#develop (SharpDevelop) で.NET

システム開発部 坂内 敦

2005年3月25日

目次

T	#develop (ShapDevelop) とは	1
2	#develop のインストール要件	2
3	#develop のインストール 3.1 ダウンロード	2 3 3 3
	3.4.1 UI Language の設定	3
	3.4.2 エディタのフォントとファイルエンコーディングの指定	5
4	Hello World (コンソール版)	5
5	Hello World (ウィンドウ版)	6
6	ユーザーコントロールの作成と追加	8
7	最後に	11

1 #develop (ShapDevelop) とは

Microsoft .NET プラットフォームで , C# と VB.NET の開発が行えるフリーな IDE 1 です .

(今のところ)デバッガは統合されていません.また,ASP.NET の開発もできません..NET の "Windows Form"(Windows 用の GUI アプリケーション)や,コンソールプログラム,クラスライブラリなどが開発できます.

VisualStudio .NET 2003 のプロジェクトファイルをインポートしたりエクスポートすることができます .

¹統合開発環境

本家: http://www.icsharpcode.net/OpenSource/SD/

日本のサイト: http://sharpdevelop-jp.sourceforge.jp/

2 #develop のインストール要件

#develop には,コンパイラ本体は含まれません.Microsoft が公開している .NET $1.1~\mathrm{SDK}$ をインストールする必要があります..NET $1.1~\mathrm{SDK}$ を利用するには,.NET フレームワークがインストールされている必要があります.

従って Windows 2000, XP のユーザーは以下の順にインストールする必要があるでしょう. 2

1. .NET Framework Version 1.1 再頒布可能パッケージ

http://www.microsoft.com/downloads/details.aspx?displaylang=ja&FamilyID=262D25E3-F589-4842-8157-034D1E7CF3A3

2. .NET Framework Version 1.1 日本語 Language Pack

http://www.microsoft.com/downloads/details.aspx?familyid=04DBAF2E-61ED-43F4-8D2A-CCB2BAB7B8EB&displaylang=ja

3. .NET Framework 1.1 Service Pack 1

http://www.microsoft.com/downloads/details.aspx?FamilyID = a8f5654f-088e-40b2-bbdb-a83353618b38&displanted by the aspect of the control of

4. .NET Framework 1.1 SP1 HotFix(MS05-004)

 $\label{lem:http://www.microsoft.com/downloads/details.aspx?FamilyId=8EC6FB8A-29EB-49CF-9DBC-1A0DC2273FF9\&displaylang=ja$

5. .NET Framework SDK Version 1.1

 $\label{lem:http://www.microsoft.com/downloads/details.aspx?familyid=9B3A2CA6-3647-4070-9F41-A333C6B9181D\&displaylang=ja$

3 #develop のインストール

3.1 ダウンロード

SorceForge の SharpDevelop-jp プロジェクトのダウンロードページ

http://sourceforge.jp/projects/sharpdevelop-jp/files/

から , SharpDevelop-X.X.X.XXXX_Setup.exe をダウンロードします . X.XXXX はバージョン番号で , 現時点での最新版は 1.0.3.1768 です .

²最新のセキュリティ情報を確認してください

3.2 セットアッププログラムの実行

ダウンロードしたセットアッププログラムを実行します.Wizard の指示に従って,ライセンスの承認とインストールディレクトリを選択すると,自動でインストールが完了するはずです.

3.3 コード補完データベースの作成

初回起動時に,コード補完データベースの作成ダイアログが表示されます.(図1参照)通常はダイアログの指示に従って(全てデフォルト選択で可)データベースを作成しておくほうがよいでしょう.

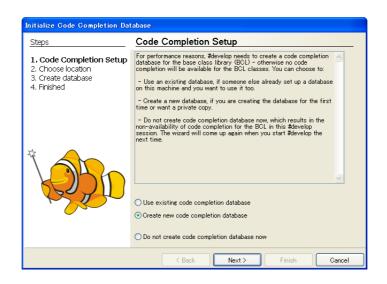


図 1: コード補完 DB の作成

3.4 日本語表示のための設定

3.4.1 UI Language の設定

メニューから Tools/Options... を選択すると, UI Language 設定ダイアログが開きます(図2参照)

Japanese を選択して OK するとメニューなどの表示が日本語に変わります.



図 2: UI Langueage の設定

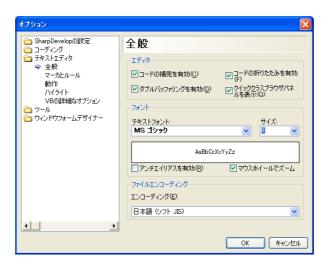


図 3: オプション (テキストエディタ)の設定

3.4.2 エディタのフォントとファイルエンコーディングの指定

ツール (T)/オプション (P)... で , テキストエディタ 全般を選択しテキストフォントを $^{\rm mm}$ ゴシック" $^{\rm mm}$ $^{$

また,現状では,エンコーディングを"日本語 (シフト JIS)"に変更する方が無難でしょう. (図 3 参照)

4 Hello World (コンソール版)

では,アプリケーションを作ってみましょう.まずは Hello, World です.

1. Start Page の "New Combine" ボタンをクリックします. (図 4 参照) ³ #develop の "Combine" は,一つ以上の Project のグループで, VisualStudio で言うところの「ソリューション」にあたります.

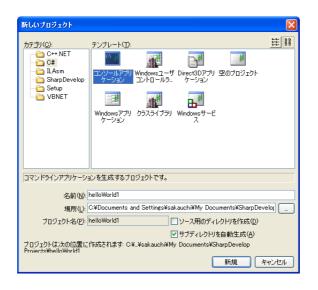


図 4: コンソールアプリケーションの新規作成

2. コンソールアプリケーションの場合, 作成されたメインソースが既に

Console.WriteLine("Hello World!");

を含んでいます. F5 を押せば4 コンパイルされ実行されます.

³または,メニューの「ファイル/新規作成/コンバイン...」を選択

⁴ツールボックスの緑の矢印をクリックしたり、デバッグメニューから「実行」を選択しても同様です

5 Hello World (ウィンドウ版)

次は Windows アプリケーションです.

1. 今度は"Windows アプリケーション"テンプレートを選択します(図5参照)

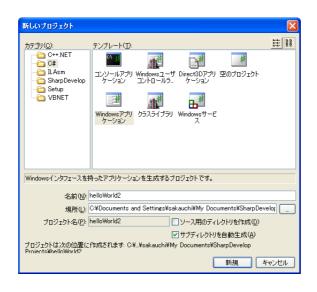


図 5: Windows アプリケーションの新規作成

2. コードが表示されている MainForm.cs ウィンドウの下に「ソース」と「デザイン」タブが表示されています「デザイン」をクリックし、MainForm.cs をフォームデザイナで編集します.

画面左下の「ツール」タブをクリックし"Windows Forms"をクリックすると,おなじみのコントロール群が顔をだします.

ボタンとテキストボックスを配置しましょう.(図6参照)

3. ボタンを押したときのイベントハンドラをコーディングします. ボタンをダブルクリックします. 自動的にイベントハンドラが作成されコードエディタ画面に切り替わります.

カーソル位置に以下をコーディングします(図7参照)

textBox1.Text = "こんにちは,世界!";

実行して、ボタンをクリックすると、テキストボックスのテキストが変わるでしょうか?

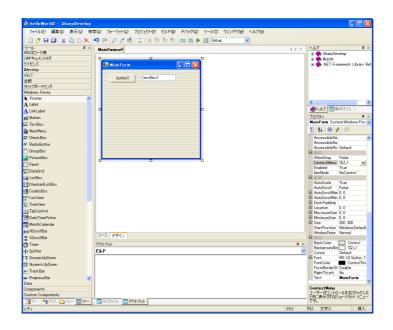


図 6: デザイナによるフォームの編集

図 7: イベントハンドラのコーディング

6 ユーザーコントロールの作成と追加

最後に,ユーザーコントロールを追加してみましょう.

1. 左下の「プロジェクト」タグをクリックし,プロジェクトツリーを表示します.helloWorld2を右クリックして「追加/新しいファイル」を選択します「ユーザーコントロール」テンプレートを選択し,MyFirstControl.cs という名前をつけ(図 8)新規をクリックします.

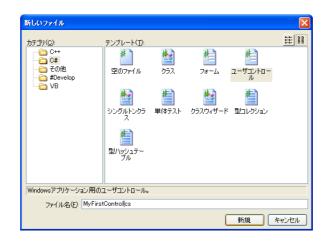


図 8: 新しいファイル

2. 「デザイン」に切り替えて,ボタンとテキストボックスを追加し,フォームの大きさを整えます(図9)

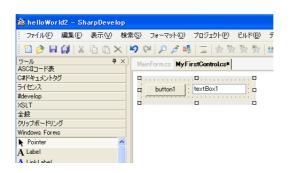


図 9: ユーザーコントロールのデザイン

3. ボタンをダブルクリックしてイベントハンドラをコーディングします.プロパティを追加して,イベントハンドラで表示するようにしましょう(図10)

図 10: ユーザーコントロールのコーディング

4. ここでコンパイル(ビルド)します . F8 を押しましょう⁵

⁵メニューから「ビルド/コンバインをビルド」を選んだり,ツールボックスからビルドを指示することもできます

5. 無事にコンパイルが成功したら, MainForm.cs のデザインに切り替えて

「ツール」から Custom Components を選択します . フォーム上に配置可能な MyFirstControl があるはずです . プロパティウィンドウには , ちゃんと message が存在します . 2 つのコントロールを配置し message プロパティに , こんにちは 」と 「こんばんわ」を設定します (図 11)

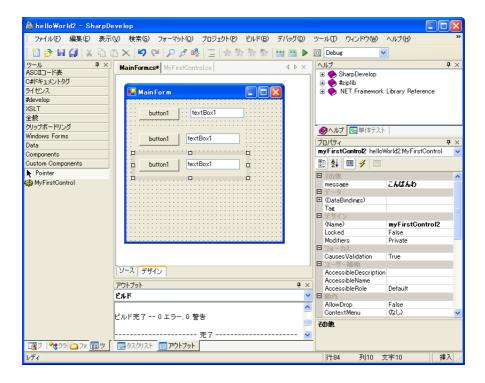


図 11: ユーザーコントロールを配置

6. 実行してボタンをクリックした様子が図 12 です.



図 12: プログラム実行中

7 最後に

駆け足ですが,#developのインストールと,その開発について紹介しました.とても簡単にユーザーコントロールを作成できることも感じていただけたでしょうか.

C# については殆どふれませんでしたが, Java に非常によくにた言語です.大きくわけると,整数や実数などのスタックにおかれる値型と,クラスや文字列などヒープに置かれる参照型の2つの型を持ちます.プログラマはメモリ管理に煩わされることはありません.コンパイラが,どこからも参照されなくなった記憶領域を回収してくれます(ガベージコレクション)

.NET のクラスライブラリは,言語に共通です.言語ごとに新たなライブラリを学ぶ必要はありません.

また,.NET は CLR(Common Language Runtme)と呼ばれる仮想マシン技術を基盤につくられています.Java と同じように (CLR さえ供給されれば)32bit なプラットフォームでも 64bit なプラットフォームでも再コンパイルの必要さえなく動作します.

Microsoft 以外の実装もすすめられつつあります . 図 13 は , Mono と呼ばれるオープンソースな CLI の実装 (CLR にあたるもの) を利用して , Linux で , helloWorld.exe が動作している様子で す . 6

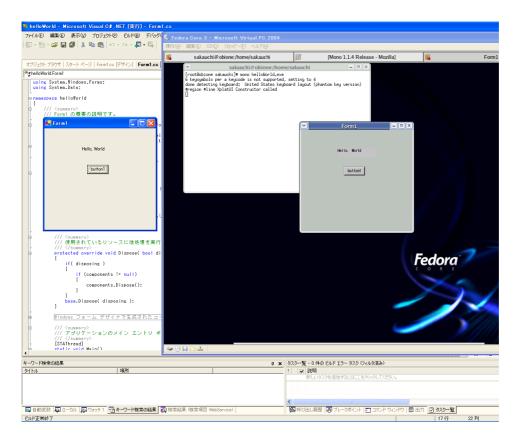


図 13: Linux で動く helloWorld.exe

⁶まだ クオリティで実用段階ではありませんが

C# と .NET は ,数年前に Java がサーバサイドのプログラマにもたらした多くの恩恵を Windows プログラマに , もたらすでしょう . 一つ問題があるとすれば , C# によるプログラミングを経験したあとでは C++ によるコーディングが苦痛になることでしょうか :p

ぜひ,C#によるプログラミングをマスターして楽になってください.