

NetMan

Components for Mail

Version 5.60

Programming guide

 **TechKnowledge**

目次

目次	2
はじめに	7
NetMan Components for Mail について	7
開発ライセンス	8
ランタイム・ライセンス	8
ユーザー・サポート	8
保証規定	10
販売元	11
開発元、ユーザーサポート	11
商標登録	11
NETMAN COMPONENTS FOR MAIL のインストール	12
システム条件	12
NetMan Components for Mail のインストール	13
NetMan Components for Mail 製品モジュール一覧	13
NameSpace について	13
サンプル	15
Visual Basic.NET サンプル	15
Visual C# サンプル	15
カスタムコンポーネントの使用法	16
Visual C#	16
Visual Basic .NET	17
コンパチビリティ	19
Version 5.5 とのコンパチビリティ	19
Version 5.0 とのコンパチビリティ	19
Version 4.0 とのコンパチビリティ	19
COM コンポーネント版 (Version 3.0 以前) とのコンパチビリティ	20
RFC	22
NETMAN SMTP COMPONENT	23
プロパティ・リファレンス	23
Bcc	23
BodyContentType	23
Cc	23
CharConvOpt	23
ConnectTimeOut	24
ContentDescription	24
ContentTransferEncoding	24
ContentType	24
DeliveryStatusNotify	25
DomainAddress	25
DocInFileName	25
DocInput	25
DocInType	26
EnvelopeID	26

FileNameEncoding.....	26
From.....	26
FromName.....	26
Header.....	27
IncludeDate.....	27
IncludeMessageId.....	27
InitSocket.....	27
LastOSError.....	27
LastSockError.....	28
LastSSLError.....	28
MessagePartial.....	28
NotifyType.....	28
PartCount.....	28
Password.....	29
PumpMessage.....	29
ReceiveTimeOut.....	29
RemoteHost.....	29
RemotePort.....	29
SendTimeOut.....	29
ShowErrorMessage.....	30
SmtpAuthType.....	30
SSLType.....	30
SSLVerify.....	31
Subject.....	31
To.....	31
TraceFileName.....	31
TraceOption.....	31
UserID.....	32
メソッド・リファレンス.....	33
AttachImage.....	33
CancelAllMessages.....	33
CancelMultiMessage.....	33
Connect.....	34
Init.....	34
IsConnected.....	35
Noop.....	35
Quit.....	35
SendMail.....	36
SendMessage.....	37
SendMultiMessage.....	37
Terminate.....	39
NETMAN SMTP COMPONENT エラー・コード一覧.....	40
NETMAN POP3 COMPONENT.....	45
プロパティ・リファレンス.....	45
AuthType.....	45
ConnectTimeOut.....	45
DocOutFileName.....	45
DocOutFilePath.....	45
DocOutput.....	46
DocOutType.....	46

Format	46
IncludeHeader	47
InitSocket.....	47
LastErrorText.....	47
LastOSError	47
LastSockError	48
LastSSLError	48
ListMailFormat	48
MessageCount	49
MessagePartial.....	49
MimeMark	49
Numbering.....	49
Password.....	50
PumpMessage.....	50
ReceiveTimeOut	50
RemoteHost	50
RemotePort	50
SendTimeOut.....	51
ShowErrorMessage	51
SSLType.....	51
SSLVerify.....	51
TraceFileName	52
TraceOption	52
UserID	52
メソッド・リファレンス.....	53
Authenticate	53
Connect	53
Delete	53
GetAttachmentInfo	54
GetUIDL	56
GetUIDLList.....	56
Init	57
IsConnected	58
ListMail	58
ListMessageID.....	60
Noop	61
Quit	62
Reset	62
RetrieveAsText.....	62
RetrieveHeader	63
RetrieveMail	65
RetrieveMailBody.....	65
RetrieveMessage	66
SaveAttachment	69
Stat.....	70
Terminate	70
NETMAN POP3 COMPONENT エラー・コード一覧	72
NETMAN IMAP4 COMPONENT リファレンス.....	78
プロパティ・リファレンス.....	78
ConnectTimeOut	78

DocOutType	78
InitSocket.....	78
LastErrorText.....	78
LastOSError	79
LastSockError	79
ListMailFormat	79
MailBoxName.....	80
Numbering.....	80
NumOfMails	80
NumOfRecents	80
NumOfUnseen.....	80
Password.....	80
PumpMessage.....	81
QuickList	81
ReceiveTimeOut	81
RemoteHost	81
RemotePort	81
Result	81
SendTimeOut.....	82
SSLType.....	82
SSLVerify.....	82
UserID	82
ShowErrorMessage	82
TraceFileName	83
TraceOption	83
メソッド・リファレンス.....	83
AppendToMailBox.....	84
Capability	85
CheckMailBox	86
CloseMailBox.....	86
Connect	87
Copy	87
CreateMailBox.....	87
DeleteMail	88
DeleteMailBox	88
ExamineMailBox.....	89
Expunge	89
FetchMail.....	90
FetchMailBody	91
FetchMailAttachment	92
FetchMailHeader	93
FetchMailStructure.....	93
Imap4Command	95
Init	95
IsConnected	96
ListMail	96
ListMailBox	97
ListSubscribed.....	98
LogIn	99
LogOut	99
MergePartialMails	100

Namespace.....	100
Noop.....	101
RenameMailBox.....	101
Search.....	102
SelectMailBox.....	104
Status.....	105
Store.....	105
SubscribeMailBox.....	106
Terminate.....	106
UnSubscribeMailBox.....	107
NETMAN IMAP4 COMPONENT エラー・コード一覧.....	108
共通クラスリファレンス.....	112
Attachment クラス.....	112
ContentDescription.....	112
ContentDisposition.....	112
ContentTransferEncoding.....	112
ContentType.....	112
FileName.....	112
Mail クラス.....	113
Attachments.....	113
Bcc.....	113
Body.....	113
BodyContentType.....	113
Cc.....	113
From.....	114
Header.....	114
ReadEml.....	114
To.....	114
WriteEml.....	114
APPENDIX A. FAQ よくあるご質問.....	116
APPENDIX.B システム・エラー・コード.....	122
2 ENOENT.....	122
9 EBADF.....	122
12 ENOMEM.....	122
13 EACCES.....	122
17 EEXIST.....	122
22 EINVAL.....	123
24 EMFILE.....	123
28 ENOSPC.....	123
34 ERANGE.....	123
36 EDEADLOCK.....	123
APPENDIX.C フラグ値.....	124
NETMAN COMPONENTS FOR MAIL 5.6 調査依頼.....	125

はじめに

NetMan Components for Mailについて

このたびは、NetMan Components for Mail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 Components for Mailはインターネット・メールを送受信する.NET Framework 2.0専用コンポーネントです。1996年に発売したNetMan Mail/OCX32, 1999年のNetMan ActiveX Control for Mail の2005年のNetMan Components for Mailのバージョン・アップ版となります。

当製品はIMAP4コンポーネント,POP3コンポーネント, SMTPコンポーネントの3つのカスタムコンポーネントから構成されます。これらのカスタム・コンポーネントにより、お客様の.NET Frameworkアプリケーションにインターネット・メール送受信機能を簡単に付加することが可能です。以下は製品の特徴です。

- .NET Framework 2.0 専用コンポーネント
- Visual Studio .NET 2005 IDE 対応
- Visual C#/VB.NET 2005 Express Edition 対応
- SSL通信対応
- .NET コンポーネントモデル準拠
- オーバーライドで複数エントリによる利便性の高いメソッド
- Genericsによるコレクションプロパティ
- DataSetオブジェクトによるメールリスト機能
- 添付ファイルの送受信、添付ファイル情報の取得
- 漢字コードの自動変換
- バイナリ・メールのMIMEエンコード/デコード
- マルチパート・メール対応
- サイズの大きいメールは自動的に分割メールで送信
- 分割メールのマージ受信機能
- SMTP AUTH(CRAM MD5/LOGIN/PLAIN認証)対応
- APOPに対応
- Imap4プロトコルに対応
- htmlメールの送受信が可能
- 配信状況通知(Delivery Status Notification)に対応
- Visual Basic.NET, Visual C#対応
- COMコンポーネント版とのコンパチビリティを確保
- EMLファイルのサポートメソッド
- ランタイム・ライセンス・フリー

開発ライセンス

開発ライセンスとは、開発者1名が1台のパーソナル・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 Components for Mail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 Components for Mail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。
 - ① NetMan Components for Mailのユーザー登録をしない場合。
 - ② 使用規定を遵守しない場合。
 - ③ 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。

ランタイム・ライセンス

本製品のランタイム・ライセンスは無料です。ただし、フリー・ソフトウェアに添付して配布することは禁止します。本製品のバージョン管理等の為に、商用アプリケーションとしてお客様のアプリケーションに添付して販売される際には、アプリケーションの登録をお願いします。販社システム・ラボ¹または開発会社テクナレッジ²までご連絡ください。お客様の社内でのご利用に関しては登録の必要はありません。

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。

¹ メール・アドレスは info@systemlab.co.jp です。

² メール・アドレスは info@techknowledge.co.jp です。

- **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、またはお手紙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、ご連絡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。途中から御担当者に変更にな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。またインシデントは4件までとさせていただきます。インシデント消費後の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。
- **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「Visual Basicでの文字の検索方法」や「Windows98XPTCP/IPの設定」などをご質問されても返答できかねる場合がございます。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く場合がございますのであらかじめご了承ください。

- 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れ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ない場合がございます。このような場合、コンポーネント操作の状況や、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もございますのであらかじめご了承ください。

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。

販売元

Systemlab®

株式会社システム・ラボ

東京都杉並区上荻1丁目5番8号 直長ビル7F

電話 03-5397-7511

FAX 03-5397-7521

E-Mail info@systemlab.co.jp

URL www.systemlab.co.jp

開発元、ユーザーサポート

TechKnowledge

株式会社テクナレッジ

東京都世田谷区駒沢2丁目16番1号 サンドービル9F

電話 03-3421-7621

FAX 03-3421-6691

E-Mail info@techknowledge.co.jp

URL www.techknowledge.co.jp

商標登録

Microsoft, Visual Basic, Windowsは米マイクロソフト社の登録商標です。本マニュアルに記載される商標、登録商標は該当会社の商標または登録商標です。

NetMan Components for Mailのインストール

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

システム条件

NetMan Components for Mail のインストールに先立って、以下の前提となるソフトウェアが必要となります。

クライアント・パソコンのオペレーティング・システムとして以下のいずれか

Microsoft Windows Vista
Microsoft Windows 2003 Server
Microsoft Windows XP
Microsoft Windows 2000

開発環境の言語として以下のいずれか

Visual Basic .NET
Visual C#
Visual J#

開発環境として以下のいずれか

Visual Studio .NET 2005
Visual C# 2005 Express Edition
Visual Basic.NET 2005 Express Edition
Visual C++ 2005 Express Edition

実行時フレームワークとして以下

.NET Framework 2.0

以上に加えて、IMAP4,POP3,SMTP をサポートする TCP/IP で接続されたインターネット・メール・サーバーが前提となります。

マニュアル記載時以降にサポートになる環境もございますので開発会社ウェブ等でご確認ください。

NetMan Components for Mailのインストール

NetMan Components for Mail 開発パッケージを CD-ROM 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。

NetMan Components for Mail をインストールする際、インストール・プログラムを実行しますがインストール・プログラムは出来るだけ Windows 環境で単独動作させてください。他のアプリケーションからの影響で NetMan Components for Mail のインストールが正常に行われない場合がございます。Windows の立ち上げ後すべてのアプリケーションを終了した状態で以下の手順で NetMan Components for Mail をインストールしてください。

- ① 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 CD-ROM のルートフォルダにある setup.exe を実行します。
- ② setup.exe の質問に答えてインストール・ボタンをクリックし、メッセージに答えると自動的にインストールが終了します。
- ③ インストールが正常に終了すると NetMan Components for Mail プログラム・グループが作成されます。
- ④ NetManMail.htm 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。インストール、対応言語に関する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。

NetMan Components for Mail 製品モジュール一覧

NetMan 製品のインストール・ディレクトリを<instdir>とした場合に、以下に NetMan Components for Mail のインストールファイル一覧を示します。

モジュール名とパス	内容
<instdir>%bin%MailLib.DLL	カスタムコンポーネントモジュール
<instdir>%man%nmMail5.pdf	オンラインマニュアル
<instdir>%man%nmMail5.html	README ファイル
<instdir>%Samples%*.*	サンプル・ファイル(詳細はサンプルのページをご参照ください)

実行時に必要なファイルは MailLib.DLL ファイルとなります。これ以外のファイルをお客様のアプリケーションに添付してランタイム環境として配布することはできません。著作権違反となる場合がございますのでご注意ください。

NameSpace について

NetMan Components for Mail の各コンポーネントは namespace MailLib に配置されます。ご利用になる各言語での namespace 宣言方法にしたがってネームスペースを宣言してコードすることをお勧めいたします。

SAMPLE

サンプル

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

Visual Basic.NET サンプル

Visual Basic.NET 用のサンプルは以下のようになります。製品インストールディレクトリの Samples¥vb.net フォルダ以下サブ・ディレクトリにあります。

フォルダ名	サンプルファイル名	概要
Pop3Sample	pop3Sample.vbproj	POP3 コンポーネント・サンプル・プロジェクト・ファイル
SmtplibSample	SmtplibSample.vbproj	SMTP コンポーネント・サンプル・プロジェクト・ファイル
Imap4Sample	Imap4Sample.vbproj	IMAP コンポーネント・サンプル・プロジェクト・ファイル

Visual C#サンプル

Visual C#用のサンプルは以下のようになります。製品インストールディレクトリの Samples¥cs フォルダ以下にあります。

フォルダ名	サンプルファイル名	概要
Pop3Sample	Pop3sample.csproj	POP3 サンプルプロジェクトファイル
SmtplibSample	SmtplibSample.csproj	SMTP サンプルプロジェクトファイル
Imap4Sample	Imap4Sample.csproj	IMAP サンプルプロジェクトファイル

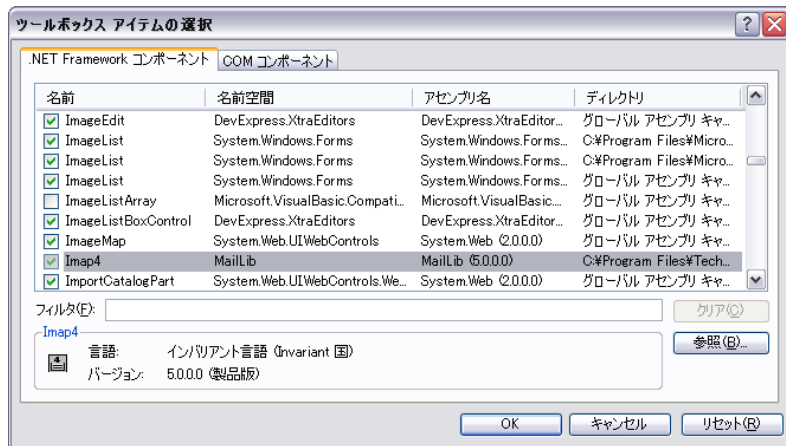
カスタムコンポーネントの使用法

この章では.NET 言語環境における NetMan Components for Mai の使い方をご説明します。

Visual C#

NetMan Components for Mail を Visual C#新規プロジェクトにて利用する手順についてご説明します。

- ① Visual Studio.NET を起動します。
- ② 「新規作成」メニューから「新規プロジェクト」を選択します。
- ③ 「新しいプロジェクト」ダイアログで Visual C#プロジェクトを選択し、「Windows アプリケーション」を選択します。プロジェクト名を設定して OK ボタンを押します。
- ④ フォームのデザインウィンドウにフォーカスをあて、ツールボックスから NetMan を追加したいタブを選択します。たとえば「全般」というタブに追加したい場合は「全般」タブを開いた状態にします。
新しく NetMan のタブを作成したい場合はツールボックスを右クリックして「タブの追加」を選択し、NetMan と入力し Enter を押します。
- ⑤ 上記で選択したタブをマウス右クリックして「アイテムの選択」を選択します。
- ⑥ 「.NET framework コンポーネント」タブを選択します。(起動時のデフォルトです)
- ⑦ 参照ボタンを押してインストールフォルダー下の BIN フォルダーにある MailLib.DLL ファイルを選択します。(インストールフォルダーのデフォルトは c:\Program Files\TechKnowledge\NetMan Components for Mail 5.60)
- ⑧ 「Imap4」「Pop3」「Smtplib」がチェックされていることを確認して OK を押します。



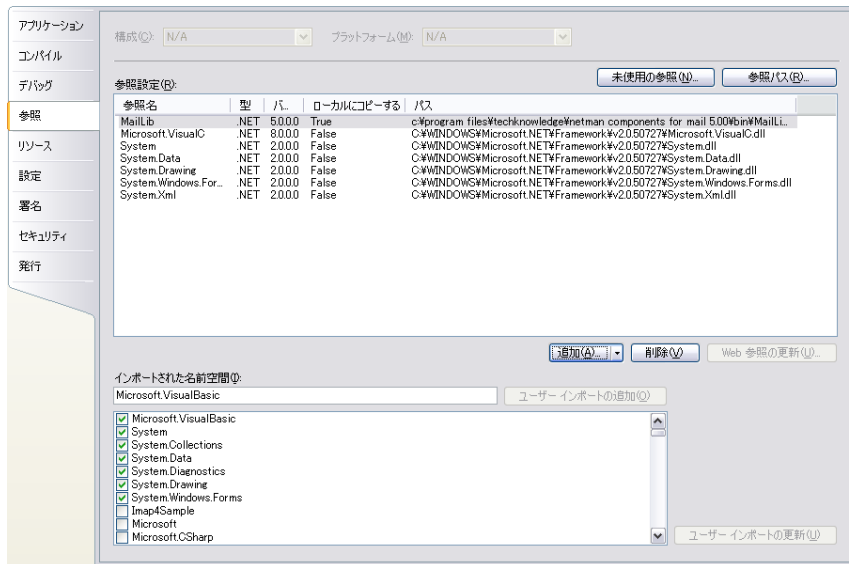
- ⑨ ツール・ボックスには以下のようにアイコンが追加されますので必要なコンポーネントを選択してフォームにドラッグすることで NetMan コンポーネントをご利用いただけます。



Visual Basic .NET

NetMan Components for Mail は Microsoft 社の Visual Basic .NET 言語からでもご利用いただけます。ご利用方法は前述の Visual C#の場合と同一で異なる点は言語の選択において Visual Basic.NET を選択する点のみの相違となります。

以下は Visual Basic.NET のプロジェクトプロパティの参照タブにて MailLib を追加した状態のサンプル画像です。



コンパチビリティ

COM コンポーネントの最終バージョン 4.0 とそれ以降の .NET 各版とのコンパチビリティについて以下に記載します。

Version 5.5とのコンパチビリティ

Version5.6 と Version5.5 の機能差異は以下になります。

- IMAP4 コンポーネントの SSL 通信対応
IMAP4 コンポーネントに SSLType,SSLVerify プロパティが追加されました。
- IMAP4 コンポーネントの機能強化
FetchMail メソッドに UIDL 指定が可能となりました。また、Imap4Command メソッドの追加により任意の IMAP4 コマンドを実行し結果を文字列に得ることができます。
- InitSocket プロパティの追加
各コンポーネントに InitSocket プロパティが追加されました。False 設定の場合には Connect メソッドで Windows Socket ライブラリの初期化をしません。複数 TCP/IP 接続を同時に保持したい場合に False 設定としてご利用ください。

Version 5.0とのコンパチビリティ

Version5.5 と Version5.0 の機能差異は以下になります。

- SSL 通信対応
Smtplib/Pop3 コンポーネントに SSLType,SSLVerify プロパティが追加されました。
- EML ファイル形式対応
従来版でも Pop3 コンポーネントの RetrieveAsText メソッドでは EML 形式のファイルを出力可能でしたが、利便性を高めるために Mail オブジェクトに ReadEml,WriteEml メソッドを追加しました。

Version 4.0とのコンパチビリティ

NetMan Components for .NET version 4.0 とは対応する .NET framework のバージョンが異なりますがプログラムソース上は上位互換性がございます。以下は製品のバージョンと対応する Microsoft .NET framework のバージョンの関係です。

製品バージョン	.NET framework バージョン
NetMan Components for .NET version 4.0	1.1
NetMan Components for .NET version 5.0	2.0
NetMan Components for .NET version 5.5	2.0

以下はプログラミング上変更が必要な点です。

- Mail オブジェクトの変更
バージョン 4.0 では Mail クラスのプロパティ、Attachments Bcc, Cc, To は 1次元配列でしたが、バージョン 2.0 からは System.Collections.Generic.List の文字列型に変更されています。
- Smtplib コンポーネントの宛先データ型変更
Bcc,Cc,To プロパティは旧版では文字列配列でしたが当バージョンでは Generics の List を使って文字列を指定します。

COMコンポーネント版(Version3.0以前)との互換性

当.NET Framework 専用版は COM コンポーネント版の上位互換製品となります。

- メソッドとプロパティはどちらの版でも同じ名前を用意されていますので従来版で記載されたアプリケーションの移行は容易です。
- たとえば配列の要素数などは.NET ではオブジェクトに Length プロパティとして取得できるようになりました。それに伴い当コンポーネントの各メソッドでこのような省略可能なパラメータがある場合はそれを省略したメソッドをオーバーロードして用意しています。また互換性を確保するため従来と同じパラメータのメソッドもオーバーロードして残しています。
- 従来版では配列は Variant 型でデータを返すものは当版ではデータ型を明示するように変更されています。.NET Framework では Variant 型はなくなった為の措置です。
- To,Cc プロパティは従来版では文字列型で複数メールアドレスはカンマで区切る仕様でしたが、当バージョンから文字列配列型に変更になりました。1メールアドレス1配列要素として指定します。
- Attachment/Mail オブジェクトの導入
従来の COM 環境ではユーザー定義のオブジェクトをメソッドのパラメータにすることが出来ない制約がありましたが.NET framework 対応の言語ではそのような制約がなくなりました。それに伴い Attachment/Mail オブジェクトを追

加しひとつのまとまったオブジェクトとしてメール情報を管理できるメソッドを追加しました。

SAMPLE

RFC

NetMan Components for Mail は以下の RFC に準拠しています。

SMTP	RFC2821,RFC2822
POP3	RFC1939
MIME	RFC1468,RFC2045,RFC2046,RFC2047,RFC2048,RFC2049
IMAP4	RFC2060
IMAP4 NAMESPACE	RFC2324
UTF-7	RFC2152
DSN	RFC1891
SMTP AUTH	RFC2554

上記ドキュメントは以下の URL から入手可能です。

<http://www.rfc-editor.org>

NetMan SMTP Component

この章ではインターネットメールを送信する NetMan SMTP カスタム・コンポーネントのプロパティ、メソッドについての説明をします。

プロパティ・リファレンス

Bcc

Bcc 宛先(blind carbon copy)を指定します。メールアドレスは Generics の List で文字列を複数指定することが可能です。このプロパティの指定は必須ではありません。

BodyContentType

メール本文のタイプを指定します。

値	シンボル	意味
0	NmPlain	Text/plain を本文の Content-Type:に指定します。通常のメールはこちらを選択します。
1	nmHtml	Text/html を本文の Content-Type:に指定します。Html メールを送信する場合にはこちらを選択します。Html メールを送信する場合には DocInput プロパティで指定する本文に html を記述します。Html 記述についてはエラー・チェック等は一切行わないで送信されますのでご注意ください。

Cc

Cc(carbon copy)宛先アドレスを指定します。アドレスは Generics の List で文字列を複数指定することが可能です。このプロパティの指定は必須ではありません。

CharConvOpt

半角カタカナの変換処理方法を指定します。以下の値を設定することができます。

値	シンボル	意味
0	nmKanaConv	半角カタカナは全角カタカナに変換されます。全角カタカナに変換することによって、1行のサイズが 1000 ³ バイトを超えた場合は、1000 バイトで一行が構成されるように CR+LF で複数行に分割されます。
1	nmKanaNoConv	半角カタカナは半角スペース文字(ascii 32)に変換されます。

ConnectTimeOut

接続タイムアウトをミリ秒単位で指定することが出来ます。0指定の場合はタイムアウト判定を実施しません。ソケット通信時に指定時間ブロックされた場合はメソッドがタイムアウトステータスを返します。

ContentDescription

マルチ・パートの内容を説明する文字列を指定します。

ContentTransferEncoding

DocInFileName プロパティでバイナリ・ファイルを指定した場合にエンコードの方法を指定します。以下の値が設定可能です。

値	シンボル	意味
0	nmCTEBase64	Base64 Encoding
1	nmCTEQP	Quoted-Printable Encoding

ContentType

マルチ・パートのメールで MIME の Content-Type タグの指定をします。このプロパティの指定は必須ではありません。指定が無い場合、DocInType プロパティに指定により処理が異なります。DocInType プロパティに 0 を指定して、DocInput プロパティで送信文を指定した場合には「text/plain」が設定されます。DocInType プロパティに 1 を指定して DocInFileName プロパティで指定さ

³ 1,000 バイトは RFC2821 で規定される1行の最大値です。

れるファイルから送信文を読み込むように指定した場合には、そのファイルの拡張子からレジストリを参照して適切な Content-Type のタグが指定されます。レジストリから適切な値が得られない場合には、「application/octet-stream」が設定されます。

DeliveryStatusNotify

配信状況通知(Delivery Status Notification)の指定をします。このプロパティは以下の値を設定可能です。

値	シンボル	意味
0	NmNoNotify	配信状況の通知をしません。通常の配布です。
1	nmNotifySuccess	配信の成功を通知します。
2	NmNotifyFail	配信の失敗を通知します。
3	NmNotifyBoth	配信の成功と失敗を通知します。

DomainAddress

送信元のドメイン名を指定します。このプロパティの指定は必須です。以下は Visual Basic.NET のサンプルコードです。

```
MailLib.Smtp.DomainAddress = "techknowledge.co.jp"
```

DocInFileName

DocInType プロパティに 1 を設定した場合に、送信するファイル名を指定します。ファイル名にはパスを含めることができます。シングル・パートのメールを送信する場合には、テキスト・ファイルを指定してください。マルチ・パートのメールを送信する場合には、バイナリ・ファイルを指定することが可能です。バイナリ・ファイルの場合には ContentTransferEncoding プロパティにより、エンコードの方法を指定する必要があります。

DocInput

DocInType プロパティに nmDocInProperty(値 0)を設定した場合に、送信するメールの本文を文字列で指定します。文章の途中で改行したい場合は改行コードを含めます。

DocInType

送信文のタイプを指定します。このタイプとは DocInput プロパティによりメモリにある文書データまたは、DocInFileName で指定されるファイル名で指定される文書データのいずれかを指定します。以下の値をプロパティに指定することができます。

値	シンボル	意味
0	nmDocInProperty	DocInput プロパティを参照する。
1	nmDocInFileName	DocInFileName プロパティを参照する。

EnvelopeID

配信状況通知プロパティ DeliveryStatusNotify を 0 以外の値に設定した場合、配信の Envelope ID をこのプロパティで指定することができます。このプロパティの最大サイズは rfc1891 により 100 バイトになります。この値はメールの送信者が返却されたメールの管理に使う目的で用意されています。Delivery StatusNotify を実行するように指定された場合でも、このプロパティの指定は必須ではありません。

FileNameEncoding

添付ファイル名に漢字が含まれている場合のエンコード方法を指定します。Microsoft Outlook Express(旧バージョン)等、従来からの JIS 漢字エンコード方法に対応しないメール・ソフトには BASE64 を選択します。

値	シンボル	意味
0	nmTradEncode	JIS 漢字エンコード。(ESC 文字を利用した kanji-in/out)
1	nmBase64Encode	BASE64 エンコード

From

送信側のユーザーID を指定します。このプロパティの指定は必須です。

FromName

送信者の名前を指定します。漢字が指定された場合には、MIME エンコードされます。

Header

メールのヘッダーに追加する文字列を指定します。以下はサンプルです。複数のヘッダ行を指定する場合には、各行の間に改行コードを指定してください。以下は設定コード例です。

```
MailLib.Smtp.Header = "X-Mailer: NetMan Component V5.50"
```

IncludeDate

このプロパティに True を設定した場合にはメール・ヘッダーに Date:を添付してメールを送信します。通常は sendmail が添付するのでこのプロパティは False に設定して問題ありません。sendmail 以外のメール・サーバー・ソフトで Date:を添付しない場合に True 設定としてください。

IncludeMessageId

このプロパティに True を設定した場合にはメール・ヘッダーに Message-Id:を添付してメールを送信します。通常は sendmail が添付するのでこのプロパティは False に設定して問題ありません。sendmail 以外のメール・サーバー・ソフトで Message-Id:を添付しない場合に True 設定としてください。

InitSocket

このプロパティはデフォルト True 設定の場合には Connect メソッド内部で Windows Socket ライブラリを初期化します。他のコンポーネントや Windows socket をライブラリを使い同時に複数の TCP/IP 接続が必要な場合には False 設定にして Connect メソッド内部からは Windows Socket の初期化を実行しない設定にして複数の同時 TCP/IP 接続を可能とします。別途 Windows Socket の初期化が必要な場合には Init メソッドを呼び出します。

LastOSError

ファイルとの I/O 時にシステムからエラーが返された値を保持します。値の詳細

細は Appendix.B のシステム・エラー・コードの値を参照してください。

LastSockError

メール・サーバーとの通信レベルのエラーが発生した場合に Winsock ライブラリが返すエラー値を保持します。保持される値は Winsock ライブラリ関数の呼出後、エラー値を検知した直後に WSAGetLastError 関数により返される値になります。通常のアプリケーションではこのプロパティを利用することはありません。弊社製品サポートのトラブル・シュート用に利用します。従って値の詳細は当マニュアルには掲載しませんが、Microsoft Developer Network C D 等の Winsock エラーに詳細の記載があります。

LastSSLError

SSL 接続が確立できなかった場合に、その理由が保持される場合があります。トラブルシュート情報です。詳細をサポートにお知らせいただく場合がございます。

MessagePartial

メールを分割して送信する場合は値1を設定します。メールは約 50K で分割して送信されます。

NotifyType

DeliveryNotifyStatus プロパティを 0 以外の値に設定し、配信状況通知メールを送る場合に返信される配信状況のタイプを指定します。

値	シンボル	意味
0	NmNotifyHeader	ヘダーのみで通知します
1	NmNotifyFull	メール全体で通知します。ただし、メールが失敗した時に限定されます。

PartCount

マルチ・パートで送信する場合、パートの数が設定されます。実行時に参照のみ可能なプロパティです。

Password

SMTP AUTH による SMTP サーバー認証を利用する場合のパスワードを指定します。同時に UserID プロパティも指定することが必須となります。

PumpMessage

旧 COM 版とのコンパチビリティのためにこのプロパティを残しています。旧版では当プロパティが True 設定の場合は長い時間を要するメソッド呼出時にウィンドウメッセージを処理して Windows の描画が止まる現象を回避できる機能を提供していましたが、.NET 環境ではスレッドの利用が可能になりましたので、この機能は廃止になりました。スレッドからコンポーネントのメソッドを呼び出すことで同現象を回避する方法でアプリケーションの実装をしてください。

ReceiveTimeout

受信タイムアウトをミリ秒単位で指定することができます。0 指定の場合はタイムアウト判定を実施しません。ソケット通信時に指定時間ブロックされた場合はメソッドがタイムアウトステータスを返します。

RemoteHost

SMTP サーバーが稼動しているホスト名を指定します。Connect メソッドの呼び出し前にこのプロパティの指定は必須です。

RemotePort

SMTP サーバーが稼動している TCP ポート番号を指定します。Connect メソッドの呼び出し前に、このプロパティの指定は必須です。デフォルトは 25 です。SSL 通信は 465、STARTTLS 通信では 587 を指定します。

SendTimeout

送信タイムアウトをミリ秒単位で指定することができます。0指定の場合はタイムアウトを実施しません。ソケット通信時に指定時間ブロックされた場合はメソッドがタイムアウトステータスを返します。

ShowErrorMessage

メソッドの呼び出し時にエラーが発生した場合、エラー・メッセージをメッセージ・ボックスに表示する場合はこのプロパティに値 1 を設定します。エラーの状態はメソッドからの戻り値で判断できます。アプリケーションでメッセージを表示したい場合は、メソッドの戻り値を検査するコードを記述して、このプロパティには値 0 を設定します。

SmtpAuthType

UserID/Password プロパティを設定した場合には SMTP 認証が実行されます。このプロパティにより SMTP 認証のタイプを設定することも必須です。SMTP 認証のタイプは現在3種類サポートされています。以下の値から選択して設定してください。

値	シンボル	意味
0	NmSmtpAuthNone	SMTP 認証しません。
1	NmSmtpAuthCRAMMD5	CRAM MD5 認証を実行します。
2	NmSmtpAuthLogin	LOGIN 認証を実行します。
3	NmSmtpAuthPlain	Plain 認証を実行します。

SSLType

SSL 通信のタイプを指定します。デフォルトは nmNoSSL で通常のソケット通信を実行します。サーバー接続時に SSL 通信タイプの指定が必要なため、SSLType プロパティは Connect メソッドによりサーバーに接続する前に設定してください。Connect 後の SSLType 設定は次回 Connect 実行時まで有効になりません。

値	シンボル	意味
-1	NmNoSSL	通常のソケット通信
0	nmSSLV23	SSL V2.3 通信
1	nmSSLV1	SSL V1 通信
2	nmSSLV2	SSL V2 通信

3	nmSSLV3	SSL V3 通信
---	---------	-----------

SSLVerify

SSL 通信時にベリファイの実行を設定します。デフォルトは nmNoVerify です。サーバー接続時にベリファイを実施するために、SSLVerify プロパティは Connect メソッドによりサーバーに接続する前に設定してください。Connect 後の SSLVerify 設定は次回 Connect 実行時まで有効になりません。

値	シンボル	意味
0	nmNoVerify	ベリファイを実行しません。(デフォルト)
1	nmVerify	ベリファイを実行します。

Subject

メールの標題を指定します。漢字が指定された場合には MIME エンコードされます。

To

メールの宛先アドレスを指定します。このプロパティの指定は必須です。アドレスは Generics の List 要素として複数指定することができます。以下は Visual Basic.NET でのサンプルです。

```
Smtplib.To.Add("r2d2@xxxx.com")
Smtplib.To.Add("drgonzo@yyyy.co.jp")
```

TraceFileName

通信トレースを出力するファイル名を指定します。ドライブ、パスを指定しない場合には実行時のカレント・ディレクトリに出力されますのでご注意ください。

TraceOption

通信トレースが必要な場合には当プロパティに値 1 をセットしてください。値 1 をセットした場合は、TraceFileName プロパティにトレース結果を保存するファイル名を指定してください。値 2 を設定した場合には値 1 と同様ですが、先頭

に以下の形式で時間が出力されます。

HH:MM:SS sss

HH = 時間

MM = 分

SS = 秒

sss = 1/1000 秒

UserID

SMTP AUTH による SMTP サーバー認証を利用する場合のユーザーID を指定します。同時に Password プロパティも指定することが必須となります。

メソッド・リファレンス

NetMan SMTP Component のカスタム・メソッドについて説明します。

AttachImage

書式

AttachImage(*FileName* As String, *ContentID* As String) As Int16

パラメータ	意味
<i>FileName</i>	イメージ・ファイル名。
<i>ContentID</i>	イメージ・ファイルの Content-ID を指定します。 Html メール の タグ等に CID: <> の形式で指定する ID です。

概要

BodyContentType を 1 (text/html) に指定した場合には本文の html から参照されるイメージをこのメソッドにてメールにて添付することが可能です。本文の html には以下のような形式で Content-ID を指定します。

```
<body background="CID:1230">  
<IMG src="CID:3210">
```

第2パラメータの ContentID には上記のタグの [CID: 以下の値を指定します](#)。

CancelAllMessages

書式

CancelAllMessages() as Int16

概要

SendMultiMessage() で添付した マルチ・パート SMTP メール の 全パートを削除します。添付されているメールがない場合には、何もしません。

CancelMultiMessage

書式

CancelMultiMessage(*PartNo* As Int16) As Int16

パラメータ	意味
<i>PartNo</i>	パート番号。SendMultiMessage で返された値を指定します。

概要

SendMultiMessage()で添付した マルチ・パート SMTP メール の 特定のパートを削除します。添付されているメールがない場合には、何もしません。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan SMTP コンポーネント・エラー・メッセージをご参照ください。

Connect

書式

Connect() As Int16

概要

RemoteHost,RemotePort プロパティで指定される SMTP サーバーに接続します。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan SMTP コンポーネント・エラー・メッセージをご参照ください。

Init

書式

Init() As Int16

概要

Windows Socket ライブラリの初期化を実行します。InitSocket プロパティを False 設定にした場合、Connect メソッド呼出前に場合に応じて当メソッドを呼び出します。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan SMTP コンポーネント・エラー・メッセージをご参照ください。

IsConnected

書式

IsConnected() As Boolean

概要

SMTP サーバーとの接続状態を返します。

リターン・コード

接続中には True 値を返します。非接続時には False を返します。

Noop

書式

Noop() As Int16

概要

SMTP サーバーに NOOP コマンドを発行することにより、SMTP サーバーとの接続状態を確認することができます。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan SMTP コンポーネント・エラー・メッセージをご参照ください。

Quit

書式

Quit() As Int16

概要

接続済みの SMTP サーバーを切断します。SMTP コンポーネントを設定したフォームを終了する前に必ず呼び出してください。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan SMTP コンポーネント・エラー・メッセージをご参照ください。

書式

SendMail(*MailObject* As Mail) As Int16

概要

パラメータで指定された Mail オブジェクト指定に従ってメールの送信を行います。このメソッドではメールの宛先情報や添付ファイル指定は Mail オブジェクトに設定します。添付ファイル指定の場合 Attachment クラスの FileName にはフルパス指定をお勧めします。ファイル名のみ指定した場合には当コンポーネントを使う実行ファイルの実行時フォルダーにあるファイルを指定したことになります。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan SMTP コンポーネント・エラー・メッセージをご参照ください。

Visual Basic.NET サンプルコード

```
Dim smtp As New MailLib.Smt
Dim att As New Attachment
Dim mail As New Mail

With att
    .FileName = "c:\My Documents\test.txt"
    .ContentType = "text/plain"
    .ContentDescription = "attachment"
End With

With smtp
    .DomainAddress = "domain.co.jp "
    .Connect()

With mail
    .Attachments.Add(att)
    .From = "from@domain.co.jp "
    .Subject = "text file test"
    .Body = "あいうえお" & vbCrLf
    .To.Add("to@otherdomain.co.jp")
End With
```

```
. SendMail (mail)
. Quit ()
```

End With

SendMessage

書式

SendMessage () As Int16

概要

メールの送信を行います。送付する際は、DocInType,DocInput または DocIn FileName を指定後、このメソッドを呼び出します。このメソッドを呼び出す際、DocInFileName プロパティで指定できるファイルはテキスト文書のみです。シングル・パート SMTP メールを送付する場合には、このメソッドのみ呼び出します。マルチ・パート SMTP メールを送付する場合には、SendMultiMessage メソッドでファイル、メモを添付してマルチ・パートを作成後にこのメソッドを呼び出します。このメソッドを呼び出す以前に、Connect メソッドで SMTP サーバーに接続済みであることが必要で、送信するメールは From,To,DomainAddress プロパティの指定が必須となります。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan SMTP コンポーネント・エラー・メッセージをご参照ください。

SendMultiMessage

書式

SendMultiMessage (PartNumber As Int16, _
ContType As String, _
ContEncode As String) As Int16

<i>PartNumber</i>	マルチ・パートの番号が返されます。
<i>ContType</i>	ファイルの拡張子から決定されたコンテンツ・タイプ文字列が返されます。この文字列はレジストリを参照して設定されます。レジストリに登録が無い場合は application/octet-stream に設定されます。
<i>ContEncode</i>	ContentTransferEncoding プロパティからシステムに登録されたエンコード方法を返します。

概要

ファイル、メモなどの添付をして、マルチ・パート SMTP メールを作成します。添付する際は、DocInType,DocInput または DocInFileName を指定後、このメソッドを呼び出します。このメソッドだけではメールの送付は行われません。実際の送付を行う場合には、SendMessage メソッドを呼び出してください。このメソッドを呼び出す以前に Connect メソッドで SMTP サーバーに接続済みであることが必要で、送信するメールは From,To,DomainAddress プロパティの指定が必須となります。

添付するドキュメントに関する情報は、DocInType,DocInFileName,DocInput,Content Type,ContentDescription,ContentTransferEncoding プロパティで設定します。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan SMTP コンポーネント・エラー・メッセージをご参照ください。

Visual Basic.NET サンプル・コード

```
Dim rc As Integer
Dim partNum As Short
Dim contType As String
Dim contEncode As String
Dim mailto(0) As String

With Smtpl
    .RemoteHost = "smtp.company.co.jp "
    .DomainAddress = "company.co.jp "
    .TraceFileName = "c:¥folder¥smtp.log"
    .TraceOption = MailLib.TraceOptionValues.nmTraceToFile
    mailto(0) = "support@systemlab.co.jp "
    .To = mailto
    .FromName = "Stieve Vai "
    .From = "vai@stieve.com "
    .Subject = "This is Test"

    rc = .Connect
    If rc <> 0 Then
        MsgBox("errr connect " + CStr(rc))
        Exit Sub
    End If
```

```

contType = "application/octet-stream"
contEncode = "base64"

rc = .SendMultiMessage(partNum, contType, contEncode)
If rc <> 0 Then
    MsgBox("errr send multi message " + CStr(rc))
End If

.DocInType = MailLib.DocInTypeValues.nmDocInProperty
.DocInput = "これはてすと。" + vbCrLf + "本文です" + vbCrLf

rc = .SendMessage()
If rc <> 0 Then
    MsgBox("errr send message " + CStr(rc))
End If

rc = .Quit
If rc <> 0 Then
    MsgBox("err quit " + CStr(rc))
End If
End With

```

Terminate

書式

Terminate () As Int16

概要

Windows Socket ライブラリの終了処理を実行します。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan SMTP コンポーネント・エラー・メッセージをご参照ください。

NetMan SMTP Component エラー・コード一覧

NetMan SMTP Component のメソッドが返すエラー値についての説明です。

メッセージ	値	概要
ME101 既に初期化されています。	101	Connect メソッドを 2 重に呼び出しました。プログラムを修正して Connect メソッドを 2 回よびださないように修正してください。
ME102 TCP/IPの初期化に失敗しました。	102	NetMan が使っているクラス・ライブラリの初期化に失敗しました。TCP/IP 関連の設定、リソースが足りないなどの理由が考えられます。お使いのパソコンで他の TCP/IP アプリケーションが動作することをご確認ください。
ME103 ソケットの生成に失敗しました。(%d)	103	ソケットを生成するのに必要なリソースが不足しています。他に TCP/IP を使うアプリケーション同時に数多く実行されているような場合に発生します。パソコンを再起動して、他の TCP/IP を使うアプリケーションが実行されていない状態で動作を確認してみてください。
ME104 ホスト名の指定に誤りがあります。	104	RemoteHost プロパティの指定がありません。正しい値を RemoteHost プロパティに設定してください。
ME105 ポート・ナンバーの指定に誤りがあります。	105	RemotePort プロパティの値が0または負の値が指定されています。正しい値を RemotePort プロパティに設定してください。
ME106 ホスト(RemoteHost)との接続に失敗しました。(%d)	106	sendmail サーバーに接続できませんでした。RemoteHost,RemotePort プロパティが正しいことを確認してください。他のインターネット・メール・クライアント・ソフトが RemoteHost,RemotePort で指定されるサーバーに接続できることを確認してください。どのメール・クライアントも接続に失敗するような場合はメール・サーバー

		の設定を確認してください。
ME107 ソケットの生成に失敗しました。	107	socket を使えるだけのメモリが不足していると思われます。ME200 メッセージと同じ状態です。ME200 メッセージの説明に従ってください。
ME108 ホスト(RemoteHost)との接続に失敗しました。	108	Connect メソッドでサーバーに接続しましたが、サーバーから不正なレスポンスが返されました。サーバーの設定を確認してください。
ME111 受信中にエラーが発生しました。(%d)	111	Socket で受信中にエラーが発生しました。括弧内の数値は LastSocketError プロパティの値が表示されます。
ME112 送信中にエラーが発生しました。(%d)	112	Socket で送信中にエラーになりました。括弧内の数値は LastSocketError プロパティの値が表示されます。
ME121 初期化されていません。	121	Connect メソッドでサーバーに接続がされていません。先に Connect メソッドを呼び出して、サーバーに接続してください。
ME133 終了処理に失敗しました。	133	QUIT コマンドを実行しましたが、サーバーがエラーを返してきました。サーバーの状態、設定に問題があると思われますのでご確認ください。
ME151 ファイルにアクセス権がありません。	151	DocInFileName プロパティで指定されるファイルをオープンする時にアクセスが否定されました。他のプログラムやプロセスがこのプロパティで指定したファイルを排他モードでオープンしていないか、確認してください。または、ファイルの属性がオープンできないように設定されています。正常にオープンできるようなファイルの属性を設定してください。
ME152 ファイル名 (DocInFileName) の指定に誤りがあります。	152	DocInFileName プロパティが空、パス名が不正、ファイル名が不正のいずれかの状態です。DocInFileName プロパティに正しい値を設定してください。
ME153 プロパティ (DocInput) が空です。	153	DocInput プロパティが空です。DocInput プロパティに正しい値を指定してください。
ME155 ファイルの読み込み時にエラ	155	ファイルが何らかの理由で読み込め

一が発生しました。(%d)		ません。エラーの詳細は括弧内の数値を Appendix-B システム・エラー・コードを参照してください。
ME161 ホスト(RemoteHost)からの応答がありません。	161	NOOP コマンドを発行しましたが、接続先が応答しません。
ME171 ドメイン(DomainAddress)が指定されていません。	171	DomainAddress プロパティに指定がありません。正しいインターネット・ドメイン・アドレスを DomainAddress プロパティに指定してください。
ME172 発信者(From)が指定されていません。	172	From プロパティに指定がありません。From プロパティに正しいインターネット・メール・アドレスを指定してください。
ME173 送付先(To)が指定されていません。	173	To プロパティに指定がありません。To プロパティに正しいインターネット・メール・アドレスを指定してください。
ME174 ドメイン(DomainAddress)の指定に誤りがあります。	174	SMTP HELP コマンドを実行しましたが、接続先のサーバーからエラーが返されました。ドメイン・アドレスの指定が不正です。正しいドメイン・アドレスを指定してください。
ME175 発信者(From)の指定に誤りがあります。	175	SMTP From コマンドを実行しましたが、接続先のサーバーで認識できませんでした。From プロパティの設定、SMTP サーバーの設定をご確認ください。
ME176 送付先(From, Cc, Bcc)の指定に誤りがあります。	176	To,Cc,Bcc プロパティの設定が不正です。正しい値を設定してください。
ME177 メールを送付に失敗しました。(サーバー返却値:%d)	177	サーバーにメールを送りましたが、エラー・ステータスを返してきました。括弧内の値を RFC821 ドキュメントを参照して、エラーの原因を取り除いてください。
ME181 宛先(%s)は、存在しません。	181	To,Cc,Bcc プロパティの設定が不正です。括弧内は存在しなかったメール・アドレスです。正しい値を設定してください。
ME191 マルチパートで設定できる最大数をオーバーしました。	191	SendMultiMessage メソッドで指定できるマルチパート数の最大値100を超えました。
ME192 マルチパートの番号の指定に	192	SendMultiMessage メソッドで返され

誤りがあります。		るマルチパート番号を CancelMultiMessage メソッドに正しく指定してください。
ME193 パラメータのデータ型が不正です。	193	メソッドのパラメータとして指定したデータの型が不正です。
ME194 認証されませんでした。	194	SMTP AUTH 認証が拒否されました。UserID/Password プロパティが正しいことを確認してください。
ME195 SSL 接続が確立できませんでした。	195	SSL サーバーへの接続が失敗しました。SSL サーバーに関連する設定をご確認ください。LastSSLError プロパティに接続が失敗した理由が保持される場合があります。
ME200 メモリ不足です。	200	プログラムの作業領域が確保できませんでした。不足しているメモリはアプリケーションのヒープ・メモリです。OS の状態によっては、同時に実行しているアプリケーションを終了させることで回避できる場合もあります。また、メモリの増設、スワップ・エリアの拡大でも回避出来る場合があります。

以下はサーバー側からのメッセージです。メソッドを呼び出した場合に表示されることがあります。

ME211 システム・ステータスまたはシステム・ヘルプの応答
ME214 ヘルプ・メッセージ、コマンドの用法について
ME220 パラメータに指定されるドメイン名のサーバの準備完了
ME221 コネクションのクローズ
ME250 リクエストされたコマンドを正常に終了しました。
ME251 メール宛て先として指定されたユーザはローカルに存在しません。応答に指定した宛て先に転送します。
ME252 VRFY コマンドでのユーザーの確認はできません。ユーザーはローカルに存在しません。しかし、メールの転送は可能です。
ME354 メールデータの入力を開始してください。最後は、<CR><LF>.<CR><LF>で終了してください。
ME421 このホストのメール・サービスは動作していません。TCP コネクションを切断します。メール転送中にサーバーがシャットダウンされました。
ME450 リクエストされたコマンドは実行できませんでした。メールボックスのビジョが原因です。

ME451 ローカル・エラーのために指定のコマンドが実行できませんでした。
ME452 リクエストされたコマンドは実行されませんでした。ファイル・システムの容量不足です。
ME500 コマンドの文法エラーです。コマンド・ラインが長すぎる可能性があります。
ME501 指定のコマンドのパラメータ・エラーです。
ME502 指定のコマンドはサポートしていません。
ME503 コマンドの実行順序に問題があります。
ME504 指定のコマンドのパラメータはサポートしていません。
ME550 宛て先のメールボックスがないなどの理由で指定のコマンドは実行されませんでした。
ME551 指定された宛て先のユーザーは、このホストに存在しません。別のパスに転送してください。
ME552 ファイル・システムの問題で、指定のコマンドは実行できませんでした。
ME553 宛て先のメールボックス名に問題があるために指定のコマンドは実行できませんでした。
ME554 メールの転送処理に失敗しました。

NetMan POP3 Component

この章ではメール・サーバーからメールを受信する NetMan POP3 カスタム・コンポーネントのプロパティ、メソッドについての説明をします。

プロパティ・リファレンス

AuthType

Authenticate メソッド実行により Pop サーバーに UserId/Password にて設定されたユーザー情報を認証しますが、当製品では通常のプレーンテキストによる認証と APOP 認証をサポートしています。当プロパティにて認証方式を切り替えます。

値	シンボル	意味
0	nmAuthPlain	通常のプレーンテキストによる認証。
1	nmAuthApop	APOP による認証を実行します。明らかにサーバーのレスポンスが APOP をサポートしないものと分かる場合にはプレーンテキスト認証を実行します。

ConnectTimeOut

接続タイムアウトをミリ秒単位で指定することが出来ます。0指定の場合はタイムアウト判定を実施しません。ソケット通信時に指定時間ブロックされた場合はメソッドがタイムアウトステータスを返します。

DocOutFileName

受信メール本文を保存するファイル名を指定します。受信メール本文をファイルで受け取る場合には DocType プロパティに値1を設定し、このプロパティにファイル名を指定して RetrieveMessage メソッドを呼び出します。ファイル名のみ指定した場合は、開発しているアプリケーションプログラムのカレント・ディレクトリに保存されます。通常はパスまで含めた完全なファイル名を指定してください。

DocOutFilePath

マルチ・パート SMTP メールを受信する場合に、各パートの展開先のディレクトリを指定します。オペレーティング・システムで正しいパス名を指定する必要があります。添付ファイル名が指定されている場合はそのファイル名で DocOutFilePath プロパティに指定されたディレクトリに展開されます。ファイル名が指定されていない場合には partXXX.YYY というファイル名で展開されます。

DocOutput

受信メール本文を文字列で受け取る場合には DocOutType プロパティに値 0 を指定して RetrieveMessage メソッドを呼び出した後、このプロパティを参照します。

DocOutType

RetrieveMessage メソッドを呼び出してメールを受け取る場合の宛先を指定します。以下の値が指定可能です。

値	シンボル	意味
0	nmDocOutProperty	DocOutput プロパティに出力
1	nmDocOutFileName	DocOutFileName プロパティで指定されるファイルに出力

Format

RetrieveMail メソッドで得られる Mail オブジェクトの Header プロパティ、Body プロパティへの格納形式をこのプロパティの設定で変更することができます。プロパティへは ContentType 定義値(nmRow または nmDecoded)を設定します。デフォルトは nmDecoded 値が設定され Header/Body とともにデコードされた状態で格納されます。nmRaw 設定時には Header/Body とともにデコードされない状態で格納されます。

Mail オブジェクトの WriteEml で EML ファイルを出力する場合には日本語を含むメールは nmRaw を設定します。

以下は POP で受信したメールを EML ファイルに出力するコード例です。

```
MailLib.Pop3 pop = new MailLib.Pop3();  
MailLib.Mail m = new MailLib.Mail();
```

```
pop.RemoteHost = "192.168.2.1";
pop.UserID = "userid";
pop.Password = "passwd";
pop.Format = MailLib.ContentFormat.nmRaw;

pop.Connect();
pop.Authenticate();
pop.RetrieveMail(1, out m);
pop.Quit();

// EMLファイルを出力。
m.WriteEml(@"c:\tmp\result.eml");
```

IncludeHeader

IncludeHeader プロパティを True に設定した場合、RetrieveMessage メソッドで得られるメール本文にヘダーを追加して返します。False 設定の倍はメール本文のみ返します。

InitSocket

このプロパティはデフォルト True 設定の場合には Connect メソッド内部で Windows Socket ライブラリを初期化します。他のコンポーネントや Windows socket をライブラリを使い同時に複数の TCP/IP 接続が必要な場合には False 設定にして Connect メソッド内部からは Windows Socket の初期化を実行しない設定にして複数の同時 TCP/IP 接続を可能とします。別途 Windows Socket の初期化が必要な場合には Init メソッドを呼び出します。

LastErrorText

POP サーバーからのレスポンスがエラーの場合、当プロパティに POP サーバーからのエラー・メッセージを保持します。エラーが発生した時点のメッセージが保持されますが、その後の正常実行で当プロパティの内容が消去されることはありません。

LastOSError

ファイルとの I/O 時にシステムからエラーが返された値を保持します。値の詳

細は Appendix.B のシステム・エラー・コードの値を参照してください。

LastSockError

メール・サーバーとの通信レベルのエラーが発生した場合に Winsock ライブラリが返すエラー値を保持します。保持される値は Winsock ライブラリ関数の呼出後、エラー値を検知した直後に WSAGetLastError 関数により返される値になります。通常のアプリケーションではこのプロパティを利用することはありません。弊社製品サポートのトラブル・シュート用に利用します。従って値の詳細は当マニュアルには掲載しませんが、Microsoft Developer Network C D 等の Winsock エラーに詳細の記載があります。

LastSSLError

SSL 接続が確立できなかった場合に、その理由が保持される場合があります。トラブルシュート情報です。詳細をサポートにお知らせいただく場合がございます。

ListMailFormat

ListMail メソッドを実行する時にリストされるメールに関する情報の表示フォーマットを指定します。書式は以下のような形式です。

%フィールド名:/表示長

フィールド名として指定できる文字列はメールのヘッダーにある以下のタグです。⁴

Cc	カーボン・コピー宛先
Date	メールの発信日
From	メールの送付元アドレス
Message-ID	メッセージ ID
Subject	メールの表題
To	メールの宛先アドレス
X-Mailer	メール作成に使ったアプリケーション名

⁴ 他にもありますが、使用頻度の高いとおもわれるものだけ記述しました。メールヘッダーの詳細については rfc をご参照ください。

以下は設定のサンプルです。

```
Pop3Ctrl.ListMailFormat = “%From:/10 [%Date:/11] %Subject:/20”
```

以下はリストされる文字列の例です。先頭にはメッセージ番号と MIME メールの場合には 'M' が表示されます。添付ファイルが無くとも、本文に漢字があれば MIME でエンコードされたメールになりますのでご注意ください。

また、MessagePartial プロパティで 1 を指定して、分割メールのサポートを指定した場合には、'M' の後ろに 'P' (大文字) または 'p' (小文字) が表示されます。分割されてきたメールがすべてサーバに届いている場合には 'P' が表示されます。一方、分割メールの一部がサーバに未着の場合には、'p' が表示されず。

```
1 M tanaka@techknowledge.co.jp Sun,20 Aug テストの件  
2 gotcha@techknowledge.co.jp Mon,21 Aug About New Test Case  
3 MP hoge@techknowledge.co.jp Tue,22 Aug Enkai
```

MessageCount

メール・サーバーに保持されているメールの数を保持します。初期値は Authenticate メソッド実行にて設定されます。また、Stat メソッドの呼出により値が更新されます。

MessagePartial

サイズの大きいメールが分割されて送られてきた場合、それらを一つにまとめて受信する場合はこのプロパティに値 1 を設定します。分割されて送られてきたメールには MIME のタグが正しく設定されていることが必要になります。

MimeMark

当プロパティに True を設定した場合は ListMail メソッドの結果に MIME 表示をします。False 設定の場合は MIME の表示はしません。MIME 表示の詳細は ListMail メソッドをご参照ください。

Numbering

True 値を設定した場合は、ListMail メソッドの結果の各行の先頭に番号を表示します。番号が必要無い場合や桁数で整形された番号を表示したい場合は当プロパティに False を設定して、ListMail メソッドの結果の各行に整形された番号等を設定してください。

Password

接続先の POP3 サーバーに定義されるユーザーのパスワードを指定します。APOP サーバーに接続された場合には自動的に UserID/Password プロパティを使い APOP 認証を実行します。

PumpMessage

旧 COM 版とのコンパチビリティのためにこのプロパティを残しています。旧版では当プロパティが True 設定の場合は長い時間を要するメソッド呼出時にウィンドウメッセージを処理して Windows の描画が止まる現象を回避できる機能を提供していましたが、.NET 環境ではスレッドの利用が可能になりましたので、この機能は廃止になりました。スレッドからコンポーネントのメソッドを呼び出すことで同現象を回避する方法でアプリケーションの実装をしてください。

ReceiveTimeOut

受信タイムアウトをミリ秒単位で指定することが出来ます。0指定の場合はタイムアウト判定を実施しません。ソケット通信時に指定時間ブロックされた場合はメソッドがタイムアウトステータスを返します。

RemoteHost

POP3 サーバーが動作しているホスト名を指定します。

RemotePort

POP3 サーバーと接続する TCP ポートを指定します。デフォルトは 110 です。SSL 通信時には 995 を指定します。

SendTimeOut

送信タイムアウトをミリ秒単位で指定することが出来ます。0指定の場合はタイムアウトを実施しません。ソケット通信時に指定時間ブロックされた場合はメソッドがタイムアウトステータスを返します。

ShowErrorMessage

メソッドの中から NetMan のエラーメッセージを表示する場合はこのプロパティに値 1 を設定します。エラー・メッセージは表示しない場合はこのプロパティに値 0 を設定します。エラー・メッセージを表示しない場合はメソッドからの戻り値を必ずチェックするようにします。

SSLType

SSL 通信のタイプを指定します。デフォルトは nmNoSSL で通常のソケット通信を実行します。サーバー接続時に SSL 通信タイプの指定が必要なため、SSLType プロパティは Connect メソッドによりサーバーに接続する前に設定してください。Connect 後の SSLType 設定は次回 Connect 実行時まで有効になりません。

値	シンボル	意味
-1	NmNoSSL	通常のソケット通信
0	nmSSLV23	SSL V2.3 通信
1	nmSSLV1	SSL V1 通信
2	nmSSLV2	SSL V2 通信
3	nmSSLV3	SSL V3 通信

SSLVerify

SSL 通信時にベリファイの実行を設定します。デフォルトは nmNoVerify です。サーバー接続時にベリファイを実施するために、SSLVerify プロパティは Connect メソッドによりサーバーに接続する前に設定してください。Connect 後の SSLVerify 設定は次回 Connect 実行時まで有効になりません。

値	シンボル	意味
0	nmNoVerify	ベリファイを実行しません。(デフォルト)

1	nmVerify	ベリファイを実行します。
---	----------	--------------

TraceFileName

通信トレースを出力するファイル名を指定します。ドライブ、パスを指定しない場合には実行時のカレント・ディレクトリに出力されますのでご注意ください。

TraceOption

通信トレースが必要な場合には当プロパティに値 1 をセットしてください。値 1 をセットした場合は、TraceFileName プロパティにトレース結果を保存するファイル名を指定してください。値 2 を設定した場合には値 1 と同様ですが、先頭に以下の形式で時間が出力されます。

HH:MM:SS sss

HH = 時間

MM = 分

SS = 秒

sss = 1/1000 秒

UserID

接続先の POP3 サーバーに定義されるユーザーID を指定します。

メソッド・リファレンス

NetMan Pop3 Component のカスタム・メソッドについて説明します。

Authenticate

書式

Authenticate() As Int16

概要

UserID,Password プロパティに設定されたユーザーを認証します。先に POP3 サーバーに Connect メソッドを使って接続済みであることが必要です。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Connect

書式

Connect() As Int16

概要

RemoteHost,RemotePort プロパティで指定される POP3 サーバーに接続します。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Delete

書式

Delete(*MessageID* As Int16) As Int16

概要

受信メールを削除します。受信メールはパラメータの MessageID で指定します。

パラメータ

<i>MessageID</i>	削除するメッセージの ID
------------------	---------------

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

GetAttachmentInfo

書式

GetAttachmentInfo (*MessageID* As Int16,
ContentTypes() As String,
FileNames() As String) As Int16

GetAttachmentInfo (*MessageID* As Int16,
Count As Int16,
ContentTypes() As String,
FileNames() As String) As Int16

概要

指定された受信メールの添付ファイル情報を取得します。

パラメータ

<i>MessageID</i>	添付ファイル情報を取得するメールのメッセージの ID
<i>Count</i>	指定したメールに添付される添付ファイルの数。
<i>ContentTypes</i>	添付ファイルの Content Type を返します。Variant 型の配列で返されます。各要素は文字列で格納されます。
<i>FileNames</i>	添付ファイルのファイル名を返します。Variant 型の配列で返されます。各要素は文字列で格納されます。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Visual Basic .NET サンプルコード

```

Dim i As Integer
Dim rc As Short
Dim count As Int32
Dim cts() As String
Dim fns() As String

```

```

With Pop
    rc = .GetAttachmentInfo(1, count, cts, fns)
    If rc <> 0 Then
        MsgBox("error " + CStr(rc))
        .Quit()
        Exit Sub
    End If

    For i = 0 To count - 1
        System.Diagnostics.Debug.WriteLine(cts(i))
        System.Diagnostics.Debug.WriteLine(fns(i))
    Next
End With

```

Visual C#サンプル・コード

```

short rc;
short id = Convert.ToInt16(textBox1.Text);
short count;
string Types[];
string filenames[];

if((rc = Pop.GetAttachmentInfo(id,
                                out count,
                                out contTypes,
                                out fileNames)) != 0)
{
    Pop.Quit();
    return;
}
//
for(int i=0; i < count; i++)
{
    listBox1.Items.Add(filenames(i));
}

```

GetUIDL

書式

GetUIDL(*MessageID* As Int16,
 UIDL As String,) As Int16

概要

指定したメールの UIDL を取得します。

パラメータ

<i>MessageID</i>	UIDL 取得するメールのメッセージの ID
<i>UIDL</i>	指定したメッセージの UIDL。パラメータ型は Variant で文字列データを含みます。UIDL は文字コード 0x21～0x7e 迄の範囲の文字列となります。

リターン・コード

正常終了ならば値 0 が返されます。0 以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

注意事項

Pop サーバーによっては RFC の記載と異なり、UIDL の先頭にメッセージ ID が添付される場合があります。Numbering プロパティを False 設定するとこのような Pop サーバーのレスポンスでもメッセージ ID を削除して UIDL のみを第 2 パラメータに返します。また、UIDL の詳細については rfc をご参照ください。

Visual C# サンプル・コード

```
。
String uidl;
short rc = Pop3.GetUIDL(1, out uidl);
if(rc != 0)
{
    MessageBox.Show(this, rc.ToString());
}
System.Diagnostics.Debug.WriteLine(uidl);
```

GetUIDLList

書式

GetUIDLList(ByRef *UIDLList()* As String,) As Int16

GetUIDLList(*Count* As Int32, ByRef *UIDLList()* As String,) As Int16

概要

現在のメール・サーバーに保持してあるメールの UIDL リストを取得します。

パラメータ

<i>Count</i>	UIDLList に含まれる配列要素の数。
<i>UIDLList</i>	UIDL リスト。VARIANT 型の配列です。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Visual C# サンプル・コード

```
String uidls();
short count;
short rc = Pop3.GetUIDLList(out count, out uidls);
if( rc != 0)
{
    MessageBox.Show(this, rc.ToString());
}
for(int i=0; i < count; i++)
{
    listBox1.Items.Add(uidls(i));
}
```

Init

書式

Init() As Int16

概要

Windows Socket ライブラリの初期化を実行します。InitSocket プロパティを False 設定にした場合、Connect メソッド呼出前に場合に応じて当メソッドを呼び出します。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コ

ンポーネント・エラー・メッセージをご参照ください。

IsConnected

書式

IsConnected() As Boolean

概要

POP サーバーとの接続状態を返します。

リターン・コード

接続中には True 値を返します。非接続時には False を返します。

ListMail

書式

ListMail(*MailList()* As String) As Int16

ListMail(*NumOfMail* As Int16, *MailList()* As String) As Int16

ListMail(*ds* As System.Data.DataSet) As Int16

ListMail(*ds* As System.Data.DataSet, *hdrList()* As String) As Int16

概要

受信メールをリストします。DataSet を指定しない形式のメソッドにつきましてはリストする項目、形式は ListMailFormat プロパティで指定します。メール番号表示の有無は Nubering プロパティで指定します。MIME メールマーク表示は MimeMark プロパティで表示の有無を設定します。

MessagePartial プロパティに値 1(分割メールのサポート)を指定した場合、分割メールは1つのメールとして表示されます。最初にサーバに到着した分割メールのメッセージ ID のみが表示され、2番目以降に到着したメールのメッセージ ID は表示されません。従って、メッセージ ID は飛び飛びに表示されることになります。MessagePartial プロパティに値 0(分割メールの非サポート)を指定した場合、分割メールは別々のメールとして表示されます。

ただし DataSet 指定のメソッドに関しましては MessagePartial の指定は無視されますのでご注意ください。

DataSet にメールをリストした場合メッセージ ID は Row の Tag にセットされません。

パラメータ

<i>NumOfMails</i>	リストされたメールの数
<i>MailList</i>	メールのリストの配列。インデックスの値は、0 から (引数 NumOfMails - 1) までが有効な値です。
<i>ds</i>	メールをリストするデータセット。
<i>hdrList</i>	データセットにメールをリストする場合のメールヘダーの項目を文字列配列で指定します。指定が無い場合はデフォルトのメールヘダーをリストします。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Visual C# サンプル・コード

```
string [] msgList=null;
pop.ListMailFormat = "%Subject:/40 %From:/20";
rc = pop.ListMail(out msgList);
if(msgList != null )
{
    for(i=0; i<msgidList.Length; i++)
    {
        System.Diagnostics.Debug.WriteLine(msgList[i]);
    }
}
```

Visual C# データセットサンプル・コード

```
string[] hdrList = { "From", "To", "Cc", "Subject" };
MailLib.Pop3 pop = new MailLib.Pop3();

pop.TraceOption = MailLib.TraceOptionValues.nmTraceToFile;
pop.TraceFileName = @"c:\tmp\%zzz.log";
pop.RemoteHost = "192.168.2.1";
pop.UserID = "mailtest";
pop.Password = "abcdefg";
pop.AuthType = MailLib.PopAuthTypeValues.nmAuthPlain;

short rc = pop.Connect();
rc = pop.Authenticate();
```

```
rc = pop.Stat(out num, out size);
msg = String.Format("num = {0} size={1}", num, size);
System.Diagnostics.Debug.WriteLine(msg);
```

```
DataSet ds = new DataSet("mailList");
rc = pop.ListMail(out ds, hdrList);
dataGridView1.DataSource = ds.Tables["mailList"];
dataGridView1.Refresh();
rc = pop.Quit();
```

ListMessageID

書式

```
ListMessageID( MsgIDList( ) As String ) As Int16
ListMessageID( NumOfMsg As Int16, _
               ByRef MsgIDList( ) As String ) As Int16
```

概要

受信メール ID をリストします。リストは、「メッセージ ID メールサイズ」という形で表示されます。MessagePartial プロパティに値 1(分割メールのサポート)を指定した場合、分割メールは1つのメールとして表示されます。最初にサーバに到着した分割メールのメッセージ ID のみが表示され、2番目以降に到着したメールのメッセージ ID は表示されません。従って、メッセージ ID は飛び飛びに表示されることとなります。メールのサイズは、分割メールの全体のサイズです。MessagePartial プロパティに値 0(分割メールの非サポート)を指定した場合、分割メールは別々のメールとして表示されます。

パラメータ

<i>NumOfMsg</i>	リストされた ID の数
<i>MsgIDList</i>	メッセージ ID の配列。インデックスの値は、0 から (引数 NumOfMsg - 1) までが有効な値です。

リターン・コード

正常終了ならば値 0 が返されます。0 以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Visual Basic.NET サンプル・コード

```
Dim rc As Int16
```

```
Dim i As Int16
Dim MsgIDNum As Int16
Dim MsgIDList As Variant
```

リストをクリアする

```
IstMailList.Clear
txtMailContent = ""
IstMimeContent.Clear
```

```
lblMailItem = "メッセージ ID サイズ"
```

```
rc = POP.ListMessageID(MsgIDNum, MsgIDList)
If rc <> 0 Then
    MsgBox "メイルリストの取得に失敗しました。" + str(rc)
    Exit Sub
End If
```

リストをクリアする

```
IstMailList.Clear
txtMailContent = ""
IstMimeContent.Clear
```

```
For i = 0 To MsgIDNum - 1
    IstMailList.AddItem MsgIDList(i)
Next i
```

```
Set MsgIDList = Nothing
```

```
If POPL.MessageCount > 0 Then
    IstMailList.ListIndex = 0
    cmdRetrieveMessage.Enabled = True
    cmdRetrieveHeader.Enabled = True
    cmdDelete.Enabled = True
End If
```

Noop

書式

```
Noop() As Int16
```

概要

POP3 サーバーに NOOP コマンドを発行することにより、POP3 サーバーとの

接続状態を確認することができます。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Quit

書式

Quit() As Int16

概要

接続済みの POP3 サーバーを切断します。POP3 コンポーネントを設定したフォームを終了する前に必ず呼び出してください。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Reset

書式

Reset() As Int16

概要

POP3 サーバーにリセットコマンドを発行します。POP3 サーバーとは接続済みかつ認証済みであることが必要です。このコマンドでは Delete メソッドによる削除や既読情報があった場合にはすべて取り消されます。このコマンド以前に取得したメッセージ ID はすべて無効になることに注意してください。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

RetrieveAsText

書式

RetrieveAsText(*MessageID* As Int16) As Int16

概要

指定されたメールをサーバーから送られてきた形式で受信します。MIME タグの解釈と漢字コードの変換はしません。メールの MIME タグの形式や漢字のエンコード状態を調査する際の用途を想定したメソッドです。

DocOutType プロパティの設定により、DocOutFileName で指定したファイルまたは DocOutput プロパティに受信結果を保存できます。

パラメータ

<i>MessageID</i>	取得するメールの ID を指定します。
------------------	---------------------

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

RetrieveHeader

書式

```
RetrieveHeader( MessageID As Int16,  
               ByRef HeaderList() As String ) As Int16
```

```
RetrieveHeader( MessageID As Int16, _  
               ByRef NumOfHeader As Int16, _  
               ByRef HeaderList() As String ) As Int16
```

概要

受信メールのヘッダーを取得します。受信メールはパラメータの MessageID で指定します。

パラメータ

<i>MessageID</i>	ヘッダーを取得するメールの ID を指定します。
<i>NumOfHeader</i>	HeaderList パラメータに返されたメール・ヘッダーの数を返します。
<i>HeaderList</i>	ヘッダーを配列で返します。インデックスの値は、0 から (引数 NumOfHeader - 1) ままで有効な値です。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Visual Basic.NET サンプル・コード

以下は Visual Basic での RetrieveHeader メソッドのサンプルです。

```
Dim rc As Int16
Dim msgID As Int16
Dim pos As Int16
Dim index As Int16
Dim Item As String
Dim i As Int16
Dim str As String
Dim HeaderNum As Int16
Dim HeaderList As Variant

'リストをクリアする
txtMailContent = ""
lstMimeContent.Clear

'選択項目がない時
index = lstMailList.ListIndex
If index = -1 Then
    End
End If

Item = lstMailList.List(index)
pos = InStr(Item, " ")
msgID = Val(Left$(Item, pos - 1))

rc = POP3CTL.RetrieveHeader(msgID, HeaderNum, HeaderList)
If rc <> 0 Then
    MsgBox "メールヘッダの取得に失敗しました:" & str(rc)
    Exit Sub
End If

'リストをクリアする
txtMailContent = ""
lstMimeContent.Clear

For i = 0 To HeaderNum - 1
    str = str + HeaderList(i) + Chr$(13) + Chr$(10)
Next i
Set HeaderList = Nothing
```



```
txtMailContent = str
```

RetrieveMail

書式

```
RetrieveMail( MessageID As Int16, _  
             ByRef MailObject () As Mail) As Int16
```

概要

メールを Mail オブジェクトとして POP3 サーバーから受信します。受信するメールは最初のパラメータで指定します。Mail オブジェクトにはヘダー関連情報、本文、添付ファイルがある場合には添付ファイル情報が設定されます。

パラメータ

<i>MessageID</i>	受信するメールの ID を指定します。
<i>MailObject</i>	受信するメールオブジェクトを指定します。

Visual C#サンプルコード

```
MailLib.Mail mail;  
rc = pop.RetrieveMail(3, out mail);  
if ( mail.Body != null )  
{  
    System.Diagnostics.Debug.WriteLine(mail.Body);  
}  
  
if (mail.Header != null)  
{  
    System.Diagnostics.Debug.WriteLine(mail.Header);  
}  
  
if (mail.Attachments != null)  
{  
    System.Diagnostics.Debug.WriteLine(mail.Attachments[0].FileName);  
}
```

RetrieveMailBody

書式

RetrieveMailBody(*MessageID* As Int16) As Int16

概要

パラメータで指定したメールの本文を DocOutPut プロパティに取得します。MIME 形式のメールの場合は最初に見つかった text/html パートまたは text/plain パートの内容となります。

パラメータ

<i>MessageID</i>	受信するメールの ID を指定します。
------------------	---------------------

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

RetrieveMessage

書式

RetrieveMessage(*MessageID* As Int16, _
ByRef *MimeHeader()* As String, _
ByRef *MimeBodyFileName()* As Variant) As Int16

RetrieveMessage(*MessageID* As Int16, _
ByRef *IsMime* As Boolean, _
ByRef *MimeHeader()* As String, _
ByRef *MimeBodyFileName()* As Variant) As Int16

RetrieveMessage(*MessageID* As Int16, _
ByRef *IsMime()* As Boolean, _
ByRef *MimePartCount()* As Int16, _
ByRef *MimeHeader()* As String, _
ByRef *MimeBodyFileName()* As Variant) As Int16

概要

メールを受信するメソッドです。受信メールはパラメータの MessageID で指定します。メールを受信した結果を格納する場所(プロパティまたはファイル)を、

DocOutType, DocOutput または DocOutFileName で指定してください。マルチ・パート SMTP を受信する場合には、DocOutFilePath プロパティを指定して、受信したファイルまたはメモが展開されるディレクトリを指定してください。

MessagePartial プロパティに値 1(分割メールのサポート)を指定した場合、分割メールは1つのメールとして受信されます。但し、分割されたメールの一部がサーバーに届いていない場合には、受信することができません。一部でも必要な場合には、MessagePartial プロパティに値 0(分割メールの非サポート)を指定して、個別にファイルを受信してください。

パラメータ

<i>MessageID</i>	受信するメールの ID を指定します。
<i>IsMime</i>	受信指定されたメールが MIME でエンコードされている場合 True 値を返します。
<i>MimePartCount</i>	マルチ・パート MIME で送られてきたメールの場合、そのパートの数を返します。
<i>MimeHeader</i>	MIME で送付されたメールのヘッダーの配列を返します。インデックスの値は、0 から (引数 MimePartCount - 1) までが有効な値です
<i>MimeBodyFileName</i>	マルチ・パート SMTP メールを受信した場合、DocOutFilePath で指定されたディレクトリにファイルが展開されます。展開されたファイル名の配列を返します。インデックスの値は、0 から (引数 MimePartCount - 1) までが有効な値です。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Visual C# サンプル・コード

```
string msg;
int i;
MailLib.Pop3 pop = new MailLib.Pop3();

pop.TraceOption = MailLib.TraceOptionValues.nmTraceToFile;
pop.TraceFileName = @"c:¥folder¥pop.log";
pop.RemoteHost = "192.168.2.1";
pop.RemoteHost = "sales.company.co.jp";
```

```

pop.UserID = "user1";
pop.Password = "password1";
pop.AuthType = MailLib.PopAuthTypeValues.nmAuthPlain;

short rc = pop.Connect();
rc = pop.Authenticate();

bool mime;
string [] MimeHeader = null;
string [] BodyFileName = null;
short partCount;

pop.DocOutFilePath = @"c:\temp\";
pop.DocOutType = MailLib.DocOutTypeValues.nmDocOutProperty;

rc = pop.RetrieveMessage(4,
    out mime,
    out partCount,
    out MimeHeader,
    out BodyFileName);

if( BodyFileName != null )
{
    for(i=0; i<BodyFileName.Length; i++)
    {
        System.Diagnostics.Debug.WriteLine(BodyFileName[i]);
    }
}
if(MimeHeader != null)
{
    for(i=0; i<MimeHeader.Length; i++)
    {
        System.Diagnostics.Debug.WriteLine(MimeHeader[i]);
    }
}

if(pop.DocOutput != null )
{
    System.Diagnostics.Debug.WriteLine(pop.DocOutput);
}
rc = pop.Quit();
}

```

SaveAttachment

書式

SaveAttachment(*MessageID* As Int16, *FileName* As String) As Int16

概要

指定したメールの指定した添付ファイルをファイルに出力します。

パラメータ

<i>MessageID</i>	受信メールの ID 設定します。
<i>FileName</i>	受信メールに含まれる添付ファイルのファイル名を指定します。添付ファイル名は GetAttachmentInfo メソッド等で得ることが出来ます。ファイル名はケース・センシティブにマッチングします。添付ファイルは DocOutFilePath プロパティで指定されるフォルダーに保存されます。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Visual Basic.NET サンプルコード

```
Dim i As Integer
Dim rc As Short
Dim count As Int32
Dim cts() As String
Dim fns() As String

With Pop
    rc = .GetAttachmentInfo(1, cts, fns)
    If rc <> 0 Then
        MsgBox("error " + CStr(rc))
        .Quit()
        Exit Sub
    End If

    rc = .SaveAttachment(1, fns(0))
    If rc <> 0 Then
        MsgBox("error " + CStr(rc))
        .Quit()
    End If
End With
```

Exit Sub
End If
End With

Stat

書式

Stat (ByRef *NumOfMail* As Int16, ByRef *MailSize* As Int32) As Int16

概要

メールの受信状況を取得します。受信したメールの数とその総バイト数を取得することができます。

パラメータ

<i>NumOfMail</i>	受信メールの数を返します
<i>MailSize</i>	受信メールのサイズをバイト数で返します。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Pop3 コンポーネント・エラー・メッセージをご参照ください。

Visual C#サンプルコード

```
short num;  
int size;  
  
rc = pop.Stat(out num, out size);  
msg = String.Format("num = {0} size={1}", num, size);  
System.Diagnostics.Debug.WriteLine(msg);
```

Terminate

書式

Terminate () As Int16

概要

Windows Socket ライブラリの終了処理を実行します。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan POP3 コンポーネント・エラー・メッセージをご参照ください。

SAMPLE

NetMan POP3 Component エラー・コード一覧

メッセージ	値	概要
PE1101 既に初期化されています。	1101	すでに接続されている状態で Connect メソッドを呼び出しました。プログラムを見直して複数回 Connect メソッドをよびださないように修正してください。別のサーバーに接続する場合は先に Quit メソッドを呼び出して POP3 サーバーとの接続を終了させてください。
PE1102 TCP/IPの初期化に失敗しました。	1102	クライアント・パソコンのリソースが不足している場合にこのエラーになります。他に動作しているアプリケーションがある場合は止めて、パソコンをリブートして動作をご確認ください。またスワップ・エリアが十分に確保されているかと、TCP/IP が正常に設定されていることをご確認ください。他に TCP/IP を使うアプリケーションがあればそちらも正常に動作することをご確認ください。
PE1103 ソケットの生成に失敗しました。(％d)	1103	同上
PE1104 ホスト名の指定に誤りがあります。	1104	RemoteHost プロパティの指定がありません。正しい値を指定してください。
PE1105 ポート・ナンバーの指定に誤りがあります。	1105	RemotePort プロパティに0または負の値を指定していません。正しい値を指定してください。
PE1106 ホスト(RemoteHost)との接続に失敗しました。(％d)	1106	POP3 サーバーと接続できませんでした。RemoteHost, RemotePort プロパティの設定、

		POP3 サービスが動作サーバー側で動作していることをご確認ください。
PE1107 ソケットの生成に失敗しました。	1107	クライアント・パソコンのリソースが不足している場合にこのエラーになります。他に動作しているアプリケーションがある場合は止めて、パソコンをリブートして動作をご確認ください。またスワップ・エリアが十分に確保されているかと、TCP/IP が正常に設定されていることをご確認ください。他に TCP/IP を使うアプリケーションがあればそちらも正常に動作することをご確認ください。
PE1108 ホスト(RemoteHost)との接続に失敗しました。	1108	POP3 サーバーとのセッションを張ることは出来ましたが、最初のデータの受信に失敗しました。POP3 サーバーが正常に動作していることを確認してください。
PE1111 受信中にエラーが発生しました。(%d)	1111	socket のデータ受信に失敗しました。理由は付録 B のシステム・エラー・コードに関する記述を参照してください。
PE1112 送信中にエラーが発生しました。(%d)	1112	socket のデータ送信に失敗しました。理由は付録 B のシステム・エラー・コードに関する記述を参照してください。
PE1121 初期化されていません。	1121	Connect メソッドが呼び出されていないか成功していない状態で他のメソッドが先によびだされています。Connect メソッドを呼び出すようにプログラムを修正してください。
PE1122 既にユーザー承認が行われています。	1122	Authenticate メソッドが重複してよびだされています。Authenticate メソッドを1度だけ呼び出すように、プログラムを見直してください。

PE1123 ユーザーID の指定に誤りがあります。	1123	UserID プロパティが指定されていません。UserID プロパティを指定した後、再度実行してください。
PE1124 ユーザー名またはパスワードの指定に誤りがあります。	1124	POP3 サーバに PASS コマンドを発行して UserID,Password の検査をしましたが、失敗しました。POP3 サーバが動作しているサーバで正しいユーザー、パスワードが UserID,Password プロパティに指定されていることをご確認ください。
PE1131 ユーザー承認が行われていません。	1131	Authenticate メソッドが呼び出されていないか、成功していません。UserID,Password プロパティを確認してください。
PE1132 メールのリストを取得できませんでした。	1132	POP3 サーバからの否定応答のためリスト系のメソッドが正常に終了しませんでした。POP3 サーバが正常に動作していることをご確認ください。
PE1133 終了処理に失敗しました。	1133	POP3 サーバに QUIT コマンドを発行しましたが、否定応答が返ってきました。POP3 サーバが正常に動作していることをご確認ください。
PE1141 メッセージ番号に誤りがあります。	1141	引数で指定したメッセージ番号に対応するメールがサーバに存在しません。
PE1142 メールを取得できませんでした。	1142	サーバとの通信に失敗しているためメールを取得できませんでした。サーバの状態を検査してください。
PE1143 メールのヘッダを取得できませんでした。	1143	メッセージ・ヘッダの内容に問題がある場合などにこのエラーが発生します。送られてきたメールのヘッダ形式に問題があると思われます。
PE1144 分割メールを取得するための	1144	メッセージに表示されたメソッ

情報が不足しています。ListMessageID または ListMail メソッドを呼び出した後、再度実行してください。		ドを再度実行してください。
PE1145 分割メールの一部がまだ未着のため、メールを取得できません。	1145	分割して送られてきたメールのすべてが到着した状態ないとメールを受信できない状態です。分割メールがすべて揃った時点で再度メソッドを実行してください。
PE1151 ファイルにアクセス権がありません。	1151	DocOutFileName, DocOutFilePath プロパティで指定したファイルが他のプロセスで使用、アクセス権が無いなどの理由でアクセスが否定されました。ファイルを使っているプロセスが同時に実行されていないこと、ファイルの属性をご確認ください。
PE1152 ファイル名 (DocOutFileName) の指定に誤りがあります。	1152	DocOutFileName の指定がオペレーティングシステムの仕様で正しいファイル名とは認識されませんでした。正しいファイル名を指定してください。
PE1153 ディスクが一杯のため、ファイルに書き込めません。	1153	ディスクに空き容量がないのでファイルを書き込むことが出来ませんでした。空きのあるディスクを DocOutFilePath プロパティで指定してください。
PE1154 ファイルの書き込みエラーが発生しました。(nn)	1154	ファイルを出力中にオペレーティング・システムがエラーを返しました。理由は付録 B システムエラー・コードをご参照ください。
PE1155 パス名 (DocOutFilePath) の指定に誤りがあります。	1155	指定したパスがパソコンのディスクに存在することをご確認ください。また、オペレーティングシステムの仕様上、正しいファイルのパスが指定されていることをご確認ください。

PE1161	ホスト(RemoteHost)からの応答がありません。	1161	POP サーバーに命令を送信しましたが、否定応答が返ってきたため、メソッドの実行が完了しませんでした。
PE1171	メモリ不足です。	1171	プログラムの作業領域が確保できませんでした。不足しているメモリはアプリケーションのヒープ・メモリです。OSの状態によっては、同時に実行しているアプリケーションを終了させることで回避できる場合もあります。また、メモリの増設、スワップ・エリアの拡大でも回避出来る場合があります。
PE1172	MIME バージョンタグがありません。	1172	受信するヘダーの形式に不正な MIME バージョンが設定されています。
PE1173	APOP 認証に失敗しました。	1173	APOP 認証サーバーに接続して Authenticate メソッド実行をしましたが APOP 認証サーバーから否定応答が返されました。UserID/Password プロパティの設定が正しいかご確認ください。また APOP 認証サーバーではユーザー単位で APOP と Plain 認証の切り替えが出来るものがありますので、APOP 認証設定されているユーザーID であることをご確認ください。APOP 認証を使わない場合は AuthType プロパティの設定を PlainText 認証に変更してください。Pop Before SMTP サーバーをご利用になる場合は POP サーバーからの初期メッセージに APOP サーバーの場合と区別できない場合がありますのでその場合も AuthType プロパティには PlainText 認証を設定してください。
PE1174	出力ファイルをオープンできま	1174	SaveAttachment メソッドを実

<p>せん。</p>		<p>行していますが、指定された添付ファイルを書き込みモードでオープンすることが出来ませんでした。ファイル名が OS で利用できる名前がセットしており、DocOutPathName プロパティの設定が正しい値であることをご確認ください。</p>
<p>PE1175 UIDL 結果が不正です。</p>	<p>1175</p>	<p>UIDL コマンドの実行結果が RFC で定義されているものと異なります。お使いの POP サーバーでは UIDL コマンドが正しく機能しない可能性があります。TraceOption/TraceFileName 指定で出力された通信トレースファイル等で UIDL コマンドのレスポンスをご確認ください。</p>
<p>PE1176 該当する添付ファイルが存在しません。</p>	<p>1176</p>	<p>SaveAttachment メソッドで指定された添付ファイルがメール中に見つかりませんでした。</p>
<p>PE1177 SSL 接続が確立できませんでした。</p>	<p>1177</p>	<p>SSL サーバーへの接続が失敗しました。SSL サーバーに関連する設定をご確認ください。LastSSLError プロパティに接続が失敗した理由が保持されている場合があります。</p>

NetMan Imap4 Component リファレンス

この章では Imap4 プロトコルをサポートする NetMan Imap4 カスタムコンポーネントのプロパティ、メソッドについての説明をします。

プロパティ・リファレンス

ConnectTimeOut

接続タイムアウトをミリ秒単位で指定することが出来ます。0指定の場合はタイムアウト判定を実施しません。ソケット通信時に指定時間ブロックされた場合はメソッドがタイムアウトステータスを返します。

DocOutType

FetchMailAttachment メソッドを呼び出して添付ファイルを受け取る場合の宛先を指定します。以下の値が指定可能です。

値	シンボル	意味
0	nmDocOutProperty	Result プロパティに出力
1	nmDocOutFileName	MIME タグで指定されるファイルまたは、DocOutFileName プロパティで指定されるファイルに出力。

InitSocket

このプロパティはデフォルト True 設定の場合には Connect メソッド内部で Windows Socket ライブラリを初期化します。他のコンポーネントや Windows socket をライブラリを使い同時に複数の TCP/IP 接続が必要な場合には False 設定にして Connect メソッド内部からは Windows Socket の初期化を実行しない設定にして複数の同時 TCP/IP 接続を可能とします。別途 Windows Socket の初期化が必要な場合には Init メソッドを呼び出します。

LastErrorText

IMAP サーバーからのタグ付きレスポンスが OK 以外の場合、当プロパティに IMAP サーバーからのエラー・メッセージを保持します。エラーが発生した時点のメッセージが保持されますが、その後の正常実行で当プロパティの内容

が消去されることはありません。

LastOSError

ファイルとの I/O 時にシステムからエラーが返された値を保持します。値の詳細は Appendix.B のシステム・エラー・コードの値を参照してください。

LastSockError

メール・サーバーとの通信レベルのエラーが発生した場合に Winsock ライブラリが返すエラー値を保持します。保持される値は Winsock ライブラリ関数の呼出後、エラー値を検知した直後に WSAGetLastError 関数により返される値になります。通常のアプリケーションではこのプロパティを利用することはありません。弊社製品サポートのトラブル・シュート用に利用します。従って値の詳細は当マニュアルには掲載しませんが、Microsoft Developer Network C D 等の Winsock エラーに詳細の記載があります。

ListMailFormat

ListMail メソッドを実行する時にリストされるメールに関する情報の表示フォーマットを指定します。書式は以下のような形式です。

%フィールド名:/表示長

フィールド名として指定できる文字列はメールのヘッダーにある以下のようなタグです。

Cc	カーボン・コピー宛先
Date	メールの発信日
From	メールの送付元アドレス
Message-ID	メッセージ ID
Subject	メールの表題
To	メールの宛先アドレス
X-Mailer	メール作成に使ったアプリケーション名

以下は設定のサンプルです。

.ListMailFormat = "%From:/10 [%Date:/11] %Subject:/20"

MailBoxName

メール・ボックス名を指定します。UTF-7 によるエンコードをサポートしていますので、漢字を指定することができます。

例) INBOX.support.本日分

Numbering

True 値を設定した場合は、ListMail メソッドの結果の各行の先頭に番号を表示します。番号が必要無い場合や桁数で整形された番号を表示したい場合は当プロパティに False を設定して、ListMail メソッドの結果の各行に整形された番号等を設定してください。

NumOfMails

SelectMailBox メソッドで選択したメール・ボックスに存在するメールの数を保持します。

NumOfRecents

SelectMailBox メソッドで選択したメール・ボックスに存在するメールで前回、メール・ボックスを選択した時から新たに到着したメールの数を保持します。

NumOfUnseen

SelectMailBox メソッドで選択したメール・ボックスに存在するメールの中で内容を参照していない最初のメール番号を保持します。内容を参照していないメールとは IMAP の FETCH コマンドを発行していないメールという意味になります。当製品では FetchMail メソッド、RetrieveMailHeader メソッド、RetrieveMail メソッドを実行すると内容を参照したメールとして識別されます。

Password

接続先の IMAP サーバーに定義されるユーザーID のパスワードを指定します。

PumpMessage

旧 COM 版とのコンパチビリティのためにこのプロパティを残しています。旧版では当プロパティが True 設定の場合は長い時間を要するメソッド呼出時にウィンドウメッセージを処理して Windows の描画が止まる現象を回避できる機能を提供していましたが、.NET 環境ではスレッドの利用が可能になりましたので、この機能は廃止になりました。スレッドからコンポーネントのメソッドを呼び出すことで同現象を回避する方法でアプリケーションの実装をしてください。

QuickList

ListMail メソッドのデータ転送量を減らすことで、ListMail メソッドのパフォーマンスを向上させます。True 設定の場合はメール・ヘダーから Received:タグを除外してリストします。

ReceiveTimeOut

受信タイムアウトをミリ秒単位で指定することが出来ます。0指定の場合はタイムアウト判定を実施しません。ソケット通信時に指定時間ブロックされた場合はメソッドがタイムアウトステータスを返します。

RemoteHost

IMAP サーバーが動作しているホスト名を指定します。

RemotePort

IMAP サーバーと接続する TCP ポートを指定します。デフォルトは 143 です。

Result

FetchMail 等のメソッドの実行結果が保持されます。当プロパティに結果を保持するメソッドの詳細はメソッド・リファレンスをご参照ください。

SendTimeOut

送信タイムアウトをミリ秒単位で指定することができます。0指定の場合はタイムアウトを実施しません。ソケット通信時に指定時間ブロックされた場合はメソッドがタイムアウトステータスを返します。

SSLType

SSL 通信のタイプを指定します。デフォルトは nmNoSSL で通常のソケット通信を実行します。サーバー接続時に SSL 通信タイプの指定が必要なため、SSLType プロパティは Connect メソッドによりサーバーに接続する前に設定してください。Connect 後の SSLType 設定は次回 Connect 実行時まで有効になりません。

値	シンボル	意味
-1	NmNoSSL	通常のソケット通信
0	nmSSLV23	SSL V2.3 通信
1	nmSSLV1	SSL V1 通信
2	nmSSLV2	SSL V2 通信
3	nmSSLV3	SSL V3 通信

SSLVerify

SSL 通信時にベリファイの実行を設定します。デフォルトは nmNoVerify です。サーバー接続時にベリファイを実施するために、SSLVerify プロパティは Connect メソッドによりサーバーに接続する前に設定してください。Connect 後の SSLVerify 設定は次回 Connect 実行時まで有効になりません。

値	シンボル	意味
0	nmNoVerify	ベリファイを実行しません。(デフォルト)
1	nmVerify	ベリファイを実行します。

UserID

IMAP サーバーに接続可能な設定をしたユーザーIDを指定します。

ShowErrorMessage

メソッドの中から NetMan のエラーメッセージを表示する場合はこのプロパティに値 1 を設定します。エラー・メッセージは表示しない場合はこのプロパティに値 0 を設定します。エラー・メッセージを表示しない場合はメソッドからの戻り値を必ずチェックするようにします。

TraceFileName

通信トレースを出力するファイル名を指定します。ドライブ、パスを指定しない場合には実行時のカレント・ディレクトリに出力されますのでご注意ください。

TraceOption

通信トレースが必要な場合には当プロパティに値 1 をセットしてください。値 1 をセットした場合は、TraceFileName プロパティにトレース結果を保存するファイル名を指定してください。値 2 を設定した場合には値 1 と同様ですが、先頭に以下の形式で時間が出力されます。

HH:MM:SS sss

HH = 時間

MM = 分

SS = 秒

sss = 1/1000 秒

メソッド・リファレンス

Imap4 Component のメソッドの概要について説明します。

- ① メソッドはすべてリターン・コードを返します。
リターン・コードはすべて 2 バイトの整数型(VB では Int16 型)で正常終了時は 0 を返します。
- ② メール番号セットについて
メール・ボックス中のメールは番号で識別されます。メソッドのパラメータでメールを指定する場合「メール番号セット」で指定可能です。数値以外の指定可能文字は以下になります。

:	範囲指定。1:4 とした場合 1 から 4 番のメールを指定。
*	終端指定。10:* とした場合 10 番から最後のメールまでの指定。
,	複数指定。1,2,5,6 のように複数のメール番号を指定できます。

③ UIDL 指定について

メールセット番号を指定できる FetchMail メソッドのうち、最後のパラメータで UIDL 指定が出来るメソッドがいくつか提供されています。UIDL 指定でメールを取得する場合には UIDL パラメータを True 設定にしてメソッドを実行してください。

AppendToMailBox

書式

```
AppendToMailBox( MailBoxName As String, _  
                 Flags As String, _  
                 DateTime As String, _  
                 MailText As String ) As Int16
```

概要

新規メールボックスを作成します。

パラメータ

MailBoxName	新規に作成するメール・ボックス名
Flags	メール・フラグ値。括弧リストで指定。 例) (¥Seen)
DateTime	メール日付け。IMAP の内部日付け形式。
MailText	メールヘダーと本文を指定。RFC822 の形式で指定。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Visual C# サンプル・コード

```
string sm = "Date: Mon, 7 Feb 1999 21:52:25 -0800 (PST)¥r¥n" +  
           "From: info@xxx.co.jp¥r¥n" +  
           "To: support@yyyy.co.jp¥r¥n" +  
           "Subject: testing2¥r¥n" +  
           "Content-Type: TEXT/PLAIN; CHARSET=US-ASCII¥r¥n" +  
           "¥r¥n" +  
           "this is test¥r¥n"+
```

```

        "end.¥r¥n" +
        "¥r¥n";
short rc;
//
imap.RemoteHost = "sales.techknowledge.co.jp";
imap.UserID = "netman";
imap.Password = "password";
imap.TraceFileName = "c:¥¥tmp¥¥imap.log";
imap.TraceOption = MailLib.TraceOptionValues.nmTraceToFile;

rc = imap.Connect();
if(rc != 0)
{
    MessageBox.Show("error " + rc);
    return;
}
rc = imap.LogIn();
if(rc != 0)
{
    MessageBox.Show("error " + rc);
    return;
}
imap.MailBoxName = "INBOX";
rc = imap.SelectMailBox();
rc = imap.AppendToMailBox("INBOX", "(¥¥Seen)", "", sm);
if(rc != 0)
{
    MessageBox.Show("error " + rc);
}
rc = imap.LogOut();

```

Capability

書式

Capability() As Int16

概要

IMAP4 の CAPABILITY コマンドを発行し、結果を Result プロパティに取得します。CAPABILITY コマンドは接続した Imap サーバーが利用可能なコマンドの一覧を返します。当メソッドは Connect メソッドによる接続後であれば発行可能となります。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

CheckMailBox

書式

CheckMailBox() As Int16

概要

現在選択されているメール・ボックスに対して CHECK コマンドを実行します。CHECK コマンドにより Imap サーバーはサーバーのメモリとディスクの同期を取るなど、ハウスキープを実行します。Imap サーバーのインプリメンテーションにより、実行される事柄は変わりますが何も実行することがない Imap サーバーのインプリメンテーションでは NOOP コマンドと同じ動作になります。

パラメータ

なし

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

CloseMailBox

書式

CloseMailBox() As Int16

概要

現在選択されているメール・ボックスを非選択状態にします。メール・ボックスの削除状態(Delete フラグがセット)に設定されているメールは実際にサーバーから削除されます。

パラメータ

なし

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Connect

書式

Connect() As Int16

概要

IMAP4 サーバーに接続します。接続時に設定が必要なプロパティは Remote Host と RemotePort です。

パラメータ

なし

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Copy

書式

Copy(*MailSetNo* As String, *TargetMB* As String) As Int16

概要

現在選択されているメール・ボックスの指定されたメールをパラメータで指定したメール・ボックスにコピーします。

パラメータ

MailSetNo	メール番号セット
TargetMB	コピー先のメール・ボックス名

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

CreateMailBox

書式

CreateMailBox(*NewBox* As String) As Int16

概要

新規メールボックスを作成します。

パラメータ

NewBox	新規に作成するメール・ボックス名
--------	------------------

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Visual Basic. NET サンプル・コード

```
Dim rc As Int16
```

```
With Imap
```

```
    rc = .CreateMailBox("INBOX.trash")
```

```
    If rc <> 0 Then
```

```
        MsgBox "メール・ボックス作成失敗" & CStr(rc)
```

```
    End If
```

```
End With
```

DeleteMail

書式

```
DeleteMail( MailSetNo As String ) As Int16
```

概要

指定したメールを削除状態にします。Imap サーバーのメール・ボックスから実際に削除するには Expunge メソッドを実行します。

パラメータ

MailSetNo	削除するメール番号セット。
-----------	---------------

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

DeleteMailBox

書式

DeleteMailBox(MailBox As String) As Int16

概要

既存メールボックスを削除します。

パラメータ

MailBox	削除するメール・ボックス名
---------	---------------

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

ExamineMailBox

書式

ExamineMailBox() As Int16

概要

MailBoxName プロパティで指定されるメール・ボックスを読みこみ専用として選択します。読みこみ専用でメール・ボックスが選択された場合には例えば Delete メソッドを使って Deleted フラグをメール設定することはできなくなります。このメソッドの実行が成功するとプロパティ Exsists,Recent,UnSeen の内容が更新されます。SelectMailBox メソッドとは違い、メールから Recent フラグは削除されません。

パラメータ

なし

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Expunge

書式

Expunge() As Int16

概要

EXPUNGE コマンドを現在選択されているメール・ボックスに対して発行します。EXPUNGE コマンドにより Deleted フラグが設定されているメールはサーバーのメール・ボックスから削除されます。

パラメータ

なし

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

FetchMail

書式

```
FetchMail( MailSetNo As String, _  
           Item As String) As Int16
```

```
FetchMail( MailSetNo As String, _  
           Item As String, _  
           bUIDL As Boolean) As Int16
```

```
FetchMail( MailNo As Int16, _  
           ByRef MailObject As Mail) As Int16
```

```
FetchMail( MailNo As Int16, _  
           ByRef MailObject As Mail _  
           bUIDL As Boolean) As Int16
```

概要

第1, 2フォームのメソッドは FETCH コマンドを発行します。実行結果は Result プロパティに保持されます。第3, 4フォームのメソッドは Mail オブジェクトにメール情報を取得します。

パラメータ

MailSetNo MailNo	メール番号セットまたはメール番号
Item	メール・アイテム名 例) (BODY[1]) メール本文を取得。 (BODY[1.MIME]) メール本文の MIME タグを取得。 (FLAGS BODY[HEADER.FIELDS (DATE FROM)]) ヘッダーの DATE: と FROM:を取得。

	詳細は RFC2060 の FETCH コマンドの message data item names をご参照ください。
MailObject	当コンポーネント共通のメール情報を保持するメールオブジェクト
bUIDL	MailSetNo または MailNo によるメール指定を UIDL で指定する場合はこのパラメータを True を指定してください。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Visual C#サンプルコード

```

short rc;
imap.MailBoxName = "INBOX";

rc = imap.SelectMailBox();
MailLib.Mail mail;

rc = imap.FetchMail(4, out mail);
if(rc != 0)
{
    MessageBox.Show("error " + rc);
}
else
{
    System.Diagnostics.Debug.WriteLine(mail.Body);
    System.Diagnostics.Debug.WriteLine(mail.Header);
}

```

FetchMailBody

書式

```

FetchMailBody ( MailNo As Long, _
                ByRef MailBody As String ) As Int16

```

概要

現在選択されているメール・ボックス中で指定したメール番号のメールの本文を取得します。メール本文にはアタッチメントは含まれません。

パラメータ

MailNo	メール番号。
MailBody	メール本文

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

FetchMailAttachment

書式

FetchMailAttachemnt(*MailNo* As Long, _
PartNo As Long) As Int16

概要

現在選択されているメール・ボックス中で指定したメール番号のメールに添付されているデータを取得します。

出力結果は DocOutType プロパティの設定により変更可能です。ファイル出力を選択した場合には Content-Disposition:タグの name 指定、または、Content-Disposition:タグの filename 指定のファイル名で DocOutFilePath プロパティで指定されたディレクトリにファイルを出力します。

DocOutType プロパティでプロパティ出力を選択した場合には、Result プロパティに結果が保持されますが、バイナリ・ファイルの場合、ご利用になる言語によってはバイナリ・データがただしく読みこめない場合もありますので、FetchMailStructure メソッドにて添付ファイルの中身がバイナリ・データではないことを確認してからプロパティに出力してください。

パラメータ

MailNo	メール番号。
PartNo	メールの添付ファイル番号。ベース値は1になります。通常のメールでは値1を指定した場合、メールの本文になります。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

FetchMailHeader

書式

FetchMailHeader(*MailNo* As Long, _
ByRef *Header* As String) As Int16

概要

現在選択されているメール・ボックス中で指定したメール番号のメール・ヘッダーを取得します。

パラメータ

MailNo	メール番号。
Header	メール・ヘッダー

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

FetchMailStructure

書式

FetchMailStructure (*MailNo* As Long, _
ByRef *PartCount* As Long, _
ByRef *ContType()* As String, _
ByRef *FileNames()* As String, _
ByRef *Encoding()* As String, _
ByRef *Sizes()* As String) As Int16

FetchMailStructure (, *MailNo* As Long _
ByRef *ContType()* As String, _
ByRef *FileNames()* As String, _
ByRef *Encoding()* As String, _
ByRef *Sizes()* As String) As Int16

概要

現在選択されているメール・ボックス中で指定したメール番号のメールの構造情報を取得します。

パラメータ

MailNo	メール番号。
PartCount	MIME マルチ・パート・メールの場合、パートの数を返します。
ContType	各パートの Content-Type:タグの値を配列で返します。インデックスは 0 になります。
FileNames	各パートの Content-Type:タグに name=の指定値を返します。インデックスは 0 になります。
Encoding	各パートの Content-Transfer-Encoding:タグに指定される値を返します。インデックスは 0 になります。
Sizes	各パートのサイズを返します。サイズは IMAP サーバーからの値で、テキストの状態でのサイズです。BASE64/Quoted-Printable でエンコードされたパートの場合、デコード後のサイズとは異なります。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Visual C# サンプル・コード

```
int count;
string [] contType;
string [] names;
string [] encodings;
string [] sizes;

imap.DocOutFilePath = "c:¥¥temp¥¥";
imap.DocOutFileName = "weekly_sales.dat";
imap.DocOutType = MailLib.DocOutTypeValues.nmDocOutFileName;
rc = imap.FetchMailStructure( 2, out count,
    out contType, out names, out encodings, out sizes);
if(rc != 0)
{
    MessageBox.Show("error " + rc);
}
else
{
    string msg;
    int i;
    for(i=0; i< count; i++)
```

```

    {
        msg = String.Format("{0} {1} {2} {3}",
            contType[i], names[i], encodings[i], sizes[i]);
        System.Diagnostics.Debug.WriteLine(msg);
    }
}

```

Imap4Command

書式

Imap4Command(*Command* As String) As Int16

概要

任意の Imap4 コマンドをサーバーに送信します。結果は文字列で Result プロパティに受信します。

パラメータ

Command	Imap4 コマンド。詳細は Imap4 Rev1 RFC などをご覧ください。
---------	--

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Visual C# サンプル・コード

```

imap.MailBoxName = "INBOX";
rc = imap.SelectMailBox();

rc = imap.Imap4Command("UID SORT (REVERSE DATE) 1:*");
if (rc != 0)
{
    MessageBox.Show("error " + rc);
}

```

Init

書式

Init() As Int16

概要

Windows Socket ライブラリの初期化を実行します。InitSocket プロパティを False 設定にした場合、Connect メソッド呼出前に場合に応じて当メソッドを呼び出します。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

IsConnected

書式

IsConnected() As Boolean

概要

Imap サーバーとの接続状態を返します。

リターン・コード

接続中には True 値を返します。非接続時には False を返します。

ListMail

書式

ListMail(NumOfMails As Long, _
ByRef TagList() As String, _
ByRef FlagList() As String) As Int16

ListMail(ByRef TagList() As String, _
ByRef FlagList() As String) As Int16

概要

現在選択されているメール・ボックスに保持されているメールとそのフラグをリストします。メソッドの実行前に ListMailFormat にリストするヘダー項目と表示サイズの指定が必須となります。

パラメータ

NumOfMails	リストされたメールの数
TagList	メールのリスト。文字列配列でベースは0になります。N

	umbering プロパティに true を設定した場合には先頭にメール番号が付加されます。
FlagList	メールに設定されたフラグのリスト。LONG 型の配列でベースは0になります。保持されるフラグの値は Apen dix.C をご参照ください。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Visual C# サンプル・コード

```

Int32 NumOfMails;
String [] MailList;
Int32 [] FlagList;

imap.Numbering = true;
imap.ListMailFormat = "%Subject:/20 %From:/15 %Date:/20";
rc = imap.ListMail(out NumOfMails, out MailList, out FlagList);
if(rc != 0)
{
    MessageBox.Show("error list mail " + rc);
}

```

ListMailBox

書式

```
ListMailBox( MailBoxName As String, _
            SearchArg As String, _
            ByRef MailBoxList() As String) As Int16
```

```
ListMailBox( MailBoxName As String, _
            SearchArg As String, _
            ByRef MailBoxList() As String, _
            ByRef NumOfMailBoxes As Long ) As Int16
```

概要

メール・ボックスをリストします。

パラメータ

MailBoxName	参照するメール・ボックス名
-------------	---------------

SeachArg	検索条件指定。ワイルド・カード文字としては*と%を指定できます。前者は階層を含んで検索条件を指定できますが、後者は階層を含みません。
MailBoxList	リスト結果
NumOfMailBoxes	リストされたメール・ボックスの数

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Visual C# サンプル・コード

```
int count;
string [] result;
short rc;
rc =imap.ListMailBox("", "%", out result, out count);
if(result != null)
{
    for(i=0; i<result.Length; i++)
    {
        listBox1.Items.Add(result[i].ToString());
    }
}
```

ListSubscribed

書式

```
Object.ListSubscribed( MailBoxName As String, _
    SearchArg As String, _
    ByRef MailBoxList() As String ) As Int16
```

```
Object.ListSubscribed( MailBoxName As String, _
    SearchArg As String, _
    ByRef MailBoxList() As String, _
    ByRef NumOfMailBoxes As Long ) As Int16
```

概要

購読状態(Subscrib/Active)に設定されたメール・ボックスをリストします。

パラメータ

MailBoxName	参照するメール・ボックス名
SeachArg	検索条件指定。ワイルド・カード文字としては*と%を指定できます。前者は階層を含んで検索条件を指定できますが、後者は階層を含みません。
MailBoxList	リスト結果
NumOfMailBoxes	リストされたメール・ボックスの数

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

LogIn

書式

Object.LogIn() As Int16

概要

Imap4 サーバーにログ・インします。ログ・イン時に参照されるプロパティは UserID, Password になります。このメソッドの実行前に Imap4 サーバーに Connect メソッドにて接続を完了している必要があります。

パラメータ

なし

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

LogOut

書式

Object.LogOut() As Int16

概要

Imap4 サーバーからログ・アウトします。アプリケーション・プログラムの終了前にかかわらず呼び出して、Imap4 サーバーとの接続を遮断してください。

パラメータ

なし

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

MergePartialMails

書式

Object.MergePartialMails(MsgNo As Long) As Int16

概要

分割して送信されたメールをひとつのメールに統合します。Imap のフォルダーでは分割されたメールは各々がフォルダーにリストされますが、一つにまとめることでリストが見やすくなります。また、分割されたメールの添付ファイルを取り出す場合にもこのメソッドを使ってから FetchMailAttachment にて添付ファイルを取り出すこととなります。

統合されたメールは新しいメッセージ番号が割り振られ、指定された分割・メールはすべてフォルダーから削除マークされ EXPUNGE が実行されます。

パラメータ

MsgNo	分割されたメールの先頭のメッセージ番号。先頭以外のメール番号は指定してもエラーとなります。
-------	---

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

注意

EXPUNGE が実行されるのでこのメソッドの前に削除フラグを設定したメールは削除が実行されますのでご注意ください。

分割されたメールとは通常以下のような Content-Type:指定がされているメールです。

```
Content-Type: message/partial; number=1; total=30;  
Id="990909123.1234@ibaniez.johnscofield.com"
```

Namespace

書式

Object.Namespace() As Int16

概要

NAMESPACE コマンドを実行しサーバーのフォルダー情報を取得します。NAMESPACE コマンドの実行結果は Result プロパティに保持されます。

NAMESPACE コマンドが利用可能なサーバーは Capability メソッドの実行結果により、確認することができます。

NAMESPACE メソッドの実行結果は IMAP サーバー・ソフトウェアの種類や設定によって異なりますので、IMAP4 RFC や IMAP サーバー・ソフトウェアのマニュアル等をご確認ください。

パラメータ

なし

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Noop

書式

Object.Noop() As Int16

概要

NOOP コマンドを実行しサーバーとの接続状態を確認します。NOOP コマンドの実行結果は Result プロパティに保持されます。

実行結果は Imap4 サーバーの状況や IMAP サーバー・ソフトウェアの種類によって異なりますので、IMAP4 RFC や IMAP サーバー・ソフトウェアのマニュアル等をご参照ください。

パラメータ

なし

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

RenameMailBox

書式

*Object.RenameMailBox(OldName As String, _
NewName As String) As Int16*

概要

メール・ボックス名を変更します。INBOX の名前変更の場合、新しいメール・ボックスに INBOX のすべてのメールが転送され、新しいメール・ボックスが作成されますが、INBOX は削除されずに残されます。(IMAP の仕様)

パラメータ

OldName	旧メール・ボックス名
NewName	新メール・ボックス名

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Search

書式

*Object.Search(SearchCriteria As String, _
ByRef Results() As String) As Int16*

*Object.Search(SearchCriteria As String, _
ByRef Results() As String, _
ByRef NumOfResults As Int32) As Int16*

概要

選択されているメール・ボックスにあるメールを検索条件に従って検索します。検索結果はメール番号の配列(インデックスは0)で返されます。

パラメータ

SearchCriteria	検索条件。検索キーと共に指定します。検索キーについてはこの章の「検索キーについて」をご参照ください。
Results	検索結果
NumOfResults	検索結果数

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

検索キーについて

Imap4 のサーチ・キーについて記載します。

ALL	メール・ボックスにあるすべてのメール。
ANSWERED	¥Answered フラグがセットされたメール。
BCC string	String で指定される文字列が BCC に指定されるメール。
BEFORE date	Date で指定される日付けより以前のメール。
BODY string	本文に string が含まれるメール。
CC string	String で指定される文字列が CC に指定されるメール。
DELETED	¥Deleted フラグがセットされたメール。
DRAFT	¥Draft フラグがセットされたメール。
FLAGGED	¥Flagged フラグがセットされたメール。
FROM string	String で指定される文字列が FROM に指定されるメール。
HEADER field string	ヘダーの field に string が存在するメール。
KEYWORD flag	Flag で指定されるキーワードがセットされたメール。
LARGER n	メールのサイズが n バイト以上のメール。
NEW	¥Recent フラグがセットされたメール。
NOT search-key	Search-key にマッチしないメール。
OLD	¥Recent フラグがセットされていないメール。
ON date	Date で指定された日付けのメール。
OR key1 key2	Key1 または key2 にマッチするメール。
RECENT	¥Recent フラグがセットされたメール。
SEEN	¥Seen フラグがセットされたメール。
SENTBEFORE date	メール・ヘダーの Date:が date 以前のメール。
SENTON date	メール・ヘダーの Date:が date のメール。
SENTSICNE date	メール・ヘダーの Date:が date 以降のメール。
SINCE date	Date 以降の日付けのメール (INTERNALDATE)
SMALLER n	サイズが n より小さいメール。
SUBJECT string	Subject:に string が含まれるメール。
TEXT string	メール・ヘダーまたはメール・ボディに string が含まれるメール。
TO string	TO フィールドに string が含まれるメール。
UID msgset	Msgset に含まれるメール。
UNANSWERED	¥Answered フラグを持たないメール。
UNDELETED	¥Deleted フラグを持たないメール。
UNDRAFT	¥Draft フラグを持たないメール。

UNFLAGGED	¥Flagged フラグを持たないメール。
UNKEYWORD flag	Flag で指定されるキーワードがセットされないメール。
UNSEEN	¥Seen フラグを持たないメール。

注意

検索文字列指定に漢字が含まれる場合はダブル・クォーテーションで囲んで文字列を指定してください。

Visual C# サンプル・コード

```
Int32 [] result;
```

```
Int32 count;
```

```
rc = imap.Search("BODY ¥井上¥", out result, out count);
if (rc != 0)
{
    MessageBox.Show("error hdr " + rc);
}
```

SelectMailBox

書式

Object.SelectMailBox() As Int16

概要

MailBoxName プロパティで指定されるメール・ボックスを選択します。選択状態のメール・ボックスにはフラグやメールの内容を書きこむことができます。このメソッドの実行が成功するとプロパティ Exsists,Recent,UnSeen の内容が更新されます。Recent フラグは削除されます。メール・ボックスの状態を変更したくない場合は Examine メソッドをご利用ください。Examine メソッドでメール・ボックスを選択した場合には Recent フラグは削除されません。

パラメータ

なし

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Status

書式

```
Object.Status(MailBoxName As String, _  
              ByRef NumOfMails As Int32, _  
              ByRef NumOfRecents As Int32, _  
              ByRef NumOfUnseen As Int32) As Int16
```

概要

MailBoxName パラメータで指定されるメール・ボックスの状態を取得します。取得できる情報はメール数、新着メール数、未読メール数となります。このメソッドの実行でメール・ボックスの状態が更新されることはありません。また、メール・ボックスが選択状態にはなりません。メール・ボックスを選択する場合には SelectMailBox または ExamineMailBox メソッドを使ってください。

パラメータ

MailBoxName	状態を取得するメール・ボックス名を指定。
NumOfMails	メール・ボックスに存在するメールの数
NumOfRecents	メール・ボックスに存在する新着メールの数
NumOfUnseen	メール・ボックスに存在する未読メールの数

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Store

書式

```
Object.Store(MailNumSet As String, Cmd As String) As Int16
```

概要

現在選択されているメール・ボックスの指定されたメールに STORE コマンドを実行します。STORE コマンドにより指定されたメールのフラグ値を変更することができます。

パラメータ

MailBox	削除するメール番号セット。
Cmd	Store コマンドのフラグ値。+または-FLAGS の

	後に括弧リストでフラグ値を指定します。詳細は IMAP4 Rev1 の RFC をご参照ください。
--	---

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Visual Basic .NET サンプル・コード

```
Dim Rc As Int16
```

```
With Imap41
```

```
    ‘ 削除状態から復帰します。
```

```
    Rc = .Store(“1:4”, “-FLAGS (¥DELETED)”)
```

```
    If Rc <> 0 Then
```

```
        MsgBox “Store fail “ & CStr(rc)
```

```
    End If
```

```
End With
```

SubscribeMailBox

書式

```
Object.SubscribeMailBox(MailBoxName As String) As Int16
```

概要

指定したメール・ボックスを購読状態(subscribe/active)に変更します。

パラメータ

MailBoxName	購読状態にするメール・ボックス名
-------------	------------------

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

Terminate

書式

```
Terminate () As Int16
```

概要

Windows Socket ライブラリの終了処理を実行します。

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan IMAP4 コンポーネント・エラー・メッセージをご参照ください。

UnSubscribeMailBox

書式

Object.UnSubscribeMailBox(*MailBoxName* As String) As Int16

概要

指定したメール・ボックスを購読状態(subscribe/active)を解除します。

パラメータ

MailBoxName	購読状態を解除するメール・ボックス名
-------------	--------------------

リターン・コード

正常終了ならば値0が返されます。0以外の値については NetMan Imap4 コンポーネント・エラー・メッセージをご参照ください。

NetMan Imap4 Component エラー・コード一覧

NetMan Imap4 Component のメソッドが返すエラー値についての説明です。

メッセージ	値	概要
IM001 すでにサーバーに接続されています。	2001	Imap4 サーバーに接続中に Connect メソッドが呼び出されました。
IM002 RemoteHost が不正です。	2002	RemoteHost プロパティの設定が空、または不正な文字が含まれています。
IM003 RemotePort が不正です。	2003	ポート番号 RemotePort プロパティにマイナス値または値 0 が設定されています。
IM004 socket 初期化に失敗	2004	TCP/IP ソケットを初期化できません。TCP/IP プロトコルのインストールまたは設定を今一度ご確認ください。
IM005 socket バージョンが違います。	2005	TCP/IP ソケット・ライブラリのバージョンが合致しません。TCP/IP プロトコルのインストールをご確認ください。
IM006 socket 生成エラー	2006	TCP/IP ソケットを生成できません。通信リソースの不足が考えられます。他に実行中のネットワーク・アプリケーションがリソースを消費していると考えられます。リポート後に作成されたアプリケーションのみ実行して動作をご確認ください。他のアプリケーションが同時に実行されていない場合には TCP/IP プロトコルのインストールが不正と思われるので今一度ご確認ください。
IM007 サーバーのアドレスが取得できません。	2007	RemoteHost プロパティで指定される IMAP サーバのアドレスが取得できません。DNS 等、アドレスを解決するプロトコルに問題がありますので、今一度 TCP/IP の設定をご確認いただくか、RemoteHost プロパティに設定された文字列をご確認ください。
IM008 サーバー接続に失敗	2008	RemoteHost で指定された Imap

		サーバーに接続できませんでした。サーバー・アドレスは解決しているのに、該当サーバーで imap4 サーバーが正常に動作していることを今一度ご確認ください。エラーの詳細は LastSocketError プロパティに保持されます。
IM009 送信失敗	2009	TCP/IP 通信で送信中にエラーとなりました。エラーの詳細は LastSocketError プロパティに保持されています。
IM010 受信失敗	2010	TCP/IP 通信で受信中にエラーとなりました。エラーの詳細は LastSocketError プロパティに保持されています。
IM011 サーバーが BAD 応答を返しました。	2011	IMAP のコマンドを実行したところ、BAD 応答がサーバーから帰りました。今一度実行したメソッドのパラメータをご確認ください。
IM012 サーバーが NG 応答を返しました。	2012	IMAP のコマンドを実行したところ、NG 応答がサーバーから帰りました。今一度実行したメソッドのパラメータをご確認ください。
IM013 サーバーが NO 応答を返しました。	2013	IMAP のコマンドを実行したところ、NO 応答がサーバーから帰りました。今一度実行したメソッドのパラメータをご確認ください。
IM014 通信バッファの上限に達しました。	2014	内部的に用意している通信バッファ・サイズを超えるデータを扱おうとしています。原因はソフトウェアがプロトコルに違反していることが想定されます。サポートに通信トレースを添えてご連絡ください。
IM015 ListMailFormat の指定が不正です。	2015	ListMailFormat プロパティの指定を今一度ご確認ください。
IM016 メモリ不足です。	2016	プログラムを実行するのに十分なメモリが確保できませんでした。メモリが増設できない場合はスワップ・エリアのサイズを増やす等をお試しください。
IM017 メール本文が空です。	2017	メール本文の指定がありません。本文を指定してください。
IM018 サーバーからの応答に期待	2018	サーバー側のレスポンスに送られ

した値(サイズ)が見つかりません。		てくるデータのサイズがありません。サーバーのプロトコル・エラーと思われます。トレースをとり、ご確認いただくか、サポートにご相談ください。
IM019 メール本文の形式を解読できません。	2019	メールの構造を解析できませんでした。トレースを取り、サポートにご相談ください。
IM020 メールの部分番号がただしくありません。	2020	分割されて送られてきたメールの先頭を MergePartialMails メソッドに指定していないか、メールの部分番号に不正な値が設定されていると思われます。MergePartialMails のメッセージ ID 指定を今一度ご確認ください。
IM021 メッセージ ID が不正または存在しません。	2021	分割されて送られてきたメールのタグが不正です。送信側のメール・ソフトまたはメール・サーバーに問題があると思われます。
IM022 ファイルが開けません。	2022	添付ファイル等を出力するファイルを開くことができませんでした。DocOutFilePath,DocOutFileName プロパティ設定を今一度ご確認ください。LastOSError プロパティにエラーの詳細があります。
IM023 ファイルの書き込みでエラーが発生しました。	2023	ディスクの破損や容量不足などのため、ファイル書き込みに失敗しました。LastOSError プロパティにエラーの詳細が保持されています。
IM024 Content-Type の指定がありません。	2024	Content-Type 指定の無いメールなので、指定されたメソッドの実行を完結することができません。トレースを確認して不正なメールの原因をご確認ください。
IM025 Message/partial で指定される分割メールではありません。	2025	MergePartialMails メソッドのパラメータで指定されたメールは、Content-Type 指定が Message/partial ではありません。分割メールをパラメータとして指定してください。
IM026 分割メールの総数が不正	2026	Message/partial の MIME タグで分割メールが送られてきていますが、total=のタグが存在しないか、

		不正な値が登録されています。
IM027 分割メールの番号が不正	2027	Message/partial の MIME タグで分割メールが送られてきていますが、number=のタグが存在しないか、不正な値が登録されています。

SAMPLE

共通クラスリファレンス

ここでは SmtP/Pop3/Imap4 コンポーネントで共通に使われている共通クラスについてのリファレンスです。Attachment クラスと Mail クラスが定義されます。どちらもメールデータ情報を保持するクラスです。基本的にはプロパティのみ定義され、メソッドは現在のところ定義されていません。

Attachmentクラス

MIME マルチパート形式のメール情報を保持します。後述の Mail クラスの中には Generics リストで定義されます。

ContentDescription

Content-Discription: タグの内容を文字列型で保持します。このタグを付けずに添付ファイルを送信するメールソフトは多いです。このタグが付けられていない場合はヌル値を保持します。

ContentDisposition

Content-Disposition: タグの内容を文字列型で保持します。添付ファイルの場合は attachment という文字列が保持されるることが多いです。

ContentTansferEncoding

MIME パートデータのエンコード方法を文字列型で保持します。Base64, Quoted-Printable などの文字列になります。テキストパートの場合は 7Bit のような文字列が設定されます。

ContentType

Content-Type: タグの内容が文字列型で保持されます。Text/plain, text/html, application/octet-stream などが代表的な設定例です。

FileName

Content-Type: の Name タグの値または Content-Disposition: タグの Filename タグの値を保持文字列型としてします。IMAP4/POP3 などの受信系コンポーネントではファイルのパスは無く、ファイル名のみ保持します。SMTP コンポーネントに指定する場合はフルパス指定が必要です。また、添付ファイルではなくテキストパート(本文など)の場合はヌルが設定されます。

Mailクラス

Mail 情報を保持するクラスです。Imap/Pop/SMTP 各クラスで共通に情報を取得したり、設定することが出来ます。受信メール情報をリプライする場合には宛先情報を逆に設定するなどの注意が必要です。

Attachments

添付ファイル情報 Attachment クラスを Generics リストで保持します。初期値はヌルになります。

Bcc

SMTP コンポーネントでメール送信する場合に Bcc 宛先メールアドレスを Generics リスト文字列で指定します。Imap4/Pop3 の受信では設定されません。

Body

メール本文を文字列で保持します。MIME マルチパートのメールの場合最初に見つかった Text/Plain パートまたは Text/Html パートを保持します。

BodyContentType

上記 Body の Content-Type:タグ設定がある場合に ContentTypeValues 定義値のいずれかがセットされます。SMTP コンポーネントで送信する場合は Content-Type の指定が出来ます。デフォルト設定は text/plain となります。

Cc

Cc 指定されたアドレスを Generics リスト文字列で保持します。送受信共に有効なプロパティです。

From

送信元のメールアドレスを文字列で保持します。

Header

メールヘダー全体を文字列で保持します。Imap4/Pop3 コンポーネントで受信した場合にはメールヘダーはデコードされない状態で保持されます。

ReadEml

Eml 形式のファイルを読み込み Mail オブジェクトを生成するメソッドです。メソッドのパラメータとして Eml ファイル名を指定します。当メソッドを実行する前のプロパティ設定は Eml の内容で上書きされます。EML ファイルの行末は CR+LF で設定されたファイルのみをサポートしています。以下はサンプルコードです。

```
mail.ReadEml(@"¥mails¥sample.eml");  
Console.WriteLine(mail.From);  
Console.WriteLine(mail.To);  
Console.WriteLine(mail.Body);
```

To

宛先アドレスを Generics リスト文字列で保持します。

WriteEml

Mail オブジェクトの内容を eml ファイル形式で出力します。メソッドのパラメータとして Eml ファイル名を指定します。Mail オブジェクトの Header/Body プロパティの設定内容を EML ファイルに出力します。POP3 コンポーネントの RetrieveMail を使う場合は Format プロパティに nmRaw を指定します。

```
pop.Format = MailLib.ContentFormat.nmRaw;  
pop.Connect();  
pop.Authenticate();  
pop.RetrieveMail(1,out mail);  
pop.Quit();
```

```
mail.WriteEml(@"c:\mail\result.eml");
```

SAMPLE

Appendix A. FAQ よくあるご質問

以下は COM コンポーネント版の NetMan Mail 関連製品におけるサポート FAQ および技術情報をまとめたものです。最新の技術情報が弊社 Web サーバーに掲載されることもありますので、以下の URL もご参照ください。

<http://www.techknowledge.co.jp/techinfo.shtml>

Smtplib コンポーネントにおける宛先の上限值

rfc821 では宛先は 100 件までと仕様が定められています。また、コマンド行の最大値は 512 バイトと定められています。

NetMan Smtplib Component で To,CC,Bcc のプロパティは RCPT TO コマンドのパラメータになりますので、これら宛先の合計が 100 件までが上限となります。また TO,CC に関しましてはプロパティの内容の先頭に TO:または CC:を付けてそれぞれのプロパティに設定された文字列を連結していますので、この文字列のサイズ 512 バイトが上限となります。BCC に関しましては BCC: のタグは設定されませんので行の上限値には関係しません。

SMTP AUTH の認証方式はどれがよいか

当製品の SMTP AUTH の認証方式として PLAIN/LOGIN/CRAM-MD3 の3方式をサポートしています。SMTP 認証はメールプロバイダーによっては複数の方式をサポートしている場合があるようです。お客様からどの認証方式が良いかご質問がありますが、この3方式の中では CRAM-MD5 が暗号化強度が高いです。次に LOGIN 認証となります。PLAIN 認証はパスワードが平文でネットワークに流れるのであまりお勧めしません。

大量のメールを送信する際の注意事項

Visual Basic 6 ではプログラムからはスレッドを使用することが出来ないため、DoEvents などを使って GUI のハングを回避する必要がありましたが、.NET Framework ではスレッドが簡単に利用できるようになりました。時間のかかるサイズの大きなメールを送信したり、スプールに沢山のメールが入っているのを取得するような場合にはスレッドを使うことでメインのフォームの動作が止まるのを回避することが出来ます。

POP Component でメールを受信したがスプールにメールが残っている

POP3 の仕様では DELE コマンドを発行するまでは、サーバーのメール・スプールから受信したメールを削除しません。POP3 Component の RetrieveMessage メソッドでメールを受信した状態では DELE コマンドは発行していないので、サーバーのメール・スプールにはまだメールが残った状態になります。DELE コマンドを発行する Delete メソッドを呼び出すことで、メールをサーバーのメール・スプールから削除することができます。

未読・既読の判断をしたい

一度 POP3 サーバーからメールデータを読み出したメールについてはメールヘッダーの STAUTS:タグに R や RO のステータスが設定され STATUS タグが存在したメールについては既読と判断できます。ほとんどの POP3 サーバーでこのタグがサポートされているようですが、一部この機能が実装されていないサーバーもありますのでお使いの POP3 サーバーで確認して STATUS タグの利用を検討してください。

RetrieveMessage した本文の拡張子がパソコン毎に異なる

メール本文のように Content-Type 指定に”plain/text”があり、ファイル名の指定や Content-Disposition:タグが無い場合、NetMan Pop コンポーネントはシステムのレジストリを検索して ContentType からファイルの拡張子を決定します。

```
HKEY_CLASSES_ROOT¥.text “Content Type” “text/plain”  
HKEY_CLASSES_ROOT¥.txt “Content Type” “text/plain”
```

大多数のパソコンには拡張子.txt の設定のみされているのが通常です。このような場合、本文を RetrieveMessage メソッドで出力すると.txt 拡張子が添付され、PART00001.txt のようなファイル名で出力されます。ところが.text の設定もされているパソコンの場合には先に.text 設定が読み込まれ、PART00001.txt のようなファイル名となります。対処方法はどちらか統一したい拡張子の登録のみ残すことです。REGEDIT.EXE でのレジストリの編集はシステム設定を破壊する恐れがあるので、十分注意して実行してください。また、プログラムからレジストリを編集するには REG で始まる WIN32 API で可能です。詳細は Microsoft Developer Network CD 等を参照してください。

IMAP Component の FetchMail コマンドで NG ステータスが返される

FetchMail コマンドの第2パラメータは括弧の前にスペースが無い場合に NG

ステータス(-2)が返されます。以下のような場合、FIELDS の後のスペースが無いとサーバーからエラーが返されますのでご注意ください。

```
(BODY[HEADER.FIELDS (FROM TO)])"
```

IMAP サーバーによるレスポンスの相違について

Imap サーバー・ソフトウェアは基本的に RFC に準拠したコマンドを受け付けますが、RFC で規定されていない部分について、たとえばサーバーからのレスポンスについては動作が異なる場合があります。

例えば uw-imapd ではリード・オンリーで選択したメール・ボックスでメールを削除した場合は、OK レスポンスを返しますが cyrus-imapd ではメールを削除した時点で BAD レスポンスが返されます。

また、UW-IMAP では Expunge をリード・オンリーのメール・ボックスに対して実行しても OK レスポンスが返るので注意が必要です。

以下は uw-imap 4.5 での Expunge 実行のログです。

```
nm16 STORE 2 +FLAGS (¥Deleted)
* 2 FETCH (FLAGS (¥Seen ¥Deleted ¥Flagged))
nm16 OK STORE completed
nm17 EXPUNGE
* NO Expunge ignored on readonly mailbox
nm17 OK EXPUNGE completed
```

Outlook Express 2000 で漢字メール・ボックス名が表示されない

Outlook Express 2000 では IMAP の漢字メール・ボックス名に対応していないため、漢字を含むメール・ボックスを CreateMailBox で作成しても、表示されません。階層の一部分に漢字が使っている場合、その上のレベルのメール・ボックスがすべて英数字でも表示されません。

INBOX.support.

INBOX.support.本日分

上記のようなメール・ボックスが存在する場合、INBOX.support も表示されないのをご注意ください。

IMAP Component や Orange Soft の WinBiff 等、UTF-7 によるエンコードに対応したメール・ソフトは表示を確認できました。(99/08 現在)

GoogleMail に接続してメールを読みたい

Version 5.5 では SSL/STARTTLS がサポートされたので Google Mail に接続することができます。POP を使う場合は GoogleMail アカウントで POP 接続を有効にする必要があります。以下は POP3 接続の C# サンプルコードです。

```
MailLib.Pop3 pop = new MailLib.Pop3();

pop.TraceOption = MailLib.TraceOptionValues.nmTraceToFile;
pop.TraceFileName = @"c:\tmp\google.log";
pop.RemoteHost = "pop.gmail.com";
pop.RemotePort = 995;
pop.UserID = "userid@gmail.com";
pop.Password = "passwd";
pop.AuthType = MailLib.PopAuthTypeValues.nmAuthPlain;
pop.SSLType = SSLTypeValues.nmTLsv1;

short rc = pop.Connect();
rc = pop.Authenticate();
DataSet ds = new DataSet("mailList");
rc = pop.ListMail(out ds);

dataGridView1.DataSource = ds.Tables["mailList"];
dataGridView1.Refresh();

pop.Quit();
```

GoogleMail に接続してメールを送りたい

上と同様に SSL/STARTTLS で GoogleMail サーバー経由でメールを送ることができます。SMTP 認証は LOGIN または PLAIN がサポートされているようです。また Quit へのレスポンスが無いので Disconnect メソッドで接続を遮断します。以下は VB.NET のサンプルコードです。

```
Dim rc As Integer
Dim mailto As New List(Of String)

With Smtpl
    .RemoteHost = "smtp.gmail.com"
    .DomainAddress = "gmail.com"
```

```

    .TraceFileName = "c:\tmp\google.log"
    .TraceOption = MailLib.TraceOptionValues.nmTraceToFile
mailto.Add("anyone@abcdefg.co.jp")
    .MailTo = mailto
    .FromName = "anyone"
    .From = "userid@gmail.com"
    .Subject = "テストメール"
    .DocInType = MailLib.DocInTypeValues.nmDocInProperty
    .DocInput = "これはテストメールです。" + vbCrLf
    .RemotePort = 465 ' SSL
    .SmtpAuthType = MailLib.SmtpAuthTypeValues.nmSmtpAuthLogin
    .UserID = "userid@gmail.com"
    .Password = "password"
    .SSLType = MailLib.SSLTypeValues.nmSSLv23
End With

rc = Sntp1.Connect
If rc <> 0 Then
    MsgBox("errr connect " + CStr(rc))
    Exit Sub
End If

rc = Sntp1.SendMessage()
If rc <> 0 Then
    MsgBox("errr send message " + CStr(rc))
End If

rc = Sntp1.Disconnect()
If rc <> 0 Then
    MsgBox("err quit " + CStr(rc))
End If

```

自己署名設定の SSL サーバーに接続できない。

自己署名設定した SSL サーバーに接続すると Connect メソッドで正常に接続できない場合があります。多くの場合 LastSSLError プロパティに"Self Signed"というような文字列が設定されます。サーバーの証明書ファイルを Windows のネットワーク接続の「証明書のインポートウィザード」でインポートすることで正常に接続が可能になります。

ASP.NET ウェブアプリケーションから使えるか

ご利用になれます。ウェブアプリケーションプロジェクトのウェブフォームに NetMan Mail Component をツールボックスからドラッグしていただくだけで、コンポーネントの初期化コードやプロパティ設定コードが自動生成されます。

VB6 で使いたい

.NET Framework のコンポーネントをラップして COM から利用する方法はマイクロソフトから提示されていますが、制約が多いので事実上は.NET framework コンポーネントを VB6 から使うのは無理と考えたほうが良いと思います。弊社では旧バージョンになりますが COM のコンポーネントも当面販売を継続する方針ですので必要であれば販売会社のシステムラボまでご連絡ください。

Appendix.B システム・エラー・コード

NetMan Components for Mail の各コンポーネントの LastOSError プロパティに格納される値で主な値を抜粋して以下に説明します。具体的には Win32 API の GetLastError からのものでありますので他の値につきまして詳細をお調べになりたい場合はマイクロソフトの技術サイト(msdn.microsoft.com)を参照してください。

2 ENOENT

ファイルまたはディレクトリがない。指定されたファイルまたはディレクトリが存在しないか見つかりません。このメッセージは、指定されたファイルが存在しないか、パス名の要素が既存のディレクトリを指定していない場合に発生します。

9 EBADF

不正なファイル番号。2 つの原因が考えられます。1) 指定されたファイル ハンドルが有効なファイル ハンドルの値ではないか、開いているファイルを参照していないとき。2) 読み出し専用で開いているファイルまたはデバイスに書き込もうとしたとき。

12 ENOMEM

メモリ不足。実行しようとした操作に十分なメモリがありません。たとえば、子プロセスを実行するためのメモリが不足している。

13 EACCES

アクセスが拒否された。ファイルのアクセス権の設定により、指定されたアクセスを行えません。このエラーは、ファイルの属性に反した方法でファイル(場合によってはディレクトリ)にアクセスしようとしたときに起こります。

17 EEXIST

ファイルが存在する。既に存在するファイルを作成しようとしています。

22 EINVAL

不正な引数。関数の引数のどれかに無効な値が指定されています。

24 EMFILE

開いているファイルが多すぎる。空いているファイル ハンドルがないので、ファイルを開けません。

28 ENOSPC

デバイスの空き領域不足。デバイスに書き込み用の領域が残っていません (たとえば、ディスクがいっぱいになった場合)。

34 ERANGE

結果が大きすぎる。数値演算関数の引数が大きすぎて、結果の有効桁数の一部または全部が失われます。このエラーは、他の関数でも、引数が想定より大きい場合に発生します。

36 EDEADLOCK

リソースのデッドロックが発生する可能性がある。

Appendix.C フラグ値

Imap4 では以下のような値を各メールにフラグとして保持します。ListMail メソッドではこれらのフラグ値を整数値の配列でアプリケーションに通知します。以下はフラグの値と通知される整数の値のリストです。

Deleted	1
Seen	2
Recent	4
Answered	8
Flagged	16
Draft	32

NetMan Components for Mail 5.6 調査依頼

Faxでのお問い合わせは03-3421-6691まで。メールの場合はSupport@techknowledge.co.jpまで、以下のフォームの内容をご記入の上メールをお願いします。

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	
使用言語とバージョン	
Windowsバージョン	
メールサーバーの種類、バージョン等	
お問い合わせ内容、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。	
SAMPLE	
添付資料	

SAMPLE

NetMan Components for Mail ver 5.60
2007年8月7日 第1版

版權・著作 株式会社テクナレッジ
Printed In Japan