

# Oracle+.NET

## 3つの新しい



株式会社サンブリッジ テクノロジーズ  
プロジェクトマネージャ  
一志 達也 ICHISHI, Tatsuya  
<http://www.sunbridg-tech.com/>

第 5 回

## データの読み出し (接続型の場合)

### Technology Tools

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

### Level



### Samples

この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥ORACLEディレクトリに収録しています。

¥APP  
今回説明した接続型のサンプルアプリケーション

### データを 読み出す方法

今回は、ADO.NETを用いたデータ読み出しについて、「接続型」と「非接続型」の2種類があることを紹介しました。それぞれの特徴についてまとめた表を表1に再掲しておきますので、これを見て思い出しておいてください。前回も述べたように、2つの方法に明確な使い分けはありません。

今回は、接続型の場合について、具体的なプログラムの書き方を紹介します。実際にプログラムを作成してみて、自分が作成しようとするアプリケーションは、どちらの方法が適切なのかを見極めてみてください。

### 接続型の プログラム

まずは接続型について、1ステッ

プずつ、そのプログラムの書き方を紹介します。今回紹介したプログラムを活用した、サンプルアプリケーションを付録CD-ROMに収録しています(図1)。そのアプリケーションの接続型フォームについて、プログラムの内容を参照していただくといいでしょう。

接続型の場合は、OracleDataReaderオブジェクトを利用し、必要なデータだけをクライアント側に転送するのが特徴です。そのため、クライアント側のメモリやネットワークへの負荷を軽減できるのですが、反面データ処理を完了するまでは(抽出したデータが不要になるまでは)データベースとの接続を解放できません。

### 手順 1 OracleConnection オブジェクトを生成し、 データベースへの接続を確立する

とにかくにも、データベースへ接続しなくては何も始まりません

表1：接続型と非接続型の違い

方法	メリット	デメリット
接続型 (DataReader)	無駄なデータを読み込む心配がない	データベースとの接続を解放できない SQLをゴリゴリ書かなければ更新できない データベース上のメモリを圧迫する心配がある
非接続型 (DataSet)	データベースとの接続を早く解放できる メソッドを呼び出すだけで簡単に更新できる	必要のないデータも読み込む心配がある メモリを余分に消費してしまう 読み込みの完了に時間がかかる

から、OracleConnectionオブジェクトを使って接続を確立します。この部分のプログラムの書き方については、すでに説明していますので、ここでは省略して先に進みます (コラム参照)。

## 手順2 OracleCommandオブジェクトに対して、手順1で取得したOracleConnectionオブジェクトを設定する

次に必要になるのは、OracleCommandオブジェクトです。このオブジェクトの役割は、データ操作を行なうために、データベースに対してSQLを発行することです。まずは、このオブジェクトのConnectionプロパティに、利用するデータベース接続の情報をセットしましょう。

図1：接続型のデータ表示画面の例

```
Dim cmd As New OracleCommand
cmd.Connection = con
```

このとき、conはOracleConnectionオブジェクトのインスタンスを格納している変数の名前、すなわちデータベース接続を表わす変数の名前と仮定してください。

## Column データベース接続は1回で済ませる

いちいちデータベースへ接続するのは、多少なりとも時間がかかるため、パフォーマンスを悪化させる原因になります。それを回避するために、ODP.NETにコネクション(接続)プーリング機能が備わっていることは、以前に紹介したとおりです。しかし、この機能はサーバー側でプーリングするわけではないので、クライアント/サーバー型のアプリケーションでは意味をなしません (図A)。

では、いちいち切断して、必要になったら再接続するほうがいいのでしょうか。当然のごとく、その答えは“No”です。サーバー側に負担はかかりますが、一度接続したら終了するまで切断しないほうが、アプリケーションのパフォーマンス上は最適です。

そのためには、接続を扱うOracleConnectionオブジェクトを格納する変数を、アプリケーション全体で共有しなければなりません。変数を共有するには、プロジェクトにモジュールをひとつ追加作成し、その中でPublic変数を宣言します。詳しくは、付録CD-ROMに収録しているサンプルアプリケーションを参考にしてください。

図A：ODP.NETのコネクションプーリング

