

# Delphi 8 オーバービュー

特集

2

いよいよ.NETに参入したDelphiの実力を探る

吉田 弘一郎

YOSHIDA, Koichiro

Borland Delphi 8 for the Microsoft .NET Frameworkがリリースされました。Webによれば、昨年のクリスマス直前にリリースされたはずなのですが、私の手元に届いたのは2月に入ってからです。Delphi 8の仕様に関しては、昨年11月に催されたBorlandカンファレンスのレポートで書いた通りですが、Borland虎の子のDelphiだけに.NET用となった第8版には、大いに期待したいのです。それでは、出来たてホヤホヤの日本語版の試用報告をご覧ください！

## Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:  
Delphi 8 Enterprise

## Level



## Samples

註) 本稿はDelphi 8 Enterpriseのフィードバック版を使用して執筆されています。そのため、実際の製品版とは異なる点があるかもしれません。



## 今さらBorland?

読者のみなさんの中には、Borlandという小さな会社の、しかもDelphiというPascal処理系に不安を感じる方も少なくないと思います。決してメジャーとは思われていませんから。そもそも、Turbo Pascalの初版が出たのは21年も昔の話。今さらPascalすることもないと感じても不思議はありません。 .NETするのであれば、寄らば大樹の陰でMicrosoftのVisual Studio .NETを用いて、C#やVisual Basic .NETを用いるのが無難であるように思えます。このような時期に、なぜBorlandのDelphiが浮上してくるのでしょうか？ 昨年11月サンノゼで催されたBorlandカンファレンスは、まさにこの疑問に答えるものでありました。

その答えとは、今までなら「開発効率」の一語に尽きます。同じVisualな開発環境にしても、Visual BasicとDelphiを比較すれば、Delphiの優位は明らかでした。でも、BorlandはMicrosoftではありません。Borlandであるこ

とに、ユーザーは不安を感じたのです。ところが、このような状況が一挙に変わりました。MicrosoftがBorlandにラブコールを送り始めたようなのです。

ここで思い出すのはDelphi 6で始めたCLXの苦い思い出。Win32用のDelphiとC++Builderは、実に見事にLinuxに移植され、Kylixとしてデビュー。Win32でも、Linux互換モードであるCLXアプリとして開発すれば、そのままKylixで再コンパイルしてLinux用アプリになりました。しかし、Kylixは「Linuxの技術的未熟さ」に足を引っ張られ、またタダ版をばら撒きすぎたせいかビジネスとしても惨敗で、昨年11月のBorlandカンファレンスでは誰も相手にしませんでした。要するにKylixは死んだのです。ですから、Borlandが何か新しいモノを始めたとも聞いても、少々及び腰になって当然。ところが、今回のDelphi 8によるDelphiの.NET化は、Microsoftとの熱い関係の中で行なわれたのです。LinuxはKylixを必要としませんでした。 .NETはDelphi 8を必要としているのです。

Microsoftは、社を上げて.NET路線を突き進んでいます。その象徴が次期

Windows “Longhorn”であることは、言うまでもありません。これを成功させるためには、Longhornがリリースされたときに、その機能を最大限活用するファンシーなアプリケーションの存在が不可欠です。そしてまさにそのために、MicrosoftはBorlandに接近してきたようなのです。とにかく、MicrosoftとBorlandの仲は、かつてなく親密なのです。ですから、Borlandが小さな会社であるということは、何の不安材料にもなりません。会社設立から20年を経て初めて、Borlandはユーザーを安心させてくれたのでした。

### ■■■ 今さらDelphi ?

Delphiが非常に効率の良い開発環境であることは認めたとしても、それはWin32での話でした。なぜにDelphiが.NET用開発環境として浮上してくるのでしょうか。往年のDelphiユーザーの枠を越えて、Delphi 8は意味を持つのでしょうか？

答えはもちろん“Yes”です。その理由は、次の一語に尽きます。

#### Win32との親和性を保った.NET化

要するに、.NET開発にはDelphi 8を用いるのが、少なくとも私の知る限りにおいて、一番楽なのです。具体的にその理由を述べれば、次のようなDelphi7との関連があるからです。

#### Win32用Delphi 7の.NET版=Delphi 8

まず「Delphi 7がWin32用最終版Delphi」であることにご注目。近年、頻繁にバージョンアップを繰り返してきたBorlandですが、Win32用DelphiはDelphi 7で「完結」したのです。そして、このDelphi 7との互換性を最大限に確保しつつ.NET用にDelphi 8を開発しました。この互換性のために、Delphi開発環境をVisualたらしめているVisual Component Library (略してVCL) までをも.NET化したのです。「Win32用の素直なDelphiプログラム」であれば、そのプロジェクトをDelphi 8でコンパイルしなおすだけで.NET用プログラムになるのですから驚きです。このような互換性は他のプログラム言語にはありません。

#### Note Win32のプログラムの.NET化は楽じゃない

- ・ Visual C++ 6.0で作ったMFCプログラムは、MFCが.NET非対応なため、そもそも.NET化できない
- ・ Visual Basic 6.0で作ったプログラムは、Visual Basic .NETに移植して.NET化する
- ・ プログラム全体の.NET化でなければ、個別にCOM化したりして、.NETに持ち込むことができる

ところが、Delphi 7のWin32プログラムであれば、うまくゆけば無変更のままDelphi 8でコンパイルしなおすだけで.NET化することができます。

### ■■■ Delphi 7のプロジェクトをDelphi 8で.NET化

論より証拠で、Delphi 7でWin32プログラムを作り、それをDelphi 8でコンパイルしなおして.NETプログラムにしてみましょう。図1のようなWin32プログラムで我慢してください。ボタンを押すとカウントがひとつ増すという実に簡単なものです。

このDelphi 7のプロジェクトをDelphi 8で開き、ただ単にコンパイルしなおすだけで、.NET化されるのです。あまりに簡単なので、だまされたような気分です。あるいは、Win32も.NETもまったく同じに思えてしまいます。実際、図2が.NET版ですが、見かけでは区別できません。

このプログラムEx01のUnit1.pasのコード（リスト1）をご覧ください。.NET化した際に「System.ComponentModel」ユニットが自動的に付加された以外、.NET的なものは何もありません。見事です。

なぜ、ここまで.NET臭を抑えることが出来たのかと言えば、それはBorlandがDelphiという言葉も、それ用のライブラリも、そっくりそのまま.NET化してくれたので、ユーザーの手を煩わさずに済んだのです。Borlandが言語もライブラリもみな自分で所有している賜物です。Win32用各ライブラリUnitの中は大いに書き換えても、同じファイル名、同じ関数名のままに.NET化してくれたので、“ユーザーのプログラ

図1：Delphi 7で作ったWin32プログラム



図2：.NET Frameworkのプログラム

