

Java Watch on the X》 JBuilder と Project Builder を併用する

Mac OS X 版の JBuilder 6 はすでにリリースされているが、Java VM 1.3.1 Update1 がリリー スされる前は、日本語のシステムで使うのはほぼ不可能に近かった。Update1 におより、 英語版として配付されている JBuilder 6 もなんとか使える範囲に入ったようなので、さっ そく活用したいところだが、パッケージ形式のアプリケーションを組む場合には Developer Tools を利用したいと考えるかもしれない。また、JBuilder 自体の動作はやはり Java で動い ているだけに、細かいデバッグは Project Builder の方が効率的かもしれない。そこで、JBuilder で GUI をデザインし、Project Builder でビルドするといった使い方を紹介しよう。

JBuilder 6 Personal を使ってみる

Java の開発ツールの中で、JDK を除くとユーザがいちばん多いのはボーランドの JBuilder であるという調査結果もあるほどのメジャーなツールだ。Java で作られたツ ールであるとは言っても、Mac 向けにリリースされたのは初めてだ。Mac OS 向けに リリースされなかったのはいろいろ理由があるとしても、JDK 1.1 ベースだったこと が大きいだろう。だが、Mac OS X は比較的新しい Java VM が組み込まれている。 Windows や Solaris との遅れがあったことは大きくばん回したが、その結果、JBuilder のリリースへとつながった。

JBuilder はいくつかのバージョンがあるものの、ボーランドのサイトでのオンライン 販売は米国などの地域だけで、基本的には日本向けには販売されていない。日本では まだ Mac OS X 版の JBuilder 6 は販売されていないので、結果的には米国の Borland の サイトからダウンロードできるフリー版の JBuilder 6 Personal だけが入手できるという ことになってしまう。Personal 版は機能制約されているが、Swing ベースの GUI をデ ザインツールで作成してアプリケーションを作る点については、制約なく利用できる。

◇JBuilder 6 Personal

JBuilder 自体の説明も必要かもしれないが、今回はおおまかに説明しよう。JBuilder も アプリケーションとなっているが、エディタ、GUI ツール、デバッグ機能などすべて 1 つのアプリケーションに統合化されている。だから、開発作業は JBuilder を起動し て作業すればそれだけで OK である。

もちろん、Java のソースのエディタなどもあるが、Java のソースは*.java ファイルに 保存するなど、いくつかのファイルを作成するのは他のツールと変わらない。また、 プロジェクトというものを 1 つのファイルを作って管理し、ある開発アプリケーショ ンなどを 1 つにまとめるような状況になっているのも同様である。ただ、通常の利用 では、class ファイルを特定のフォルダに生成してそれを実行する形式となっている。 だから、1 つのプロジェクトでは、ファイルやフォルダがたくさん用意されることと なる。

JBuilder でプロジェクトを作成する

実際に JBuilder でアプリケーションを作ってみよう。プロジェクトを作成するには、File メニューから New Project を選択するのが 1 つの方法だ。以下のように、Project Wizard のダイアログボックスが出て来るので、必要な情報を入力する。最初は、プロジェク ト名を入力するが、JBuilder では、ホームフォルダの直下に jbproject フォルダを作り、 そこのプロジェクトを集めておくといったことがデフォルトになっている。 Project Wizard の最初の画面

000	P	roject Wizard - Step 1 of 3			
	Select name and template for your new JBuilder project				
	Enter a name for your project and the path to the directory where it is to be saved. You can optionally choose an existing project as a template f or the initial default values used in this wizard.				
	N <u>a</u> me:	JBTest	<u>T</u> ype: jpx 🔻		
	Directory:	/Users/msyk/jbproject/JBTest	•		
	Tem <u>p</u> late: (Default project)				
	Cenera	te project notes file			
	e	< Back Next > ? Canc	el Finish		

Project Wizard の2つ目のダイアログボックスではいろいろな設定があるが、基本的に はこのままでいいだろう。ここで、Output Path は生成した Class ファイルが作られる 場所であるが、プロジェクトのフォルダの下にある classes フォルダとなっている。ま た、ソースはプロジェクトのフォルダの下にある src フォルダとなっているが、こう したフォルダ構成が JBuilder でのプロジェクトの基本となっている。

Project Wizard の2つ目の画面

000	Project Wizard - Step 2 of 3				
Specify projec	t paths				
Edit the paths s can be chang	and settings here to help define your new project. The ged after the project is created.	ese and other pro	pertie		
JDK: java 1.3.1-root-010902-18:51					
Output path: /Users/msyk/jbproject/JBTest/classes					
Bac <u>k</u> up path:	/Users/msyk/jbproject/JBTest/bak				
Working directory:	/Users/msyk/jbproject/JBTest				
1	Source Documentation Required Libraries	<u> </u>			
Default Test	Path	Add	\supset		
O /Use	rs/msyk/jbproject/JbTest/test	Edit	D		
	Remove	D			
		Move Up	D		
1		Mova Down			

そして、Project Wizard の 3 つ目の画面では、JavaDoc のフィールド設定などがあるが、 ここでは Encoding で SJIS つまり Shift JIS を選択しておくのが基本だろう。そして、Finish ボタンをクリックすることで、プロジェクトが新たに作られる。

Project Wizard でエンコードの設定を行う

00	Project Wizard - Step 3 of 3				
Specify general project s	ettings				
Enter settings here to he nged after the project is	lp define your new project. These and other properties can be cha created.				
Encoding: SJIS					
Enable assert <u>k</u> eyword					
Automatic source packages	<u> </u>				
Enable source package d	iscovery and compilation				
Deepest package exposed:	3				
_ 1 1 2 1					
Class Javadoc fields:					
Label	Text				
Title:					
Description:					
Copyright:	Convright (c) 2002				
Copyright: Company:	Copyright (c) 2002				
Copyright: Company: @author	Copyright (c) 2002				
Copyright: Company: @author @version	Copyright (c) 2002				
Copyright: Company: @author @version	Copyright (c) 2002				
Copyright: Company: @author @version Include references from pr	Copyright (c) 2002 1.0 oject library class files				
Copyright: Company: @author @version Include references from pr	Copyright (c) 2002 1.0 oject library class files				
Copyright: Company: @author @version Include references from pr	Copyright (c) 2002 1.0 roject library class files				

こうして作られたプロジェクトのフォルダ構成を Finder でチェックしていただきたい。 プロジェクトのフォルダにはいくつかファイルがあるが、プロジェクトのファイルは JBuilder アプリケーションに関連付けられていないので、プロジェクトのファイルを ダブルクリックして開くといったことができない。これはリリースのアップで改善さ れることを期待するしかないだろう。

プロジェクトにアプリケーションを追加する

JBuilder ではアプリケーションについても基本的なひな形を簡単に作成する機能が用 意されている。File メニューの New (Command+N) などで Object Gallery を呼び出し て行うことができる。以下のように、Personal 版は限られたオブジェクトしか作るこ とはできないが、幸い Application の作成はできる。

New	Project	Web	Object Ga	CORBA Er	terprise	Test
			A	-		7
Application	Applet	t Arch	ive Builde	er Javadoc	Class	Interface
0					l	
JavaBean	Dialog	Frame	Panel	Data Module	e Data I Appli	Module cation

Application を作成する

そうすると、Application Wizard が呼び出される。最初の画面では、パッケージ名とク ラス名を入力する。デフォルトでもかまわないけども、JBuilder ではパッケージをき ちんとつけるというのがスタイルである。Project Builder ではパッケージの指定は任意 的な感じだが、JBuilder ではきちんとパッケージを指定するのが基本である。デフォ ルトでは、プロジェクト名を全部小文字にするなど、システム的には自動的に作って くれる。

アプリケーションのクラスと組み入れるパッケージを指定する

00	Application Wizard - Step 1 of 2			
	Enter application class details Fill in the following fields to quickly define and create a n ew Java application. The application will consist of a main application class and a Frame class that you can customize using the visual designer. Package: jbtest Class: Application1 ✓ Generate header comments			
<	Back Next > ? Cancel Finish			

2 つめのダイアログボックスでは、アプリケーションのメインクラスに加えて、ウイ ンドウを表示するためのクラスの定義が行われる。クラス名などは適当に入力すると しても、Option として、メニューバーやステータスバーなどさまざまなものが自動的 に作られる。ある意味では便利だが、生成されたソースを一度は解析しておく必要は あるだろう。

ウインドウの諸定義を行う

00	Application Wizard - Step 2 of 2
	 Enter frame class details Fill in the following fields to define the frame class for your new Java application. Once created, this frame class can be customized using the visual designer. Class: Frame1 Title: Frame Title Options © Generate menu bar © Generate toolbar © Generate status bar © Generate About dialog © Center frame on screen
	< Back Next > ? Cancel Finish

こうしてプロジェクトには、アプリケーションの本体とも言える Application1 と、ウ インドウを表示するための Frame1 というクラスが定義された。プロジェクトではそ れらのファイルが見えているが、アバウトボックスのウインドウのクラスや、ツール バーのアイコンの GIF ファイルなども見えている。



プロジェクトに作られたいくつかのクラスや画像ファイル

ここで、エディタ部分で利用するフォントについてはカスタマイズできるので、設定 を行っておこう。Tools メニューの Editor Options を選択して、Display のタブを選択す ると、エディタで使うフォントを指定できる。もちろん、好みでかまわないが、 Osaka-Mono で 14 ポイントあたりが無難なところだと思われる。

00		Editor Options		
Editor D	isplay Color	CodeInsight	Templates	Java Structure
Right margin: <u>V</u> isible rig	ht margin Right <u>i</u>	margin: 80		ŧ
Editor font:				
Font family:	Osaka-Mono		Fo	nt <u>s</u> ize: 14 🔽
Sample:				
		AaBbXxZz:		
Reset		0	? Canc	el OK

エディタで使うフォントを指定した

JBuilder のデザインツールを使う

さっそく、JBuilder の GUI 生成機能を使ってみよう。ここではウインドウは Framel ク ラスであるので、そのクラスを開いておき、下側のタブで Design を選択すると、最初 は時間がかかるが、GUI のデザイン画面に切り替わる。デザイン機能の使い方もここ では省略するが、ポイントはレイアウトの機能をうまく理解して、結果として生成さ れるソースをある程度確認しながらオブジェクとを配置するというあたりだろう。な お、自由にレイアウトしたいなら。とりあえずはレイアウトを null にするという方法 もある。ここで、配置したボタンの Text プロパティに日本語文字列を設定してみたの だが、プロパティのリストでも、デザインの画面でも、文字化けが発生した。

<u>вее.</u> в п л <i>-</i> ~	Users/msyk/jbproject/jBTest/src/jbtest/Frame1.java	
□ □ □ □ · · · □ □ □ · · · · · · · · · ·		0 m.+ &
 JBTest.jpx Application 1.java closeFile.gif France 1.java 	Swing Swing Containers DataExpress dbSwing	More dbSwing dbSv ↔
S Frame1, Java		prefer 84, 26
 openFile.gif this contentPane (BorderLayout) borderLayout1 is jButton1 iButton2 		reque True rollov False rollov selected False select text toolTi verifyl True
jButton3	₽	Properties

一方、このまま、Source のタブを選択して、ソース表示にすると、日本語はまったく 問題なく利用できる。だから、ボタン名を日本語にしたい場合には、Design では適当 な文字を設定しておいて、正しい文字列をソースの側で設定するという方法で問題な く、日本語のダイアログボックスは作成できる。なお、ソースエディタは、Java 1.3.1 Update1 以前だと編集がおかしくなっていたのだが、Update1 により何の問題もなく、 きちんと日本語の編集ができるようになっている。



Project Builder のプロジェクトと統合する

次に、JBuilder のプロジェクトで生成された Java のソースを Project Builder で利用でき るようにする。Project Builder でプロジェクトを作るとき、ここでは「空白のプロジェ クト」を選択してみた。



そして、JBTestPB と言う名前でとりあえず、JBuilder のプロジェクトのフォルダと同 じ位置にここでは作っておく。Project Builder のプロジェクトは必ずフォルダを作るよ うなので、作ってから後で Finder を使って移動するなどする。

N 8!	J名で保存: JBTestPB
プロジェク 保存	Documents Download Documen for_System_Reinstall FTPSyncList.txt FTPSyncLog.txt jbproject keiko_face.png Library Library MoreFilesXTest
	新規フォルダ よく使う項目に追加 キャンセル 選択

とりあえず適当な名前と場所にプロジェクトを作る

ここでプロジェクトを閉じるか、Project Builder を終了するなどしてから、Finder で作 ったプロジェクトのファイルを、JBuilder のプロジェクトのフォルダに移動しておく。 単にドラッグすればいい。ここでは Project Builder のプロジェクトファイルだけが欲 しいので、作成されたフォルダは削除しておいてもいいだろう。



プロジェクトのファイルを JBuilder 側のフォルダに移動しておく

そして、Project Builder のプロジェクトファイルをダブルクリックして、プロジェクト を開く。「プロジェクト」メニューの「新規ターゲット」を選択して、新しくターゲ ットを作成する。ターゲットの種類は、Java の Application を選べば良いだろう。また、 名前ももちろん適当につけておく。

Java のアプリケーションのターゲットを追加する

) 🛃 JBTestPB.pbproj	0
新規ターゲット	2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
結合 ▼Cocoa Application	
▼ Java Applet Application Package	
Tool V Kernel Extension Generic Kernel Extension IOKit Driver	
アプリケーション ツール パンドル フレームワーク ライブラリ	÷
キャンセル 戻る 次へ	(元7)

続いて、JBuilder が生成したソースを、Project Builder 側にも登録する。もちろん、Finder からドラッグ&ドロップする方法でも、メニューから選択しても良いが、いずれにし ても、プロジェクトのターゲットに登録しておくようにする。JBuilder では、クラス 定義にパッケージを含めるが、パッケージの階層に応じたフォルダ構成を自動的に作 る。これは Java の作法だが、Project Builder を使う上ではパッケージをあまり意識し ないかもしれない。だが、JBuilder ではパッケージ階層とフォルダの階層が合ってい ないといけないので、ソースファイルの位置は移動させないでおこう。また、Project Builder でソースを追加するときには気をつける必要があるポイントだ。 また、GIF ファイルについても基本的にはプロジェクトに登録しておき、生成したパ

ッケージに含めるようにするのが基本だが、これについては後で注釈する。

ソースファイルを Project Builder にも登録する。

000	🛃 J	BTestPB.pbproj
5 🗞 🔿	📃 🗐 📜	
◎ グループとウァイル ■ IBTestPB	11.java	
Productsava	a R	🔾 🖸 💰 ビルドスタイル:開発 🗢 🔣 🙆 🙆 😣
		ルド設定
Frame1t	Box.java	名前值
	K	COPY_PHASE_STRIP = NO
	6	00
	000	j j btest 🤤
	 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
	jbtest	▶ j Application1.java
		😰 closeFile.gif
		j Frame1.java
		Frame1.jbx
		j Frame1tBox.java
		openFile gif
		i opennegn

Project Builder 側では、ターゲットとして Java の Application を選択した場合でも、い くつかの設定は追加しなければならない。「プロジェクト」メニューの「アクティブ ターゲットの編集」(Command+option+E)を選択して、「アプリケーション設定」の タブを選択し、「詳細設定」のボタンをクリックする。ここで、Java という辞書クラ スのプロパティを作成し、さらに ClassPath と MainClass のプロパティを設定する。設 定の方法については、以前に説明した通りであるが、作成される jar ファイルをクラ スファイルとして参照すると同時に、メインクラスは、パッケージ名を含めたクラス 名を記述しなければならない。

アプリケーション設定を追加する

000		🛃 JBTestPB.pbpr	oj	0
🔨 🧞 📵 🧯	L -	👌 🐔 💿 JBTest	÷	
ダループとファイル			Q	検索 🥄 ビルド 📜 要実行 🗼 📜 デバッグ
JBTestPB	7	O O Ø ターゲット:JBTest \$		0 0 2 2
Application1.java	5	קב	ビード部中	マプリケーション部中 東谷司ポファイリ
Frame1_AboutBox.jav			ヒルド設定	アプリリーション 設定 美川可能アアイル
🗹 👳 openFile.gif				簡易設定 詳細設定
CloseFile.gif	K			
Products	4	下位に新規作成 削	除)	
- Inoducis	0	プロパティリスト	クラフ	結
	\prec	CEBundleDevelopmentBegion	文字列	≜ English
	5	CEBundleExecutable	文字列	↓ English
		CFBundleGetInfoString	文字列	i l
	2	CFBundleIconFile	文字列	÷
	S	CFBundleIdentifier	文字列	÷.
	a	CFBundleInfoDictionaryVersion	文字列	\$ 6.0
	2	CFBundleName	文字列	÷
		CFBundlePackageType	文字列	↓ APPL
	2	CFBundleShortVersionString	文字列	÷
	7	CFBundleSignature	文字列	\$ 7777
	4	CFBundleVersion	文字列	\$ 0.0.1d1
		▼Java	辞書	
	\prec	ClassPath	文字列	\$JAVAROOT/JBTest.jar
	1	MainClass	文字列	jbtest.Application1
	2	NSJavaNeeded	文字列	\$ YES
	5	▶ NSJavaPath	配列	↓ 1 個の順序付きオブジェクト
	6	NSJavaRoot	文字列	Contents/Resources/Java
	1			
	n			
	1			
ビルドは問題なく完了しました				

これでビルドをすると、Mac OS X のアプリケーションパッケージとなったアプリケーションが生成される。

Project Builder での生成されたアプリケーション

まず、未だに Project Builder で生成された Swing のアプリケーションは、日本語文字 列はそのままでは文字化けする。CFBunldeDevelopmentRegion を Japanese にするとい う方法もあるが、Mac でしか使わないのなら、あっさり、フォント設定するのがとり あえずは手軽だろう。ソースでいちいちと思うと面倒だが、JBuilder の Design 機能で、 オブジェクとをまとめて選択して、Osaka フォントに設定すればいいことなので、作 業的にも面倒は少ない。

こうして 2 つの開発ツールで同一のソースファイルを編集するとなると、競合の問題 が発生してしまう。したがって、JBuilder から Project Builder に移るときには、ソース ファイルをいったん閉じる。そして、改めて開く。逆も同様にする。これが基本だ。 JBuilder では閉じるキーボードショートカットが Command+W なのに対して、Project Builder は Command+Shift+W となっている。Project Builder で Command+W をするとプ ロジェクトのウインドウ自体が閉じられてしまう。ちょっとうっとおしいけど、すぐ になれるだろう。

JBuilder でアプリケーションを作るときにツールバーを指定すると、アイコンの GIF ファイルを用意する。GIF ファイル自体は、ウインドウのクラスと同じフォルダにあ るという仮定でプログラムが組まれている。プログラムをそのまま生かす形にしたい のなら、生成される jar ファイルの中に GIF ファイルをアーカイブしなければならな い。たとえば、

jar uf build/JBTest.app/Contents/Resources/Java/JBTest.jar -C src jbtest/closeFile.gif

のようにして、closeFile.gif という GIF ファイルを、フォルダ階層に応じて jar ファイ ルにパッケージできる。これを必要なファイルの数だけ実行するのだが、ビルドフェ ーズでシェルスクリプトを実行するようにすると、良いだろう。

JBuiler で作ったアプリケーションをパッケージとして生成したい場合では、以上の要にうまう使い分ければそれほど難しいことではないだろう。

いいいいので、以上いいいいのの「新居雅行」いいいいのの