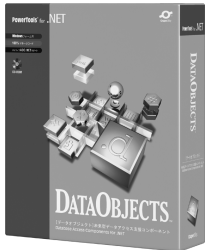


NEW PRODUCTS

Visual Studio.NET/.NET Framework関連ツール新製品レビュー



DataObjects for .NET

大容量結果セットの高速化、クエリテーブルへのトリガ機能追加、分散配置を可能にした、ADO.NET代替のデータアクセスコンポーネント



秋月巖ソリューション事務所
<http://www.akizuki.co.jp/>
秋月 巖 AKIZUKI, Iwao

問
合
先

グレープシティ株式会社

TEL : 022-777-8211

FAX : 022-777-8233

URL : <http://www.grapecity.com/>

MAIL : sales@grapecity.com

Technology Tools

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

Environment

ターゲットOS
Windows 2000/XP/Server 2003

対応開発環境
Visual Studio .NET 2002/2003
.NET Framework 1.0 (SP2以上) / 1.1

価格
81,900円 (パッケージ版)
65,100円 (ダウンロード版)

※ともに1開発ライセンス。サーバーライセンスあり (サーバーコンポーネントとして運用する場合、Webサーバーごとに別途サーバーライセンスが必要)

何のための製品なのか？

最初にこのDataObjects for .NETのニュースリリースを受け取ったとき、一体、何のためのものなのか、はっきりとした製品イメージをつかめなかった。ただ、.NETに関する開発製品であり、データベースに関連するものだということはわかったので興味を持った。

この時点ですでに、私は本誌2004年1月号～3月号の連載「.NET Framework アイデアノート」において簡単なプログラムコードでデータベースにアクセスできるUDA (Universal Data Access) クラスを発表していた。このクラスはADO.NETのラッパークラスであり、できることはADO.NETと何も変わらない。プログラミングスタイルを変えるためのクラスである。その後、同連載の2004年11月号・12月号ではADO.NETでSQL Serverのサーバーカーソルを利用するためのUDAsクラスを発表し、2005年1月号・2月号では高機能DataSetを実現するTable Module 用抽象クラス (TableModule クラス) を発表した。

これらはすべて本製品の機能や内容を知らずに行なってきたことである。

今回、本レビューの執筆のために製品を評価してみて、私のこれらの仕事と本製品が目指すものが大きく重なっていることを発見したのである。.NET Frameworkを使ったデータベースプログラミング (つまり、ADO.NET) で感じる問題は共通だからだろう。

ADO.NETの代替製品

まず、私が本当にこのレビュー記事を書くのにふさわしい人間であるかを考える必要がある。なぜなら、私は本製品の競合製品ともいえるようなクラスを発表しているからである。確かに、私は自分が発表したクラスがどれだけ利用されるかによって収入を得るわけではない。だから、読者がDataObjects for .NETを使うか、あるいは私が発表したクラスを使うかで、私に利害があるわけではない。とはいえ、やはり、自分が作ったものはかわいいものである。問題点があるとしても、その回避の仕方がわかるし、それになんといっても新たに学習する必要がない。だから、もし、私のDataObjects for .NETに対する評価が厳しいものになったとしても、その点を割り引いて考える必要が

ある。競合製品を提供している技術者による他社製品評価だからである。その一方で、同じことに問題を感じ、その解決策を提供してきたという理由で、私こそがこの製品を評価するのにふさわしいという気持ちもある。

DataObjects for .NETとは何かということを一言でいえば、この製品は“ADO.NETの代替製品”である。ADO.NETの代わりに用いて、各種データベースアクセスを行なうことができる。Webサイトでは「ADO.NETのデータアクセス機能をパワーアップさせる基幹コンポーネント」とうたっているの、内部的にはADO.NETが利用されているのだと思うが、この製品を使うときに、開発者はADO.NETのプログラムコードを書かなくても済む。

図1：DataObjects for .NETのコンポーネントが登録されたツールボックス

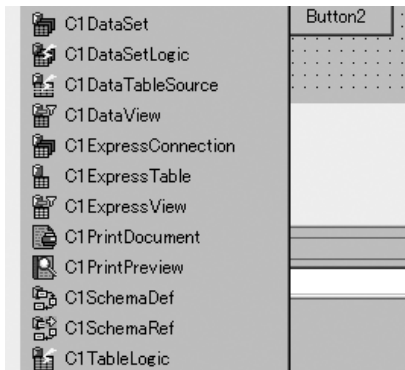
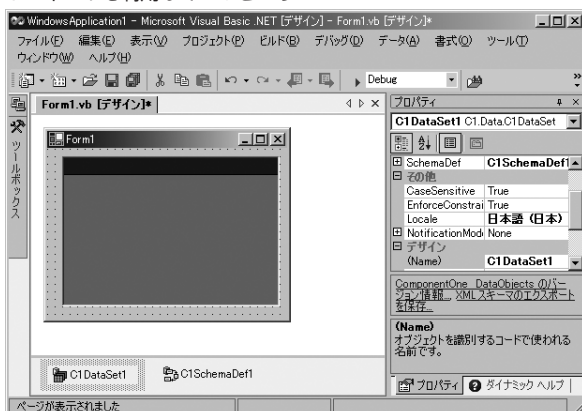


図2：VS.NETでC1SchemaDefコンポーネントとC1DataSetコンポーネントを利用しているところ



■ 利用することで得られるメリット

この製品を使うことで、ADO.NETではできなかった機能を持つプログラムができるということではない。ADO.NETと違うのは、

- ①プログラミングスタイルが簡単になる
- ②結果セットの行数が多い場合にパフォーマンスが高くなる
- ③クエリテーブル指向の設計がしやすい
- ④運用スタイルが柔軟になる

点である。これらについて、もう少し詳しくみてみよう。

■ プログラミングスタイル

DataObjects for .NETで、データベースアプリケーションを作るには、次のような手続きを行なう。

Visual Studio .NETのツールボックス(図1)でC1SchemaDefコンポーネントを選択し開発環境に配置する(図2)。

C1SchemaDefコンポーネントを右クリックしコンテキストメニューから「スキーマデザイナー」を選択し、データスキーマデザイナーを起動する(図3)。データスキーマデザイナーのインポートウィザードで動作中のデータベースに接

続し、データベースの構造を取得する。取得したデータベース構造を元にクエリビルダを使ってクエリを作成したあとでスキーマ全体を保存する。クエリビルダの使用方法は、AccessやSQL Server Enterprise Managerのクエリビルダと大きくは変わらない。ただし、生成されたSQL文はあまり強調されない。つまり「SQLを意識しなくともデータベースシステム開発ができる」というのは、DataObjects for .NETのコンセプトのひとつのようである。

続いてVisual Studio .NETのツールボックスでC1DataSetコンポーネントを選択し開発環境に配置する。C1DataSetコンポーネントのSchemaDefプロパティに先ほどスキーマを作成したC1SchemaDefコンポーネントを指定する。C1DataSetコンポーネントはデータグリッドのデータソースとなるため、データグリッドとの連結が可能となる。

このままでは指定したスキーマのすべての行を取得してしまうため、C1DataSetコンポーネントのBeforeFillイベントプロシージャで抽出条件を指定する。そのコードはたとえば、次のようなものになる。

```
e.Filter.Add(
    New C1.Data.FilterCondition(
        dataSetDef.TableViews(
```

図3：クエリをビジュアルに作成するスキーマデザイナー

