Oracle Data Provider for .NETを使った RDBMS Webアプリケーション

認証とOracleへの接続／更新を学ぶ

初音 玲 *HATSUNE, Akira*

Technology Tools

- ☑ Visual Basic .NET
- ❑ Visual C# .NET
- ❑ SQL Server 2000
- ☑ Oracle 9i
- ❑ Access 2002
- ☑ ASP.NET
- ☑ Internet Information Services
- ☑ Other:
  - Oracle 9i Client R9.2.0.1.0
  - Oracle 9i Database EE R9.2.0.1.0

*) なお、記事の作成にあたってクライアントにはWindws XP Professional（SP1）、サーバーにはWindows 2000 Server（SP3）とMiracle Linux Standard Edition V2.1を使用しました。

Samples

・以下のサンプルは本誌付録CD-ROMの\DOTNET\ORADBディレクトリに収録しています。

- **WEB_NTAUTH**：NT認証
- **WEB_FRAUTH**：フォーム認証
- **WEBODPCONN**：ODP.NET接続
- **WEBMSORACONN**：MS製DataProvider接続
- **WEBOLEDBCONN**：Oracle製DataProvider接続
- **WEBODPLOGON**：ODP.NETログオン
- **WEBODPREADER**：ODP.NETデータ参照
- **WEBODPGRID**：ODP.NET一覧参照
- **WEBODPEDIT**：ODP.NET一覧更新
- **HTTP_AUTH**：認証用HTTPモジュール
- **WEBODPAUTH**：ODP.NET接続
- **DATASET_ODP**：Oracle Command Builder利用例（Winアプリ）
- **SETUP**：Webアプリのセットアップ

はじめに

インターネットによる情報発信がもてはやされた当初、企業が公開するWebサイトは商品情報や所在地などを記載したパンフレットのような静的Webページが中心であった。その後、時刻の表示やマウスカーソルの位置を取得することなどで、Webページの表示内容が変化するような動的Webページが登場した。

ただし、ここまでの技術では、あらかじめ想定していた情報以外は表示されず、また、掲示板のようにデータが変化するものも多くなってきたが、あくまでも閉じたデータの随時更新であり、基幹システムと連携して、基幹システム上を流れるデータをリアルタイムで表示するようなものではなかった。

しかし、RDBMSが一般的になり、また各RDBMSメーカーも“無制限接続ライセンス”のように、インターネットで不特定多数が使うときのライセンス体系の整備などにより、状況が変わってきた。つまり、中核にRDBMSを配置し、そのRDBMS上のデータを基幹システムでもWebサイトでも使おうという動き、いわゆるRDBMS Webアプリケーションが台頭し始めているのだ。

RDBMS Webアプリケーション

RDBMS Webアプリケーションとは、一体どのようなシステムになるのだろう。

もし、イントラネット（社内利用限定）であれば、Webサーバー上のプログラムから基幹システムのRDBMSを直接呼び出すことでデータを取得し、それをWebブラウザにHTMLとして返却するのが一般的な形態だろう。

もし、インターネットで公開するのであれば、RDBMSとWebサーバー、さらにWebブラウザの関係はイントラネットと同様だが、RDBMSを社内ではなく、DMZ（非武装地域）と呼ばれるネットワーク上の領域に設置し、インターネット側から社内の基幹システムに直接接続できないようにするのが

Oracle Data Provider for .NETを使った
RDBMS Webアプリケーション

現在のところ一般的な形態のひとつとなる。

システム構成上の違いはあるかもしれないが、イントラネットで使うWebアプリケーションとインターネットで使うWebアプリケーションにはプログラム的な相違点はない。ただし、大抵のWebアプリケーションは、ユーザーを特定するために認証を行ない、承認（認証が通った状態）した利用者のみ使用できるように作成されるため、この認証周りが重要となる。したがって、インターネットからの接続を意識したWebアプリケーションの構築をする際の認証という面に注目した時のキーポイントは、次の3つとなる。

キーポイント 1

IISは通常、統合Windows認証により利用者を承認する。統合Windows認証の場合は、ドメインのユーザー管理情報としてユーザーIDとパスワードを管理している。そのためイントラネットの中では、社内Windowsドメインにログオンしていれば、自動的に認証→承認が行なわれるので、非常に便利だ。もちろん、ASP.NETでも統合Windows認証により利用者を承認する。しかし、インターネット経由の利用者を社内Windowsドメインに登録して統合Windows認証を使う[注1]ことは、セキュリティを考えると難しいだろう。

キーポイント 2

統合Windows認証以外の認証方式の場合、ユーザーIDとパスワードをどのような形でどこに保存しておくかを決定しなければならない。

キーポイント 3

承認後、Webアプリケーションの各Webページが承認済利用者からのアクセスかどうかを判断する必要がある。

それでは、各キーポイントをASP.NETの認証方式から見てゆくことにしよう。

NT認証
Sample：Web_NTAuth

ファイルやフォルダに対してアクセス権が設定されている場合、IISは統合Windows認証により利用者を識別し、アクセス可否を判定する。ASP.NETでも、あるWebアプリケーションフォルダ配下のファイルについて統合Windows認証で承認された利用者以外はアクセスできないように設定できる。この設定を行なえば、直接URLを指定されたとしても統合Windows認証で承認されていなければ「このページを表示する権限がありません」エラーとなる。

このような設定を行なうには、Webアプリケーションのプロジェクトに含まれている“Web.config”ファイルの「authentication要素」と「authorization要素」に対してNT認証の設定を行なうだけでよい（図1）。

▶設定

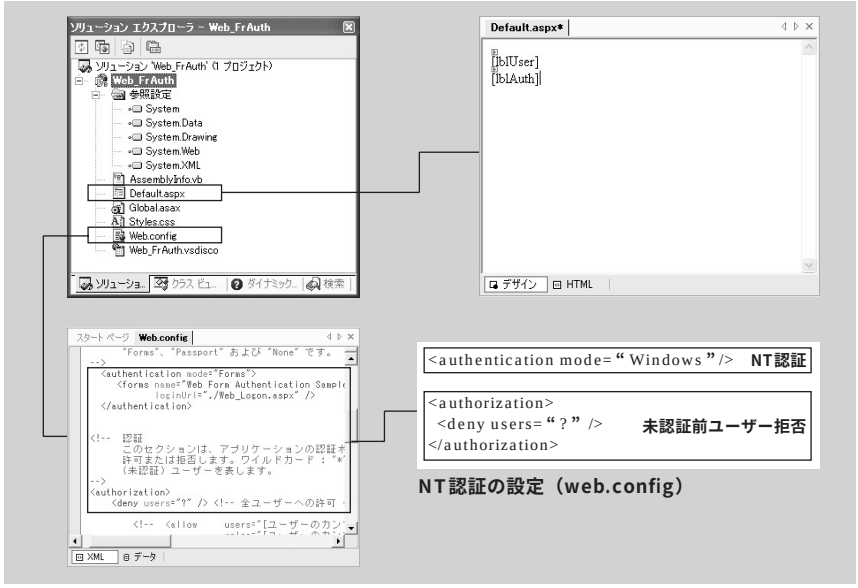
・authentication要素

Webアプリケーションの認証モードを指定する要素。ここに「Windows」と記述。

・authorization要素

Webアプリケーションの利用者の許可

図1：ASP.NETのNT認証



NT認証の設定（web.config）

注1）エクストラネットのように、限られた利用者が比較的少数であったり、VPNなどでお互いの会社のドメインを相互認証していたりする場合には、統合Windows認証でも問題はない。