第16章 Visual Basic によるプログラミング

前章ではVisual C++を使ってxPC Targetのプログラムを作りました. プログラム作法を理解するためにコンソール・アプリケーションで作ってみましたが,ユーザの立場からいうと使い勝手の良いものではありません.

第16章では、WindowsのGUIを利用するためにVisual Basicを使ってプログラムを作ります.

Visual C++の際に述べましたが, xPC Targetが提供する APIは Visual Basic 6.0対応であり, .NET フレームワークには対応していません.

しかし Visual Studio .NET 2003, あるいは Visual Studio 2005は, 直接 COM を参照することができ るので, Visual Basic 6とほとんど同じ手順でプログラムを作ることができます. もちろんプログラミ ングの作法の細部は多少異なります.

ここでは、まず、Visual Basic 6を使ってプログラムを作成します.

Visual Basic 6のプログラムをVisual Studio .NET 2003の自動変換エンジンを使ってVisual Studio .NET 2003のプログラムに変換します.

自動変換のプログラムを参考にして、.NET対応のプログラムを新規に作成します.

以下において Visual Basic を VB と略称します.

■ 16.1 準備作業

VB 6を使ってxPC Targetのプログラミングを行うに当たり、いくつかの準備作業が必要です.それらの準備作業について述べます.

xPC Targetのプログラムを開発するために必要な APIは,

xpcapi.dll

です.

このファイルに必要なすべてのクラスとメソッドが格納されています.

このAPIは、Visual C++のプログラムを作る際に使用しました.

VB6はCOMのラッパーを介してxpcapi.dllにアクセスします.

このために, xpcapi.dllに加えてCOMのラッパーとタイプ・ライブラリ,





画面16.1 コマンド プロンプト

xpcapiCOM.dll

xpcapi.tlb

が必要になります.

私のシステムにおいてこれらのファイルは,

C:¥Program Files¥MATLAB704¥toolbox¥rtw¥targets¥api

ディレクトリに格納されていました.

xpcapiCOM.dll はシステム・ファイルなので,

c:¥WINDOWS¥system32

フォルダヘコピーします.

Visual C++によってプログラムを作った際に xpcapi.dllをコピーしたので、結局、

xpcapi.dll

xpcapiCOM.dll

の二つのファイルをc:¥WINDOWS¥system32へコピーしたことになります.

Windowsの規約によってCOMはレジストリに登録する必要があります.

xpcapiCOM.dllをレジストリへ登録します.

画面16.1に示すように, [コマンド プロンプト]を立ち上げて,

xpcapiCOM.dll

xpcapi.tlb

ファイルが格納されているフォルダへ移動します.

[コマンド プロンプト]のコマンドラインから,

>> regsvr32 xpcapiCOM.dll

と入力します.

画面16.2に示すように、レジストリの登録は成功します.



		2084/12/18 2084/12/18 2085/02/11	08:15 04:33 17:40	182,400 12,054 61,440	хакаані хакаані хакаані	di i Regioniti		
		2085/02/11 2084/03/01 2084/01/22	17:40 22:80 13:54	18,016 3,736 20,704	xpcapi xpcapi xpcini	۹	upageCOMult & DiPagiateCarve 12823U	1.6.
		C-IReserve	71日のフ 11日のデ 11日のデ	マイル マレクトリー2 1949-0010-0042	(T. 885 .	ant clouch	Configuration (COM 411	
画面16.2	登録が成功	CHP regram P (Included Department of Children P (Andre State Children Chi						

これで準備作業は完了しました.

■ 16.2 VB 6のプログラミング

VB6のプログラムを作成します.

読者はVBの操作を理解していると仮定します.ここでは複雑なプログラムを作るわけではないので、 VBの専門的な知識は必要ありません.

まず, VB6を立ち上げます.

画面16.3に示した[新しいプロジェクト]のダイアログが開きます.

[新規作成]のタブをクリック選択して(画面16.3では選択済),中央のペインにおいて[標準EXE]を 選択して(画面16.3では選択済), [開く]ボタンをクリックします.

画面16.4に示すVB6の画面が開きます.

メニューから[ファイル]→[名前をつけてプロジェクトの保存]とクリックします.

画面16.5に示すように[名前をつけてファイルの保存]のダイアログが開きます.

プロジェクトを格納するフォルダを選択し、フォームの名前を書き込んで[保存]ボタンをクリック します.

画面16.6に示すように[名前をつけてプロジェクトの保存]のダイアログが開きます.

プロジェクトを格納するフォルダを選択し、プロジェクトの名前を書き込んで[保存]ボタンをク リックします.

私の場合はプロジェクト vb6.vbp は,

C:¥work¥Chapt16¥vb6

フォルダに格納しました.

画面16.7に示すように[ソース・コード管理]のダイアログが開きます.

私の場合ソース・コードの管理は不要なので,[いいえ]ボタンをクリックしました.皆さんの事情 に応じて適当なボタンを選択してください.

フォームが選択されている間にフォームのプロパティ [Caption] を,

Form1 → vb6

と変更します.

画面16.8に示すようにフォーム上へ6個のボタンを配置します.







9- X 3- FEM (X) 		表16.1 各ボタン [Caption]	の[オブジェクト名]と		
		オブジェクト名	Caption		
		Command1	ターゲットへ接続		
画面16.7 [ソース・コート管理]のダイアログ		Command2	ターゲットの切断		
		Command3	ダウンロード		
	9-931-488 9-931-0488	Command4	アンロード		
		Command5	スタート		
	97,A-F 7,A-F	Command6	ストップ		
King and a set of the set o	<u></u> 画面16.8 フォーム	この元 リック 込む手	このボタンをク リックするとプ ログラムを書き 込む段階に入る		

画面16.9 [参照設定]のダイアログ

各ボタンの[オブジェクト名]と[Caption]は,表16.1とします.

コードを書く準備をします.

VB 6のメニューから[プロジェクト]→[参照設定]とクリックします.

画面16.9に示したように[参照設定]のダイアログが開きます.

[参照可能なライブラリファイル]を最下段へスクロールして,

xPC Target API COM Type Library

にチェック・マークを入れて[OK] ボタンをクリックします (画面16.9においてチェック・マークが 入っているが,最初は空欄になっている).

これでプロジェクトにCOMのタイプ・ライブラリが登録されました.

タイプ・ライブラリ登録の操作はプロジェクトごとに必要です.

タイプ・ライブラリが登録されると、プログラムの書き込む際にインテリ・センスが働くのでキー入 力が楽になり、同時にエラーも少なくなります.

それではプログラムの書き込みを始めます.

VB 6の画面 (**画面16.4**)の右側上部の [プロジェクト]のウインドウのツール・バーの [コードの表示] ボタン (**画面16.10**)をクリックします.

画面16.11 に示すように[Project-Form1 (コード)]の編集ウインドウが開きます. 上部左側のリストボックスにおいて,



画面16.10 プロジェクト 📗



画面16.12 オブジェクトブラウザ

画面16.13 クラスとメンバー関数

(General)

が選択されていて,かつ上部右のリストボックスにおいて,

(Declarations)

が選択されていることを確認します.

編集ウインドウに,画面16.11に示すように使用するグローバル変数の名前を書き込みます.

念のために、編集ウインドウに書き込んだ宣言文をリスト16.1に示します.

これらの文はC言語でいえば型の宣言に当たります.

Protocol, Target, Scopesというオブジェクトを宣言しました.

XPCProtocol, xPCTarget, xPCScopesなどのオブジェクト名は登録したCOMに記載されています. 登録されているオブジェクト名をチェックします.

メニューから[表示]→[オブジェクト ブラウザ]とクリックします.

画面16.12の[オブジェクト ブラウザ]のダイアログが開きます.

画面にも示したように 〈すべてのライブラリ〉をクリックして、 プルダウン・リストの最下行の、

XPCAPICOMLib

を選択します.

画面16.13に示すようにライブラリにおいて定義されているクラスとメンバ関数が表示されます. リスト16.1において使用したクラスがタイプ・ライブラリに記載されていることが確認できます.



プログラムを進めます.

編集ウインドウ(**画面16.11**)の上部左のリスト・ボックスにおいて[Form]を選択し、その右側において[Load]を選択して、**画面16.14**に示すようにプログラムを書き込みます.

プログラムをリスト16.2に示します.

ここではフォームがロードされた時点(プログラム実行時の最初の時点)で実際のオブジェクトを生成して、それらを初期化します.

Protocolオブジェクトはターゲットへ接続しないと有効にならないので、この時点でターゲット2へ 接続して、その結果をTargetとScopesの初期化に使用します.

したがってこのプログラムを実行する直前に、ターゲットの電源をONにし待機状態にする必要があります.

続いて、ボタンがクリックされたときに実行するプログラムを書き込みます.

プログラムを書き込んだ画面を画面16.15に示します.

ボタンのクリックの処理も含めて、全体のプログラムをリスト16.3に示します.

