# #develop (SharpDevelop) で .NET

# システム開発部 坂内 敦

### 2005年3月25日

# 目 次

1	#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	<b>2</b>
	3.1 ダウンロード	2
	3.2 セットアッププログラムの実行	3
	3.3 コード補完データベースの作成	3
	3.4 日本語表示のための設定	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 SDK を インストールする必要があります..NET 1.1 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.<sup>2</sup>

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

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

2. .NET Framework Version 1.1 日本語 Language Pack

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

3. .NET Framework 1.1 Service Pack 1

http://www.microsoft.com/downloads/details.aspx?FamilyID = a8f5654f-088e-40b2-bbdb-a83353618b38& displaying the second structure of the second struc

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

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

5. .NET Framework SDK Version 1.1

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

### 3 #develop のインストール

### 3.1 ダウンロード

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

から, SharpDevelop-X.X.X.XXXXX\_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.

<sup>2</sup>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 "MS ゴ シック" 9 ポイント 等に変更します.

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.

Start Page の "New Combine" ボタンをクリックします. (図4参照)<sup>3</sup> #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.

<sup>&</sup>lt;sup>3</sup>または,メニューの「ファイル/新規作成/コンバイン…」を選択

<sup>&</sup>lt;sup>4</sup>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)

新しいプロジェクト					
カラゴリ(Q): C++,NET C# C# C# C# C# C# C# C# C# C#	デンプレート(I):       ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・				
Windowsインタフェースを打	きったアプリケーションを生成するプロジェクトです。				
名前(W): helloWorld2 場所(W): (○*Documents and Settings¥sakauchi¥My Documents¥SharpDevelox) フロジェクト名(P): helloWorld2 ソース用のディレクトリを作成(D) ジ サブディレクトリを自動生成(A) プロジェクトは次の位置(こ作成されます: C¥_¥sakauchi¥My Documents¥SharpDevelop Proients¥BhelloWorld? 新規 「キャンセル					

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.

 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.

新しいファイル					
_カテゴリ( <u>O</u> ):	テンプレート(工):				5-5- 5-5- 5-5- 5-5- 5-5- 5-5- 5-5- 5-5
	どうしゃう そうしゃう どうしゃう どうしゃう どうしゃう どうしゃ かくしゅう かく かく かくしゅう かく	<b>惨</b> クラス	<u>惨</u> フォーム	ב-ידבי-רם- א	
<b>-</b>	シングルトンクラ ス	たいしょう しんしょう しんしょ しんしょ	り クラスウィザード	<u></u> 型コレクション	
	<b>生</b> 型ハッシュテー ブル				
Windowsアプリケーション	用のユーザコントロール。				
ファイル名( <u>F</u> ) M	FirstControlles				
			0	新規	キャンセル

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)

```
void Button1Click(object sender, System.EventArgs e)
{
        textBox1.Text = _message;
}
private string _message;
///<summary>
/// 表示するメッセージ
/// </summary>
public string message {
        get {
                return _message;
        }
        set {
                _message = value;
        }
}
```



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<sup>5</sup>

<sup>&</sup>lt;sup>5</sup>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)

📸 helloWorld2 - SharpDe	eve lop	
: ファイル(E) 編集(E) 表示	₩ 検索(S) フォーマット(Q) プロジェクト(P) ビルド(B) デバッグ(Q)	ツール① ウィンドウѠ ヘルプ(屮) 🂙
1 🖸 🤌 🖬 🎒 🗶 🗅 (	🛚 🗙 🔎 🔎 🎜 🦉 📋 😓 数 数 🔛 🕮 🕨	O Debug
ツール 早×	MainFormes* MyEirstControles 4 D ×	ヘルプ <b>キ</b> ×
ASCIコード表		🗉 🔶 SharpDevelop
C#ドキュメントタグ		🗈 🔶 #ziplib
ライセンス	🔜 🖬 MainForm 📃 🗖 🔀	🗄 🏀 .NET Framework Library Reference
#develop		
XSLT	hutton1 textBox1	
全般		
クリップボードリング	• • • • • • • • • • • • • • • • • • • •	ペルプ M型体テスト
Windows Forms	button1 textBox1	
Data		7U/\77 4 ×
Components		myFirstControl2 helioworld2.MyFirstControl
Custom Components	button1 textBox1	🔡 🛃 🔲 🗲 📼
📐 Pointer		□ 子の他
🤬 MyFirstControl	. L	message こんぱんわ
	· · · · · · · · · · · · · · · · · · ·	ヨ データ
		⊞ (DataBindings)
		Tag
		□ デザイン
		(Name) myFirstControl2
		Locked False
		Private
		Causes)/alidation True
		□ 7-ザー補助
	ソース デザイン	AccessibleDescription
	Trade 1 Trade 1	AccessibleName
	PJFJ9F ¥ X	AccessibleRole Default
	Enk 📉	動作
	^	AllowDrop False
	ビルド完了 0 エラー, 0 警告	ContextMenu 🕼 🛛
		その他
	完了	
🛐 🦄 🖄 🖓 🖓 🖓	,	-
レディ		· 行:84 列:10 文字:10 挿入 ::

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.

🖳 Main Form		
button1	こんにちは世界!	
button1	こんにちは	
button1	こんばんわ	

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.<sup>6</sup>



図 13: Linux で動く helloWorld.exe

<sup>&</sup>lt;sup>6</sup>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.