<u>スクール COBOL2002 GUI 構築パック</u>

操作入門書



(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。

3/45

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。

<u>仕様(GUI1.exe)</u>

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。

Hello!!		Hello!!
反転	\rightarrow	反転
[form[]		form0
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 $\mu(\underline{F})$ 」の「新規作成(\underline{N})」、をクリックします。新規作成画面が表示されます。

	■費用マネージャ for COBOL2002		
	711ジェクトマスタの10 クール(10 ヘルフで円)		
1	新規作成(型)。 開((型)。 Chri+O] ≝ ≌ ∰ ≌ ≿ ≪ #] ⊴ ≌	
	デフォルオフジョンの設定(型)		
	最新の7.276		
	開発マネージャの終了(2)		
	新規にPO開発環境用プロジェ外マスタファイルを	作成する	NUM //

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。

ንግንትታንትዋスタの作成			×	
プロジェクトマスタ名(<u>M</u>):	GUM I			
ን"ロジェクトマスタフォルダ名(<u>F</u>):	D:¥Temp¥GUI¥GUI1	参照(<u>R</u>)]	
		完了▲キ	*ンセル	

図2.4 新規作成画面

スクール COBOL2002 GUI 構築パック

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。

開発マネーシ	°r for COBOL2002 X	
⚠	D¥Temp¥GUI¥GUI1(は存在しません。作成しますか?	
	OK キャンセル	

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。

ንግን፣ታሉወ作成 🔀	
プロジェクト名(<u>N</u>): SAMPLE1 ◀	
プロジェクトフォルダ(<u>P</u>): D:¥Temp¥GUI¥GUI1¥SAMPLE1 参照(<u>R</u>)	
作業フォルダ): 参照(E)	
▼ フロジェクトフォルダと作業フォルダを同一にする。①	
OK 🔺 ++>tell	

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。

開発マネージ	tor COBOL2002	
	D:¥Temp¥GUI¥GUI1¥SAMPLE1(は存在しません。作成しますか?	
	OK キャンセル	

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。

ንግንዥታሉመ追加:COBOLንግንዥታሉ			×	
最終生成物の種類を選択してください。				
 ● 実行形式プログラム(E) ○ ダイナミックリンクライブラリ(D) ○ 標準ライブラリ(L) 				
	< 戻る(<u>B</u>)	次へ(№)>◀	キャンセル	

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。

プロジェクトの追加:COBOLプロジェクト			×	
プロジュクトの種類を選択してください。				
System指定メインプログラム V3指定メインプログラム メインプログラムなし OLEアウトオブプロセスサーバプログラム CGIメインプログラム			A	
			T	
	く戻る(<u>B</u>)	次へ(10) > 一	キャンセル	

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。

ንግጋንኳንትመንጀክክ : COBOLንግጋንኳንኑ	×	
出力ファイル名(FXF)を入れてください。(O)		
GUII	参照(<u>R</u>)	
〈 戻る(B) 次へ(N) >	キャンセル	

図2.10 プロジェクトの追加画面3

8/45

GUI 定義ファイルを入力し、「完了」ボタンを押します。

ንግንንድንዞመንጀክበ፡ COBOLንግንንድንዞ	×
GUI定義ファイル名を入れてください。(G)	
SAMPLE1 参照(R)
, 	
< 戻る(B) 完了 キャ	シセル

図2.11 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。

jizz∰発マネージャ for COBOL2002 - GUI1.hmf	<u>- 🗆 ×</u>
ר אָראָלאָדאָלאָאָאָאָאָאָאָאָאָאָאָאָאָאָאָא	
3 2 2 3 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3	
GUII.hmf	
🔁 gun	
SAMPLE1 - GUI1.exe	
□ SAMPLE1.gcb	
ντ ^s r	

図2.12 「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2.13 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。

jizzg開発マネージャ for COBOL2002 - GUI1.hmf	
ר איז	
3 2 2 3 3 2 4 1 2 2 3 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2	
GUII.hmf	
🔁 GUN	
SAMPLE1 - GUII.exe	
	J
ντ [*] ί	

図2.14 「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。

🔜 SAMPLE1.gcb - COBOL GUI for COBOL2002	
ファイル(E) 手続き編集(P) テスト表示(T) 表示(V) ツール(L) ヘルプ(H)	
form0 🔹 Activate 💽	

図2.15 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(<u>F</u>)」から「フォームエディタの起動(<u>F</u>)」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2.16 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。

form0	
ファイル(E) 編集(E) フォーム(Q) 表示(V) ヘルプ(H)	
•	
Té an Ol	
	· · · · · · · · · · · · · · · · · · ·

図2.17 フォームエディタ

スクール COBOL2002 GUI 構築パック

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。

7ォームフ°ロハ⁰ティ	×
わジェ外名(Q) form0 表示文字列(Q) □ わジェ外を表示する(V) □ お性にする(E) □ 知び表示する(D) □ かのまー(C用) (C 上部(4))	フォームサイズ*(型): OK ウィント*ウサイズ* ・ 任意 ・ A4 210 × 297 mm ・ A5 148 × 210 mm ・ B4 257 × 364 mm ・ B5 182 × 257 mm ・ A3 297 × 420 mm ・ ハガ*1 100 × 148 mm ・
 ステータスパーを表示する(S) アイコンファイル名(E) アイコンファイルを埋め込む(A) 参照(Q) 背景色(B) 前景色(G) 	Point(1/72インチ) ■ オフジェクトのトロップを可能にする(K) マウスホペインタ(P): Windows既定値 ▼

図2.18 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(<u>B</u>)...」、「前景 色(<u>G</u>)...」で変更します。ここでは「背景色(<u>B</u>)...」について説明しますが「前景色(<u>G</u>)...」 の設定も「背景色(<u>B</u>)...」と同様に設定します。

「背景色(<u>B</u>)...」をクリックします。

フォームフ´ロバティ オブジェクト名(Q) [form(0) 表示文字列(Q) 「オブジェクトを表示する(V) 「オブジェクトを表示する(V) 「活性にする(E) 「対`の表示(位置) 「ア(ロンファイル名(E))	× 7オームサイス [*] (2): アオームサイス [*] (2): のK キャンセル キャンセル へルフ [*] (H) のき ● 縦(T) ● 横(P) スケールモート [*] (M): Point(1/72インチ) ▼ 「オンジェクトのトロッフを可能にする(K)
アイコンファイル名(E)	□ オフジェクトのトロップを可能にする(K)
「アイコンファイルを埋め込む(A) 参照(Q)…	マウスポインタ(P):
背景色(B)… 前景色(G)…	Windows既定値

図2.19 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2.20 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2.21 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。

ᡔ᠋ᡒ᠆᠘᠋᠋᠋ᡗ [®] ᡔ᠋ᡝ	×	1
	フォームサイズ(2): OK ウイントウサイズ ● 白さ(1) ● 「白さ(1) ● 「白さ(1) ● 「白さ(1) ● 「白さ(1) ● 「クイルトート*(M): ● 「クロホイート*(M): ● 「オフジシェクトのトロッフ*を可能にする(k) マウスホインタ(P): 「Windows既定値<	

図2.22 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。

form0
ファイル(E) 編集(E) フォーム(Q) 表示(V) ヘルプ(H)
[formD]
NUM //

図2.23 色の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(<u>F</u>)」から「部品パレットの起動(<u>P</u>)」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。

🕌 部品パ	レット					
7711(E)	編集(<u>E</u>)	表示⊙	カテコ⅌@	シーオプション	Z©) ∧I	·7°(<u>H</u>)
X	GØ		<u>B</u>	U 🖭 🕇	3 😺	?
	\boxtimes		AB			4
Button	Check	Combo	Label	ListBox	Radio	Scroll
	ABI		m		9-0-0 1 0-0	***
Spin	Text	Table	Graph	ListView	IreeVie	w Image
【◀【◀】▶】▶】 [GUI] [表] A P 制御]						

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。

<u>ファイル(E)</u> 編集(E) フォーム(Q) 表示(V) ヘルブ(H)
Label1 🗨
Button1
Dunom
(form0)
NUM //

図2.26 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。

form0	
ファイル(E) 編集(E) フォーム(Q) 表示(V) ヘルプ(H)	
□ □ □ A \ -> ■ ■ ■ == == ==	
Label1	
Button1	
[formD]	
Rect(74, 39, 269, 62)	NUM ///

図2.27 サイズ変更と配置変更

17/45

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(<u>G</u>)...」を選びます。

form0	
ファイル(E) 編集(E) フォーム(Q) 表示(V) ヘルプ(H)	
<u></u> ケリッド(G)	r
ケリットに配置(S)	
LaDell	
Button1	
[form0]	
クリッドのフロハティを設定 NUM //	

図2.28 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(<u>V</u>)」から「グリッドに配置(<u>S</u>)」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。

יכ	ァイル(E) 編集(E) フォーム(Q) 表示(V) ヘルプ(H)
<u>-</u>	
<u>ケリット*</u> 区 ブリット*を表示する(V) OK	Label1
ケリット*の種別 キャンセル ○ 線(L) ヘルフ°(H) ○ 点(D)	
	Button1
10 10 がリッドの色(C) Re	form0

図2.29 グリッドダイアログ 図2.30 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2.31 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。

<u>プロパティ シート ー</u>	Label1 2
Label1 Label	·
Alignment	2 - 上中央
AutoResize	0 - 高さ、幅を指定する
BackColor	
BorderColor	
BorderStyle	0 - なし
BorderWidth	1
CancelEvent	FALSE
Caption	Hello!!
Caption3D	0 - なし
Enabled	TRUE
Font	System
ForeColor	
Format	
Height	28
المحمد	00

図2.32 プロパティシート

19/45

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。

ንግሥንት እት -	Label1 🛛 📉
Labell Label	
Alignment	2 - 上中央
AutoResize	0 - 高さ、幅を指定する
BackColor	
BorderColor	
BorderStyle	0 - なし
BorderWidth	1
CancelEvent	FALSE
Caption	Hello!!
Caption3D	0 - なし
Enabled	TRUE
Font	System
ForeColor	
Format	
Height	28
l	<u></u>

図2.33 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。

form0		
ファイル(E) 編集(E)	フォーム(2) 表示(⊻)	∧μフ°(<u>H</u>)
<u>8</u> 60 0		
	Hello!!	
	Button1	
	───── ┩┤	
[form0]		
,		

図2.34 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。

プロパティシートー	Label1 🛛 🔟
Label1 Labe	
Alignment	2 - 上中央
AutoResize	0 - 高さ、幅を指定する
BackColor	
BorderColor	
BorderStyle	0 - なし
BorderWidth	1
CancelEvent	FALSE
Caption	Hello!!
Caption3D	0-なし
Enabled	TRUE
Font	System
ForeColor	
Format	
Height	28
	90 ×

図2.35 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。)

フォント	×
フォント名(E): MS ゴシック な MS UI Gothic か MS ゴシック な MS 明朝 な OCRB O Palatino Linotype Roman Script	スタイル(½): サイズ(S): 太字 18 標準 斜体 14 キャンセル 大字 18 20 22 24 マ 26 ▼
文字飾り 「取り消し線(<u>K</u>) 「下線(<u>U</u>)	サンブル Aaあぁアァ亜宇 書(本語 マ

図2.36 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2.37 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2.38 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。

SAMPLE1.gob - COBOL GUI for COBOL2002	
ファイル(E) 手続き編集(<u>P</u>) テスト表示(<u>T</u>) 表示(<u>V</u>)	_ ツール (L) ヘルフ [°] (H)
□ ➡ 単通手続きの編集(C)	
Button1 イベント手続きの編集でん。 Other メンットとフロハディの表示(V)。 Ctrl+M	
バーチ続きの編集(M)	
form 初期処理(E) ▶	
ファイル(終了処理(L) ・	(H)
■ [] 環	
<u></u> 名称チェック(<u>H</u>)	
Hello!!	
┠┼┼┼┼┼┼╡ _{┏╴╧╴} ┠┼┼	
● □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	

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。

選択した部品で使用できるイベントが「イベント(⊻)」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(<u>V</u>)」の「Click」を選択して、「編集(<u>E</u>)」をクリックします。

イベント手続き編集		×
わジュウト名称(<u>O</u>)	<u>1^୬⊦</u> ₩	12°-(C)
Button1		買用を行う
メソッド(<u>M</u>)	DblClick GotFocus KeyDown	[[編集(E)]]
7ใ⊔∧°ティ(<u>P)</u> _Value		<u>- 閉じる</u>

図2.40 イベント手続き編集ダイアログと説明

23/45

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2.41 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。

24/45

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2.42 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'FFFFF'(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

くプロ	コパティの設定>					
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'000FF':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。

図2.43 反転ボタン Click イベントのコーディング全体

26/45

2.8 GUI構築パックの終了

COBOL エディタの「ファイル(<u>F</u>)」から「エディタの終了(<u>X</u>)」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(<u>F</u>)」から「GUI エディタの終了(<u>X</u>)」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

27/45

スクール COBOL2002 GUI 構築パック

「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2.47 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。

・手続きを間違っ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(<u>B</u>)」から「ビルド(<u>B</u>)」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。

☞ 開発マネージゃ for COBOL2002 - GUI1.hmf				
プロジェクトマスタ(<u>M</u>) ファイル(<u>E</u>) プロジェクト(<u>P</u>)	ビルド(B) ツール(T)	ウィントንን₩)	∿ルフ°(<u>H</u>)	
	ビルト*(B) クイックビルト*(Q) コンパイル(G)		Ctrl+B	
Gutt	<u>ሃ</u> ピʹルト [≈] (<u>R</u>)			
GUII.exe	すべてのプロジェクト(対する操作但) •	
□ ソースファイル ■ SAMPLE1.gcb ● 依存ファイル	ビルドの中止(S) 依存関係更新(A)			
	デンベッガ(<u>D</u>)			
<u> </u>	実行(<u>E</u>) 実行支援(<u>L</u>)		Ctrl+E	
必要なファイルのみビルトを行う				

図2.48 ビルド選択

28/45

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。

民テストデバッグ一覧			<u>- 0 ×</u>
名称	状態	属性	
📮 GUN.exe	 ! !	実行可能ファイル	
i i i form0	1	GUI定義	
📙 🖳 Button1	1	GU 部 品	
Click	, ,	イベント手続き	
			•
1			

図2.49 テストデバッガー覧画面

図2.50 実行部の表示をクリック

29/45

「実行部の表示」は次のようになります。

SAMPLE1	
0000785	LINKAGE SECTION.
0000787 0000788 0000789 0000789	PROCEDURE DIVISION . * 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。 SET 前景色 TO 'ForeColor' WITH Label1.
0000791 0000792 0000793 0000793	IF 前景色 = 0 THEN SET 'ForeColor' WITH Label1 TO H'FFFFFF' ELSE SET 'ForeColor' WITH Label1 TO H'000000'
0000795	END-IF.

図2.51 実行部の表示

入り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、「入り口中断点の設定(<u>S</u>)」をクリックします。

🔮 テ ス ト デバッガ for COBOL2002 - GUIT.exe	
ラストデバッかの刺繍(E) 爆集/表示(E) ブログラムの刺繍(C) 中新(E) デー)の操作(D) ソール(D)	917F7W 4F77W
· · · · · · · · · · · · · · · · · · ·	10 TO 1 + 1 + 1 + 1 + 1 11 11 11 11
	7/73.51
SAMPLE1	
0000785 LINKAGE SECTION.	
0000787 0000787 0000788 PROCEDURE DIVISION . 0000789 * ハンが起きた場合に実行される手続き 0000790 SET 前景色 TO 'ForeColor' WITH	E 22hF/N'y)/一覧 E 1□ × A称 状態 居住 日 GUID.exe すorm0 GUID語 GUID語 GUIDE
0000791 IF MJ392E2 = 0 THEN 0000792 SET 'ForeColor' WITH Label1 T 0000793 ELSE 0000794 SET 'ForeColor' WITH Label1 T 0000795 END-IF.	0 支稿部の表示(E) 実行部の表示(M) 入口中断点の設定(E) 出口中断点の設定(E)
KUUUUUU41-1 GUIフロクラムです。オフション設定タイアロクホック	マスの指定規則で、COBOL85の規則と
入口に中断点を設定する	NUM //

図2.52 入口中断点の設定

30/45

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2.53 データ値の表示

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

データ値の表示		×
デ━タ名:	Label1 マ OLE7泊ハディの表示(P)	OK キャンセル

図2.54 データ値の表示ダイアログ

31/45

スクール COBOL2002 GUI 構築パック

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。

11 E19			
名称	Caption		-
룊	文字列型:8		
値	Hello!!		_
名_称	Caption3D		-1
1	823000 · 71		

図2.55 プロパティの表示結果

ワンポイントレッスン

部品名、バリアントデータ項目、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2.56 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。

32/45

2.11 実行

「開発マネージャ」の「ビルド(<u>B</u>)」から「実行(<u>E</u>)...」を選択します。

延開発マネージャ for COBOL2002 - GUI1.hmf			
プロジェクトマスタ(M) ファイル(E) プロジェクト(P)	ビルド(<u>B)</u> ツール(I) ウイントウ(W)	· ^//フ°(<u>H</u>)	
	ビルト*(B) クイックビルト*(Q)	Ctrl+B	. ?
GUI1.hmf	コンハイル(<u>C</u>) ルビット(CD)		
🔁 GUII	7C7015 (<u>E</u> V		
🖻 🗐 SAMPLE1 – GUII.exe	すべてのプロジェクトに対する操作	(<u>P</u>) •	
□	ビルドの中止(S) 依存関係更新(A)		
	デバッガ(<u>D</u>)		
<u>µ</u>	実行(E)	Otrl+E	
	実行支援([]		
実行可能ファイルを起動する			NUM ///

図2.57 「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。

実行	1	×
実行可能ファイル名(<u>N</u>):	GUI¥GUI1¥sample1¥GUI1.exe ▼ 参照(B)]
作業フォルダ(<u>F</u>):	✓ 参照(B)]
引数(<u>A</u>):		
	ОК 4 ++>±#	

図2.58 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)	整数型	2バイト
	USAGE IS COMP		(-32768 ~ 32767)
Long	PIC S9(9)	長整数型	4バイト
	USAGE IS COMP		(-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
Curency	PIC X(領域長)	通貨型
	領域長はプロパティの値に依存します。	
String	PIC X(領域長)	文字列型
	領域長はプロパティの値に依存します。	
Date	PIC X(領域長)	日付型
	領域長はプロパティの値に依存します。	
Boolean	PIC 1(1) USAGE BIT	ブール型
Object	USAGE IS OBJECT REFFERENCE OLE	オブジェクト型
Variant	USAGE IS VARIANT	バリアント型
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(<u>H</u>)」から「トピックスの検索(<u>H</u>)」を選択します。

form0	
ファイル(E) 編集(E) フォーム(Q) 表示(V)	^ルフ°(<u>H</u>)
	トピックの検索(H) F1 🖉 💻
	ハーション情報(A)
Helloll	
+++++++++++++++++++++++++++++++++++++++	
<u>╶</u> ┤┊┤┊╎┊╎┊ <mark>╶╶╶╸╸╸╸</mark>	
form	
kt.hote	
(1.770)19(米)	

図3.1 フォームエディタの「トピックスの検索(<u>H</u>)」を選択

トピックの検索: APPGALLERY (開発環境) ヘルプ	?
──▶ 目次 キーワード テキスト検索	
トピック をクリックし、 次に [表示] をクリックしてください。または、 [キーワード] など別のタ クしてください。	げをクリッ
 2 マニュアルの紹介 ◆ アプリケーションの基礎 [解説] 	-
 アプリケーションの開発 [解説] アプリケーションの開発に利用する部品と機能 [解説] 	
 スタッフの利用[通電算] 第二、中国祭 [通空算] 	
▶ アプリケーション開発ツール[操作]	
● スクリフト[文法] ● 式(テーブル部品)[文法]	
 シリファレンス スタッフの関発「通知員) 	
▼ ブロクラミンクトビック ◆ 用語解説	-
	キャンセル

35/45

[「]目次」タグをクリックします。

「リファレンス」をダブルクリックします。

トピックの検索: APPGALLERY (開発環境) ヘルプ	? ×
目次 キーワード テキスト検索	
	1
本。をクリックし、次に「開く」 をクリックしてください。または、「キーワード」 など別のタブをクリックして	
	-
2 マニュアルの紹介	
📎 アプリケーションの開発 [解説]	
🔷 🎨 アプリケーションの開発に利用する部品と機能 [解説]	
🔷 🔖 スタッフの利用 [解説]	
🔷 📀 部品の開発 [解説]	
🔷 📎 アプリケーション開発ツール[操作]	
🔷 🔷 スクリプト[文法]	
🔷 🔷 式(テーブル部品)[文法]	
געזדע 🔶 🚽 אראדע אדע איז א אראדע אדע איז א געזידע א איז א געזידע א איז א געזידע געזידעגעזידע געזידע געזידעגעזידע געזידע געזידעגעזידעגעזידעגעזידעגעזידעגעזיגעזידעגעזיגעזעזיגעזיגעזעזיגעזעזעזיגעזעגעזעזיגעזעזעגעזיגעזעגעזיגעגעזעזעגעזעג	
- 🔷 🔷 スタッフの開発[解説]]
📀 旧バージョンからの移行 [解論]	
📀 プログラミングトピック	
- 用語解説	
開((<u>Q</u>) 印刷(<u>P</u>) キャンセ	ll I

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。

PE990/Rest for decent will be sky of the	즤
目次 キーワード テキスト検索	
,	
本 をクリックし、 次に 「購く」 をクリックしてください。または、 [キーワード] など別のタブをクリックして ください。	
U1 U1 U	
🔷 🔷 スクリプト関数	
🔹 🔷 ウィンドウ/メニュー作成で使用する部品	
🔹 🔷 🕪 GUI作成で使用する部品 🔫 🛶 👘	
● データベースアクセス部品	
🔹 🔷 🔷 ネイティブデータベースアクセス部品	
🔹 🔷 テーブル部品	
▶ アプリケーション制御部品	
シーク処理部品	
▼ コード·ID→覧	
↓ V エラーコード→覧	

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。

トピックの検索: APPGALLERY (開発環境) ヘルプ	<u>?</u> ×
目次 キーワード テキスト検索	
トピック をクリックし、 次に [表示] をクリックしてください。または、 [キーワード] など別のタブをクリ クしてください。	ツ
U177 U27 U27 U2 U	.
🔷 🔍 シファレンスの見方	
🔹 🔷 、 スクリプト関数	
🔹 🔷 🔷 ウィンドウ/メニュー作成で使用する部品	
🚺 🚺 GUI作成で使用する部品	
2 オブジェクト階層	
👔 👔 オブジェクト一覧 🗲	
2 วียไว้ระ	
אַעא 🗿 אין	
2 エラーコード一覧	
▼ データベースアクセス部品	
◆ ネイティブデータベースアクセス部品	
↓ ● テーブル部品	<u> </u>
	216

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。

	▲PREALLERY 開発環境)への オ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	
	フドルク 編集の しめの オリルバリ ヘルカロ	ADS AUGUSSS
	目次(2) 4-ワーF(0) 二(2) 印刷(2) 月間(3) ≤< ≥ 高橋(1) 略称(2) M(7(3))	
	GUI作成で使用する部品-オブジェクト一覧	
	Based 3 z Z = 2h	1
	Batten/12/274/2E%1	
	RheshButton 7: 72 a 21: W1	
	SelvenHedges=Ld22cale	
95 - C	Gelwontheater2:323.425	
	GomboRox过过2421年1	
	Dielog21252x2E	
	Fontdutstauth	
	In set Listオプジェクト※3	
	Late (社)ジェクト用1 第2	
Labelオナ	∮ジェクトGUI作成で使用する部品)	
70/17		
-CON		
1		
	Nodescal./25/mL/	
	Nede 21 22 2 2 2 1	
	Badie Brist JELA ST W1	
	Constitue 3 - 41 - 26	-

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。

◆ APPGALLERY (開発環 ファイル(F) 編集(F) し は	(境) ヘルプズ おり(M) オブション(D) ヘルプ(H)
 57 m @ mmx @ 00 目次(©) キーワード(0) 終了(※) 	_ 戻る(B) 印刷(P) 用語(G) << ≥ 商標(T) 略称(A)
Labelオブジェクト	-ブロバティー覧
<u>Alignment</u>	ラベルに文字列を表示するときの位置を指定・取得します。
<u>AutoResize</u>	ラベルに表示するビットマップ又は文字列のサイズを,調整するかどうかを指定・取得 します。
BackColor	ラベルの背景色を指定・取得します。
BorderColor	ラベルの境界線の色を指定・取得します。
BorderStyle	ラベルの境界線の種類を指定・取得します。
BorderWidth	ラベルの境界線の幅を指定・取得します。
<u>CancelEvent</u>	実行時にオブジェクトのGUI操作やオートメーションで発生するすべてのイベントを抑止 するかどうかを指定・取得します。
Caption	ラベルに表示する文字列を指定・取得します。
Caption3D	ラベルに表示する文字列を3-D(立体)とするかどうかを指定・取得します。
Enabled	ユーザの操作によって発生したイベントを有効とするかどうかを指定・取得します。
Font	Fontオブジェクトを指定・取得します。
ForeColor	ラベルの前景色を指定・取得します。
Format	ラベルに文字列を表示するときの書式を指定・取得します。
FormManager	FormManagerオブジェクトを取得します。
<u>Height</u>	ラベルの高さを指定・取得します。
hWnd	Windowsのオブジェクトを識別したり、アクセスしたりするためのハンドルを取得します。

図3.7 「ForeColor」をクリック

ForeColor プロパティのデータ型(Long)情報。

ファイル(F) 編集(E) しおり(M) オブション(D) ヘルブ(H) 目次(C) キーワード(D) 戻る(B) ED刷(P) 用語(G) 公 ▷ 商標(T) 略称(A) 終了(A) ForeColorプロパティ 設定・参照 ガタ Board, Button, CheckButton, ComboBox, Label, ListBox, ListView, RadioBox, SpinButton, TextBox 健能 オブジェクトの前景色を指定・取得します。 標準値は、フォームのAmbientプロパティのForeColorプロパティに設定されている色です。ただし、リストビューの場合の標準値は、コンテナの前景色になります。 解説 ■前泉色の指定方法 前景色は、16進数の32ビットの値は、最下位パイトから赤、緑、青の順に輝度を表します。 輝度は、16進数00~255)の範囲で指定します。 色の指定例を次に示します。 8h00000008 黒 &h00000008 黒 &h0000FF008 緑
目次(①) キーワード(①) 戻る(B) 印刷(P) 用語(Q) 公 >> 商標(D) 略称(A) 終了(20) ForeColorプロパティ 設定・参照
 ForeColorプロパティ 設定・参照 Board, Button, CheckButton, ComboBox, Label, ListBox, ListView, RadioBox, SpinButton, TextBox # Board, Button, CheckButton, ComboBox, Label, ListBox, ListView, RadioBox, SpinButton, TextBox # <i>Tiジェク</i>トの前景色を指定・取得します。 欄準値は、フォームのAmbientプロパティのForeColorプロパティに設定されている色です。ただし、リス トビューの場合の標準値は、コンテナの前景色になります。 # 前景色の指定方法 前景色は、16進数の32ビットの値又はスクリプト関数の<u>RGB関数</u>で指定します。 ● を表ます16進数の32ビットの値は、最下位パイトから赤、緑、香の順に輝度を表します。 色の指定例を次に示します。 ● の指定例を次に示します。 & h0000FFF008 & & & & & & &
ForeColorプロパティ 設定・参照 Board, Button, CheckButton, ComboBox, Label, ListBox, ListView, RadioBox, SpinButton, TextBox な 構築 オブジェクトの前景色を指定・取得します。 標準値は、フォームのAmbientプロパティのForeColorプロパティに設定されている色です。ただし、リス トビューの場合の標準値は、コンテナの前景色になります。 解説 ■前景色の指定方法 前景色は、16進数の32ビットの値又はスクリプト関数の <u>RGB関数</u> で指定します。 色を表す16進数の32ビットの値は、最下位バイトから赤、緑、香の順に輝度を表します。輝度は、 16進数00~FF(10進数の0~255)の範囲で指定します。 色の指定例を次に示します。 &h0000FFFFFF& 白 &h000FF000& 春 &h000FF000& 緑
対象 Board, Button, CheckButton, ComboBox, Label, ListBox, ListView, RadioBox, SpinButton, TextBox てオブジェクトの前景色を指定・取得します。 標準値は、フォームのAmbientプロパティのForeColorプロパティに設定されている色です。ただし、リストビューの場合の標準値は、コンテナの前景色になります。 解説 ■前景色の指定方法 前景色は、16進数の32ビットの値又はスクリプト関数のRGB関数で指定します。 色を表す16進数の32ビットの値又はスクリプト関数のRGB関数で指定します。 色の指定例を次に示します。 & h000000008 黒 & h000FFFFF系 白 & h000FF0008 春 & h000FF008 緑
73% Board, Button, CheckButton, ComboBox, Label, ListBox, ListView, RadioBox, SpinButton, TextBox 桜を オブジェクトの前景色を指定・取得します。 標準値は、フォームのAmbientプロパティのForeColorプロパティに設定されている色です。ただし、リストビューの場合の標準値は、コンテナの前景色になります。 解説 ■前景色の指定方法 前景色は、16進数の32ビットの値又はスクリプト関数のRGB関数で指定します。 在を表す16進数の32ビットの値は、最下位バイトから赤、緑、春の順に輝度を表します。輝度は、16進数00~FF(10進数の0~255)の範囲で指定します。 をの指定例を次に示します。 &h0000000& 黒 &h00FFFFFF& 白 &h00FFF000& 春 &h000FF00& 緑
Board, Button, CheckButton, ComboBox, Label, ListBox, ListView, RadioBox, SpinButton, TextBox 標準値は, フォームのAmbientプロパティのForeColorプロパティに設定されている色です。ただし, リス トビューの場合の標準値は, コンテナの前景色になります。 解説 ■前景色の指定方法 前景色は, 16進数の32ビットの値又はスクリプト関数の <u>RGB関数</u> で指定します。 色を表す16進数の32ビットの値は, 最下位バイトから赤, 緑, 春の順に輝度を表します。 輝度は, 16進数00~FF(10進数の0~255)の範囲で指定します。 色の指定例を次に示します。 &h000FFFFFF& 自 &h00FFF000& 春 &h00FF000& 緑
 株配 オブジェクトの前景色を指定・取得します。 標準値は、フォームのAmbientプロパティのForeColorプロパティに設定されている色です。ただし、リストビューの場合の標準値は、コンテナの前景色になります。 解説 ■前景色の指定方法 前景色は、16進数の32ビットの値又はスクリプト関数のRGB関数で指定します。 色を表す16進数の32ビットの値は、最下位バイトから赤、緑、香の順に輝度を表します。輝度は、16進数00~FF(10進数の0~255)の範囲で指定します。 色の指定例を次に示します。 &h0000000& 黒 &h00FFFFF系 &h00FF000& 春 &h000FF00& 緑
ボラシェウトの前条色を指定・4kHcます。 標準値は、フォームのAmbientプロパティのForeColorプロパティに設定されている色です。ただし、リストビューの場合の標準値は、コンテナの前条色になります。 ドビューの場合の標準値は、コンテナの前条色になります。 「 新各色の指定方法 前条色は、16進数の32ビットの値又はスクリプト関数のRGB関数で指定します。 色を表す16進数の32ビットの値は、最下位バイトから赤、緑、香の順に輝度を表します。輝度は、 16進数00~FF(10進数の0~255)の範囲で指定します。 色の指定例を次に示します。 &h0000000& 黒 &h00FFFFF系 白 &h00FF000& 春 &h00FF008 緑
Kimit (1) A = A (0) Ambient (2) A f (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
FELECOUSEON 編準値は、コンチナの前条色になります。 解説 ■前条色の指定方法 前条色は、16進数の32ビットの値又はスクリブト関数の <u>RGB関数</u> で指定します。 色を表す16進数の32ビットの値は、最下位バイトから赤、緑、香の順に輝度を表します。輝度は、 16進数00~FF(10進数の0~255)の範囲で指定します。 色の指定例を次に示します。 &h0000000& 黒 &h000FFFFF系 白 &h00FFF00& 禄
★認 ■前景色の指定方法 前景色は、16進数の32ビットの値又はスクリプト関数の <u>RGB関数</u> で指定します。 色を表す16進数の32ビットの値は、最下位バイトから赤、緑、香の順に輝度を表します。 輝度は、 16進数00~FF(10進数の0~255)の範囲で指定します。 色の指定例を次に示します。 &h0000000& 黒 &h00FFFFFF& 白 &h00FF000& 春 &h00FF008 緑
 前景色は、16進数の32ビットの値又はスクリブト関数の<u>RGB関数</u>で指定します。 色を表す16進数の32ビットの値は、最下位バイトから赤、緑、香の順に輝度を表します。 渡度は、 16進数00~FF(10進数の0~255)の範囲で指定します。 色の指定例を次に示します。
 前景白は,10進級の32ビットの値とは、スクリット開設の<u>NGOLAD 国政</u>で指定します。 色を表す16進数の32ビットの値は,最下位バイトから赤,緑,春の順に輝度を表します。 16進数00~FF(10進数の0~255)の範囲で指定します。 色の指定例を次に示します。 &h0000000& 黒 &h00FFFFFF& 白 &h00FF0000& 春 &h000FF008 緑
16進数00~FF(10進数の0~255)の範囲で指定します。 色の指定例を次に示します。 &h0000000& 黒 &h00FFFFF& 白 &h00FF0000& 春 &h000FF00& 緑
色の指定例を次に示します。 &h00000000& 黒 &h00FFFFF& 白 &h00FF0000& 春 &h000FF00& 緑
&h0000000& 黒 &h00FFFFF& 白 &h00FF0000& 春 &h000FF00& 緑
&h00FFFFF& 白 &h00FF0000& 春 &h0000FF00& 緑
&h00FF0000& 春 &h0000FF00& 緑
&h0000FF00& 緑
&h000000FF& 赤
&h00FFFF00& シアン
&h0000FFFF& 黄
&h00FF00FF& マゼンタ
■リストビューの前景色の標準値について
リストビューの場合の標準値は,コンテナの前景色です。ただし,リストビューの前景色を変更し
た場合、コンテナの前景色を変更してもリストビューの前景色には反映されなくなります。また、
前景色の指定はColumnHeaderオブジェクト以外の領域に有効です。
データ型
エラーコード
・Button, CheckButton, ComboBox, Label, ListBox, RadioBox, SpinButton, TextBoxの場合

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
- Q1: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。
- A1: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。

- Q2:「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、「GUI エディタ」が 起動しません。
- A2: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。
- Q3:「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。
- Q4: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。
- A4: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ませんか? 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。
- Q5: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。
- A5: 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。

40/45

3.3.2 GUI アプリケーション実行時の注意事項

- Q1:「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。
- A1: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。
- Q2:「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。
- Q3: 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。
- Q4: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。
- A4: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。
- Q5: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SET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」の言語仕様にない機能を 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。

43/45

バリアントデータ項目

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。

44/45

《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。