

NEW

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

PRODUCTS



iNetSuite for .NET

インターネットツール開発コンポーネント集



精進湖計算機
立中 秀樹 TATENAKA, Hideki

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

開発環境
Visual Studio .NET 2002/2003
(VB.NET/C#)

価格
パッケージ版 : 102,900円
ダウンロード版 : 81,900円

はじめに

常時接続の一般化に伴い、企業はもちろん個人でもメールやFTPを使ったデータのやり取りが頻繁に行なわれるようになり、その重要度を増しています。筆者が開発を行なうアプリケーションでも「メールやFTPの機能を組み込んでほしい」という要望が多くなっています。たとえば、FTPサーバーからファイルを定期的にダウンロードしたり、アプリケーションから遠隔地のユーザーにメール添付形式でデータを送るようになりといった内容です。また、FTTHなどの高速なインターネットの常時接続も可能となり、ネットワークにつながった遠隔地のPC間で直接通信を行なうような案件も出てきています。

当然、ネットワーク通信を行なうアプリケーションの構築には、各機能に適したプロトコルの知識とそれに対応するアプリケーション構築技術が必要になります。そして、通信系のアプリケーションは“接続”や“切断”などの通信手順やエラー時の処理などに多くの工数を取られがちです。

そこで、今回は、このような通信手

順やエラー処理などをメソッドやイベントなどで簡単に実現できる、ネットワーク通信コンポーネント集「iNetSuite for .NET」をご紹介します。

iNetSuite for .NETの概要

iNetSuite for .NETは、次の3種類のコンポーネントで構成されています。

- ① FTP for .NET : ファイル転送プロトコルとして一般に用いられるFTPをアプリケーションに実装できるコンポーネント
- ② Mail for .NET : SMTP、POP3に加え、IMAP4にも対応したメール送受信のコンポーネント
- ③ SSL Sockets for .NET : SSL認証付きのソケット通信をアプリケーションに実装できるコンポーネント

各コンポーネントの機能を簡単に表1にまとめておきます。特にソケット通信は、暗号化機能が組み込まれていて外部のネットワークを中継して通信を行なう場合でも安心して使用できます。

誌面の都合ですべてのコンポーネントについて紹介することはできません

表1: iNetSuite for .NETに含まれるコンポーネント

コンポーネント	機能	特徴
FTP for .NET	FTPによるファイル交換	<ul style="list-style-type: none"> ・自動ログオン機能 ・プロキシサーバー経由での転送 ・メモリ上での送受信 ・イベントによる転送状況表示
Mail for .NET	SMTP、POP3、IMAP4によるメールの送受信	<ul style="list-style-type: none"> ・MIMEエンコード対応 ・HTMLメールの送信 ・メールの返信と転送 ・イベントによる送受信状況表示
SSL Sockets for .NET	SSLプロトコルをサポートしたソケット通信	<ul style="list-style-type: none"> ・クライアント、サーバー双方のコントロールを用意 ・複数のデータ切り出し方法 ・マルチキャスト対応のUDPコントロール

が、今回はソケット通信を用いた簡単なチャットサーバーとクライアントアプリケーション、そしてFTPを用いてファイルをFTPサーバーに自動でアップロードするアプリケーションを作成してコンポーネントの使用感についてまとめてみたいと思います。

チャットアプリケーションの概要

さっそく、SSL Sockets for .NETのTcpオブジェクトとServerオブジェクトを用い、ソケット通信でチャットを行なうアプリケーションを構築してみましょう。

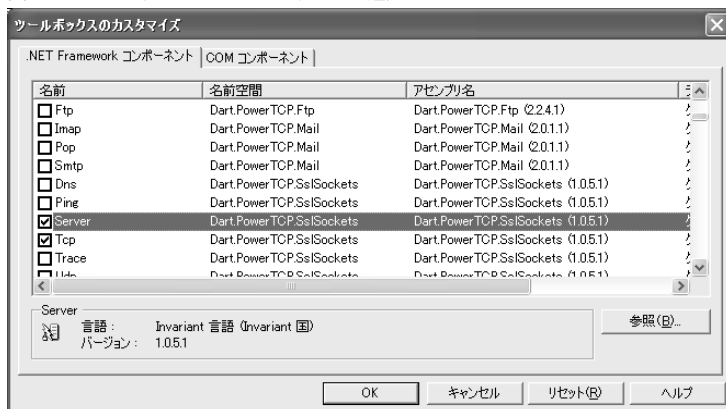
ひとつのサーバーアプリケーションがクライアントアプリケーションからの接続を待ち受ける、という構成です。サーバーアプリケーションに複数のク

ライアントアプリケーションが接続を行ない、各クライアントから送られてきた文字列を受け取ったサーバーはすべてのクライアントに返すという形をとります。

クライアントは自分自身の書いた文字列もいったんサーバーに送り、チャットログにサーバーから受け取った文字列を蓄積することで、データの送信部分と受信/ロギング部分を切り離して構築します。

はじめに、アプリケーションを構築する準備として、使用するコンポーネントを「アイテムの追加と削除」でツールボックスに追加しておきましょう(図1)。サーバーで使用するコンポーネントは「Server」、クライアントで使用するコンポーネントは「Tcp」、名前空間はDart.PowerTCP.SslSocketsです。

図1: ツールボックスにコンポーネントを追加



チャットサーバーの作成

まずは、通信を受け取り、各クライアントに配信するサーバーアプリケーションを構築します。

プロジェクトを新規作成し、ツールボックスの「Server」アイコンをフォームにドラッグ&ドロップします。このほかフォームには、以下のコントロールを配置します(図2)。

- ・新規待ち受けを開始/停止する [開始] ボタン (CmdOpen)
- ・すべての接続を切断する [中断] ボタン (CmdAbort)
- ・現在接続中のセッションを表示する リストボックス (ListBox1)
- ・クライアントが書き込んだ文字列のログを表示する リストボックス (ListBox2)
- ・現在のセッションの状態を定期的に取得するTimerコントロール (Timer1)

コーディングのポイント

続いてコーディングですが、チャットサーバーのソースコードはリスト1のようになります。ポイントとなる部分について説明してゆきましょう。

ボタンのClickイベントプロシージャ 待ち受けを開始する [開始] ボタン

図2: チャットサーバーのデザイン

