



.NET Framework 2.0には、さまざまなコントロールやクラスライブラリが追加され、以前よりも簡単にアプリケーションが作れるようになりました。

そこで本特集では、Visual C# 2005を使って、少し実用的なアプリケーションを作るには、どのようにすればよいかを説明します<sup>[注1]</sup>。

この特集で例として取り上げるのは、SMTPでの送信やPOP3での受信ができるメールソフトです (図1)。

レベル >>> Level

1 2 3 4 5

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.shoeisha.com/mag/windev/>からダウンロード可能です。

# メール送信で知っておきたいさまざまなこと

## SMTPと文字コードを理解してメール送信アプリケーションを作る

大澤 文孝 OSAWA, Fumitaka

この特集では、次の点に着目していきます。

### ① Visual Studio 2005や .NET Framework 2.0の新機能を使って楽をする

Visual Studio 2005や .NET Framework 2.0には、ユーザーインターフェイスを手早く作成する機能や、特定の処理を簡単にするクラスが備わっています。

注1) とはいえサンプルですから、使いやすさは期待しないでください。とくに今回のサンプルでは、非同期通信をサポートしていないので、メールの送受信中はアプリケーションの操作ができず、中断もできません。そのため長いメールを送受信する場合には、実用的ではありません。

言語 >>> Language

• Visual C#

ツール >>> Tool

• Visual C# 2005 Professional

ます。

それらの機能やクラスを使って、コーディング量を減らす方法を説明します。

### ② Smtplibクラスを使ってメールを送信する

.NET Framework 2.0では、SMTPを使ってメールを送信する、Smtplibクラスが新たに提供されました<sup>[注2]</sup>。

そこで、Smtplibクラスを使ってメールを送信する方法を説明します。

また、SMTP-AUTHによる認証が必要な場合の認証方法や、Smtplibクラスで送信するメールの書式の問題点についても説明します。

注2) Smtplibクラスは、従来のSmtplibクラスを置き換えるものです。SmtplibクラスはCDO (Collaboration Data Objects for Windows 2000) コンポーネントを用いてメールを送信していましたが、SmtplibクラスはCDOコンポーネントをしません。すなわち、CDOコンポーネントがインストールされていないコンピュータでも、メールを送信できます。

### ③ POP3による受信の手順を理解する

残念ながら、POP3を使ってメールを受信するためのクラスは、.NET Frameworkにはありません。

そこで、POP3プロトコルのやりとりを実装して、メールを受信する方法を示します。

TCPで通信するためには、従来から提供されているTcpClientクラスを使います。

### ④ メールをデコードする

受信したメールは、さらに、日本語のデコードや、添付ファイルの解析などの処理が必要です。

そこで受信後にデコードして、読めるようにする方法を説明します。

幸い、.NET Frameworkには、正規表現や文字コード変換、Base64デコードなどのクラスが備わっているので、それらを駆使すれば、比較的短いコードでデコード処理を実装できます。

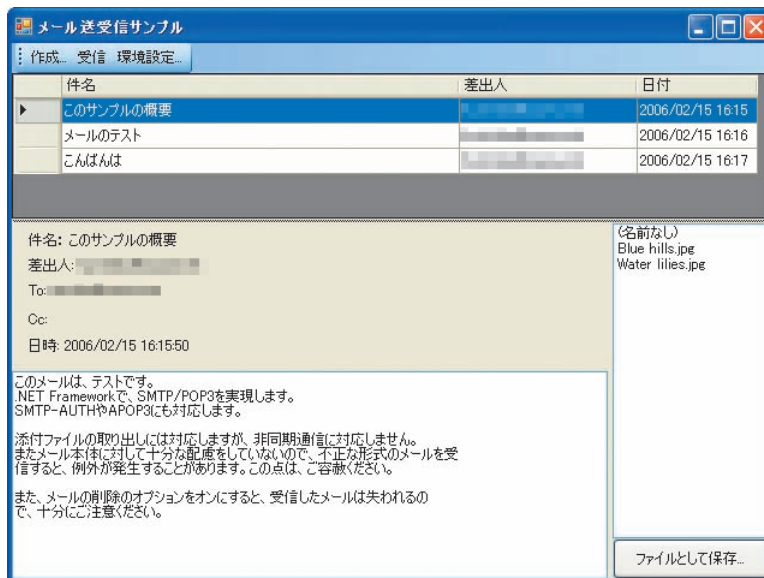
本稿では、まず①と②の部分の説明し、③以降は別稿で解説します。

## アプリケーション設定を使った環境設定

メールアプリケーションでは、「SMTPサーバー名」「メールアドレス」「POP3サーバー名」「ユーザー名」「パスワード」など、さまざまな設定情報を必要とします。

これらの設定情報は、起動するたびに入力するのではなく、ユーザーの環境情報として保存しておき、次回起動したときに、それを読み込めば、ずっ

図1：本特集で作成するメール送受信のアプリケーション



と使いやすくなります<sup>注3)</sup>。

.NET Frameworkでは、これらの情報を保存する場合、独自のファイルやレジストリなどに書き込むのではなく、「アプリケーション設定」として保存することが推奨されています。

### アプリケーションのデータとユーザーのデータ

アプリケーション設定は、XML形式のファイルです。

Windowsフォームアプリケーションの場合、アプリケーション設定のデータは、「アプリケーションデータ」と「ユーザーデータ」の2種類に分類できます。

#### ・アプリケーションデータ

アプリケーション全体の設定を示します。

アプリケーションと同じフォルダに、「アプリケーション名.config」という名前のファイルとして記述します。

たとえばMailApp.exeというアプリケーション名なら「MailApp.exe.config」

という名前になります。

このファイルに書かれたデータは、読み取り専用です。

#### ・ユーザーデータ

ユーザーごとに異なるデータを示します。「user.config」というファイル名です。

このファイルは、ユーザーごとに作成され、どの場所に保存されるのかは、アプリケーションをどのようにインストールしたかによって異なります。

ほとんどの場合、「C:\¥Documents and Settings¥<ユーザー名>\¥Local Settings¥Application Data¥<会社名>¥<製品名>¥<バージョン番号>」です。

アプリケーションの実行時には、このフォルダ名を、Application.LocalUser AppDataPathで取得できます。

注3) 「パスワード」を環境情報として保存すべきかどうかは、検討すべき事項です。本稿のサンプルでは平文でアプリケーション設定ファイルにパスワードを書き込むため、セキュリティ的に、少し問題があります。本来なら、暗号化して書き込むべきでしょう。