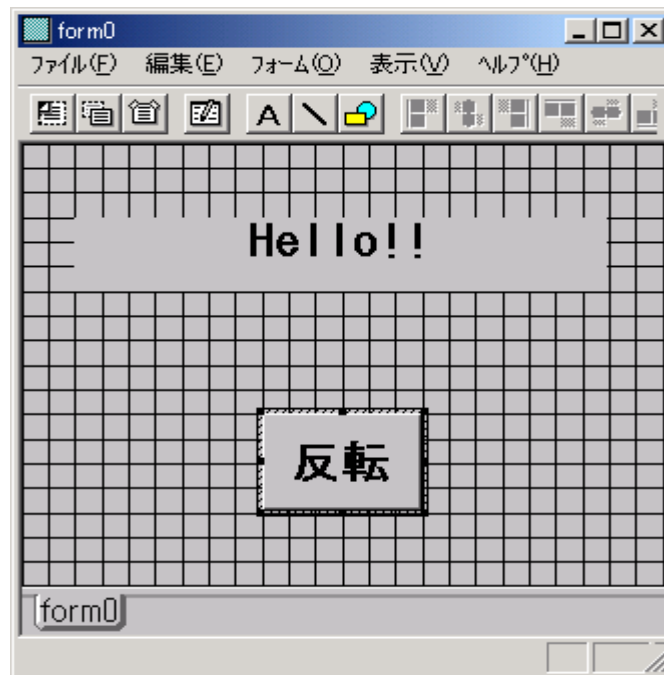


図 2 . 3 8 ボタンプロパティ設定後の表示

2.6 イベント手続きの登録

フォームエディタ上のボタン (Button1) をクリックし、GUI エディタの「手続き編集(P)」から「イベント手続きの編集(E)...」を選択します。

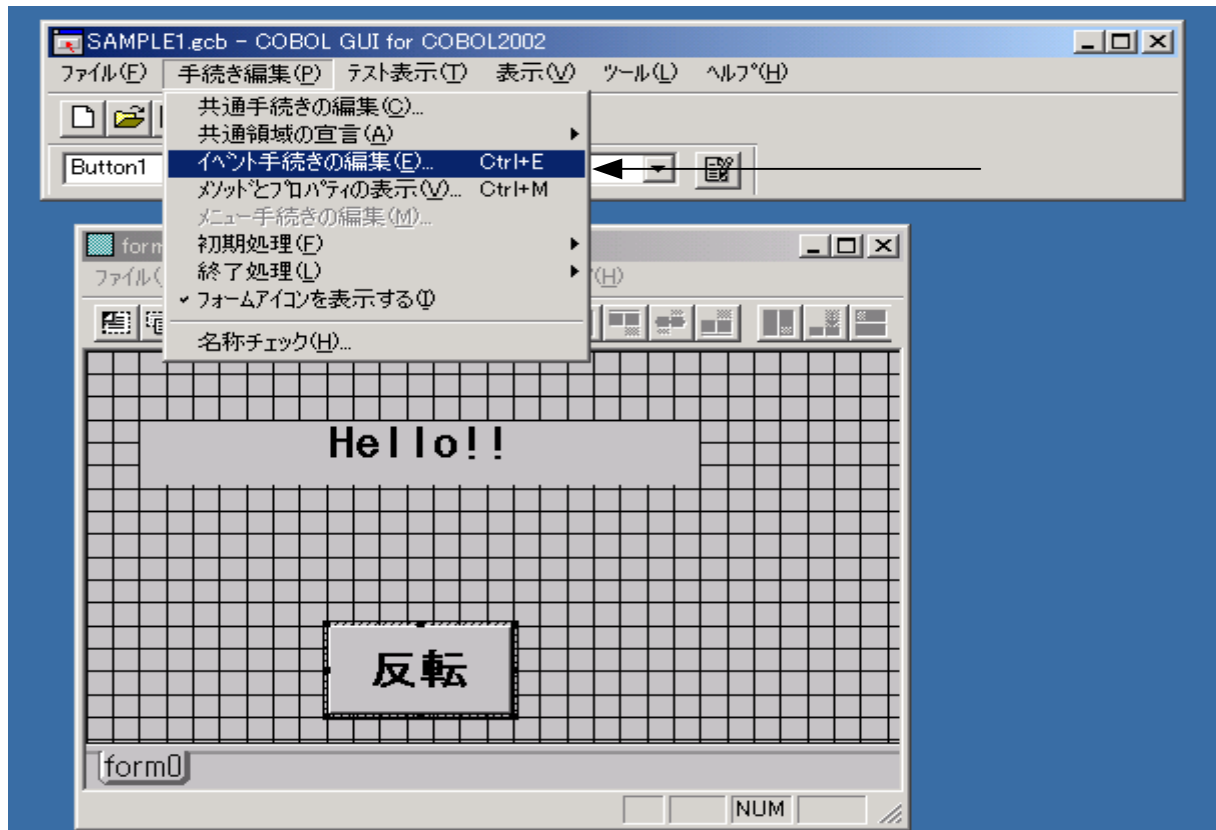


図 2.39 フォームエディタと GUI エディタの選択中

イベント手続き編集ダイアログが表示されます。

選択した部品で利用できるイベントが「イベント(V)」に表示されています。

イベントの詳細は「2.7 イベント手続きを記述しましょう」で説明するヘルプを参照してください。

ボタンがクリックされたときのイベント手続きを記述します。

「イベント(V)」の「Click」を選択して、「編集(E)」をクリックします。

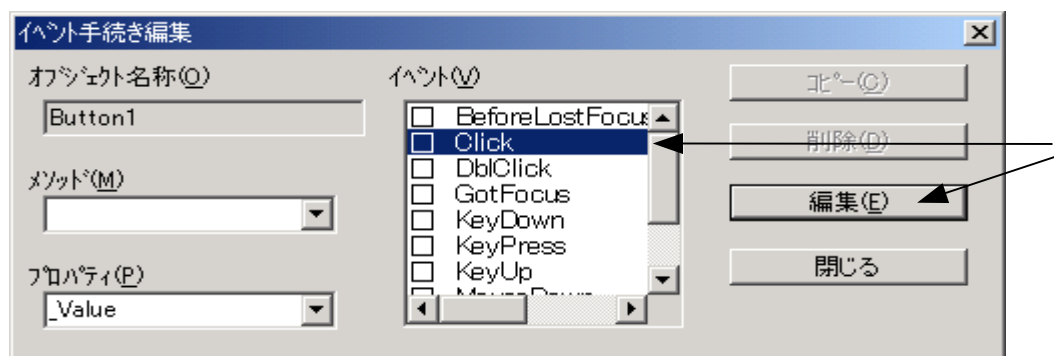


図 2.40 イベント手続き編集ダイアログと説明

ワンポイントレッスン

イベント手続きは GUI エディタの「オブジェクト選択ツールバー」からも編集ができます。GUI アプリケーションの作成に慣れてきたら、こちらの方法を使うことをお勧めします。「オブジェクト選択ツールバー」の詳細な使い方はマニュアル「COBOL2002 GUI 構築パック ユーザーズガイド」を参照してください。

エディタが起動します。

(ここではデフォルト設定の「COBOL エディタ」で説明します。なお、開発マネージャの設定を変更して、使い慣れたエディタに切り替えることもできます。詳細はマニュアル「COBOL2002 操作ガイド」をご覧ください。)

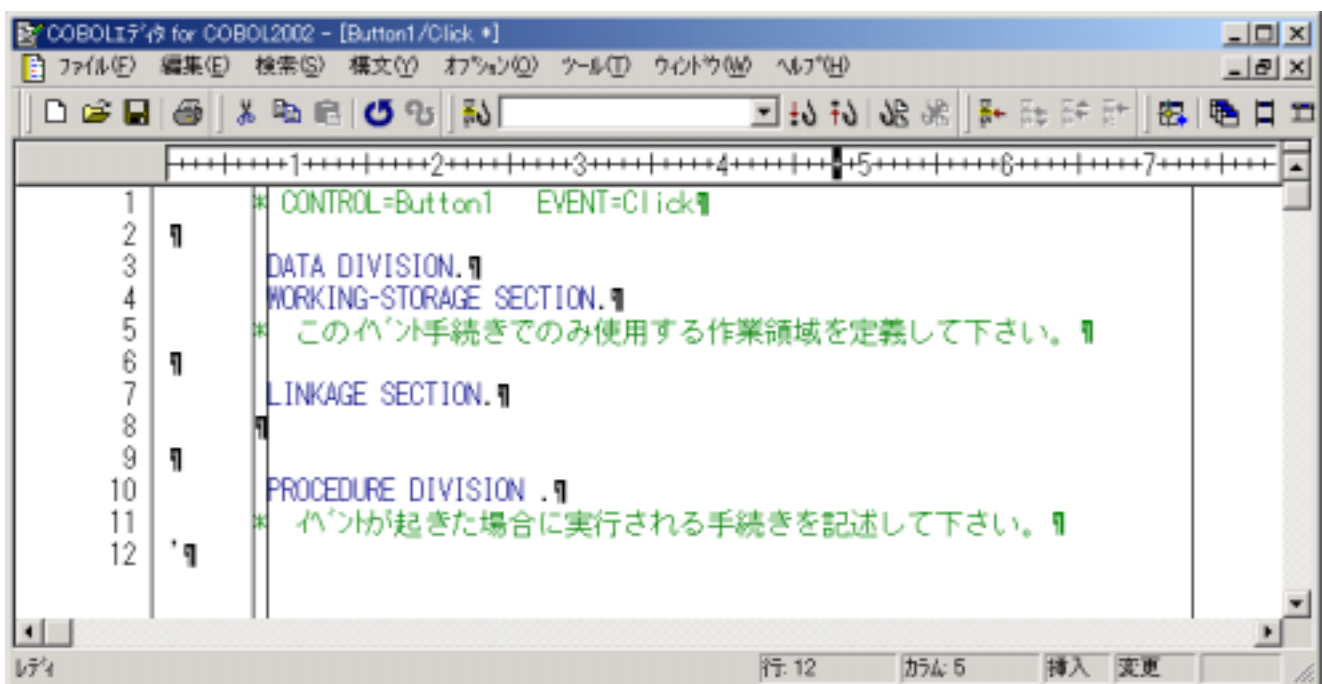


図 2 . 4 1 エディタ起動

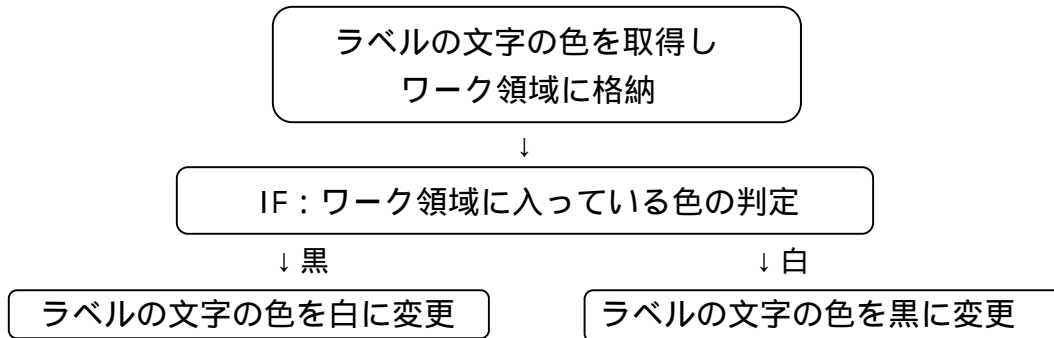
ワンポイントレッスン

イベント手続きの削除はイベント手続き編集ダイアログでイベントを選択して、削除ボタンを押すと削除できます。

イベント手続きは間違って削除すると元に戻せませんので注意してください。

2 . 7 イベント手続きの記述

メッセージラベルの文字の反転処理を次の手順で行います。



ラベル文字の色のデータの型を COBOL の型に合わせてコーディングします。
データ型は Long ですので COBOL での定義は「PIC S9(9) USAGE COMP」となります。

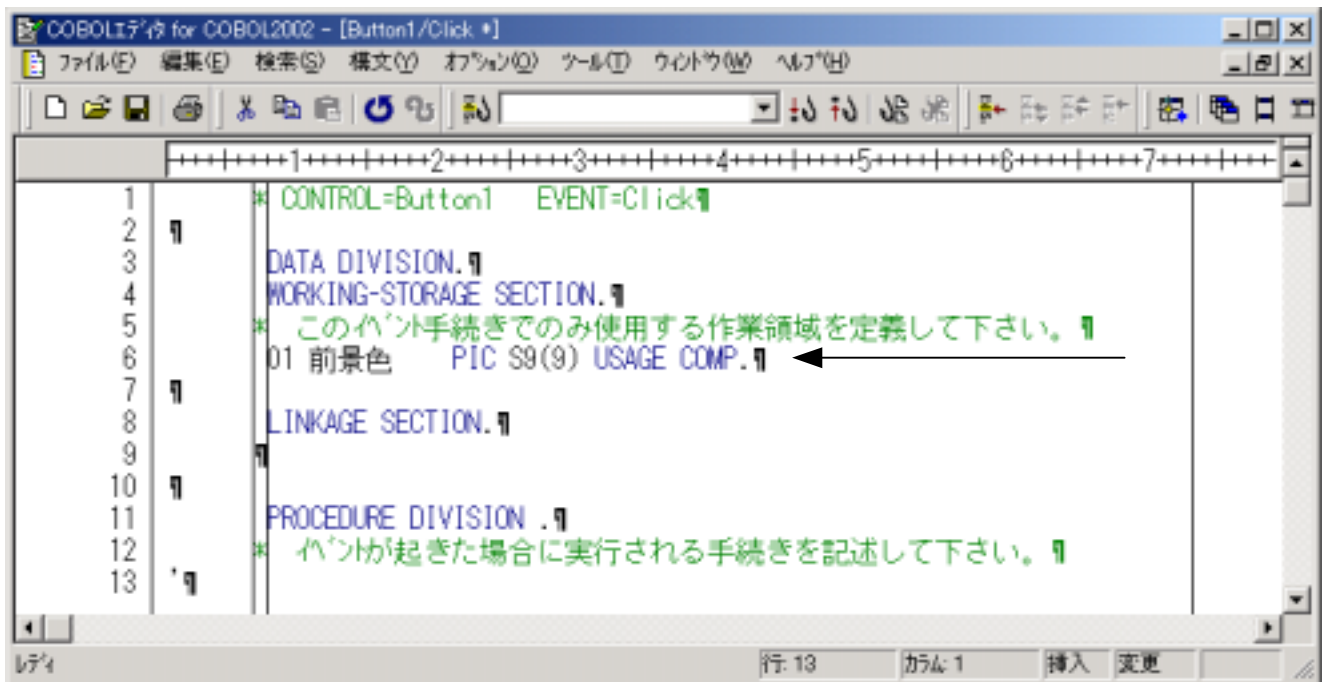


図 2 . 4 2 エディタでの定義の記述（データ名：前景色）

ワンポイントレッスン

部品のデータ型と COBOL のデータ型の対応については「3 . 1 部品の型と COBOL の型の対応」を参照してください。

プロパティの値を参照、代入するには SET 文を使用します。

データ項目の「前景色」にラベルのプロパティの前景色を格納します。

・SET 前景色 TO 'ForeColor' WITH Label1.

'ForeColor'に6けたの16進数数字定数でH'FFFFFF'（白を表します。）を設定します。

・SET 'ForeColor' WITH Label1 TO H'FFFFFF'.

ワンポイントレッスン

<プロパティの設定>

SET プロパティ名 WITH オブジェクト名 TO 送り出しデータ.

<プロパティの取得>

SET 受け取り領域名 TO プロパティ名 WITH オブジェクト名.

ワンポイントレッスン

イベント手続きで文字や GUI 部品の色を設定する場合には 16 進数数字定数を使います。基本色は以下の通りです。

H'FFFFFF' : 白	H'000000' : 黒	
H'FF0000' : 青	H'00FF00' : 緑	H'0000FF' : 赤
H'FFFF00' : シアン	H'FF00FF' : マゼンタ	H'00FFFF' : 黄

全体のコーディングは次のようになります。

```

1  # CONTROL=Button1  EVENT=Click
2
3  DATA DIVISION.
4  WORKING-STORAGE SECTION.
5  * このイベント手続きでのみ使用する作業領域を定義して下さい。
6  01 前景色    PIC S9(9) USAGE COMP.
7
8  LINKAGE SECTION.
9
10
11  PROCEDURE DIVISION .
12  * イベントが起きた場合に実行される手続きを記述して下さい。
13  SET 前景色 TO 'ForeColor' WITH Label1.
14  IF 前景色 = 0 THEN
15      SET 'ForeColor' WITH Label1 TO H'FFFFFF'
16  ELSE
17      SET 'ForeColor' WITH Label1 TO H'000000'
18  END-IF.
  
```

図 2 . 4 3 反転ボタン Click イベントのコーディング全体

2.8 GUI構築パックの終了

COBOL エディタの「ファイル(F)」から「エディタの終了(X)」を選択して、COBOL エディタを終了します。

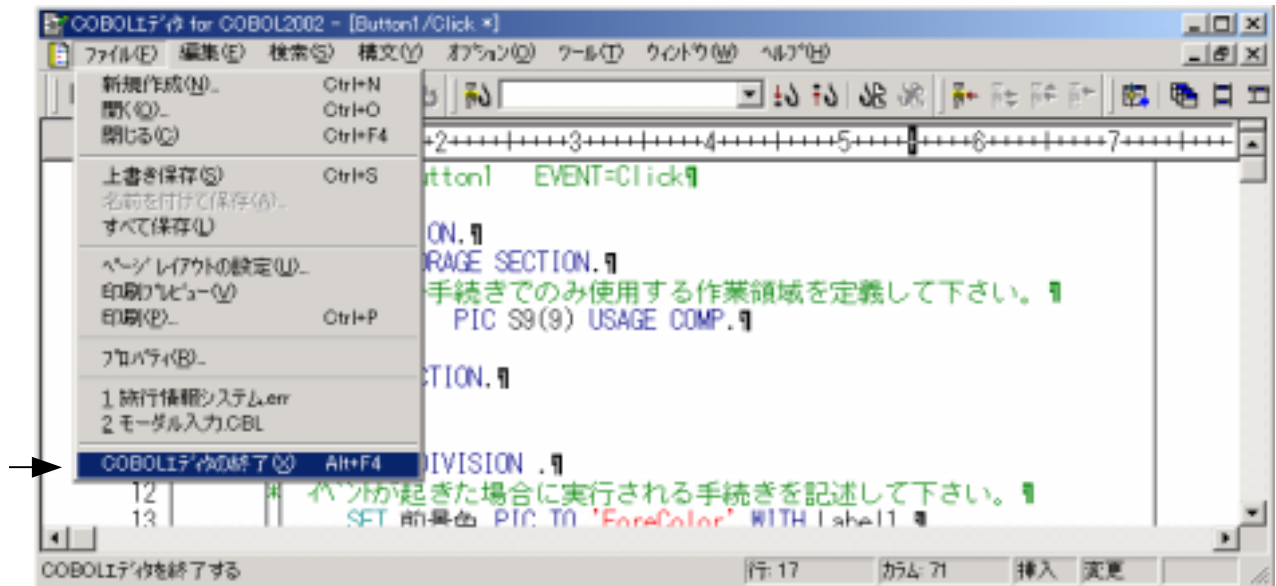


図 2.44 エディタの終了

「変更を保存」ダイアログが表示されたら「はい(Y)」をクリックしてください。



図 2.45 Button1/Click の「変更を保存」ダイアログ

GUI エディタの「ファイル(F)」から「GUI エディタの終了(X)」を選択し、GUI エディタを終了します。



図 2.46 GUI エディタの終了

「変更を保存」ダイアログが表示されたら「はい(Y)」をクリックしてください。

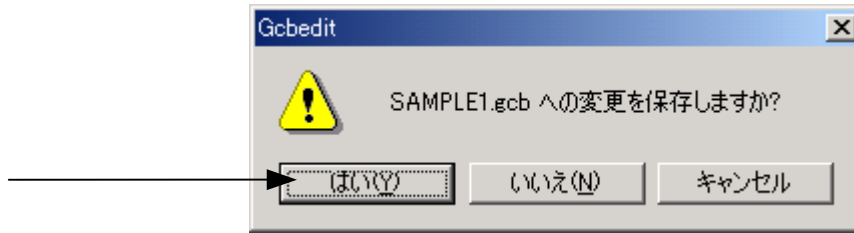


図 2 . 4 7 SAMPLE1.gcb の「変更を保存」ダイアログ

ワンポイントレッスン

- ・ GUI エディタの「変更を保存」では GUI 定義ファイルが書き込まれます。
- ・ 書き込みを行わないとエディタで行った変更が反映されませんので注意してください。
- ・ 手続きを間違えて削除してしまった場合などは、GUI エディタの「変更を保存」ダイアログで「いいえ」を選択すれば、前回に変更を保存したときの状態に戻すことができます。

2 . 9 コンパイル方法

「開発マネージャ」の「ビルド(B)」から「ビルド(B)」を選択して、アプリケーションを作成します。コンパイルエラーが発生したときの修正手順は「COBOL2002 操作入門書」を参照してください。

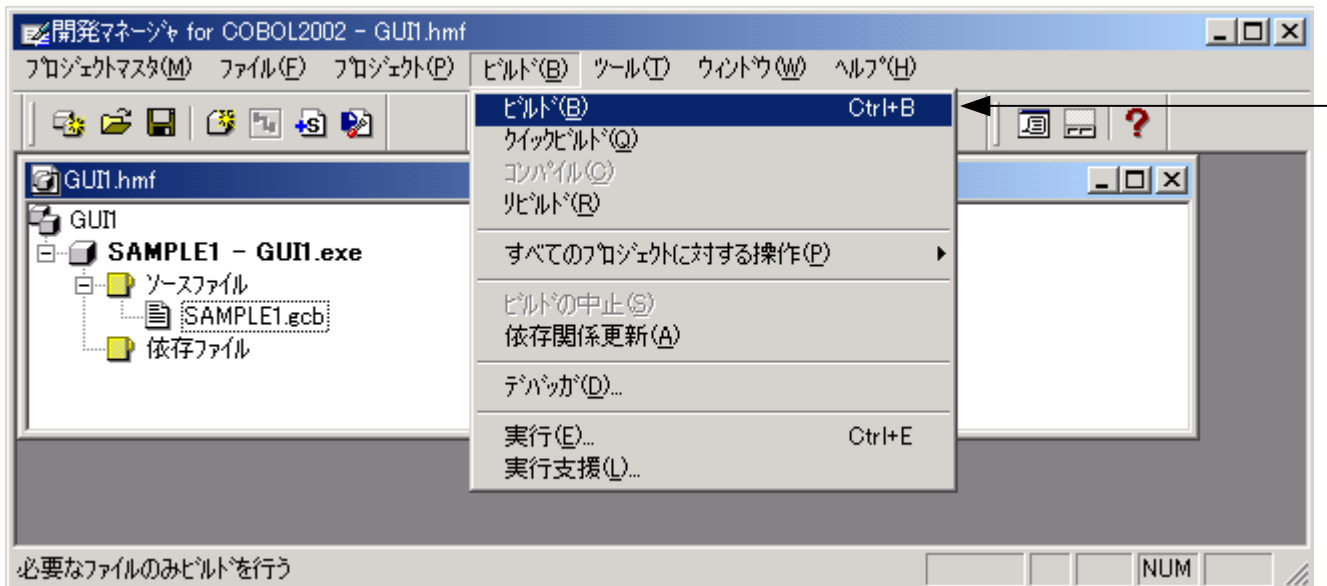


図 2 . 4 8 ビルド選択

2.10 デバッグ方法

ここでは GUI アプリケーションのデバッグ時のテストデバッガの画面と表示の意味を説明します。デバッグの詳細な手順については「COBOL2002 操作入門書」を参照してください。

テストデバッガー一覧画面はこのように表示されます。名称の左側が「+」になって隠れているときは、「+」をクリックすると表示されます。また、イベント手続き名を右クリックして「定義部の表示(F)」や「実行部の表示(M)」をクリックするとソースを表示できます。

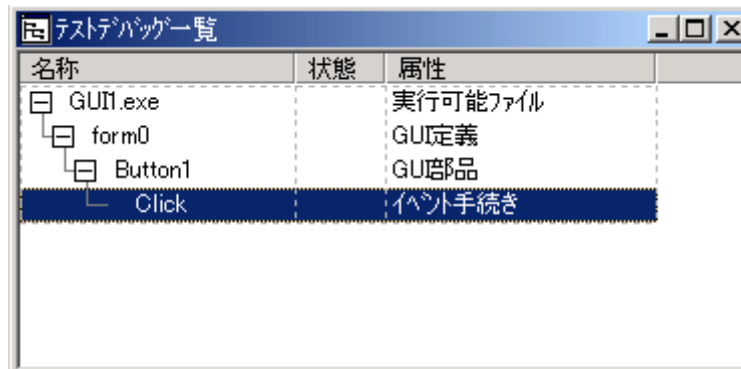


図 2.49 テストデバッガー一覧画面

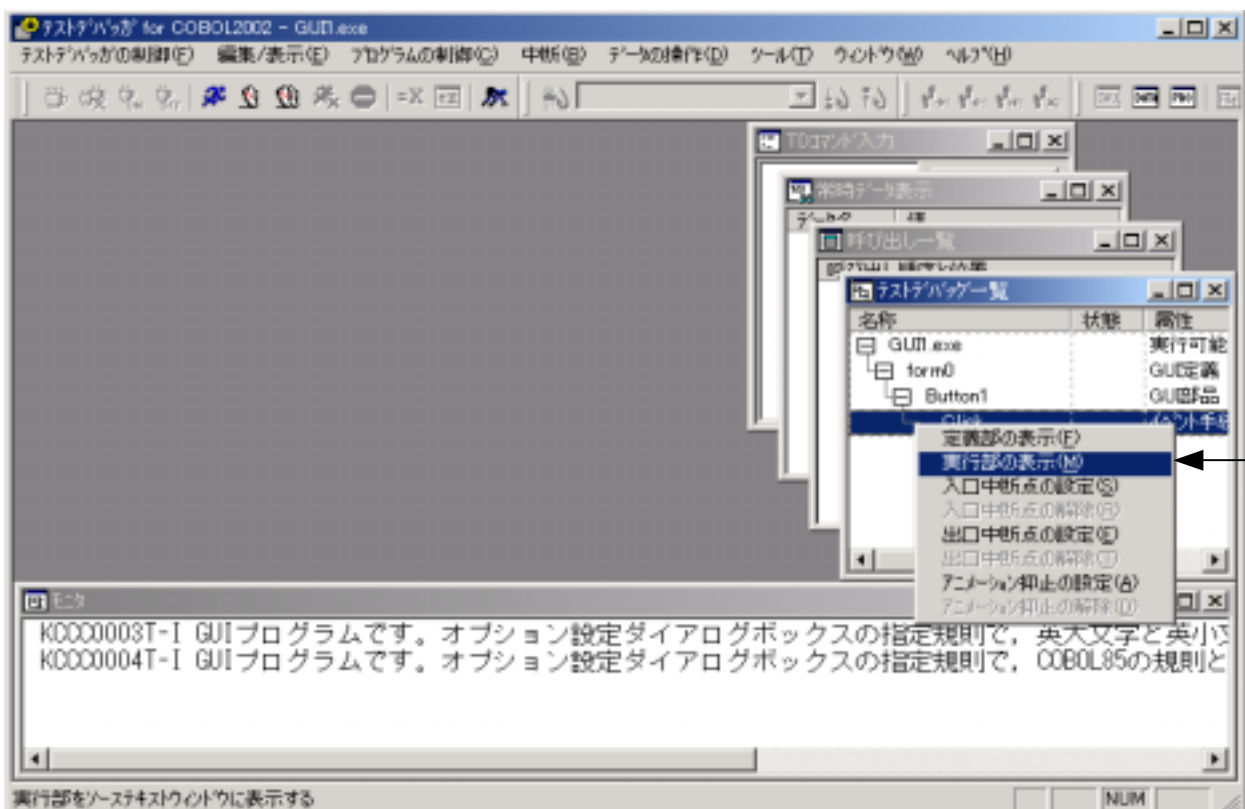


図 2.50 実行部の表示をクリック

「実行部の表示」は次のようになります。



図 2 . 5 1 実行部の表示

入入口中断点を設定してみましょう。イベント手続き名を右クリックし、「入入口中断点の設定(S)」をクリックします。

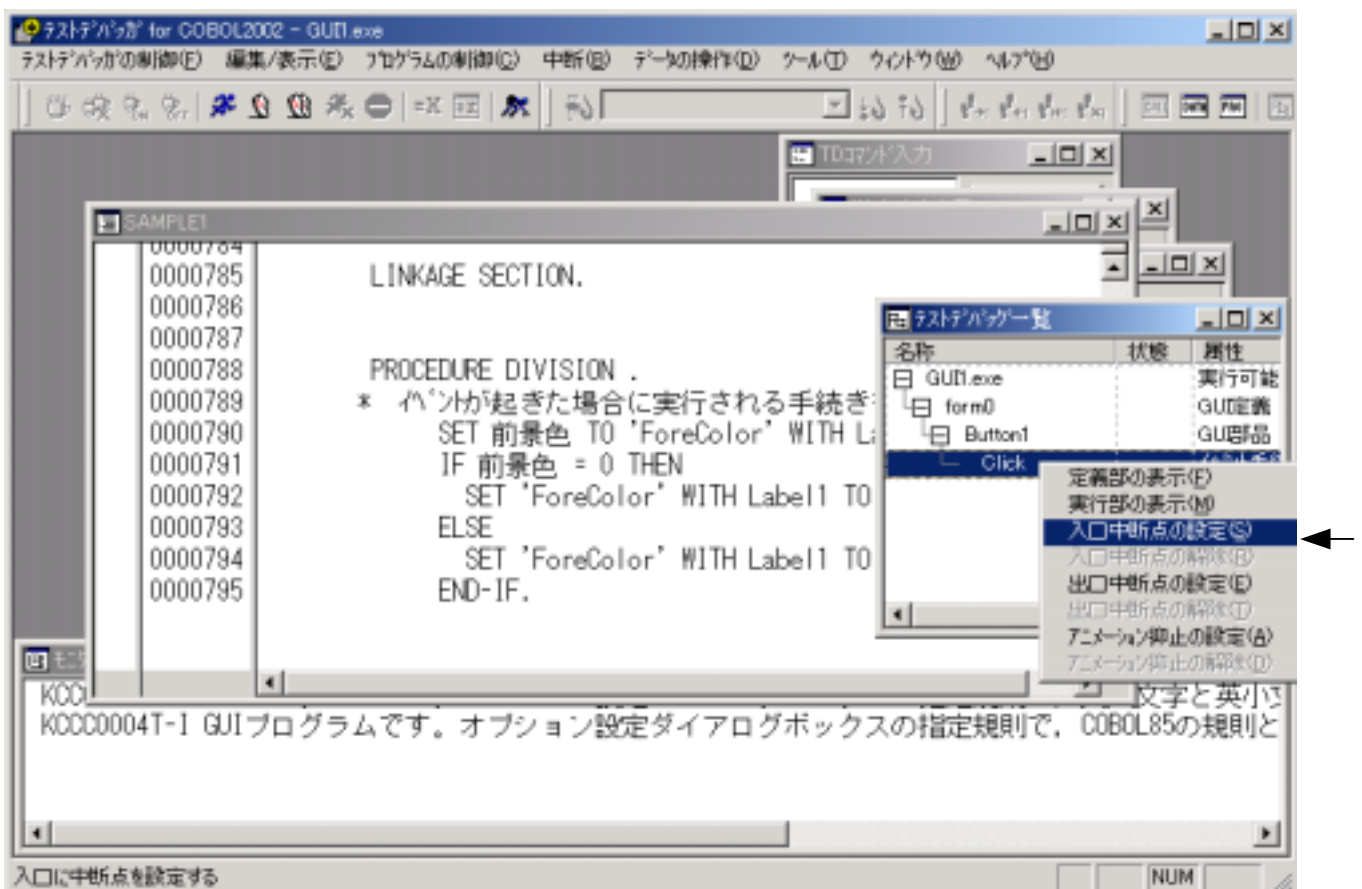


図 2 . 5 2 入入口中断点の設定

続いて実行すると GUI アプリケーションの画面が表示されます。

GUI アプリケーションの「反転」ボタンをクリックすると、テストデバッガの画面が表示されます。

テストデバッガの実行画面で、部品名「Label1」を選択して右クリックし、「データ値の表示(D)」を選択すると「データ値の表示」ダイアログが表示されます。

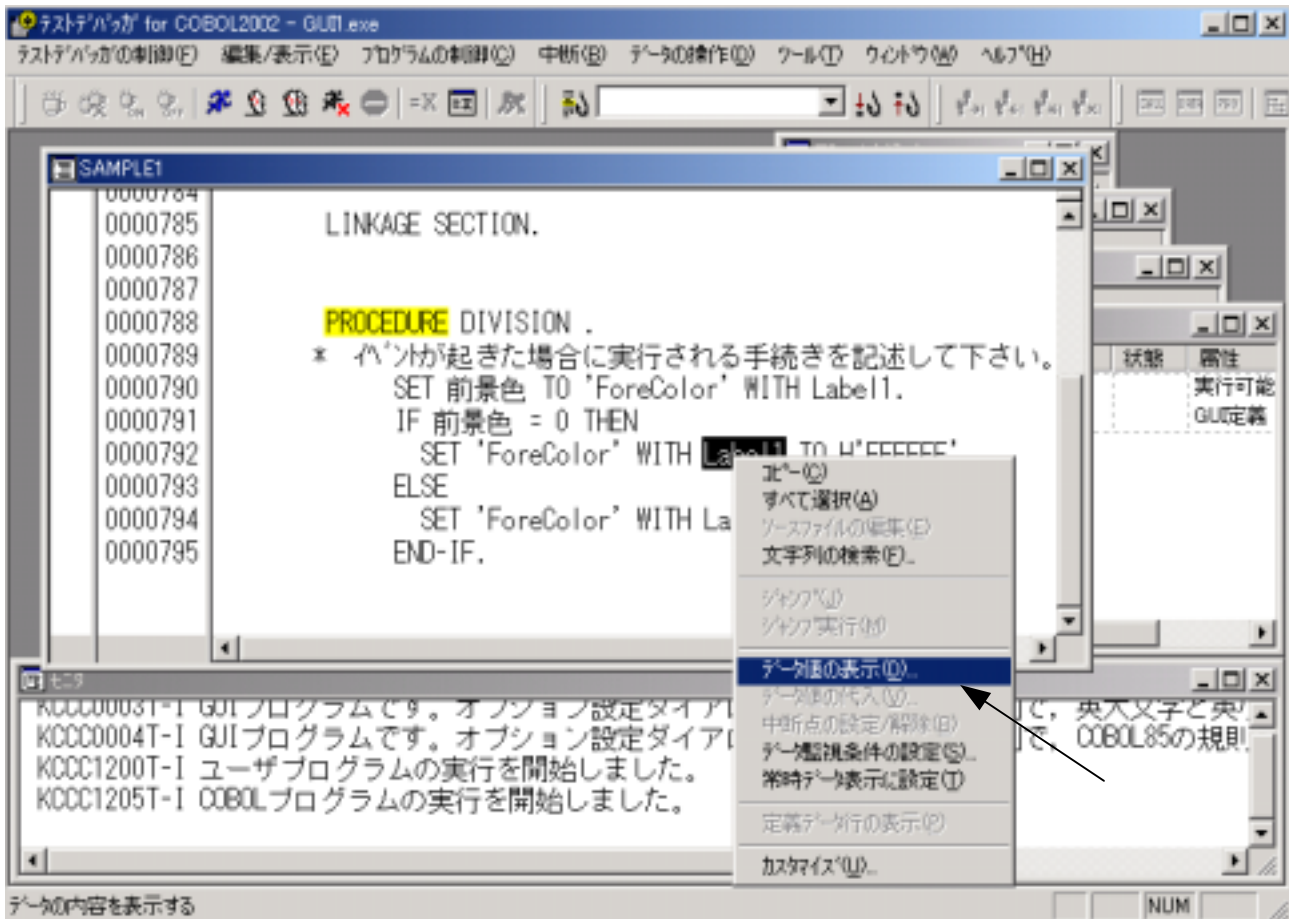


図 2 . 5 3 データ値の表示

「データ値の表示(D)」ダイアログで「OLE プロパティの表示(P)」をチェックして「OK ボタンをクリックします。



図 2 . 5 4 データ値の表示ダイアログ

モニタ画面にプロパティの内容が表示されます。全てのプロパティがアルファベット順に表示されるので、スクロールバーを使って見たい所に位置付けてください。

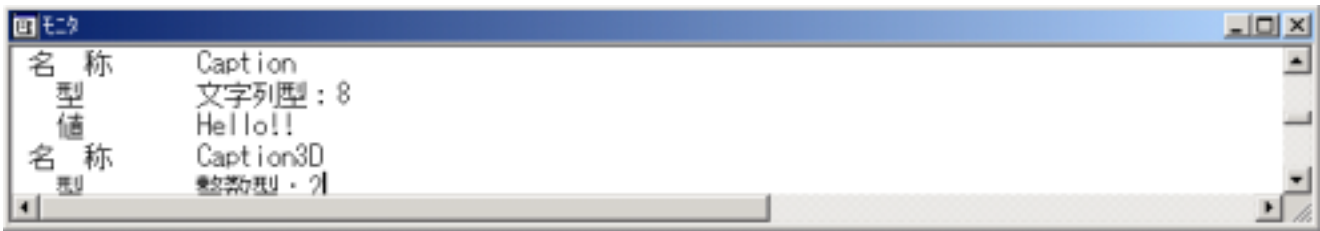


図 2 . 5 5 プロパティの表示結果

ワンポイントレッスン

部品名、バリエーションデータ項目、オブジェクト参照データ項目にはテストデバッガから値の設定はできません。

続けて実行してみましょう。GUI アプリケーションの画面に制御が渡りますが、テストデバッガの画面からは自動的に切り替わりません（画面左下の状態表示は、「プログラム実行中」になっています）。

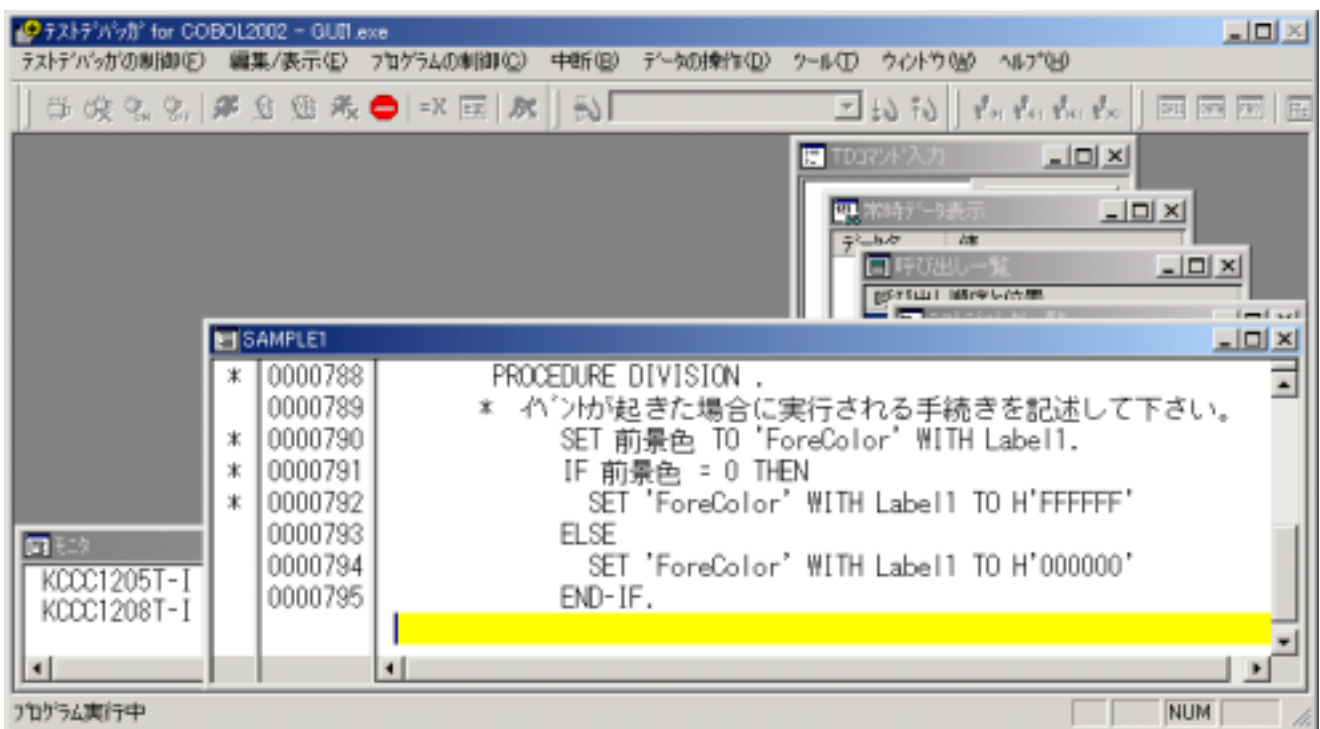


図 2 . 5 6 テストデバッガの画面（実行中）

ワンポイントレッスン

テストデバッガから GUI アプリケーションへ処理が渡るときには、画面は自動的に切り替わりません。利用者がマウスを使用して GUI アプリケーションの画面に切り替える必要があります。ただし、テストデバッガに制御が渡っているときには GUI アプリケーションに画面を切り替えることはできません。

2 . 11 実行

「開発マネージャ」の「ビルド(B)」から「実行(E)…」を選択します。

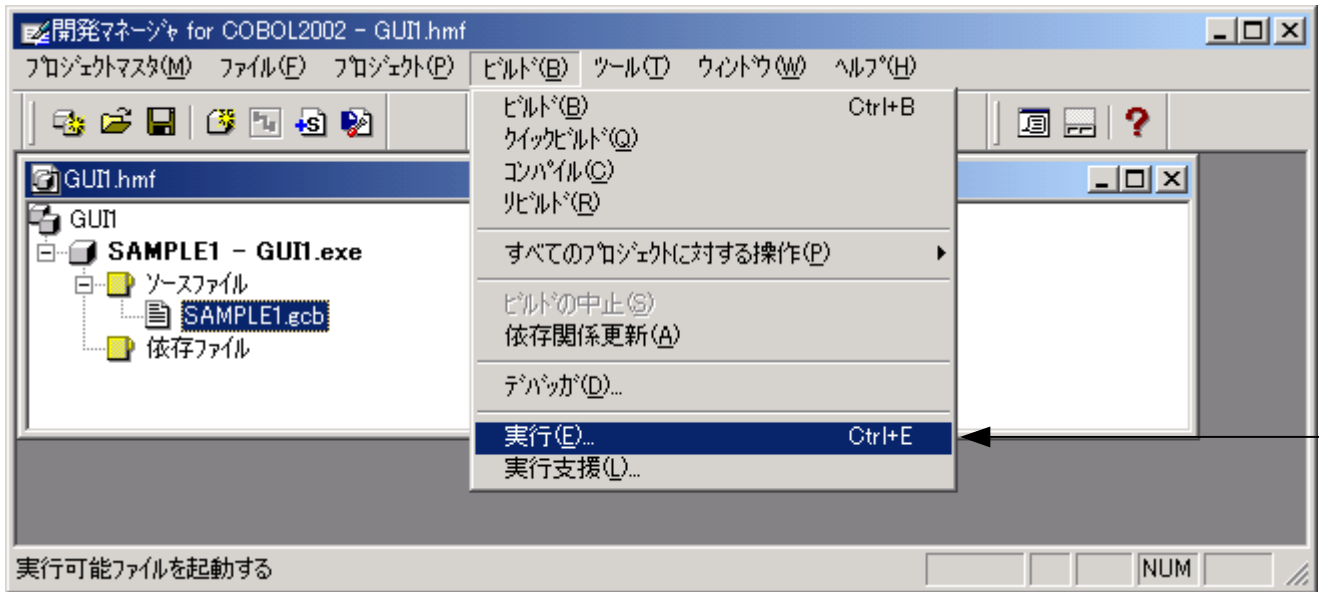


図 2 . 5 7 「開発マネージャ」から実行

実行ダイアログの「OK」ボタンで GUI アプリケーションを実行します。

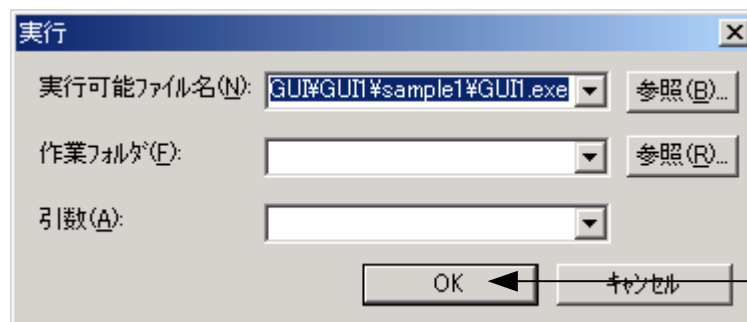


図 2 . 5 8 実行ダイアログ

GUI アプリケーションの画面が表示されます。

GUI1.exe の仕様の通り動作すれば完成です。

終了させるときは、GUI アプリケーションの画面右上の「閉じる」ボタンを押してください。

3 . 補足資料

3 . 1 部品のデータ型とCOBOLのデータ型の対応

< 数値データ型 >

部品のデータ型	COBOL のデータ定義	データ型	バイト数 (値の範囲)
Integer	PIC S9(4) USAGE IS COMP	整数型	2 バイト (-32768 ~ 32767)
Long	PIC S9(9) USAGE IS COMP	長整数型	4 バイト (-2147483648 ~ 2147483647)
Single	USAGE IS COMP-1	単精度浮動小数点型	4 バイト (-3.402823E38 ~ -1.401298E-45 1.401298E-45 ~ 3.402823E38)
Double	USAGE IS COMP-2	倍精度浮動小数点型	8 バイト (-1.79769313486231E308 ~ -4.94065645841247E-308 4.94065645841247E-308 ~ 1.79769313486231E308)

< その他のデータ型 >

部品のデータ型	COBOL のデータ定義	データ型
Currency	PIC X(領域長) 領域長はプロパティの値に依存します。	通貨型
String	PIC X(領域長) 領域長はプロパティの値に依存します。	文字列型
Date	PIC X(領域長) 領域長はプロパティの値に依存します。	日付型
Boolean	PIC 1(1) USAGE BIT	ブール型
Object	USAGE IS OBJECT REFERENCE OLE	オブジェクト型
Variant	USAGE IS VARIANT	バリエーション型
Null	USAGE IS VARIANT	NULL 型
Empty	USAGE IS VARIANT	EMPTY 型

3.2 部品のデータ型の検索方法

フォームエディタの「ヘルプ(H)」から「トピックの検索(H)」を選択します。



図3.1 フォームエディタの「トピックの検索(H)」を選択

「目次」タグをクリックします。

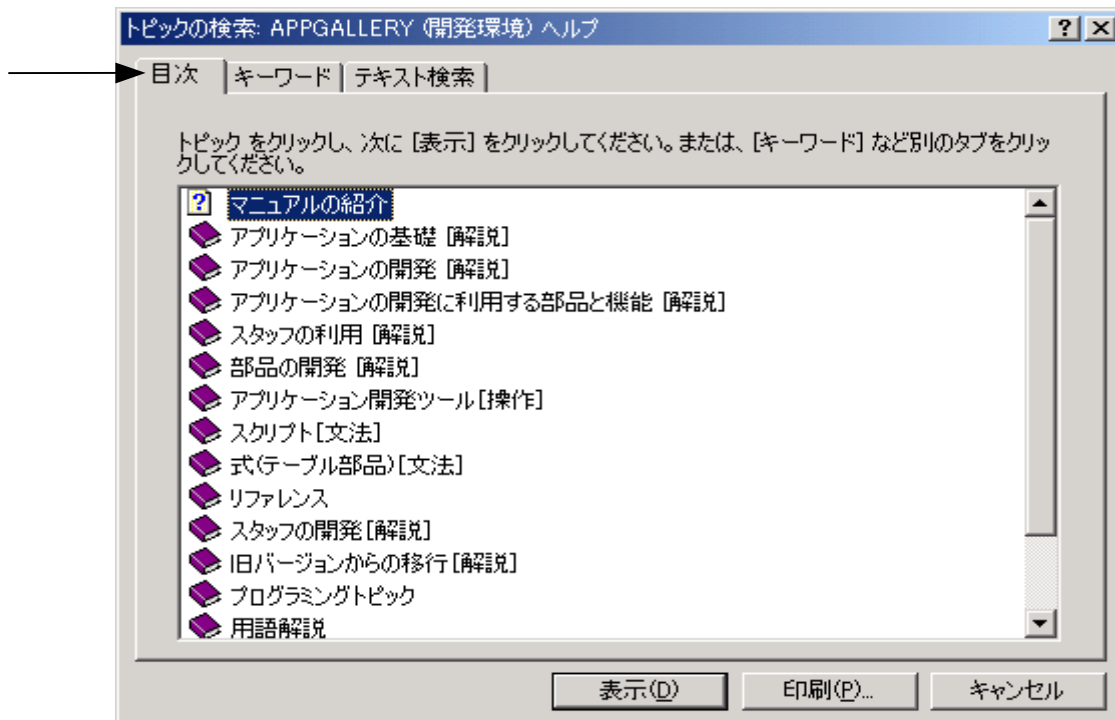


図3.2 「目次」タグをクリック

「リファレンス」をダブルクリックします。

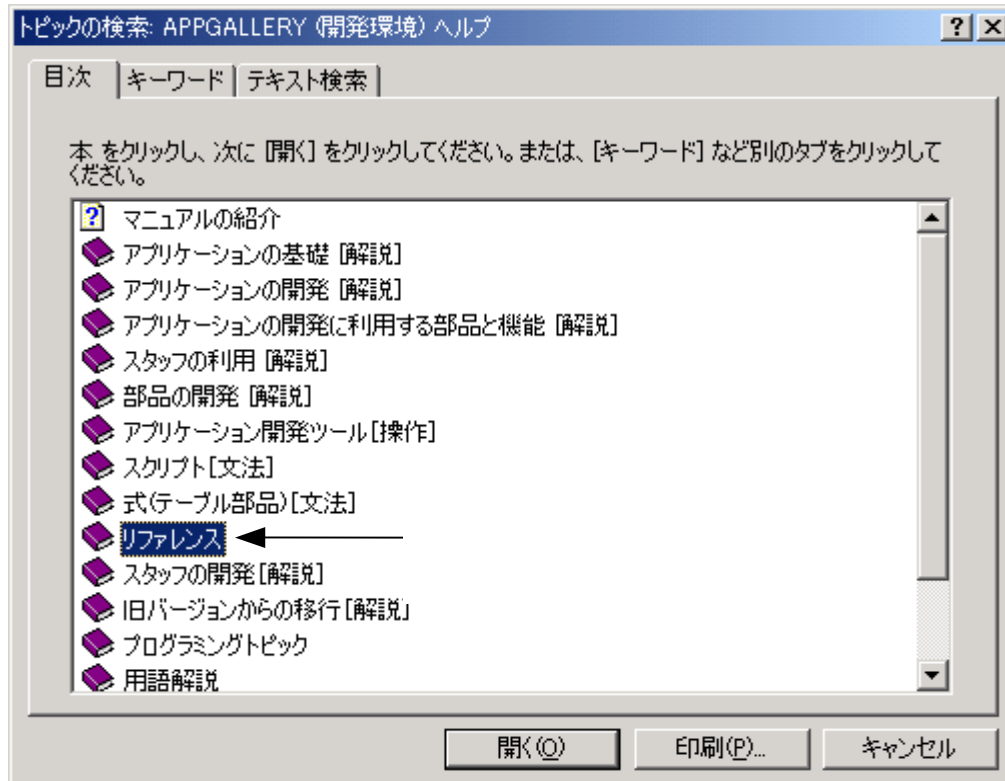


図 3 . 3 「リファレンス」をダブルクリック

「GUI 作成で使用する部品」をダブルクリックします。

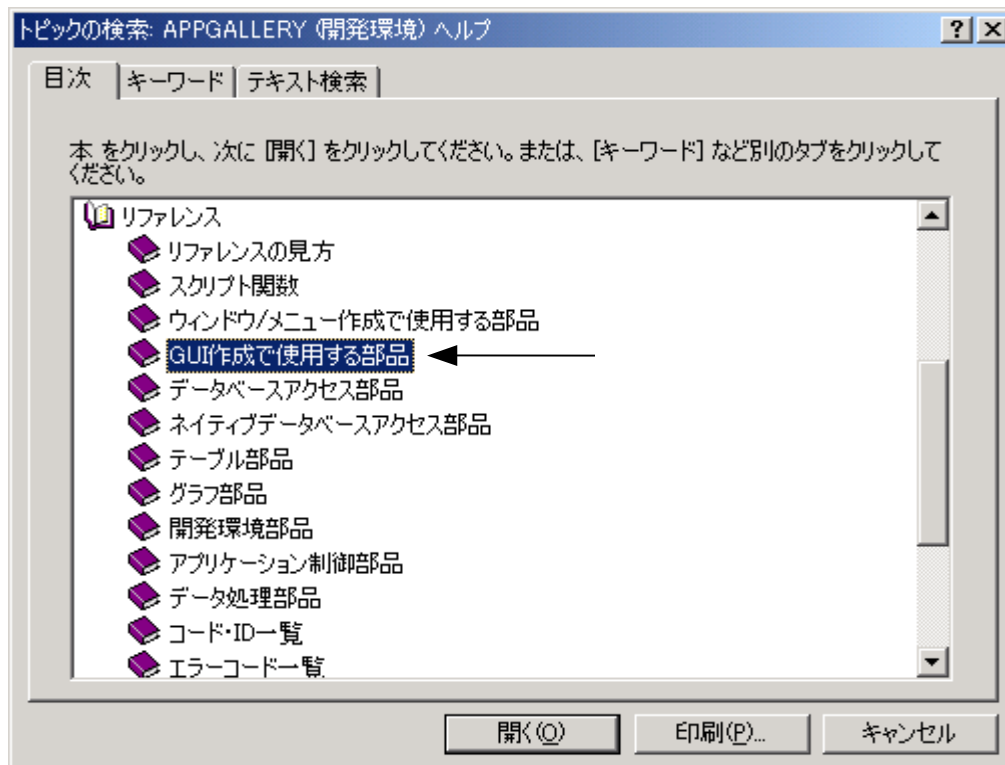


図 3 . 4 「GUI 作成で使用する部品」をダブルクリック

「オブジェクト一覧」をダブルクリックします。

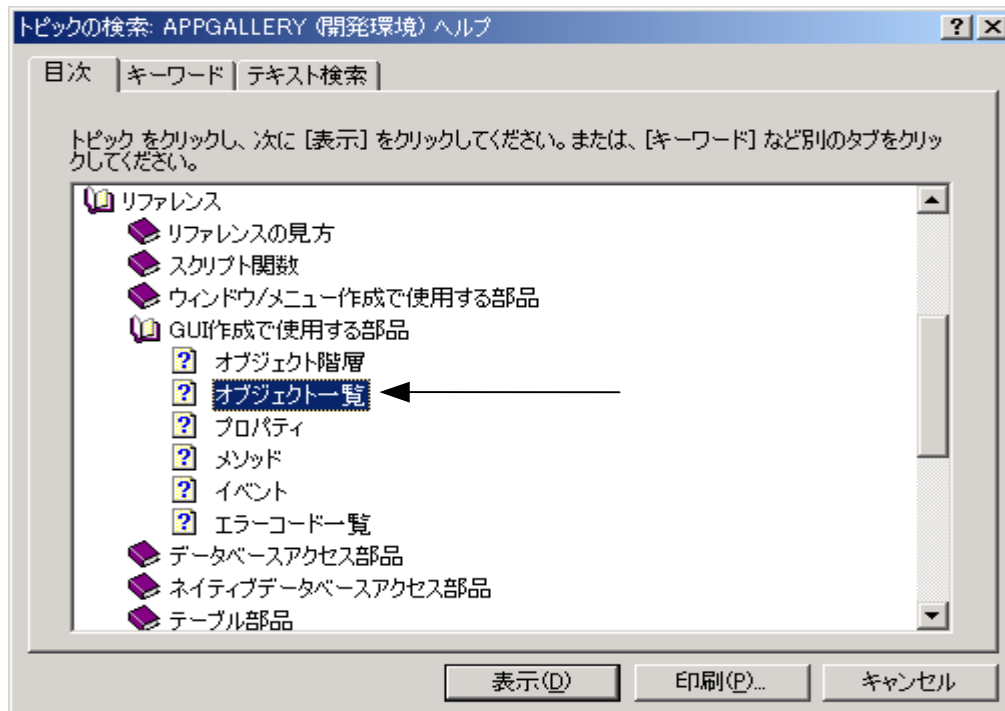


図 3 . 5 「オブジェクト一覧」をダブルクリック

「Label オブジェクト」の「プロパティ」をクリックします。

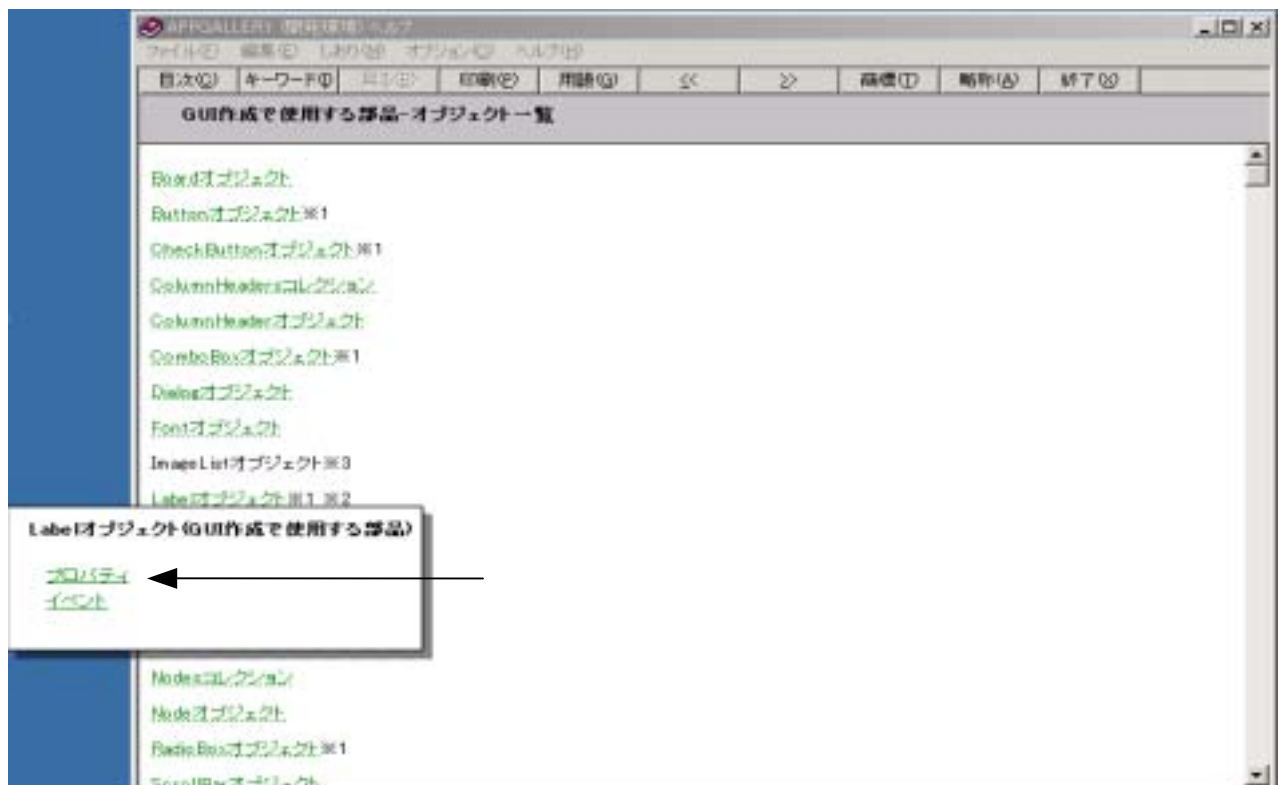


図 3 . 6 「プロパティ」をクリック

ラベルの文字の色は「前景色」を変更することにより変わります。

「前景色」のプロパティは「ForeColor」です。「ForeColor」をクリックします。

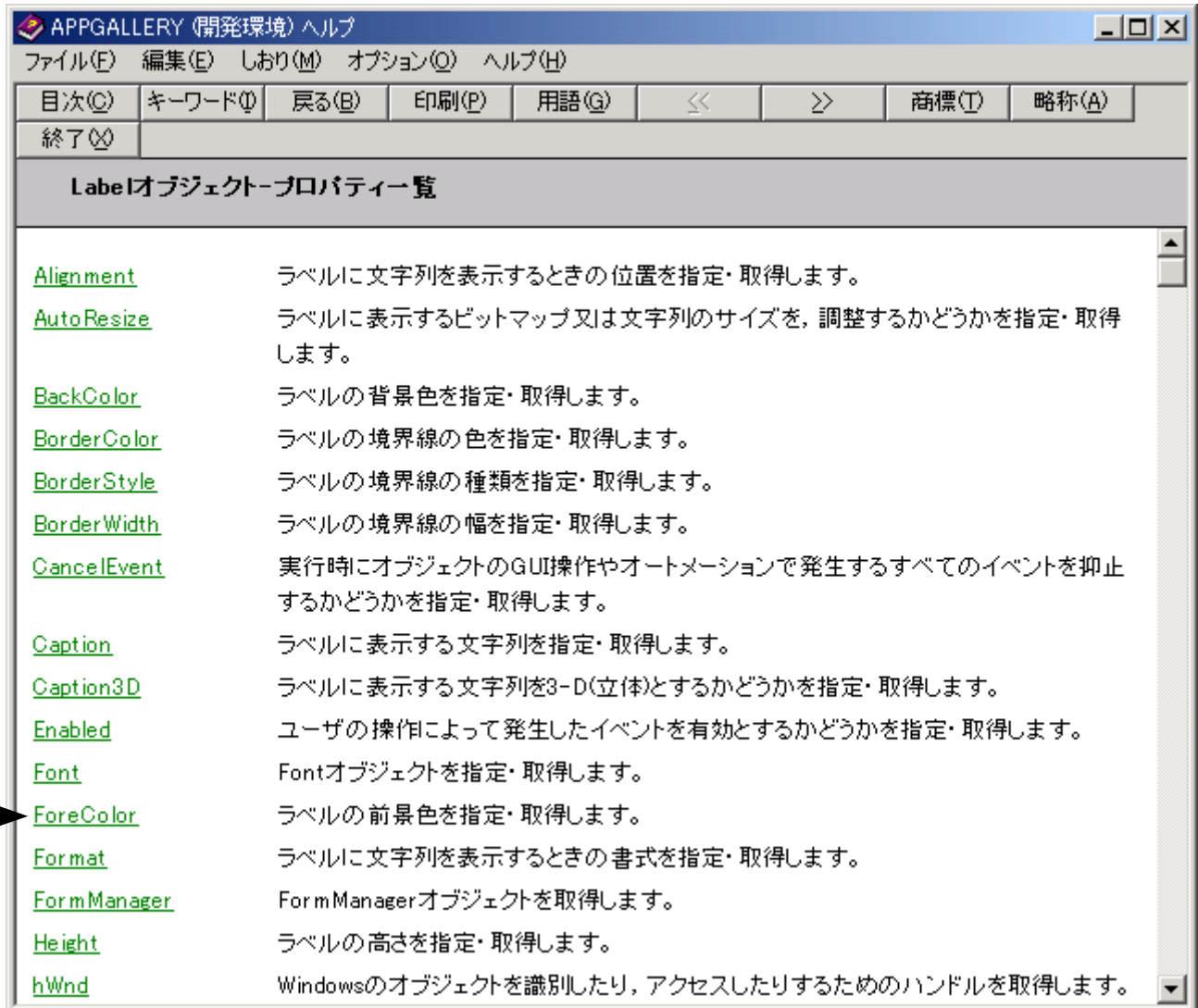


図 3 . 7 「ForeColor」をクリック

ForeColor プロパティのデータ型(Long)情報。

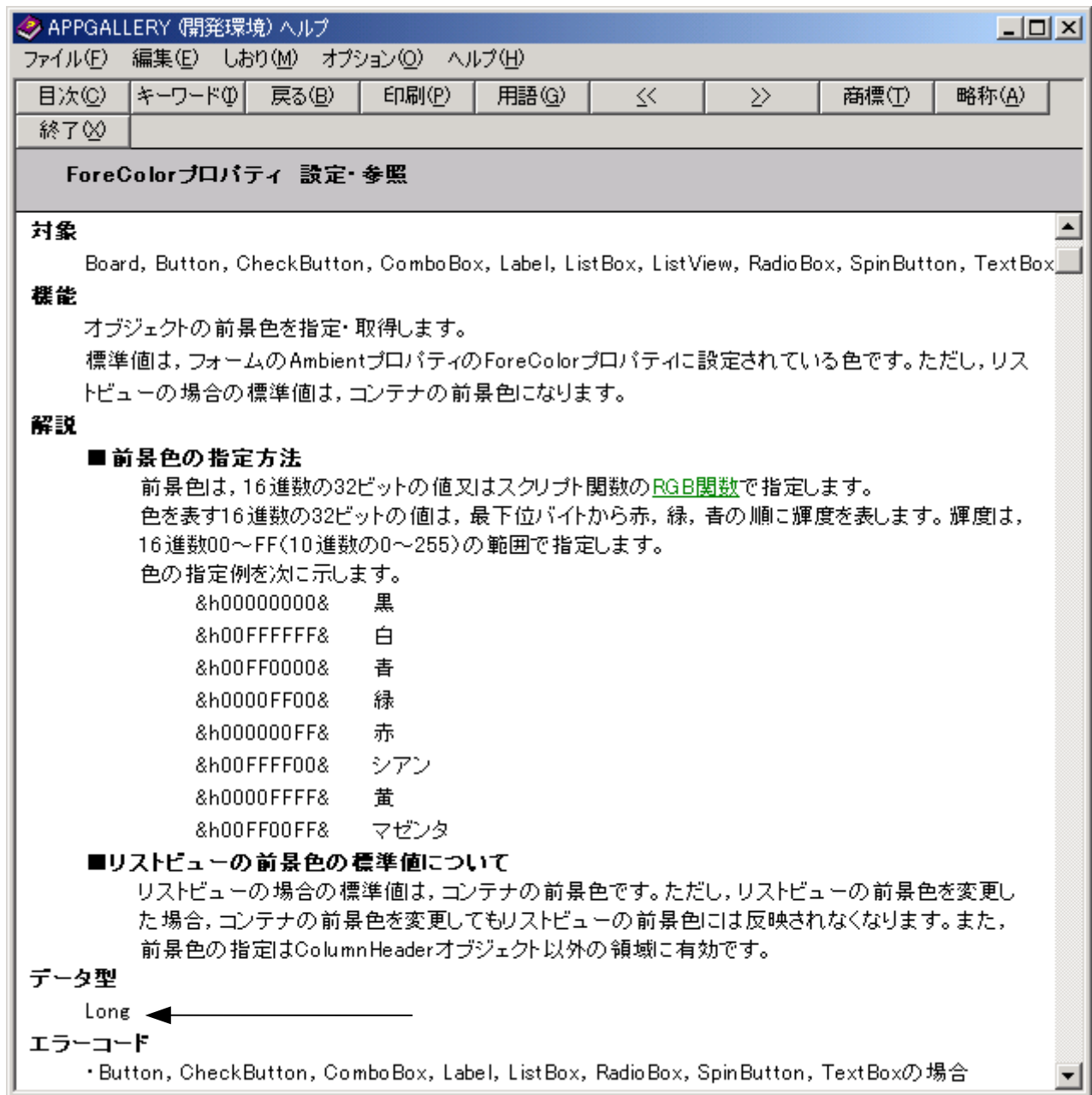


図 3 . 8 「ForeColor」プロパティ情報

3 トラブルシューティング

3.3.1 GUI アプリケーション作成時の注意事項

Q 1 : イベント手続きを保存して、再コンパイルしましたが、アプリケーションのイベント手続きの内容が変更されません。

A 1 : GUI エディタで GUI 定義ファイルを保存しましたか？

「GUI 構築パック」では、イベント手続きの保存には 2 段階の操作が必要になります。

(1) イベント手続きを保存する。

(2) すべての編集終了後に GUI エディタで GUI 定義ファイルを保存する。

GUI 定義ファイルを保存をしないと、GUI 定義ファイルが変更されませんので、イベント手続きの変更は無効となります。GUI エディタで GUI 定義ファイルを保存して、再コンパイルしてください。

Q 2 : 「フォームが既に関かれています」とメッセージが表示され、「GUI エディタ」が起動しません。

A 2 : 修正対象のアプリケーションが実行中ではありませんか？

「GUI 構築パック」で作成する 2 種類のファイルのうちフォームファイルは実行時にも使用されます。したがって、修正対象のアプリケーションが実行中である場合、排他処理によって GUI エディタの起動が抑止されます。

修正対象のアプリケーションを終了させてから GUI エディタを起動してみてください。

Q 3 : 「STOP RUN」文はなぜ使用できないのですか。

A 3 : GUI アプリケーションの手続き中に「STOP RUN」を使用できません。

これはアプリケーション終了時に、自動生成した終了処理を呼ばなくてはならないためです。GUI アプリケーション終了時の記述方法は、マニュアル「COBOL GUI オプション 操作入門」の「3.5 GUI アプリケーションの終了を記述する」を参照してください。

Q 4 : コンパイル時に「自分自身を呼び出せません」というエラーメッセージが表示されてしまうのですが。

A 4 : 画面の切り替え処理の時に自分のフォーム名を呼び出していないですか？

画面の切り替え処理の時に自分自身への切り替えはできません。

別のフォームを呼び出すように修正してください。

Q 5 : コンパイル時に「同一オブジェクトファイル指定エラー」メッセージボックスが表示されて、リンクできません。

A 5 : 複数のフォームに同じ名前を設定していませんか？ 重複した名前を変更してください。

3 . 3 . 2 GUI アプリケーション実行時の注意事項

Q 1 : 「フォームが既に開かれています」とメッセージが表示され、アプリケーションがエラーになってしまうのですが。

A 1 : フォームエディタでアプリケーションを修正中ではありませんか？

「GUI 構築パック」で作成する 2 種類のファイルのうち、フォームファイルは実行時にも使用されます。したがって、フォームエディタで修正中である場合、排他処理によってアプリケーションの実行を中止します。フォームエディタを終了させてからアプリケーションを実行してください。

Q 2 : 「AGF ファイルがありません」のメッセージが表示されて実行できません。

A 2 : 実行ファイル (.exe ファイル) と同じフォルダにフォームファイルはありますか？
作業フォルダを設定した場合、フォームファイルは実行ファイルと異なるフォルダに作成されますので、実行ファイルと同じフォルダにコピーする必要があります。

Q 3 : TextBox に文字列を入力しないで処理をすると英数字項目への転記で実行時エラーとなってしまいます。

A 3 : 文字列の長さが 0 の場合、Variant 型では文字列データではなく、EMPTY データとして扱うため、データ型の不一致でエラーとなります。TextBox の TYPE プロパティを「1 - 文字列」に設定するか、マニュアル「COBOL GUI オプション ユーザーズガイド」の「12.7 その他の手続きの記述例」の「(4) データ型が変わるプロパティ値の参照の例」を参照してコーディングしてください。

Q 4 : TextBox から英数字項目に文字列を受け取ったときに余計な文字が入ってしまいます。

A 4 : 英数字項目の領域はクリアしていますか？

SET 文ではデータの長さ分の転記しかしませんので、受け取り側領域をクリアしないと文字が残ってしまいます。INITIALIZE 文などを使って受け取り側領域をクリアしてください。

Q 5 : Label に英数字項目から値を設定すると余計な空白が入ってしまいます。

A 5 : Variant 型のプロパティに英数字項目から SET 文で転記した場合は英数字項目の領域長だけ転記されます。必要な長さだけ送りたい場合には部分参照を使って設定してください。

3 . 4 コーディングする際の注意事項

(1) 予約語は英大文字で記述しなければならない

COBOL の予約語は英大文字で記述しないとコンパイルエラーになります。

(2) 利用者語での英大文字と英小文字は同じ文字として扱われない

「GUI 構築パック」を使って COBOL の手続きを記述する場合に、利用者語の英大文字と英小文字は異なる文字として扱われます。例えば、データ名「A」と データ名「a」は異なるデータ名として扱われます。

(3) プロパティの設定や参照は S E T 文を使う

GUI 部品はプロパティというデータを持っていますが、プロパティの設定や参照には MOVE 文は使えません。プロパティの設定や参照をするには SET 文を使って記述する必要があります。

SET 文では数値データ以外のデータでは MOVE 文の様に「初期化してから転記」という処理はされません。そのため、受け取り側項目の初期化などを意識したコーディングが必要です。

(4) 利用者語に使用できない語

「GUI 構築パック」では「GCB」および「CBL」で始まる文字列を利用者語に使うことはできません。

3 . 5 GUI 構築パックの用語説明

イベント

画面上の GUI 部品に対する操作のことです。たとえば、ボタン上でマウスをクリックした場合には、「Click」イベントが発生します。

イベント手続き

「GUI 構築パック」で、イベントが発生したときに対応して呼ばれる COBOL の手続きのことです。

オブジェクト

データとそのデータを処理する手続きが一体となった固まりのことです。GUI 部品はデータ（プロパティ）とデータを処理する手続き（メソッド）とイベントを持つオブジェクトです。

オブジェクトデータ項目（オブジェクト参照データ項目）

GUI 部品などのオブジェクトを扱うためのデータ項目です。

開発マネージャ（COBOL 開発マネージャ）

コンパイル対象となるファイルやコンパイラオプションなどコンパイルやリンケージに必要な情報を管理するツールです。

共通手続き

一つの画面の複数のイベント手続きから呼ばれる処理のことです。GUI エディタのメニューから設定できます。

共通領域

一つの画面の複数のイベント手続きから使用できる領域のことです。GUI エディタのメニューから設定できます。

サービスルーチン

COBOL2002 および「GUI 構築パック」の言語仕様がない機能を CALL 文で呼び出すプログラムとして提供しています。これをサービスルーチンといいます。

終了処理

一つの画面の終了時に必ず実行される処理のことです。GUI エディタのメニューから設定できます。

初期処理

一つの画面の作成時に必ず実行される処理のことです。GUI エディタのメニューから設定できます。

スクリプト

Visual Basic や Java での手続きのことです。

遷移

アプリケーションの画面が切り替わったとき、切り替わった画面が表示されて、元の画面は消去されている状態のことです。

バリエーションデータ項目

OLE で規定される Variant 型の値を格納するためのデータ項目です。メソッドやプロパティの操作のときに使用します。

フォーム

GUI 画面のインタフェースを配置する土台となる部品のことです。

フォームエディタ

GUI アプリケーションの表示に関する設定を編集するツールです。

フォームファイル(.agf)

フォームエディタで作成したフォームの 1 画面分の情報が格納されているファイルです。編集時だけでなくアプリケーション実行のときにも必要なファイルです。

部品パレット

GUI 部品を表示するツール画面のことです。ここからドラッグアンドドロップでフォームエディタの上に GUI 部品を貼り付けて、GUI アプリケーションの画面を作ります。

プロジェクトファイル(.hmf)

コンパイル対象となるファイルやコンパイラオプションなどコンパイルやリンケージに必要な情報を管理するファイルです。

プロパティ

GUI 部品やフォームが持つ、データのことです。

メソッド

GUI 部品やフォームが持つ、データ（プロパティ）を処理するための手続きのことです。

モーダル

アプリケーションの画面が切り替わったとき、切り替わった画面には操作ができ、元の画面には操作できなくなっている状態のことです。

モードレス

アプリケーションの画面が切り替わったとき、切り替わった画面にも、元の画面にも操作できる状態のことです。

AGF ファイル

フォームファイルと同義です。フォームファイルの項を参照してください。

GUI エディタ（主メニュー域）

ボタンやラベル、テキストボックスなどのイベント手続きの登録をして、画面の情報を管理するツールです。（この操作入門書では「GUI エディタ」として記述します。）

GUI 定義ファイル(.gcb)

GUI エディタで作成したフォームの 1 画面分の手続き情報が格納されています。編集時、コンパイル時に必要なファイルです。

GUI 部品

ボタンやラベル、テキストボックスなど画面のインタフェースを構成するための部品のことです。

《他社所有名称に対する表示》

- OLE は、米国 Microsoft Corp.が開発したソフトウェア名称です。OLE 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。
- 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。