

# **NetMan**

**UpLoader**

**for**

**ASP**

Version 2.00

Programming Manual

***TechKnowledge***

# 目次

---

目次.....	2
はじめに.....	4
NetMan UpLoader for ASP について.....	4
開発ライセンス.....	4
サーバー・ランタイム・ライセンス.....	5
クライアント・ランタイム・ライセンス.....	5
保証規定.....	5
ユーザー・サポート.....	5
販売元.....	9
開発元、ユーザーサポート.....	9
インストール.....	10
システム条件.....	10
インストールの実行.....	10
アンインストール.....	12
サンプルの実行.....	14
サーバーコンポーネントの使用法.....	15
クライアント・コンポーネントの利用法.....	18
サーバー・コンポーネント・リファレンス.....	21
FILE オブジェクト.....	21
プロパティ.....	21
メソッド.....	22
FORM オブジェクト.....	25
プロパティ.....	25
UPLOADER オブジェクト.....	26
プロパティ.....	26
メソッド.....	31
クライアント・コンポーネント・リファレンス.....	35
FILE オブジェクト.....	35
プロパティ.....	35
FILES オブジェクト.....	36
プロパティ.....	36

メソッド .....	36
UPLOADER オブジェクト .....	37
プロパティ .....	37
メソッド .....	38
<b>FAQ - よくあるご質問 .....</b>	<b>39</b>
<b>エラー・コード一覧 .....</b>	<b>41</b>
<b>APPENDIX-A .....</b>	<b>44</b>
<b>VISUAL BASIC アップロード・サンプル・コード .....</b>	<b>45</b>
<b>NETMAN UPLOADER FOR ASP 調査依頼票 .....</b>	<b>46</b>

## はじめに

---

### NetMan UpLoader for ASPについて

このたびは、NetMan UpLoader for ASPをお買い上げくださり、まことにありがとうございます。NetMan UpLoader for ASPはhtmlフォームによるファイル・アップロード機能を簡単に実現するMicrosoft Internet Information Server用サーバー・コンポーネントです。以下は製品の特徴です。

- 複数ファイルのアップロードが可能
- アップロード・ファイルの上限サイズが設定可能
- ファイルの保存パスを指定可能
- ファイルのパスは絶対パス・相対パスで指定可能
- アップロードされたファイルのサイズ、元ファイル名などをオブジェクトを通じて取得可能
- アップロードしたファイルのコピー、移動、削除、属性設定が可能。
- 同一フォームにある- ATL 3.0で作成したサイズの小さいサーバー・コンポーネント
- ADO経由でデータ・ベースへの保存が可能
- マッキントッシュ・クライアントのサポート
- SSL環境でのアップロードに対応。
- クライアント・コンポーネントによりアプリケーション・プログラムからのアップロードに対応。
- アップロードファイルのロギングが可能。

### 開発ライセンス

開発ライセンスとは、開発者1名が1台のコンピュータ・システムで開発環境を利用することが出来る権利です。複数のコンピュータ・システムへのインストール、複数人での使用は著作権法違反となりますので、御注意ください。以下、開発者ライセンス使用権を単に使用権と記述します。

- NetMan UpLoader for ASPの使用権はいかなる方法によっても第三者に譲渡および貸与することは出来ません。
- 使用権はNetMan UpLoader for ASPパッケージを開梱したときに発効します。

- 使用権は以下のいずれかの事由が起こった場合に消滅します。

NetMan UpLoader for ASPに同封されているユーザー登録書を返送しない場合。

使用規定に違反した場合。

プログラム・ディスク、印刷物などを使用権の範囲外の目的で複製した場合。

## サーバー・ランタイム・ライセンス

本製品には1サーバー・ランタイム・ライセンスが含まれています。本製品1ライセンスをご購入になって複数のサーバーにインストールしてWebサイトを運用する場合は別途サーバー・ライセンスをご購入ください。サーバー・ライセンスの詳細につきましては販売会社までお問い合わせください。

## クライアント・ランタイム・ライセンス

本製品のアップロード用クライアント・コンポーネントはランタイム・ライセンス・フリーとします。

## 保証規定

当製品、および付随する著作物に対して商品性及び特定の目的への適合性などについての保証を含むいかなる保証もそれを明記するしないに関わらず提供されることはありません。

当製品の著作者及び、製造、配布に関わるいかなる者も、当ソフトウェアの不具合によって発生する損害に対する責任は、それが直接的であるか間接的であるか、必然的であるか偶発的であるかに関わらず、負わないものとします。それは、その損害の可能性について、開発会社に事前に知らされていた場合でも同様です。

## ユーザー・サポート

- ユーザー登録はがき  
この製品には、ユーザー登録はがきを添付しています。お買い上げのあと、できるだけ早い機会に、必要事項をご記入の上、販売会社システム・ラボまでご返送ください。このユーザー登録が行われていないと、ユーザー・サポート

が受けられない場合があります。必ずご返送をお願いいたします。

- お問い合わせの方法  
どうしても解決できない問題が発生した場合には、技術サポートをご利用ください。あらかじめ後ページの調査依頼書にお問い合わせ事項を記入していただき、それをファックス、またはインターネット・メールでお送りいただければ、折り返しご連絡をさせていただきます。当製品につきましては、複雑な内容のお問い合わせになることが多い為、電話によるユーザーサポートはいたしておりませんので、ご了承をお願いいたします。また、問い合わせの内容によっては、調査などのために、回答に時間がかかる場合がありますので、かさねてご了承をお願いいたします。
- 登録内容の変更について  
転居などによるご住所や電話番号など登録内容に変更が生じた場合には、郵送またはファックスにて、販売会社システム・ラボまでご連絡をいただきますようお願いいたします。なお、電話による口頭での連絡変更は受けかねますので、よろしくをお願いいたします。
- 併用される他社製品について  
当社製品と併用される、他社製品の使い方等についてのご質問をお受けすることがあります。しかし、他社製品に関しましては、お答えできない場合があります。他社製品につきましては、該当開発・販売会社にご連絡ください。
- サポート対象  
ご質問はご登録ユーザー様に限定させていただきます。ご登録ユーザー様以外からのご質問にはお答えできません。当ソフトウェアの料金にはご登録ユーザー様1名に限りサポート料が含まれています。途中から御担当者が変わられる場合は別途サポート料金について御相談ください。
- サポート期間  
製品のユーザー登録後、初回のお問い合わせから90日間は無償サポート期間とさせていただきます。有償サポートにつきましては販社システム・ラボにてお取り扱いしております。キャンペーン製品などディスカウント販売に該当する製品では無償サポート期間の設定が短くなる場合があります

ますのであらかじめご了承ください。

- 最新版のご提供について  
弊社webにて最新版の実行モジュールや技術情報、サンプル・コードの提供をしておりますのでサポートにご連絡になる前に弊社webをご参照いただけるようお願いいたします。URLは<http://www.techknowledge.co.jp>となります。
- ご質問の内容について  
製品サポートは当製品に関連するご質問に限定させていただきます。例えば「ASPでの文字の検索方法」や「Windows NTでのTCP/IPの設定」などをご質問されても返答いたしません。あらかじめご了承ください。
- 連名でサポート依頼される場合  
連名でサポート依頼される場合はご質問の人数分のシリアル番号を依頼用紙に明記してください。1シリアル番号に対して複数の連名でサポート依頼されるお客様、1シリアル番号で複数の開発者の方からのサポート依頼されるお客様につきましては、不正コピー使用とみなし、サポートを打ち切らせていただきます。
- サポート依頼について  
ご質問はできるだけ詳細に状況をお伝えください。「アプリケーション・エラー-xxxx:yyyyが出た。」とだけ記述してご質問なさるお客様がいらっしゃいますが、これだけではWindowsの環境では原因の診断ができません。このような場合オペレーション、ソース・コード等の詳細をご記述ください。場合によっては問題解決の為、ソース・コード等、再現可能な環境一式をお送りいただくことをお願いする場合がございます。そのような場合、お送りくださったフロッピー等は返却いたしません。ご了承ください。また、弊社サポートは処理の順序を「First Come, First Served」ベースで行っております。「至急、至急返答、納期が近い」等の記述は不要です。
- 環境変更後のエラー等について  
「xxxxというソフトをインストールしたらNetManの動作がおかしくなった」というようなご相談を最初に弊社サポートにご相談になる前に環境を変えてしまった元のソフトのサポートに先にご相談いただけますようお願いいたします。



## 販売元

### **System. Lab.**

(株) システム・ラボ

東京都杉並区上荻 1 丁目 5 番 8 号 直長ビル 7 F

電話 03-5397-7511

FAX 03-5397-7521

Internet-Mail [info@systemlab.co.jp](mailto:info@systemlab.co.jp)

URL [www.systemlab.co.jp](http://www.systemlab.co.jp)

## 開発元、ユーザーサポート

### ***TechKnowledge***

(株) テクナレッジ

東京都世田谷区駒沢 2 丁目 1 6 番 1 号 サンドービル 9 F

電話 03-3421-7621

FAX 03-3421-6691

Internet-Mail [info@techknowledge.co.jp](mailto:info@techknowledge.co.jp)

URL [www.techknowledge.co.jp](http://www.techknowledge.co.jp)

## 商標登録

本マニュアルに記載される商標、登録商標は該会社の商標または登録商標です。

## インストール

---

NetMan UpLoader for ASPのインストールについて説明します。

### システム条件

NetMan UpLoader for ASPのインストールに先立って、以下の前提となるマクロソフト製品が必要となります。

サーバー・システムとして以下のいずれか

Microsoft Windows NT 4.0 + IIS 4.0

Microsoft Windows95/98/Me + Personal WEB Server

Microsoft Windows 2000 + IIS 5.0

Windows環境のブラウザとして以下のいずれか

Microsoft Internet Explorer 5.5

Microsoft Internet Explorer 5.0

Microsoft Internet Explorer 4.0

Netscape Communicator 4.5

Macintosh環境のブラウザとして以下のいずれか

Microsoft Internet Explorer 5.0

Netscape Navigator 4.0

### インストールの実行

NetMan UpLoader for ASP 開発パッケージをフロッピーディスクからハードディスクにインストールする手順について説明します。

Windows のプログラム・マネージャーまたはエクスプローラ  
ー等から NetMan のインストールディスクットの1枚めにあ

る setup.exe を実行します。  
 setup.exe の質問に答えてインストール・ボタンをクリックし、メッセージに答えてディスクットを入れ替えると、自動的にインストールが終了します。インストールが正常に終了すると NetMan のプログラム・グループが作成されます。  
 NetManUploader.TXT ファイルにはマニュアルに記述されていない最新情報が記述されています。最新情報が記述される場合もありますので、必ずご一読ください。

## 製品モジュール一覧

---

Windows のインストール・ディレクトリを c:\%windows, NetMan 製品のインストール・ディレクトリを c:\%NetMan とした場合に、本製品がインストールされるファイルの一覧を示します。

モジュール名とパス	内容
c:\%winnt%\system32\nmUpload.dll	サーバーコンポーネント
c:\%NetMan%\nmUpload.pdf	当マニュアル
c:\%NetMan%\nmUpload.htm	README ファイル
c:\%NetMan%\regsvr32.exe	OLE 登録ユーティリティ
c:\%NetMan%\Samples\*. *	サンプル・ファイル
C:\%NetMan%\nmUpload.reg	ログ情報開始用レジストリ・データ
C:\%winnt%\system32\nmHttpUploader.ocx	クライアントコンポーネント

## イベントログ設定

---

システムにアップロード・イベントを記録する場合は regedt32.exe を実行して「復元」メニューから nmUpload.reg ファイルを指定してレジストリにデータを登録する必要があります。ファイル・アップロード・ログを通常のテキスト・ファイルに出力する場合はこの設定は必要ありません。

## アンインストール

---

### 自動アンインストール

コントロール・パネルの「アプリケーションの追加と削除」メニューから NetMan Uploader for ASP を選択することでアンインストールが可能です。以下は操作手順です。

「設定」メニューから「コントロール・パネル」を選択  
アプリケーションの追加と削除をダブル・クリック  
「セットアップと削除」タブのリスト・ボックスから「NetMan Uploader for ASP」を選択  
「追加と削除」ボタンをクリック  
「追加と削除」ボタンをクリック  
モジュール名を表示して削除を質問される場合は、NMRAS.OCX の削除を指定。それ以外は削除しないでください。

### 手動アンインストール

間違えてインストールしたフォルダーを削除した場合や、上書きインストールして自動アンインストール出来なくなった場合には以下の手動でアンインストールしてください。

MS DOS プロンプトを起動します。  
インストール・ディレクトリに移動します。デフォルト・インストールでは c:\Program Files\techknowledge\NetMan Uploader for ASP となります。  
regsvr32 /U c:\winnt\System32\nmupload.dll を実行します。この操作でレジストリから情報が削除されます。ディレクトリは WindowsNT のデフォルト・インストール時のものです。別ディレクトリにインストールしている場合は適宜変更してください。  
システム・ディレクトリからモジュールを削除します。  
Del c:\winnt\system32\nmloader.dll  
インストール・ディレクトリを削除します。  
rmdir /s c:\Progra~1\nmUpload  
を実行してください。  
NetMan メニューを削除します。ショートカットなどを作成した場合も同様に削除します。  
イベント・ログ設定をした場合には以下のキーを削除します。  
HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Application\nmUpload

以下はクライアント・コンポーネントの手動アンインストール方法です。

MS DOS プロンプトを起動します。

Windows NT/2000 の場合は `regsvr32 /U c:%windnt%\System32\nmHttpUploader.ocx` を実行します。この操作でレジストリから情報が削除されます。ディレクトリは Windows NT/2000 のデフォルト・インストール時のものです。別ディレクトリにインストールしている場合は適宜変更してください。

Windows 9x/Me の場合は `regsvr32 /U c:%windows%\System\nmHttpUploader.ocx` を実行します。この操作でレジストリから情報が削除されます。ディレクトリは Windows 9x/Me のデフォルト・インストール時のものです。別ディレクトリにインストールしている場合は適宜変更してください。

## サンプルの実行

---

この章では NetMan Uploader for ASP に添付されるサンプルの実行方法を説明します。

NetMan UpLoader for ASP には以下のサンプルが添付されます。  
IIS の Web ディレクトリにコピーして拡張子.html のファイルを指定して実行します。

ファイル名	概要
Sample1.html	複数ファイルのアップロードを指定するフォームの例です。実行する前にはファイルを置いたディレクトリの下に upload という名前であらかじめディレクトリを作成してください。
Sample1.asp	Sample1.html のデータを処理して、コメントとファイル名をテーブルに表示します。
UploadImage.html	GIF/JPEG ファイルをアップロードして表示するサンプルです。
ShowImage.asp	UploadImage.html のデータを処理します。
ShowImage2.asp	GetBinaryData メソッドのサンプルです。
Attribute.inc	ファイルの属性値を定義したファイルです。
VBSample.vbp	Visual Basic 用クライアントアップロード・サンプル・プロジェクト
frmClientUpload.frm	Visual Basic 用サンプル・フォーム・ファイル

## サーバーコンポーネントの使用法

---

### アップロード・フォームの作成

---

Web サイト利用者がファイルをアップロードする html form を作成します。html form には ENCTYPE 指定で"multipart/form-data"を指定します。この指定は rfc1867 に準拠した形式でデータをアップロードすることを指定していますので必須になります。アップロードするファイルは<input type="file">タグで指定します。以下はサンプルになります。

```
<form method=post enctype="multipart/form-data" action="save.asp">  
アップロードファイル:<br>  
<input type="file" name="datafile"></input>  
<br>  
<input type="submit" value="アップロード"></input>  
</form>
```

### サーバー・サイド・スクリプトの作成

---

作成した html フォームからポストされる ASP ファイルを作成します。このファイルでは NetMan オブジェクトの作成、動作モードの指定、ファイルの保存を実行します。ファイルの保存パスは FilePath プロパティで指定します。UseVirtualPath を指定した場合、Server.MapPath の結果によりファイル保存位置を相対指定することも可能になります。

Save.asp ファイルのスクリプト部分は以下のようになります。

```
<%  
Dim NetMan, rc  
  
Set NetMan = Server.CreateObject("NetMan.Uploader.1")  
With NetMan  
    .MaxSize = 1000  
    .Truncate = False  
    .FilePath = c:¥company¥profile¥data1"  
    rc = .SaveFile  
    If rc <> 0 Then
```

```
        Response.Write "ファイル保存に失敗しました " & CStr(rc)
    End If
End With
Set NetMan = Nothing
%>
```

## ファイル情報の取得

---

html form から指定されたファイル名、サーバー側での保存状態やサイズなどを File オブジェクトにより取得することができます。以下はサンプル・スクリプトです。

```
<%
Dim f

For Each f in NetMan.Files
    Response.Write f.FileName
    If f.Status = 0 Then
        Response.Write "は正常に保存されました。<br>"
    Else
        Response.Write "は保存されませんでした。<br>"
    End If
Next
%>
```

## フォーム情報の取得

---

アップロードする html form に定義される<input>タグの値は NetMan の Forms コレクションを通じて取得することができます。以下はサンプル・コードです。

```
<%
Dim f

For Each f In NetMan.Forms
    Response.Write f.Name & " = "
    Response.Write f.Value
    Response.Write "<br>"
Next
%>
```

また以下のような形式でhtml form上のinputタグのNameの値を指定してValueを取得することも可能です。

```
<%  
With Response  
  .Write "input FullNameの値は " & NetMan.Form("FullName")  
  .Write "になります<br>"  
End With  
>%
```

## クライアント・コンポーネントの利用法方

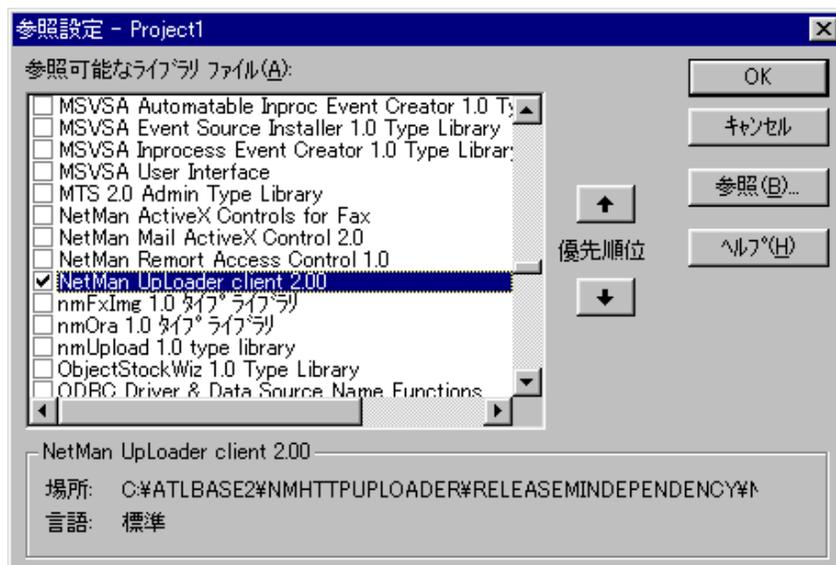
---

クライアントコンポーネントのご利用方法について説明いたします。  
ここでは代表的な ActiveX Control ホスト言語として Visual Basic 6.0 を中心にご説明いたします。Visual Basic 以外の言語につきましてはのご利用方法は各言語マニュアル等の ActiveX Control または OLE オートメーションの利用方法をご参照ください。

### 参照設定

---

参照設定によりクライアント・コンポーネントを利用可能とします。



(上記画像の場所：につきましてはインストールする OS やアプリケーションの配布方針によって異なります)

### クライアントオブジェクトの宣言

---

Visual Basic では以下のように File オブジェクトと uploader オブジェクトを宣言します。

```
Dim fs As New NMHTTUPLOADERLib.Files  
Dim uploader As New NMHTTUPLOADERLib.uploader
```

## 送信ファイルのセットアップ

---

Files オブジェクトには最初に送信するファイル数を Allocate メソッドにて通知します。

```
Call fs.Allocate(2)
```

次にアップロードするファイルを指定します。

```
fs.Item(0).FileName = "c:¥tmp¥greeting.txt"  
fs.Item(0).Name = "upload1"  
fs.Item(1).FileName = "c:¥tmp¥thanks_letter.txt"  
fs.Item(1).Name = "upload2"
```

## ファイルのアップロード

---

以下は URL を指定してアップロードを実行するサンプル・コードです。

```
With uploader  
  .URL = "http://localhost/Sample1.asp"  
  .Files = fs  
  rc = .Upload  
  If rc <> 0 Then  
    MsgBox CStr(rc)  
  Else  
    Debug.Print .Response  
  End If  
End With
```

```
Set fs = Nothing  
Set uploader = Nothing
```

コード全体は Appendix-B に掲載してありますのでご参照ください。

## フィールド値のアップロード

---

ファイルと Text フィールドの値が混在しているフォームに対してアップロードすることも可能です。フィールド値は Value プロパティに文字列で指定します。Value プロパティを使う場合は FileName プロパティ

イには空文字列(“”)を指定してください。

```
Call fs.Allocate(2)
fs.Item(0).FileName = "c:¥tmp¥invoice.doc"
fs.Item(0).Name = "upload1"
fs.Item(1).FileName = ""
fs.Item(1).Name = "comment1"
fs.Item(1).Value = "特にコメントはありません"
```

## アップロード結果の取得

---

アップロード結果は Upload メソッドからのリターン・コードが 0 の場合でもファイル単位でアップロード出来ていない場合がありますので注意が必要です。Upload メソッドからの戻り値は主に通信エラーなどが返されます。ファイル単位のアップロード結果は Sent プロパティを参照することで可能です。また http プロトコル・レベルのエラーは Response プロパティに返されます。URL で指定したサーバーの asp ファイルが存在しない場合などは Response プロパティにエラーが返ります。

## クライアント・コンポーネントの配布

---

クライアント・コンポーネント実行時に必要なファイルは nmHttpUploader.ocx になります。通常はシステム・ディレクトリ等に配布します。依存するファイルは以下のようになります。アプリケーションと一緒に配布するにはインストーラーで一般的な ActiveX Control として配布することで問題無く配布できますが前提として Internet Explorer 4.0 以降が必要になります。レジストリへの登録は regsvr32.exe を使っても可能ですが通常はインストーラーが自動的にレジストリ登録を実行しますので特別な処理等は不要です。

```
SHLWAPI.dll
WININET.dll
OLEAUT32.dll
ole32.dll
ADVAPI32.dll
GDI32.dll
USER32.dll
KERNEL32.dll
```

## サーバー・コンポーネント・リファレンス

---

この章では NetMan UpLoader for ASP サーバー・コンポーネントのプロパティ、メソッド、イベントについての説明をします。

### File オブジェクト

#### プロパティ

##### FileName

---

###### 概要

サーバー側に保持されたファイル名を保持します。

###### データ型

String

##### FileSize

---

###### 概要

サーバー側に保持されたファイルサイズを保持します。Uploader オブジェクトの Truncate プロパティの設定が True の場合はオリジナルのサイズが保持されます。

###### データ型

Long

##### Status

---

###### 概要

ファイルの保存状態を保持します。

###### データ型

Long

###### 値

0	正常に保存された
1	ファイルサイズが MaxSize を超えたため保存されなかった。
2	ファイルサイズが MaxSize を超えたため MaxSize まで保持された。
4	サーバーの同一ファイル名のため改名されて保存された。

上記の値は論理和を取って設定されることがあります。例えば、Truncate=True で MaxSize 設定がされている状態で、同一ファイルが存在すると値 6 が Status プロパティに設定されます。

## UserFileName

---

### 概要

ユーザーが指定したファイル名を保持します。フルパス指定の場合、パスはクライアント・パソコンのパスになります。

### データ型

String

## メソッド

### Copy

---

### 概要

アップロードされたファイルを指定されたパスにコピーします。

### 書式

*Object*.Copy( *NewPath* As String ) As Integer

### パラメータ

NewPath	コピー先のファイル・パス
---------	--------------

### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

## Delete

---

### 概要

アップロードされたファイルを削除します。

### 書式

*Object.Delete* () As Integer

### パラメータ

なし

### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

## GetAttribute

---

### 概要

アップロードされたファイルの属性を取得します。

### 書式

*Object.GetAttribute*() As Long

### パラメータ

なし

### 戻値

ファイルの属性値が返ります。値の詳細は Appendix-A をご参照ください。

## GetBinaryData

---

### 概要

アップロードされたファイルの内容を取得します。

### 書式

*Object*.GetBinaryData ( ) As Variant

#### パラメータ

なし

#### 戻値

ファイルの内容が 1 バイト・バイナリの配列として返ります。

### InsertToDb

---

#### 概要

アップロードされたファイルを ADO 経由でデータ・ベースに保存します。

#### 書式

*Object*.InsertToDb( *ConnectStr* As String,  
*Sql* As String,  
*Col* As String ) As Long

#### パラメータ

ConnectStr	ADO のデータ・ベース接続文字列。事前にアップロード・オブジェクトの ConnectToDb メソッドを使って接続をする場合はこのパラメータにはヌル文字列を指定します。
Sql	インサートを実行する sql 文。イメージをバインドするカラムは?をセットします。
Col	イメージを格納するカラム名

#### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

### Move

---

#### 概要

アップロードされたファイルを指定されたディレクトリに移動し

ます。

### 書式

*Object*.Move ( *NewPath* As String ) As Long

### パラメータ

NewPath	移動先のファイル・パス
---------	-------------

### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

## SetAttribute

---

### 概要

アップロードされたファイルの属性を設定します。

### 書式

*Object*.SetAttribute( *NewAttr* As Long ) As Integer

### パラメータ

NewAttr	設定するファイル属性。設定可能な値につきましては Appendix-A をご参照ください。
---------	---

### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

## Form オブジェクト

### プロパティ

Name

---

### 概要

アップロード html フォームに設定された<input>タグの name の値です。参照のみ可能なプロパティです。

### データ型

String

Value

---

### 概要

アップロード html フォームに設定された<input>タグの Value 値です。通常はブラウザから入力した値が設定されます。参照のみ可能なプロパティです。

### データ型

String

## Uploader オブジェクト

### プロパティ

DbErrorText

---

### 概要

ADO 経由でデータ・ベースの操作に失敗した場合、ターゲット・データ・ベースからのエラー・メッセージが保持されます。

### データ型

String

FilePath

---

### 概要

アップロードされたファイルを保存するパスを指定します。指定するパスは存在する必要がありさらにサーバーの IIS 実行プロセスがアクセス可能なディレクトリを指定します。

### データ型

String

## Files

---

### 概要

アップロードされたファイルに関する情報を保持する File オブジェクトのコレクションです。

### データ型

File オブジェクト

## Form

---

### 概要

アップロード指定された html フォームにある<input>タグの値を参照可能とするプロパティです。数値をパラメータとする場合は、Form オブジェクトを返します。ベースは1となります。

### データ型

String

### サンプル・コード 1

```
ExchangeRate = Uploader.Form("ExchangeRate")
```

### サンプル・コード 2

With Response

```
.Write Uploader.Form(1).Name & "<br>"
```

```
.Write Uploader.Form(1).Value & "<br>"
```

End With

## Forms

---

### 概要

アップロード指定された html フォームの<input>タグに関する情報のコレクションです。

### データ型

Form オブジェクト

### サンプル・コード

```
Dim f
```

```
For Each f In Uploader.Forms
    Response.Write f.Name & " = "
    Response.Write f.Value & "<br>"
Next
```

## LastOSError

---

### 概要

Win32 API 呼出失敗時のシステム・エラー・コードを保持します。具体的な値は Win32 API の GetLastError からのものであります。詳細はマイクロソフトの Platform SDK 等にありますのでご参照ください。

### データ型

Long

## LogFileName

---

### 概要

ログをファイルに保存する場合はログファイル名をフル・パスで設定します。このパス及びファイルには IIS の実行時オーナーが書き込み可能に設定する必要があります。ログ・ファイルにはアップロードされた日付け、時間、ファイル名等がテキスト・ファイル形式で保存されます。当プロパティにヌル文字列を設定した場合はログは出力されません。

### データ型

String

## Logging

---

### 概要

ログの方法を指定します。当プロパティの設定によりテキストファイルまたはシステムのイベント・ログにファイル・アップロード・ログを記録することが出来ます。

プロパティ値	ログ処理
0	ログ出力しません。

1	アップロード・ログを LogFile Name プロパティで指定されるファイルに出力します。
2	アップロード・ログをシステム・イベントログに出力します。事前にレジストリ・設定を実行しておく必要があります。

### データ型

Integer

MaxSize

---

### 概要

アップロード・ファイルの最大サイズをバイト単位で指定します。

### データ型

Long

OverWrite

---

### 概要

アップロード・ディレクトリに同一ファイル名が存在する場合の処理方法を指定します。True 設定の場合は同じファイル名で上書きされます。False 設定の場合は新しくファイル名が生成され保存されます。

ファイル名の生成ルールは以下のようにファイルの拡張子の前に通し番号を添付する形式になります。

Filename.serial.extension

例えば a.txt がすでに存在する場合、重複したファイルは、

a.1.txt

と新しく名前が割り振られます。

a.txt

a.1.txt

が存在する場合にまた a.txt がアップロードされたときには、a.2.txt という名前で新たに保存されます。結果としてアップロード・ディレクトリには以下の 3 ファイルが保持されます。

a.txt  
a.1.txt  
a.2.txt

データ型  
Boolean

## Truncate

---

### 概要

MaxSize プロパティで指定されたファイル・サイズを持つファイルについてアップロード処理方法を指定します。True に設定した場合はファイルは MaxSize までのデータが保存されます。False 設定とした場合にはファイルは保存されません。SaveFile メソッドにて保持された File オブジェクトの Status プロパティに保存状態が記録されます。

データ型  
Boolean

## VirtualPath

---

### 概要

True 設定の場合 Upload オブジェクトの FilePath プロパティの指定を相対パスとみなします。パスのマッピングは IIS の Server オブジェクトの MapPath に従います。False 設定の場合、Upload オブジェクトの FilePath プロパティは絶対パスとみなします。

データ型  
Boolean

### サンプル・コード

```
Dim NetMan
```

```
Set NetMan = Server.CreateObject("NetMan.Uploader.1")
```

```
With NetMan
    .VirtualPath = True
    .FilePath = "images"
    .SaveFile
End With
```

上記のコードが c:\inetpub\wwwroot\sample で実行された場合、クライアントからのファイルは c:\inetpub\wwwroot\samples\images に保存されます。

## メソッド

### ConnectToDb

---

#### 概要

データ・ベースに接続します。当製品では File オブジェクトの InsertToDb メソッドでアップロードしたデータをデータ・ベースに格納することができますが、InsertToDb メソッドの第一パラメータの接続文字列を指定した場合には、InsertToDb メソッドを発行する毎にデータ・ベースと接続が確立され、データを登録後にデータ・ベースから遮断されます。同じ接続先に複数のデータを格納するような場合、何度も接続・遮断を繰り返すのはパフォーマンスの問題が発生することが考えられます。そのような場合にはこのメソッドを使って事前にデータ・ベースに接続することで、接続・遮断の回数を減らすことができます。このメソッドを使って事前に接続を実行した場合には InsertToDb メソッドの第一パラメータはヌル文字列を指定します。

#### 書式

*Object*.ConnectToDb(*ConnectStr* As String) As Integer

#### パラメータ

ConnectStr	ADO の接続文字列を指定します。
------------	-------------------

#### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

### DisconnectFromDb

---

### 概要

データ・ベースとの接続を遮断します。

### 書式

*Object.DisconnectFromDb* () As Integer

### パラメータ

なし

### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

## Copy

---

### 概要

サーバー上のファイルを指定されたパスにコピーします。

### 書式

*Object.Copy*(*FromFile* As String,*NewPath* As String) As Integer

### パラメータ

FromFile	コピー元のファイル名。VirtualPath=True の場合は相対パス指定になります。
NewPath	コピー先のファイル・パス。VirtualPath=True の場合は相対パス指定になります。

### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

## Delete

---

### 概要

指定されたファイルを削除します。

#### 書式

`Object.Delete( FileName As String ) As Integer`

#### パラメータ

FileName	削除するファイル名。VirtualPath=True の場合は相対パス指定となります。
----------	---

#### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

### GetAttribute

---

#### 概要

指定されたファイルの属性を取得します。

#### 書式

`Object.GetAttribute( FileName As String ) As Long`

#### パラメータ

FileName	属性を取得するファイル名。VirtualPath=True の場合は相対パス指定となります。
----------	--

#### 戻値

ファイルの属性値が返ります。値の詳細は Appendix-A をご参照ください。

### Move

---

#### 概要

指定されたファイルを指定されたディレクトリに移動します。

## 書式

```
Object.Move( FileName As String,NewPath As String ) As Long
```

## パラメータ

FileName	移動するファイル名の指定。VirtualPath=True の場合は相対パス指定となります。
NewPath	移動先のファイル・パス。VirtualPath=True の場合は相対パス指定となります。

## 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

## SetAttribute

---

## 概要

アップロードされたファイルの属性を設定します。

## 書式

```
Object.SetAttribute( FileName As String, _  
NewAttr As Long ) As Integer
```

## パラメータ

FileName	属性を設定するファイル名。VirtualPath=True の場合は相対パス指定となります。
NewAttr	設定するファイル属性。設定可能な値につきましては Appendix-A をご参照ください。

## 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

## クライアント・コンポーネント・リファレンス

---

### File オブジェクト

#### プロパティ

FileName

---

##### 概要

アップロードするファイルを指定します。フルパス指定しない場合はアプリケーションの実行時ディレクトリからの相対パスになります。

##### データ型

String

Name

---

##### 概要

サーバー側で参照する<input>タグの name の値を設定します。

##### データ型

String

Sent

---

##### 概要

ファイルまたはフィールド値の送信が完了した場合には True がセットされます。True にセットされるのは Upload メソッド実行後になります。

##### データ型

Boolean

Value

---

### 概要

アップロードするフィールド値を指定します。

### データ型

String

## Files オブジェクト

### プロパティ

Item

---

### 概要

当オブジェクトに含まれる File オブジェクトです。配列でアクセスします。配列はゼロ・ベースとなります。

### データ型

File Object

### メソッド

Allocate

---

### 書式

*Object*.Allocate( *NumOfItem* As Long ) As Integer

### パラメータ

NumOfItem	アップロードするファイルとフィールドの合計数を設定します。
-----------	-------------------------------

### 概要

Files オブジェクトを利用する場合に最初に呼び出してアップロードするファイル数を設定します。当メソッド内部ではリソースを配置しアップロード情報を保持する準備をします。

### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・

リファレンスをご参照ください。

## Uploader オブジェクト

### プロパティ

#### Files

---

##### 概要

送信するファイル、フィールド情報を設定した Files オブジェクトを設定します。

##### データ型

Files Object

#### LastError

---

##### 概要

Upload メソッドがエラーを返した場合に Win32 API レベルのエラー・コードを保持します。

##### データ型

Long

#### Response

---

##### 概要

http/https で送信後、サーバーからのレスポンスが保持されます。

##### データ型

String

#### URL

---

##### 概要

送信先の URL を指定します。http または https で始まる URL を指定します。

##### データ型

String

## メソッド

Upload

---

### 書式

*Object*.Upload() As Integer

### パラメータ

なし

### 概要

http/https アップロードを実行します。

### 戻値

正常終了時には 0 が返ります。それ以外の値はエラー・コード・リファレンスをご参照ください。

## FAQ – よくあるご質問

---

SaveFile の結果 1003 のエラーになる。

---

Microsoft Internet Explorer 5.x からファイルをアップロードする場合、<input>タグに name まで定義しないと、正常な形式でデータがアップロードされません。かならず<input>タグには name = "xxxx" の形式で name の定義をしてください。

### InsertToDb のサンプル・コード

---

以下は ADO を使って Oracle データ・ベースにデータを登録するサンプルです。

```
Dim x As Object, rc As Integer, fl
Dim sql As String, conn As String

conn = "Provider=MSDAORA;Data Source=srv1;User ID=scott;Password=tiger;"

Set x = CreateObject("NetMan.Uploader.1")
x.OverWrite = False
x.Truncate = True
x.MaxSize = 0
x.FilePath = "c:¥temp"
rc = x.SaveFile

If rc = 0 Then
  For Each fl In x.Files
    sql = "insert into imagetest (fname,image) values ('"
    & fl.FileName & "',?)"
    rc = fl.InsertToDb(conn, sql, "image")
    If rc <> 0 Then
      Response.Write "err " & x.DbErrorText
    End If
  Next
End If
Set x = Nothing
```

## ADO 2.5 と Oracle 8.1.x の組み合わせで接続に失敗する

---

ADO 2.5 は Oracle 8.1.x の oci.lib とリンクしていないことが原因と推測されますが、Oracle への接続に失敗します。毎回接続に失敗することもあります。まれに正常に接続できることもあるようです。ADO が Oracle 8.1.x に対応することを待つしかないと思います。(2000/05/19 現在)

## エラー・コード一覧

---

NetMan UpLoader for ASP メソッドが返すエラー値についての説明です。サーバー・コンポーネントとクライアント・コンポーネントで別々の値を返すことにご注意ください。

### サーバー・コンポーネント・エラーコード

---

値	シンボル	意味
1001	ERR_NO_MEMROY	メモリ不足。ローカル・ヒープに必要なメモリが確保できませんでした。
1002	ERR_SEPARATOR	アップロードされるデータのセパレータが不正でした。Html form の enctype 指定が正しいことをご確認ください。
1003	ERR_UNKNOWN_FORMAT	未知の形式です。ブラウザがサポートされる製品バージョンであることをご確認ください。また、html form の enctype 指定が正しいことをご確認ください。ブラウザによっては<input タグに name= の指定が必須となりますのでご確認ください。
1004	ERR_ARRAY_ACCESS	配列データにアクセスすることができません。他のプロセスが同時にアクセスしている可能性があります。
1005	ERR_OPEN_FILE	ファイルがオープンできません。エラーの詳細は LastOSError プロパティに保持されています。
1006	ERR_DELETE_FILE	ファイルが削除できません。エラーの詳細は LastOSError プロパティに保持されています。
1007	ERR_COPY_FILE	ファイルをコピーできません。エラーの詳細は LastOSError プロパティに保持されています。
1008	ERR_MOVE_FILE	ファイルを移動できません。エラーの詳細は LastOSError プロパティに保持されています。

1009	ERR_SET_ATTRIBUTE	ファイルの属性が設定できません。エラーの詳細は LastOSError プロパティに保持されています。
1010	ERR_GET_ATTRIBUTE	ファイルの属性を得ることができません。エラーの詳細は LastOSError プロパティに保持されています。
1011	ERR_NO_LICENSE	プログラムを実行するライセンスがありません。正常にパッケージをインストールしていないものと思われる。
1012	ERR_PARAMETER	メソッドを呼び出す際に必要なパラメータが設定されいません。またはパラメータのデータ型に問題があります。
1013	ERR_FILE_LOAD	内部的に保持していると思われたファイルのロードに失敗しました。アップロード・ディレクトリにあるファイルが消去されたり、別のプロセスでオープンされている可能性があります。
1014	ERR_CREATE_INSTANCE	ADO の呼出に必要な OLE オブジェクトの生成に失敗しました。ADO が正常に動作する環境を整えてください。
1015	ERR_DB_HANDLING	データ・ベース操作に失敗しました。データ・ベースからのエラー・メッセージは DbErrorText プロパティに保持されている場合がありますので、そちらの値を参照して sql 文や接続文字列を訂正してください。
1016	ERR_ALREADY_CONNECTED	ConnectToDb メソッドがすでに接続されている状態で再度呼び出されました。プログラミングのミスと思われるので、今一度ロジックの見直しをしてください。

クライアント・コンポーネント・エラー・コード

100	ERR_EXCEPTION	ソフトウェア例外が発生しました。Wininet.dll のバージョン等を添えてサポートにご相談ください。
101	ERR_SEND_REQUEST	サーバーに send request を発行できませんでした。LastError プロパティの値を Win32 エラー・コードと照らし合わせて原因を特定してください。
102	ERR_END_REQUEST	サーバーに end request を発行できませんでした。LastError プロパティの値を Win32 エラー・コードと照らし合わせて原因を特定してください。
103	ERR_OPEN_URL	URL プロパティで指定された URL に接続が出来ませんでした。LastError プロパティの値を Win32 エラー・コードと照らし合わせて原因を特定してください。
104	ERR_OPEN_INET	Wininet 環境が正常に設定されていないと思われます。クライアントにインストールされている Internet Explorer のバージョンや TCP/IP プロトコルの設定等をご確認ください。
105	ERR_WRITE_FILE	通信中にデータを出力できないようです。通信エラーの詳細につきましては LastError プロパティをご参照ください。
106	ERR_NO_MEMORY	一時的にデータを移動させるためのメモリ領域が不足しています。他のアプリケーションが多く実行されているような場合はそれらを終了させてメモリを空けてから再実行してください。

## Appendix-A

---

ファイルの属性値は以下のような値になります。

シンボル	値 (hex)	意味
FILE_ATTRIBUTE_ARCHIVE	20	アーカイブファイルまたはアーカイブディレクトリです。この属性は、ファイルのバックアップや削除のためのマークとして使われます。
FILE_ATTRIBUTE_COMPRESSED	800	圧縮ファイルまたは圧縮ディレクトリです。圧縮ディレクトリ内に新しく作られるファイルまたはディレクトリは、デフォルトで圧縮状態になります。
FILE_ATTRIBUTE_DIRECTORY	10	ディレクトリです。
FILE_ATTRIBUTE_HIDDEN	02	隠しファイルまたは隠しディレクトリです。
FILE_ATTRIBUTE_NORMAL	80	特に属性はありません。
FILE_ATTRIBUTE_OFFLINE	1000	ファイルのデータは、すぐには利用できません。ファイルのデータが別のオフラインの記憶装置に移されていることを示します。
FILE_ATTRIBUTE_READONLY	01	読み取り専用です
FILE_ATTRIBUTE_SYSTEM	04	オペレーティングシステムのファイルの一部です。または、オペレーティングシステム専用のファイルです。
FILE_ATTRIBUTE_TEMPORARY	100	テンポラリファイルです。

サンプル・ディレクトリには上記の Const 宣言ファイルが保存してありますのでご利用ください。ASP ソースにインクルードするには以下のように記述をします。

```
<!-- #INCLUDE FILE="attribute.inc" -->
```

## Visual Basic アップロード・サンプル・コード

```
Dim fs As New NMHTTUPLOADERLib.Files
Dim uploader As New NMHTTUPLOADERLib.uploader

Call fs.Allocate(2)
fs.Item(0).FileName = "c:¥tmp¥greeting.txt"
fs.Item(0).Name = "upload1"
fs.Item(1).FileName = "c:¥tmp¥thanks_letter.txt"
fs.Item(1).Name = "upload2"

With uploader
    .URL = "http://localhost/Sample1.asp"
    .Files = fs
    rc = .Upload
    If rc <> 0 Then
        MsgBox CStr(rc)
    Else
        Debug.Print .Response
    End If
End With

Set fs = Nothing
Set uploader = Nothing
```

## NetMan Uploader for ASP 調査依頼票

---

日付	
会社名	
登録ユーザー名	
製品シリアル番号	
製品バージョン	
電話番号	
ファックス番号	
電子メールアドレス	
使用パソコン機種	
ホスト言語とバージョン	
OSバージョン	
IISバージョン	
お問合わせ内容、問題記述など、具体的に再現可能なようにご記入ください。	
添付資料	

NetMan UpLoader for ASP version 2.00  
プログラミング・マニュアル

第1版

2001年9月5日

版權・著作 株式会社テクナレッジ  
Printed In Japan