

# 実践! 次世代



# C/S

クライアント/サーバー

# システム

スマートクライアント&Webサービスの実際

中垣 健志  
NAKAGAKI, Kenji  
株式会社CSKシステムズ  
IT生産技術部

新連載

## スマートクライアント&Webサービスの基礎



### 今こそスマートクライアント&Webサービス

蒸気機関の登場が近代社会における産業革命を引き起こしたように、インターネットの登場は現代社会における情報革命を引き起こしました。今やインターネットは電子メール、オンライン決済など企業活動において、なくてはならないものとなっています。

インターネットが爆発的に普及する前は、会社の業務を支える基幹システムはクライアント/サーバー型のアー

キテクチャで構築されていました。このアーキテクチャは、急激に安価にかつ高性能に進化していたパーソナルコンピュータに多くの機能を詰め込んだ、いわゆる「リッチクライアント」と呼ばれるものが主流でした。Windowsの操作性を存分に活かしたアーキテクチャは幅広く普及しましたが、情報システム部では普及による管理コストの増大という問題が発生しました。企業によっては何千台もの全国に散らばるコンピュータに対して、バージョンアップなどの管理作業を行わなければならなくなっ

たのです。

インターネットが普及すると、「シンクライアント」アーキテクチャが台頭してきました。これは、90年代前半にWWWの普及と共に爆発的に広まったブラウザを用いて、ユーザーインターフェイスを構築するアーキテクチャです。ほぼすべての実装がサーバー側に置かれるために、何千台マシンが存在しようとも、管理コストを低く抑えることができます。しかしブラウザを利用するシンクライアントでは、リッチクライアントに対する操作性の低下が問題となりました。情報システム部の管理コストが減る代わりに、エンドユーザーの作業時間が増えてしまったのです。

リッチクライアントのような使いやすい操作性を備え、かつシンクライアントのように管理しやすいシステムが欲しい。そんな要望に対してMicrosoftは「スマートクライアント」という回答を用意しました。スマー

レベル >>> Level

1 2 3 4 5

言語 >>> Language

Visual Basic  
C#

ツール >>> Tool

Visual Studio 2005 Professional  
SQL Server 2005

サンプル >>> Sample

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、  
<http://www.shoehisha.com/mag/windev/>  
からダウンロード可能です。

トクライアントで作成されたシステムは、クライアントに配布される「Windowsアプリケーション」とサーバー側の「業務ロジック」の2つから構成されます。そしてこの2つのプログラムは、「Webサービス<sup>[注1]</sup>」を用いて連携します。

そこで本連載では、スマートクライアント&Webサービスを使ったシステムをどのように構築していくかについて紹介していきます。



## 開発環境について

この連載では、実際の開発とほぼ同じ手順を紹介していきます。そこでVisual Studio 2005やSQL Server 2005のどのエディションを使うのか、ここで検討します。

Visual Studio 2005の廉価版であるExpress Editionでは、残念ながら今後の連載で紹介する手順を再現できません。なぜなら、同一ソリューションにWindowsアプリケーション/クラスライブラリ/ASP.NET Webサービスの3つのプロジェクトを含めることができないからです。Visual Web Developer 2005 Express Editionも併せて利用すれば同等のシステムを構築できるかもしれませんが、WindowsアプリケーションとWebサービスを同時にデバッグできないなどの不便が発生してしまいます。

また、SQL Server 2005については、Express Editionで連載の手順を十分再現できます。ただし、データベース上にテーブルやビューを作成する手順はSQL Server 2005とSQL Server 2005 Express Editionでは違いがありますので、ご注意ください。

以上を踏まえ、本連載では以下の構成を推奨します。

注1) ここでは「Webサービス」という言葉をMicrosoftが定義する「XML Webサービス」と同じ意味で使っています。XML Webサービスは「XML、HTTP、SOAPなどのインターネット標準技術を利用して、異なるプラットフォーム上のアプリケーションと統合することが可能なソフトウェアの総称」として定義されています。

注2) <マイプロジェクト>は、Visual Studio 2005のデフォルトのプロジェクト保存フォルダです。「場所」テキストボックスの右側にある「参照」ボタンをクリックすると表示されるダイアログを使って移動することができます。本連載では便宜上このフォルダにソリューションを保存することにしますが、実際は環境に合わせて別のフォルダに保存することも可能です。

- ・最小構成：Visual Studio 2005 Standard Edition + SQL Server 2005 Express Edition
- ・推奨構成：Visual Studio 2005 Professional Edition + SQL Server 2005 Developer Edition



## まずは簡単なサンプルから

### \*ソリューションとプロジェクトの作成

はじめに簡単なサンプルアプリケーションを作成します。このサンプルアプリケーションは、Webサービスに対する問い合わせを行なって結果を表示するWindowsアプリケーションプロジェクトと、クライアントからの要求を受け付けて結果を返すASP.NET Webサービスプロジェクトの2つから構成されます。Visual Studio 2005では、このように複数のプロジェクトから構成されるアプリケーションをまとめる単位として、ソリューションという概念を用意しています(図1)。

では、まずソリューションとプロジェクトを作成しましょう。

**手順①** Visual Studio 2005を起動し、メニューから「ファイル」-「新規作成」-「プロジェクト」を選択する。

**手順②** プロジェクトの種類から「その他のプロジェクトの種類」の「Visual Studioソリューション」を、テンプレートから「空のソリューション」を選択。ソリューションの情報を次のとおり設定する(図2)。設定できたら「OK」ボタンを押す。

- ・プロジェクト名：HelloSolution
- ・場所：<マイプロジェクト><sup>[注2]</sup>

**手順③** ソリューションエクスプローラ上で、作成した「HelloSolution」を右クリックし、「追加」-「新しいプロジェクト」を選択する。

**手順④** プロジェクトの種類から「Visual Basic (または Visual C#)」の「Visual Studioソリューション」を、テンプレートから「Windowsアプリケーション」を選択。プロジェクトの情報を次のとおり設