

Windows アプリケーションで使う Webサービス

DataSetをクライアントに保存し、
差分だけを転送する

大澤 文孝

OSAWA, Fumitaka

Technology Tools

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

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¥F01ディレクトリに収録しています。

¥DBCREATE

サンプルDB作成用のSQL

¥PRODUCTWEBSRV

Webサービスプロジェクト

¥PRODUCTWEBAPP

Webアプリケーションプロジェクト

¥PRODUCTWINAPP

Windowsアプリケーションプロジェクト

¥PRODUCTEXCEL

Excelワークブックプロジェクト

※サンプルの利用/実行方法については、F01ディレクトリのReadme.txtを参照してください。

はじめに

Windowsアプリケーションでも、もちろん、Webアプリケーションと同じ方法で、Webサービスを使うことができます。

しかしWindowsアプリケーションは、Webアプリケーションと違って、大きなリソースを利用できます。そのため、設計上の工夫が少し違ってきます。

DataSetを クライアントに保存する

Windowsアプリケーションでは、クライアントがパソコンであるため、メモリが豊富であることを期待できます。

そのためWebサービスから得たDataSetオブジェクトをクライアントにずっと保持したままにしても、大きな問題となりません。

むしろ、いちど取得したデータは、積極的にクライアント側に保存してキャッシュとして使い、次にWebサービスからDataSetオブジェクトを取得する

ときには、以前取得したとき以降に更新された差分だけを取得したほうが、効率が良いと言えます。

また、ユーザーが編集したあと、データベースに書き戻すときには、すべてのレコードをWebサービスに送信するのではなく、変更されたレコードだけを送信するようにすることで、データ量を抑えることができます。

さらに、取得したDataSetオブジェクトをクライアント側にファイルとして保存すれば、ネットワークに接続していないオフライン状態でも、編集作業ができるように構成できます (図1)。

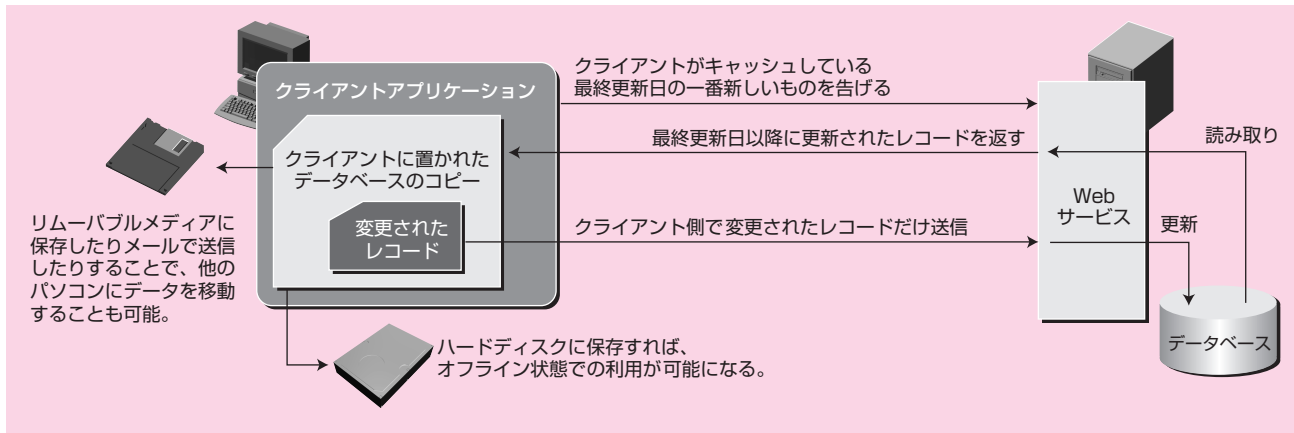
本稿のサンプル

実用的な例としては、注文を外出先で入力して、帰社してからデータベースに書き戻せるようにするサンプルというのが一番良いのですが、そうすると扱うテーブルが多くて、少し処理が難しくなります。

そこで本稿では、話を簡単にするために、図2のように、商品情報のみを編集するサンプルを考えます。

Windowsアプリケーションで使うWebサービス

図1: DataSetオブジェクトをクライアントに保存すれば、オフライン状態でも利用できる



このサンプルでは、次のように操作できるようにします。

■データベースとの同期

[同期] ボタンがクリックされたときには、クライアントのデータとデータベースのデータとを同期させます。具体的には、次の2つの処理をします。

・変更されたレコードの書き戻し

クライアントで変更されたレコードをWebサービスに送り出してデータベースを更新します。

・データベースの最新レコードを取得

以前にWebサービスから取得した日時よりもあとに更新されたレコードをデータベースから取得し、クライアント上のデータを最新の情報に更新します。

図2: 商品を編集するサンプル



■ファイルへの保存

[ファイルに保存] ボタンがクリックされたときには、DataSetオブジェクトをファイルに保存しておきます。

そして、[ファイルから読み込み] ボタンがクリックされたときには、そのファイルを読み込めるようにしておきます。

そうすることによって、ユーザーはWebサービスから得たデータを編集するのではなく、ローカルに保存されているデータを編集することができ、オフライン状態でもレコードを編集することができるようになります。

Webサービス側のメソッドを作る

では実際に、図2のアプリケーションを実装してゆきます。アプリケーションを構築するには、Webサービス側に、次の2つのメソッドが必要になります。

■指定された日時以降に変更されたレコードを返すメソッド

まずは、指定された日時以降に変更

されたレコードを返すメソッドを用意します。

ここまで、レコードの最終更新日時を、lastupdate列に記録するように実装してきました。

そこで、lastupdate列が指定された日時よりも大きいレコードだけを返すように実装します (リスト1)^[注1]。

■更新のメソッドを作る

次に、クライアント側で変更されたレコードを受け取り、それをデータベースに書き戻すためのメソッドを用意します (リスト2)。

リスト2では、SqlDataAdapterオブジェクトのUpdateメソッドを使ってデータベースを更新しています。

このとき、競合の例外が発生しないように抑制し、また、トランザクションとして実行するようにしています。

リスト2では、競合があったときに、そのエラーを伝えるため、更新後のDataSetオブジェクトをXML文字列と

注1) deleted列が1であるもの(削除済みとしたもの)も含めて返すようにしてあります。その理由は後述します。