

MAILOCX

WILL

株式会社ウィル

白紙ページ

白紙ページ

- Microsoft、Windows、Windows NT、Visual Basic、ActiveX、Office、Access、Excel は、米国 Microsoft Corporation の米国ならびに各国における登録商標です。
- その他本書に掲載されている会社名、製品名はそれぞれ各社の商標又は登録商標です。

目次

はじめに.....	5
商品に含まれるもの.....	7
動作環境について.....	7
インストール.....	8
ライセンスの登録.....	10
サンプルを見る.....	13
サポートについて(無償).....	14
バージョンアップについて(無償).....	15
再配布について.....	17
POP コントロール.....	19
▼ プログラミング概要.....	21
▼ 状態遷移図.....	25
▼ プロパティ.....	29
Copyright プロパティ.....	31
DecodeMimeHeadInMailBody プロパティ.....	32
Port プロパティ.....	33
APOP プロパティ.....	34
DeleteOnError プロパティ.....	35
Column***.....	36
▼ メソッド.....	37
POP メソッド.....	39
Abend メソッド.....	40
Column***.....	41
▼ イベント.....	43
Done イベント.....	45
Trace イベント.....	46
Stat イベント.....	47
UidlFirst イベント.....	48
UidlDone イベント.....	49
TopFirst イベント.....	50

TopNext イベント	51
TopDone イベント	52
RetrFirst イベント	53
RetrNext イベント	54
MailHeadFirst イベント	55
MailHeadNext イベント	56
MailHeadDone イベント	57
MultiPartOpen イベント	58
MimeHeadFirst イベント	59
MimeHeadNext イベント	60
MimeHeadDone イベント	61
MimeDataFirst イベント	62
MimeDataNext イベント	63
MimeDataDone イベント	64
MultiPartClose イベント	65
RetrDone イベント	66
Column***	67
SMTP コントロール	69
▼ プログラミング概要	71
▼ 状態遷移図	75
▼ プロパティ	79
Copyright プロパティ	81
MimeMultiPart プロパティ	82
Port プロパティ	83
Hostname プロパティ	84
AUTH プロパティ	85
USER プロパティ	86
PASS プロパティ	87
AngleBracketType プロパティ	88
KanjiIn プロパティ	89
KanjiOut プロパティ	90
Fold78 プロパティ	91
Charset プロパティ	92
Column***	93

▼ メソッド	95
Smtp メソッド	97
Column***	98
▼ イベント	99
Done イベント	101
Trace イベント	102
MoreData イベント	103
MimeHead イベント	104
MimeBody イベント	105
Column***	106
Winsock エラーコード一覧	109
サンプル	113
WillMAIL	115
SMTP_VC5	118
SimpleDirectMail	119
索引	121

白紙ページ

はじめに

はじめに

白紙ページ

商品に含まれるもの

1. CD-ROM
 - Willware.exe
 - Cryptdll.exe
(暗号 DLL 専用・実行環境用セットアップキット)
 - readme.txt
2. フロッピーディスク
 - レジストリファイル
 - readme.txt
3. 使用許諾契約書
4. マニュアル

動作環境について

■対応 OS

MAILOCX は、以下に示す OS で動作確認を行っております。

Microsoft Windows 95、Microsoft Windows 98、
Microsoft WindowsNT 4.0、Microsoft Windows 2000、
Microsoft Windows XP、Microsoft Windows 2003

■開発に必要なソフトウェア

MAILOCX をご使用いただくには、以下のいずれかのソフトウェアが必要です。

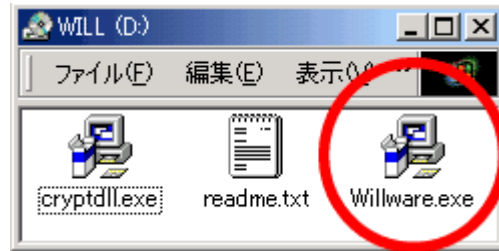
Microsoft Visual Basic Ver 5.0
Microsoft Visual Basic Ver 6.0
Microsoft Office 2000 (Access、Excel)

MAILOCX は、Microsoft Visual Basic Ver 5.0 で作成しています。サンプルは、Microsoft Visual Basic Ver 5.0 及び Ver 4.0 で作成しています。

※ 本製品は日本語環境のみの対応となります。

インストール

製品の CD-ROM に含まれているセットアップキット (Willware.exe) をダブルクリックします。

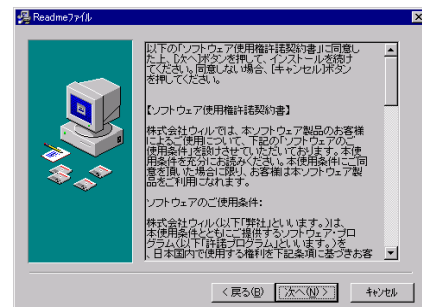


画面にしたがって、インストールを進めて下さい。

1. インストールを始めます。「次へ」をクリックして下さい。



2. 使用許諾契約書です。内容に同意される場合は「次へ」をクリックして下さい。



3. インストール先のフォルダを指定します。初期設定でよろしければ「次へ」をクリックして下さい。別のフォルダを指定したい場合は「参照」をクリックし、フォルダを指定して下さい。



4. インストール中に置換されるファイルのバックアップを作成できます。そのバックアップファイルの保存先フォルダを指定します。初期設定でよろしければ「次へ」をクリックして下さい。



5. WILLWARE Components を登録するスタートメニュー又はプログラクマネージャのグループフォルダを指定します。初期設定では、新規に「WILLWARE Components」の名前でフォルダを作成します。特に指定する必要がなければ、初期設定をお勧めします。



6. プログラムのコピーを開始します。「次へ」をクリックして下さい。



7. プログラムのコピーをしています。中断する場合は、「キャンセル」をクリックして下さい。



8. インストールが完了しました。「完了」をクリックし、インストールを終了して下さい



はじめに

ライセンスの登録

■レジストリファイルから登録する

ライセンスを登録します。製品に含まれているフロッピーディスクのレジストリファイル (EMXXXXXXXXX.reg) をダブルクリックして下さい。(「XXXXXXXXXX」は、任意の数字がファイル名として付けられています。)

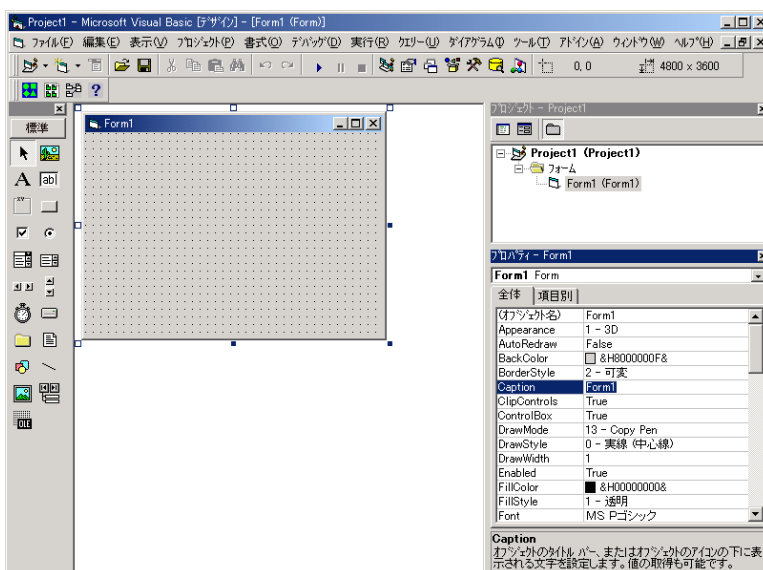


以下のメッセージボックスが表示され、ライセンスがレジストリに登録されます。



■手動で登録する

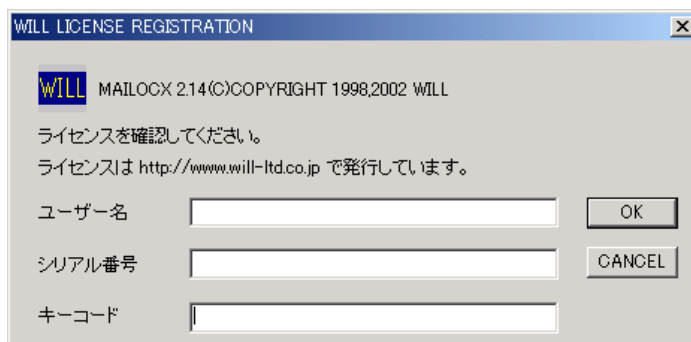
あらかじめ電子メールで通知しているライセンス情報を利用してライセンスを登録する等、レジストリファイルを利用しない場合は、VisualBasic 起動後に新規プロジェクトを選択し以下のデザイン画面を開きます。



ツールバーの「プロジェクト」から、「コンポーネント」を選択し、「コンポーネント」画面を開きます。次にコントロールタブの一覧から MAILOCX を選択して「OK」をクリックすると、MAILOCX がツールボックスに追加され、アイコンが表示されます。



ツールボックスに追加された MAILOCX を選択し、フォームにアイコンを貼り付けると、以下の「WILL LICENSE REGISTRATION」画面が表示されます。ここで、ユーザー名、シリアル番号、キーコードをそれぞれ入力してライセンス登録を行います。



はじめに

■トライアルライセンスから正規ライセンスへの移行

既にトライアルライセンスが登録されている場合には、デザイン画面にある MAILOCX のプロパティで「バージョン情報」をクリックして下さい。



「WILL LICENSE REGISTRATION」画面が表示されますので、ここで正規ライセンスを入力して下さい。



■ライセンス入力時のご注意

※ライセンスが入力できない!?

入力したライセンスにスペースが含まれていないか確認して下さい。(ライセンスに、スペースは使用していません。)

※登録したライセンスを認識しない!?

ライセンスを登録しても、オブジェクトが新規ライセンスを認識していない場合は、MAILOCX のアイコンを少し動かして下さい。この作業により、オブジェクトにライセンスが記憶されます。

※トライアルライセンスで作成したアプリケーションはどうする!?

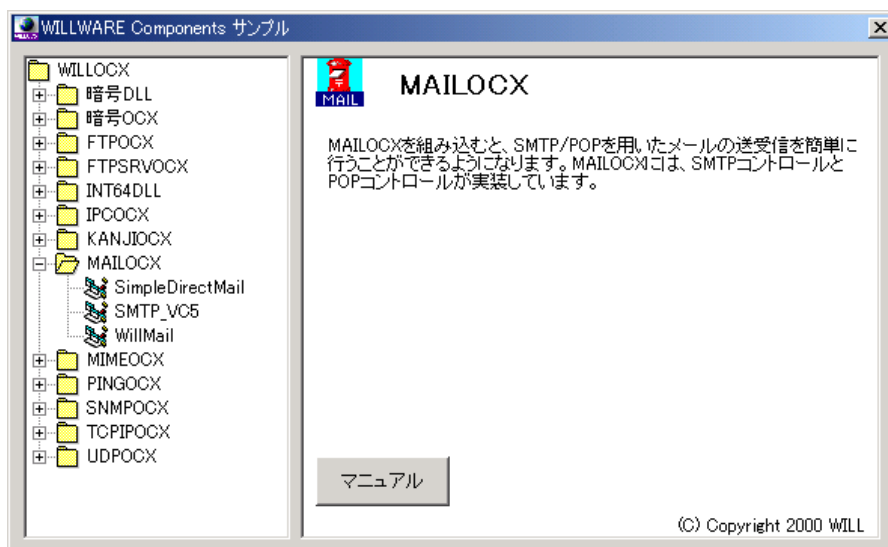
既にトライアルライセンスで作成したアプリケーションは、正規ライセンスを登録した後、再コンパイルする必要があります。

サンプルを見る

インストールが完了すると、スタートメニューに「WILLWARE Components」が追加されます。



「WILLWARE Components」の「サンプル」を起動すると「WILLWARE Components サンプル」画面が表示されます。サンプルの起動、またはそれぞれのソースを開くことができます。但し、ソースを開くにはライセンスが必要です。トライアルライセンス又は、正規ライセンスを登録してご利用下さい。(ライセンスの登録方法は前項の「ライセンスの登録」をご覧ください。)



はじめに

サポートについて(無償)

サポートは基本的に電子メールで受け付けております。サポートは無償でご利用いただけます。

■お問い合わせの前に

サポート作業を円滑に行うために、お問い合わせの際には以下の情報をご用意下さい。

1. 製品名及びバージョン
2. 開発環境(OSの種類及びバージョン、サービスパッケージの種類)
3. 開発ツール及びバージョン
4. サーバの種類
5. 問題点
 - (1) エラー内容又は、エラー状況のハードコピー
 - (2) 問題点となる部分のサンプルソースコード。

■FAQ

弊社ホームページの「サポート」のページで、キーワードを入力して FAQ を検索できます。休業日などサポートの対応が遅れる場合もありますので、まずはこちらをご確認下さい。

■お問合せ先

info@will-ltd.co.jp

バージョンアップについて(無償)

製品のバージョンアップは、すべて無償です。

■バージョンアップ情報の入手方法

バージョンアップの情報は、弊社ホームページの新着情報で通知し、各商品のページの更新履歴で更新内容を掲示致します。

■最新バージョンの入手方法

最新バージョンのプログラムは、弊社ホームページ(<http://www.will-ltd.co.jp/>)のダウンロードのページよりダウンロードすることが出来ます。ダウンロードするファイルは、以下のバージョンアップの目的により異なりますのでご注意ください。

- **WILLWARE Components(全製品用)セットアップキットを利用してバージョンアップ**
ファイル名 : 「Willware.exe」

WILLWARE Components(全製品用)セットアップキットは全ての製品をインストールするためのものです。そのため本製品以外の製品及びサンプル、マニュアルも同時にバージョンアップされます。

- **各コンポーネント毎のセットアップキットを利用してバージョンアップ**
ファイル名 : 「○○○ocx.exe」

各コンポーネントのファイル(ocx、dll)及び、依存ファイルのみバージョンアップされません。サンプル及びマニュアルはバージョンアップされませんのでご注意ください。

はじめに

■バージョンアップをする前に

各セットアップキットを利用してバージョンアップをする前に、以下のことにご注意ください。

● WILLWARE Components(全製品用)セットアップキットを利用してバージョンアップする場合は、古いバージョンをアンインストールしてから、最新バージョンをインストールすることをお勧めいたします。

※ アンインストールの方法は、スタートメニューから「設定」→「コントロールパネル」→「アプリケーションの追加と削除」の画面で、「WILLWARE Components」を選択し、画面の指示に従って行って下さい。

● 各コンポーネント毎のセットアップキットを利用してバージョンアップする場合は、最新バージョンをそのままインストールして下さい。古いファイルは上書きされます。

※ 弊社製品を複数ご利用いただいている場合、いずれか1つをバージョンアップしても他の製品に影響はありません。

■バージョンアップの方法

セットアップキットをダブルクリックし、画面の指示に従ってインストールを進めて下さい。

再配布について

■作成したアプリケーションの配布時

MAILOCX を利用して作成したアプリケーションの配布時のランタイムライセンスはフリーです。但し、開発ライセンスの配布はできません。

■再配布時に必要な配布可能ファイル

MAILOCX を利用して作成したアプリケーションを配布する場合には、以下のファイルを添付する必要があります。()内は推奨バージョンです。

- ・ MAIL.OCX
- ・ TCPIP.OCX
- ・ MIMEO.OCX
- ・ KANJI.OCX
- ・ MD5.OCX
- ・ VB5JP.DLL (Ver 5.0.4319)
- ・ MSVBVM50.DLL (Ver 5.2.8244)
- ・ MFC42.DLL (Ver 4.21.7022)
- ・ MFC42LOC.DLL (Ver 4.21.7022)
- ・ MSVCRT.DLL (Ver 5.00.7022)
- ・ OLEPRO32.DLL (Ver 5.0.4118)
- ・ OLEAUT32.DLL (Ver 2.20.4118)

※ セットアップウィザードを使用する場合

MAILOCX をインストールすると、自動的に OCX の依存ファイルが以下のディレクトリにインストールされます。

C:\Windows\system (Windows95, Windows98 の場合)

C:\WINNT\system32 (WindowsNT4.0, Windows2000, Windows2003 の場合)

C:\Windows\system32 (WindowsXP の場合)

セットアップウィザードを実行すると自動的にアプリケーション配布時に必要な OCX (内部で利用している OCX)と、DLL ファイルが Setup.lst ファイルに追加されます。

■著作権

- ・ MAILOCX およびこれに付随するマニュアルの著作権は株式会社ウイル(横浜市保土ヶ谷区)にあります。
- ・ 本ソフトウェアおよびマニュアルを運用した結果については、当社は一切責任を負いません。
- ・ 本ソフトウェアの仕様またはマニュアルに記載されている事項は予告無く変更することがあります。
- ・ マニュアルなどに記載されている会社名、製品名は、各社の商標および登録商標

はじめに

です。

- ・ MAILOCX を利用するアプリケーションは MAILOCX の著作権表示を行わなければなりません。Copyright プロパティに MAILOCX の著作権を示す文字列があります。アプリケーションまたはドキュメントのいずれかにこの文字列を表示して、MAILOCX を使用していることを示してください。

POP コントロール

▼ プログラミング概要

▼ 状態遷移図

▼ プロパティ

▼ メソッド

▼ イベント

白紙ページ

▼ プログラミング概要

白紙ページ

pop メソッドを使って POP サーバに接続し、その後発生するイベントに従って読み出したメールの処理を行ってください。

pop メソッドを呼び出すと、最初に、

```
Stat(Many%, MuchAll&)
```

が発生します。

Many に受信可能なメール総数が入っています。各メール毎に以下のイベントが順に発生します。

```
UidlFirst(id%, CancelUidl%)
UidlDone(id%, Uidl$, Delete%)
TopFirst(id%, CancelTop%)
TopNext(id%, Data$)
TopDone(id%, Delete%)
RetrFirst(id%, Much&, CancelRetr%)
RetrNext(id%, Data$)
MailHeadFirst(id%)
MailHeadNext(id%, FieldName$, FieldValue$)
MailHeadDone(id%)
MultiPartOpen(id%, ParentPart&, ContentType$, _
    ContentName$, ContentCharset$, TransferEncoding$, Tag$)
MimeHeadFirst( id%, Part&)
MimeHeadNext( id%, Part&, FieldName$, FieldValue$)
MimeHeadDone( id%, Part&)
MimeDataFirst( id%, NewPart&, ContentType$,ContentName$,
    ContentCharset$, TransferEncoding$, IsText As Boolean)
MimeDataNext( id%, Part&, Data$)
MimeDataDone( id%, Part&)
MultiPartClose( id%, Tag$)
RetrDone(id%, CancelDelete%)
```

なお、UildDone,TopNext,TopDone イベントは、POP サーバが UIDL コマンドおよび TOP コマンドをサポートしている場合だけ発生します。

すべてのメールを処理したら、

```
Done(Code&, Msg$)
```

が発生します。これはエラーの場合も発生します。Code が 0 であれば正常に処理が終わっています。

POP コントロール

その他のイベントとして、

Trace(Msg\$)

があります。これは、POP コントロールと POP サーバ間のやり取りをアプリケーションに知らせるものです。デバッグに利用します。

《注意》

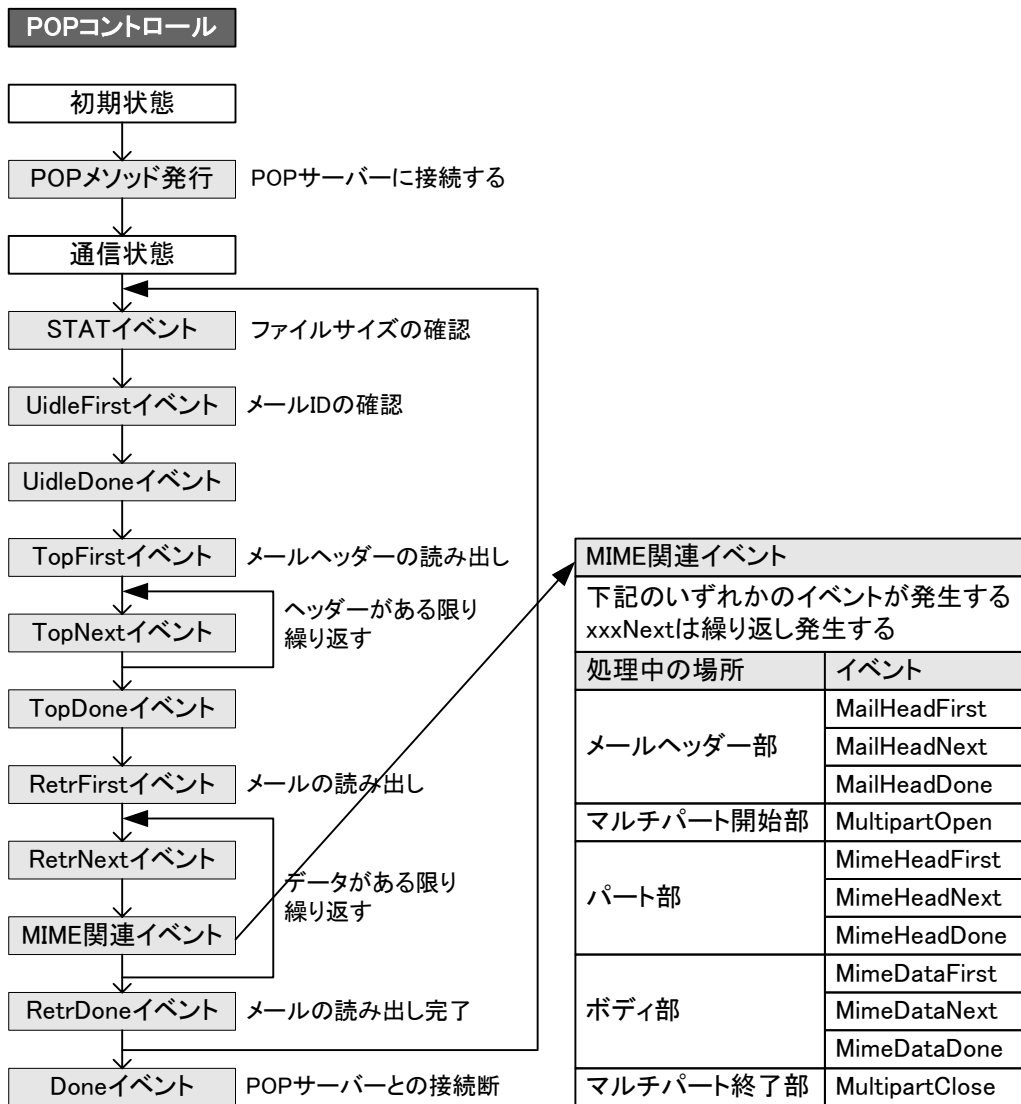
UIDL コマンドおよび TOP コマンドを発行した際にエラーになっても処理は中断しません。UIDL および TOP コマンド以外のコマンドがエラーを返した場合、RSET コマンドを発行して QUIT します。

RSET コマンドを発行すると、削除の指示は無効となります。

▼ 状態遷移図

白紙ページ

状態遷移図



白紙ページ

▼ プロパティ

白紙ページ

Copyright プロパティ

■機 能

MAILOCX のバージョンと著作権情報です。参照のみ可能です。

■構 文

Object.Copyright

Copyright プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。

■データ型

文字列(String)

DecodeMimeHeadInMailBody プロパティ

■機 能

これは、以前のバージョンと互換性を持たせるためのものです。
MAILOCX2.01 で追加された機能を使う場合は、これを必ず False にしてください。

■構 文

Object.DecodeMimeHeadInMailBody[=Value]

DecodeMimeHeadInMailBody プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Value	バージョンの互換性を指定します。次の設定値を参照してください。

■設定値

Value の設定値は次のとおりです。

(値)	(説 明)
True	旧バージョンとの互換性を持たせます。(デフォルト)
False	新機能を使います。

■データ型

ブール型(Boolean)

Port プロパティ

■機能

接続する POP サーバのポート番号です。

■構文

Object.Port[=Value]

Port プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Value	接続する POP サーバのポート番号を指定する整数式または文字列式(サービス名称)です。デフォルトは 110 です。

■データ型

長整数(Long)、文字列(String)

APOP プロパティ

■機能

APOP 認証を行うかどうか指定します。

■構文

Object.APOP [=Value]

APOP プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Value	APOP を使用するかどうかを指定します。次の設定値を参照してください。

■設定値

Value の設定値は次のとおりです。

(値)	(説明)
True	APOP 認証を使用します。
False	APOP 認証を使用しません。(デフォルト)

■データ型

ブール型(Boolean)

DeleteOnError プロパティ

■機能

POP コントロールはエラーが発生するとそれまでに DELETE するように指示したことをキャンセルして処理を終えますが、DeleteOnError プロパティを True にすることで、エラー発生時でも、それまでに指示した DELETE 処理を継続することが可能になります。

■構文

Object.DeleteOnError[=Value]

DeleteOnError プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Value	エラー発生時に DELETE マークのついたメールを削除するかどうかを指定します。次の設定値を参照してください。

■設定値

Value の設定値は次のとおりです。

(値)	(説明)
True	エラー発生時に DELETE マークのついたメールを削除します。
False	削除しません。(デフォルト)

■データ型

ブール型(Boolean)

Column...

SMTP 認証

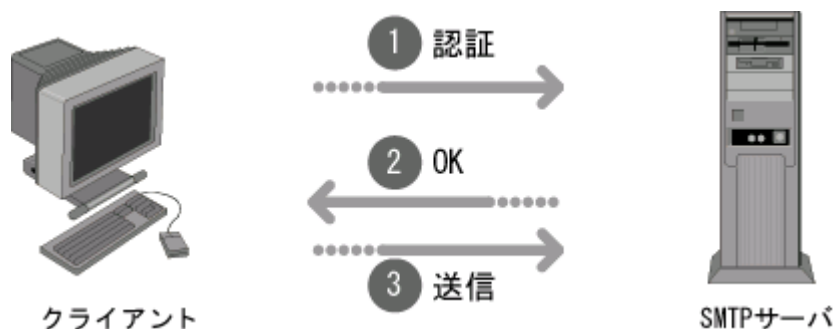
当初 SMTP にはクライアントを認証する手段が用意されていませんでした。そのため、第三者によるスパムメール(迷惑メール、不正中継)という問題の原因となっていました。

その後、POP3 の認証を利用した認証方法「POP before SMTP」が提供されるようになりました。さらに RFC2554 に定義された「AUTH」コマンドの拡張により、クライアントを認証する手段が提供されました。これが「SMTP Authentication」と呼ばれるメール送信時認証です。現在ではまだ「POP before SMTP」が主流と思われませんが、今後はメールクライアントおよびメールサーバの対応にともない、「SMTP Authentication」が広まるものと思われます。

「SMTP Authentication」

SMTP Authentication とはメールを送信する際に、SMTP サーバ(メール送信用サーバ)でユーザアカウントとパスワードの認証を行います。認証がされた場合のみメールの送信を許可します。この認証ができない場合はメールを送信することができません。

SMTP Authentication は 「AUTH-LOGIN」、「AUTH PLAIN」、「AUTH CRAM-MD5」と呼ばれる認証方式があります。平文でない安全な認証プロトコルとしては「CRAM-MD5」だけがほぼ共通で使えるようです。さらに詳細をお調べになりたい方は RFC2554 をご覧下さい。



▼ メソッド

白紙ページ

POP メソッド

■機 能

POP サーバに接続してメールの受信を開始します。

■構 文

Object. POP(xserv As String, xuser As String, xpass As String, xlines As Integer)
POP メソッドの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
xserv	POP サーバアドレスです。
xuser	POP3 の USER コマンドを発行する際に指定する引数です。 (ユーザーアカウント)
xpass	POP3 の PASS コマンドを発行する際に指定する引数です。 (ユーザーパスワード)
xlines	POP3 の TOP コマンドを発行する際に指定する引数です。 (取り出すメール本文の行数) 必ず 0 以上を指定してください。

■戻り値

なし

Abend メソッド

■機 能

メールの受信をキャンセルします。また、キャンセルする際、削除マークのついたメールの削除をするかどうかを指定します。

■構 文

Object.Abend([CancelDelete As Boolean = True])

Abend メソッドの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
CancelDelete	True にすると、削除マークのついたメールを削除しません。(デフォルト) False にすると削除マークのついたメールを削除します。

■戻り値

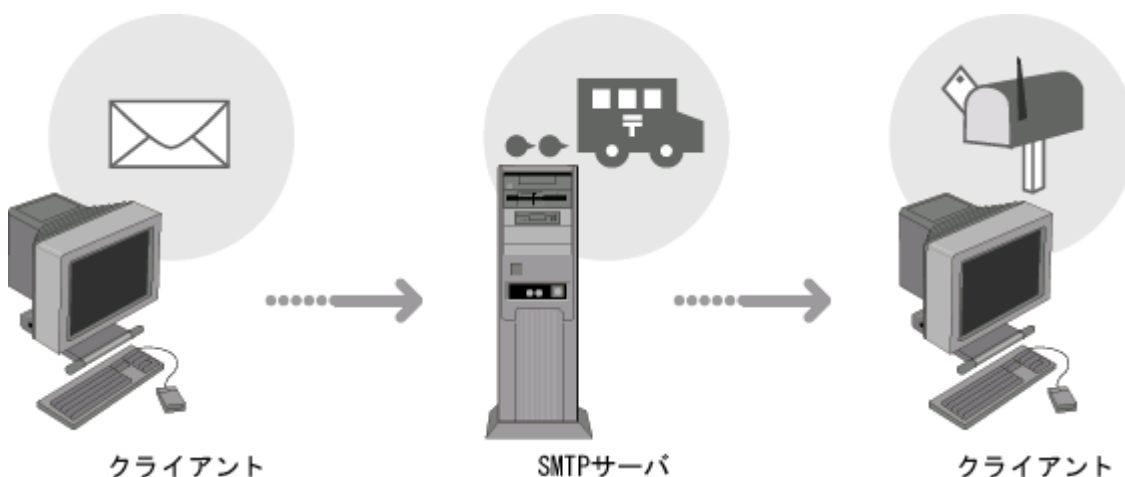
なし

Column...

電子メールの構成

メール送受信のプログラミングを行うにあたって、電子メールの構成要素をしっかりと把握することは大切なことです。

メールの仕組みを理解するには実際の手紙のやり取りと置き換えてみるとわかりやすいと思います。メールは送信者(差出人)→SMTP サーバ(郵便局)→受信者(受取人)という順に伝わって行きます。



このときそれぞれが利用する(できる)情報は限られています。これらの情報は大きく3つの部分で構成されています。

電子メール	実際の手紙
エンベローブ	封筒の外側に示されている宛先、差出人の情報
メールヘッダ	手紙の本文に記述されている宛名など
本文	手紙の本文

実際の手紙で、郵便局の人はメールヘッダや本文を覗き込んだり書き換えたりすることはできません。ただエンベローブにかかれています宛先に手紙を届けるのみです。

同様に電子メールの世界でも SMTP サーバはメールヘッダをまったく利用しません。エンベローブにかかれています情報のみを参照してメールを届けます。そのため、受信者のメールソフトで表示されている送信者と実際の送信者とが異なることがあります。エンベローブの情報は SMTP コマンドである MAIL と RCPT を発行するときに指定するものです。また、メールヘッダの情報は From: や TO: など送信者が任意に指定できるものです。

白紙ページ

▼ イベント

白紙ページ

Done イベント

■機能

POP の処理がすべて終了したときに発生します。

■構文

Private Sub Object_Done(Code As Long, Msg As String)

Done イベントの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Code	0 → 正常終了したことを示します。 0 以外 → 異常終了したことを示します。
Msg	異常終了した場合の理由を示す文字列です。

Trace イベント

■機 能

POP コントロールと POP サーバ間のやり取りをアプリケーションに知らせます。
">>"という文字列で始まるものは送信したコマンドを示します。
"+"で始まるものはコマンドに対するサーバの応答です。

■構 文

Private Sub Object_Trace(Msg As String)
Trace イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Msg	発行したコマンドと受信したステータスです。

Stat イベント

■機 能

POP3 の STAT コマンドの結果を知らせます。

■構 文

Private Sub Object _Stat(Many As Integer, MuchAll As Long)

Stat イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Many	サーバにあるメール総数です。32000 通以上ある場合は、32000 となります。
MuchAll	メールの総バイト数です。

UidlFirst イベント

■機 能

POP3 の UIDL コマンドを発行する直前に発生します。UIDL コマンドが成功した場合の次に続くイベントは IdleDone イベントです。CancelUidl を True にすると UIDL コマンドを発行しないため処理が少しだけ速くなります。POP サーバが UIDL コマンドをサポートしていない場合、または CancelUidl を True にした場合の次に続くイベントは TopFirst イベントです。

■構 文

Private Sub Object_UidlFirst(id As Integer, CancelUidl As Integer)

UidlFirst イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
CancelUidl	True にすると UIDL コマンドを発行せず、次に TopFirst イベントが発生します。 False にすると UIDL コマンドを発行します。(デフォルト)

UidlDone イベント

■機 能

POP3 の UIDL コマンドが成功した場合に発生します。ここでの主な処理はメッセージを受け取る前に、そのメールを削除するかどうかを判断し、その指示を行うことです。Delete に False を指定した場合の次に続くイベントは TopFirst イベントです。Delete に True を指定した場合、POP3 の TOP コマンドおよび RETR コマンドを発行しないため Top 関連のイベントおよび Retr 関連のイベントは発生しません。

■構 文

Private Sub Object_UidlDone(id As Integer, Uidl As String, Delete As Integer)

UidlDone イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Uidl	メールを一意に識別する文字列です。
Delete	True にするとメールを削除します。 False にするとメールを削除しません。(デフォルト)

TopFirst イベント

■機 能

POP3 の TOP コマンドを発行する直前に発生します。TOP コマンドが成功した場合の次に続くイベントは TopNext イベントです。CancelTop を True にすると TOP コマンドを発行しないため処理が少しだけ速くなります。POP サーバが TOP コマンドをサポートしていない場合、または CancelTop を True にした場合の次に続くイベントは RetrFirst イベントです。

■構 文

Private Sub Object_TopFirst(id As Integer, CancelTop As Integer)

TopFirst イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
CancelTop	True にすると Top コマンドを発行しません。 False にすると Top コマンドを発行します。(デフォルト)

TopNext イベント

■機 能

POP3 の TOP コマンドにより読み出されたメールヘッダーおよびメール本文の一部 (または全部) について 1 行読み込む度に発生します。次に続くイベントは TopDone イベントです。

■構 文

Private Sub Object_TopNext(id As Integer, Data As String)

TopNext イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Data	メールヘッダ 1 行分のデータです。改行コードは含まれません。

TopDone イベント

■機 能

POP3 の TOP コマンドによるデータの読み出しがすべて終了したときに発生します。ここでの主な処理はメッセージの本文を受け取る前に、そのメールを削除するかどうかを判断し、その指示を行うことです。Delete に False を指定した場合の次に続くイベントは、RetrFirst イベントです。Delete を True にした場合、POP3 の RETR コマンドを発行しないため Retr 関連のイベントは発生しません。

■構 文

Private Sub Object_TopDone(id As Integer, Delete As Integer)

TopDone イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Delete	True にするとメールを削除します。 False にするとメールを削除しません。(デフォルト)

RetrFirst イベント

■機 能

POP3 の RETR コマンドを発行する直前に発生します。CancelRetr を True にすると RETR コマンドを発行しないため処理が少しだけ速くなります。RETR コマンドが成功した場合の次に続くイベントは RetrNext イベントです。CancelRetr を True にした場合の次に続くイベントは、まだ受信できるメールがある場合は STAT イベント、すべてのメールの受信処理が終了した場合は Done イベントです。

■構 文

Private Sub Object_RetrFirst(id As Integer, Much As Long, CancelRetr As Integer)

RetrFirst イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Much	このメールのバイトサイズです。
CancelRetr	True にすると RETR コマンドを発行しません。 False にすると RETR コマンドを発行します。(デフォルト)

RetrNext イベント

■機 能

RETR コマンドにより読み出されたメールヘッダおよびメール本文について 1 行読み込む度に発生します。ここでの主な処理は、POP サーバから受け取ったデータをそのまま記録することです。メールヘッダ、パートヘッダ、パートデータを別々に取り出す場合はそれぞれ MailHeadXXXX イベント、MimeHeadXXXX イベント、MimeDataXXXX イベントで処理を行います。

■構 文

Private Sub Object_RetrNext(id As Integer, Data As String)

RetrNext イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Data	ヘッダ又は本文 1 行分の文字列式です。

MailHeadFirst イベント

■機 能

メールのヘッダを読み込む直前に発生します。

■構 文

Private Sub Object_MailHeadFirst(ByVal id As Integer)

MailHeadFirst イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。

MailHeadNext イベント

■機 能

メールのヘッダを1行読み込む度に発生します。それぞれのヘッダは項目名とその値に分けて渡されます。

■構 文

```
Private Sub Object_MailHeadNext(ByVal id As Integer, ByVal FieldName As String,  
    ByVal FieldValue As String)
```

MailHeadNext イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
FieldName	メールヘッダの項目名です。コロンは含みません。
FieldValue	フィールドの値です。改行コードは含まれず、複数行に渡るものは1行にまとめられます。

MailHeadDone イベント

■機 能

メールヘッダの読み込みが終了したときに発生します。

■構 文

Private Sub Object_MailHeadDone(ByVal id As Integer)

MailHeadDone イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。

MultiPartOpen イベント

■機 能

マルチパートの読み込みを開始する直前に発生します。

ParentPart は、マルチパートの一部がさらにマルチパートで構成されるような MIME マルチパートの場合、開始されたマルチパートの ID を含む値となります。

Tag は、記録用の汎用変数として利用できます。Tag に値を設定すると MultiPartClose でその値を得ることができます。

MIME 形式でないメールの場合、MIME マルチパートに含まれる 1 つのパートであるかのようにイベントが発生します。このことにより、プログラム利用者は MIME 形式のメールかどうかを気にすることなく、処理を行うことができます。

■構 文

```
Private Sub Object_MultiPartOpen(ByVal id As Integer, ByVal ParentPart As Long,
    ByVal ContentType As String, ByVal ContentName As String,
    ByVal ContentCharset As String, ByVal TransferEncoding As String,
    Tag As String)
```

MultiPartOpen イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
ParentPart	開始されたマルチパートを含むマルチパートの ID です。 初期値は 1 です。
ContentType	multipart/mixed, multipart/alternative, multipart/parallel のいずれかが入ります。
ContentName	パートの名前です。
ContentCharest	使用している文字コードです。
TransferEncoding	そのパートで使われるエンコード方式です。 BASE64、QuotedPrintable、7-bit、8-bit の場合にはエンコードや文字コード変換を行い、その他は何もしない。
Tag	記録用の汎用変数です。

MimeHeadFirst イベント

■機 能

MIME ヘッダを読み込む直前に発生します。

■構 文

Private Sub Object_MimeHeadFirst(ByVal id As Integer, ByVal Part As Long)

MimeHeadFirst イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Part	パートの番号です。

MimeHeadNext イベント

■機 能

マルチパートに含まれる1つのパートが始まると、MimeHeadFirst()が通知されます。パートのデータタイプなどは、MimeHeadNext()で通知されます。パートのヘッダ部が終わると、MimeHeadDone()が通知されます。

■構 文

```
Private Sub Object_MimeHeadNext(ByVal id As Integer, ByVal Part As Long, _  
    ByVal FieldName As String, ByVal FieldValue As String)
```

MimeHeadNext イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Part	パートの番号です。
FieldName	MIME ヘッダの項目名です。コロンは含みません。
FieldValue	フィールドの値です。改行コードは含まれず、複数行にわたるものは1行にまとめられます。

MimeHeadDone イベント

■機 能

MIME ヘッダの読み込みが終了したときに発生します。

■構 文

Private Sub Object_MimeHeadDone(ByVal id As Integer, ByVal Part As Long)

MimeHeadDone イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Part	パートの番号です。

MimeDataFirst イベント

■機 能

MIME ボディを読み込む直前に発生します。

■構 文

```
Private Sub Object_MimeDataFirst(ByVal id As Integer, ByVal NewPart As Long,
    ByVal ContentType As String, ByVal ContentName As String,
    ByVal ContentCharset As String, ByVal TransferEncoding As String,
    IsText As Boolean)
```

MimeDataFirst イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
NewPart	パートの番号です。
ContentType	パートの MIME タイプです。
ContentName	パートの名前です。
ContentCharset	使用している文字コードです。
TransferEncoding	未対応の MIME 方式の場合はそのエンコード方式名です。 (x-uuencode、binhex など) デコードできた場合は""を返します。
IsText	True にすると、デコードしたデータの漢字変換処理を行います。 (デフォルト) False にすると、デコードしたまま漢字変換処理を行いません。

MimeDataNext イベント

■機 能

MIME ボディを読み込んだ時に発生します。Data は MimeDataFirst イベントの IsText で True が設定されていると Unicode 文字列に変換されます。

■構 文

Private Sub Object_MimeDataNext(id As Integer, Part As Long, Data As String)
MimeDataNext イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Part	パートの番号です。
Data	デコードされたデータです。バイナリの可能性もあります。

MimeDataDone イベント

■機 能

MIME ボディの読み込みが終了したときに発生します。

■構 文

Private Sub Object_MimeDataDone(ByVal id As Integer, ByVal Part As Long)

MimeDataDone イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Part	パートの番号です。

MultiPartClose イベント

■機 能

マルチパートの読み込みが終了したときに発生します。

■構 文

Private Sub Object_MultiPartClose(ByVal id As Integer, ByVal Tag As String)

MultiPartClose イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Tag	MultiPartOpen で渡した Tag の値です。

RetrDone イベント

■機 能

POP3 の RETR コマンドによるデータの読み出しが終了したときに発生します。

■構 文

Private Sub Object_RetrDone(id As Integer, CancelDelete As Integer)

RetrDone イベントの構文の指定項目は次のとおりです。

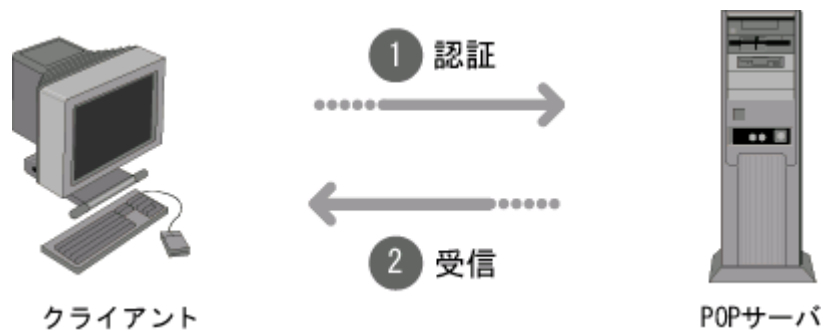
(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
CancelDelete	True にするとメールを削除しません。 False にするとメールを削除します。

Column...

POP3(Post Office Protocol Version 3)

POP はメールサーバ(POP サーバ)に到着し保存されているメールをサーバから取得するためのプロトコルです。パソコンでのメールソフトのほとんどは、このプロトコルを使用してメールを取得しています。

POP3 の大きな特徴としてユーザ認証機能が挙げられます。POP3 ではあらかじめ POP サーバに登録しておいたユーザ ID とパスワードでの認証を行った後に受信のための処理が行われます。



実際の POP3 サービスはサーバがポート番号 110 を常に listen することによって行われています。クライアントはポート番号 110 に接続しに行くことで POP3 サービスを利用することができるようになります。

白紙ページ

SMTP コントロール

▼ プログラミング概要

▼ 状態遷移図

▼ プロパティ

▼ メソッド

▼ イベント

白紙ページ

▼ プログラミング概要

白紙ページ

Smtplib を使って SMTP サーバに接続し、メールを送信します。送信すべきデータが多い場合は、その後に発生する MoreData イベントの Data に送信すべき追加のデータを書き込んでください。MoreData イベントは、Data に "" を格納しないかぎり何度も発生するので注意してください。

単純な送信の場合は、XData 変数にメールヘッダおよびメール本文を RFC821 に従って格納して、mail メソッドを発行してください。

実際の送信者とメールヘッダに記述されている From: のアドレスが異なる場合(転送の場合など)は、xFromAddr に実際の送信者(SMTP サーバ上でのユーザ ID)を指示してください。

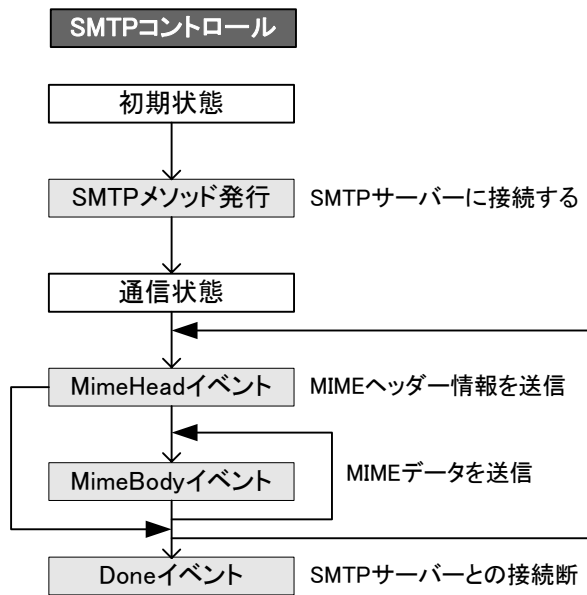
実際の受取人とメールヘッダに記述されている To,cc,Bcc 行のアドレスが異なる場合(転送の場合など)は、xToAddr に実際の受取人を指示してください。xToAddr を指定した場合、メールヘッダの To,cc,Bcc 行を参照しません。

白紙ページ

▼ 状態遷移図

白紙ページ

状態遷移図



白紙ページ

▼ プロパティ

白紙ページ

Copyright プロパティ

■機 能

MAILOCX のバージョンと著作権情報です。参照のみ可能です。

■構 文

Object.Copyright

Copyright プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。

■データ型

文字列(String)

MimeMultiPart プロパティ

■機 能

MIME 形式でメールを送信する場合のサブタイプを指定してます。ここで指定するサブタイプにより、受け手のメールソフトの処理が変わります。通常の添付ファイルの場合は“mixed”、2 つ以上の方法で同じデータを送信する場合は“alternative”、音声と映像を送る場合は“parallel”を使用します。

■構 文

Object.MimeMultipart[=Value]

MimeMultipart プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Value	MIME 形式でメールを送信する場合に指定するサブタイプです。次の設定値を参照してください。

■設定値

Value の設定値は次のとおりです。

(値)	(説 明)
mixed	通常の添付ファイルを送信する場合に使用します。
alternative	2 つ以上の方法で同じデータを送信する場合に使用します。
parallel	音声と映像を送る場合に使用します。

■データ型

文字列(String)

Port プロパティ

■機能

接続する SMTP サーバのポート番号を指定します。

■構文

Object.Port[=Value]

Port プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Value	接続する SMTP サーバのポート番号または、サービス名を指定します。デフォルト値は 25 です。

■データ型

長整数(Long)、文字列(String)

Hostname プロパティ

■機 能

送信元 PC のホスト名を指定します。デフォルトでは、PC のホスト名が格納されています。PC のホスト名に漢字を使っているとメールサーバによってはエラーになるケースがあります。このプロパティにより任意のホスト名を名乗ることができます。

■構 文

Object.Hostname[=Value]

Hostname プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Value	送信元 PC のホスト名です。デフォルト値は PC のホスト名です。

■データ型

文字列(String)

AUTH プロパティ

■機能

認証方法を指定します。

■構文

Object.AUTH[=Value]

AUTH プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Value	認証方法を指定します。次の設定値を参照してください。

■設定値

Value の設定値は次のとおりです。

(値)	(説明)
0	認証なし。(デフォルト)
1	AUTH CRAM-MD5 認証を使用します。
2	AUTH LOGIN 認証を使用します。
3	AUTH PLAIN 認証を使用します。

■データ型

長整数(Long)

USER プロパティ

■機能

ユーザ ID を指定します。

■構文

Object.USER[=Value]

USER プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Value	ユーザ ID(名)を指定します。 AUTH プロパティが 0 以外の場合に使用されます。

■データ型

文字列(String)

PASS プロパティ

■機 能

パスワードを指定します。

■構 文

Object.PASS[=Value]

PASS プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Value	パスワードを指定します。 AUTH プロパティが 0 以外の場合に使用されます。

■データ型

文字列(String)

AngleBracketType プロパティ

■機 能

SMTP プロトコルにおいて、受取人及び差出人のメールアドレスを指定するときに、ブラケット(⟨⟩)を付けるか付けないかを指定します。

■構 文

Object.AngleBracketType[=Value]

AngleBracketType プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Value	エンベロープにおけるメールアドレスを⟨⟩でくるかどうかを指定します。次の設定値を参照してください。

■設定値

Value の設定値は次のとおりです。

(値)	(説 明)
0	エンベロープにおけるメールアドレスを⟨⟩でくります。(デフォルト)
1	エンベロープにおけるメールアドレスを⟨⟩でくりません。

■データ型

整数(Integer)

KanjiIn プロパティ

■機能

1文字あたり2バイトのエスケープシーケンスを指定します。参照も可能です。エスケープ文字を除いたものを指定します。

■構文

Object.KanjiIn[=Value]

KanjiIn プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Value	エスケープシーケンスのモードを指定します。次の設定値を参照してください。

■設定値

Value の設定値は次のとおりです。

(値)	(説明)
\$@	<ESC> @\$@
\$B	<ESC> \$B RFC 1468 で定義されている JIS X 0208-1983 文字セットです。 (デフォルト)
\$(D	<ESC> \$(D

■データ型

文字列(String)

KanjiOut プロパティ

■機 能

1文字あたり1バイトのエスケープシーケンスを指定します。参照も可能です。エスケープ文字を除いたものを指定します。

■構 文

Object.KanjiOut[=Value]

KanjiOut プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Value	エスケープシーケンスのモードを指定します。次の設定値を参照してください。

■設定値

Value の設定値は次のとおりです。

(値)	(説 明)
(J)	<ESC> (J RFC 1468 で定義されている JIS X 0201-1976 (“Roman” set)文字セットです。(デフォルト)
(B)	<ESC> (B
(I)	<ESC> (I

■データ型

文字列(String)

Fold78 プロパティ

■機能

ヘッダ行の長さを 78 バイト(改行コードを含まず)に納める処理を行うかどうかを示します。

■構文

Object.Fold78[=Vaue]

Fold78 プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Value	改行を指示する長整数です。デフォルト値は 1 です。次の設定値を参照してください。

■設定値

Value の設定値は次のとおりです。

(値)	(説明)
1	行が 78 バイトを超えると直前のカンマの位置で改行します。
2	行が 78 バイトを超えると直前の空白の位置で改行します。
4	行が 78 バイトを超えるとその位置で改行します。

■データ型

長整数(Long)

■解説

上記の設定値は組み合わせることができます。組み合わせの場合は数字の小さいものが優先されます。

Charset プロパティ

■機能

マルチパートで送信する場合のメールヘッダー部の文字コードを指定します。

■構文

Object.Charset[=Value]

Charset プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Value	メールヘッダー部の文字コードです。デフォルト値は“iso-2022-jp”です。

■データ型

文字列(String)

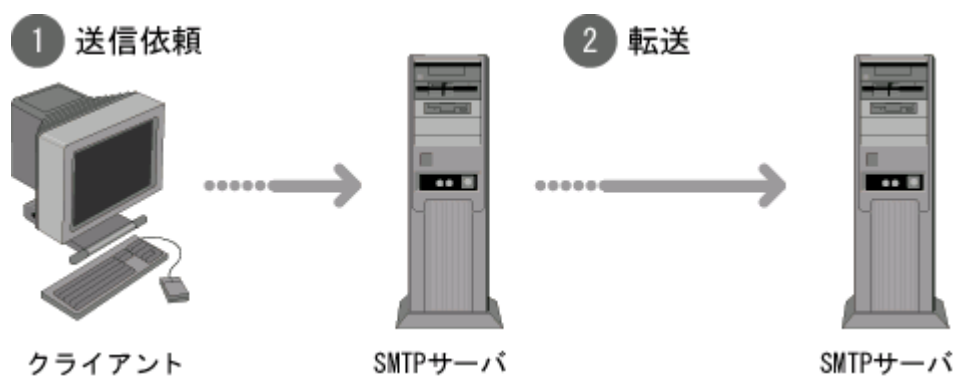
■解説

Value に "" を設定すると charset オプションを設定しません。この場合、受信したメールは "us-ascii" が設定されたものと取り扱う約束になっています。

Column...

SMTP(Simple Mail Transfer Protocol)

SMTP は電子メールを送信するためのプロトコルです。クライアントからメールを受け取った SMTP サーバは、メールアドレスを確認して適切な SMTP サーバへメールを転送します。



実際は、まずクライアントと SMTP サーバとの間で TCP のコネクションが張られ、次に FTP と同じ要領でデータを転送 します。SMTP サーバ上からメールを仲介して他の SMTP サーバに転送する役割をするソフトを MTA(Message Transfer Agent)と呼びます。代表的なものとして sendmail があげられます。以下の図は SMTP によるメール転送の流れを表したものです。

実際の SMTP サービスはサーバがポート番号 25 を常に listen することによって行われています。クライアントはポート番号 25 に接続しに行くことで SMTP サービスを利用することができるようになります。

白紙ページ

▼ メソッド

白紙ページ

Smtplib メソッド

■機能

SMTP サーバに接続してメールの送信を開始します。

■構文

Object.Smtplib(xserv As String, xData As String, [xFromAddr As String], [xToAddr])
Smtplib メソッドの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
xserv	SMTP サーバアドレスです。
xData	送信する内容です。MIME 形式のメールではメールヘッダのみです。
xFromAddr	SMTP の MAIL コマンドを発行する際に指定する引数(送信者のアドレス)です。省略するとメールヘッダに記述した From:を参照します。 ※通常はメールヘッダに送信者のアドレスを指定し、この引数は省略します。
xToAddr	SMTP の RCPT コマンドを発行する際に指定する引数(受信者のアドレス)です。省略するとメールヘッダに記述した To,cc,Bcc 行を利用します。複数の宛先に送る場合は文字配列を渡してください。 (配列番号 1 から使用します。) ※通常はメールヘッダに受信者のアドレスを指定し、この引数は省略します。

■戻り値

なし

Column...

送信時に指定する 代表的なメールヘッダについて(その1)

送信時に指定できるメールヘッダには以下のようなものがあります。

To:

送信先のメールアドレス。一度に複数指定することが可能です。複数指定する場合には“,”カンマで区切ります。

From:

発信者のメールアドレス。ただし、必ず正しい情報が入っているとは限りません。書式は決まっておらず以下の4種類のパターンがあります。

(ア) From:hoge@will-ltd.co.jp

(イ) From:<hoge@will-ltd.co.jp>

(ウ) From:nickname<hoge@will-ltd.co.jp>

(エ) From:hoge@will-ltd.co.jp (nickname)

(ウ)や(エ)の方法で指定することで、受信時にメールソフトによっては、送信者の部分に hoge と表示することができます。

CC: (carbon copy)

To:に複数のアドレスを指定する場合と同じように、複数のあて先に同じメールを送ることができます。これは本来メールの内容を通知したい相手とは別に、そのメールを読んでおいてもらいたい受信者を指定します。

BCC: (blind carbon copy)

CC:と同じ目的で使用します。CC:とは異なり To:や CC:で指定した受信者には、誰あてに写し(コピー)を送ったかをわからなくすることができます。

Subject:

メールの表題。日本語には対応しておらず、日本語対応のメールソフトでない場合文字化けが発生してしまいます。日本語対応のメールソフト間でのやり取りでは、送信側でいったんMIMEコードに変換し、受信側で再び元の文字コードに戻すといった処理を行って日本語を表示させています。

※ MAILOCX では送信する際に上記の MIME コード変換を行っております。

Reply-To:

From:で指定したアドレス以外に返信して欲しい場合、この項目に返信用アドレスを指定します。メーリングリストなどで、From:に実際の送信者のアドレス、Reply-To:にメーリングリストのアドレスを入れたりして利用されています。

▼ イベント

白紙ページ

Done イベント

■機 能

SMTP の処理がすべて終了したときに発生します。

■構 文

Private Sub Object_Done(Code As Long, Msg As String)

Done イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Code	0 → 正常終了したことを示します。 0 以外 → 異常終了したことを示します。
Msg	異常終了した場合の理由を示す文字列です。

Trace イベント

■機 能

SMTP コントロールと SMTP サーバ間のやり取りをアプリケーションに知らせます。
">>"という文字列で始まるものは送信したコマンドを示します。
"+"で始まるものはコマンドに対するサーバの応答です。

■構 文

Private Sub Object_Trace(Msg As String)

Trace イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Msg	発行したコマンドと受信したステータスです。

MoreData イベント

■機 能

SMTP メソッド発行直後、MimeMultipart プロパティに値が設定されていない場合、Data に””を指定するまで繰り返し発生します。Data に””を指定した場合の次に続くイベントは Done イベントです。

ここでは主に送信すべきデータ(メッセージ本文)を指定します。

※MIME 形式に対応する必要がない場合に利用します。

■構 文

Private Sub Object_MoreData(id As Integer, Data As String)

MoreData イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
id	メールを識別する一時的な番号です。
Data	送信すべき追加のデータです。 何も入力しなければ送信を完了します。

MimeHead イベント

■機能

SMTP メソッド発行直後、MimeMultipart プロパティに値が設定されている場合に、送信するデータ(本文またはファイル)を読み込む直前に発生します。

ここでの主な処理はデータごとのヘッダを生成することです。OriginalDataFormat で”Binary”を指定した場合、データはそのまま TransferEncoding にしたがってエンコードされます。

値を ContentType に指定した場合の次に続くイベントは MimeBody イベントです。ここで、”multipart/”で始まる値を指定した場合は、新たなマルチパートを開始します。ContentType に””を指定した場合、データの読み込みを終了し、Done イベントが続きます。

■構文

```
Private Sub Object_MimeHead(ByVal Depth As Integer, ByVal Part As Integer,
    OriginalDataFormat As String, ContentType As String,
    ContentName As String, ContentCharset As String,
    TransferEncoding As String, AdditionalHeader As String)
```

MoreData イベントの構文の指定項目は次のとおりです。

(指定項目)	(内容)
Object	MAILOCX オブジェクトです。
Depth	マルチパートのネストの深さです。
Part	パートの番号です。
OriginalDataFormat	送信するデータタイプです。 ”Binary”または”Unicode”を指定してください。
ContentType	送信するデータの種類です。 ”plain/text”、”image/jpeg”などです。
ContentName	送信するデータの名前です。
ContentCharset	送信するデータの文字コードです。 日本語の場合は通常”iso-2022-jp”を指定してください。
TransferEncoding	そのパートで使われるエンコード方式です。 (”7bit”、”8bit”、”quoted-printable”、”base64”、”binary”)
AdditionalHeader	追加するヘッダ情報をです。 複数行ある場合は改行を vbCrLf で区切ったデータを渡します。 最後の追加項目の後の改行は不要です。

MimeBody イベント

■機 能

MimeHead イベントの発生直後、Data に””を指定するまで繰り返し発生します。Data に””を指定した場合の次に続くイベントは MimeHead イベントです。

ここでは主に各マルチパートごとのデータ(メッセージ本文またはファイルのバイナリデータ)を指定します。

■構 文

```
Private Sub Object_MimeBody(ByVal Depth As Integer, ByVal Part As Integer, _  
    Data As String)
```

MimeBody イベントの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	MAILOCX オブジェクトです。
Depth	マルチパートのネストの深さです。
Part	処理中のパートです。
Data	送信するデータです。

Column...

送信時に指定する 代表的なメールヘッダについて(その2)

メールヘッダの中でも **Date**: および **Message-ID**: は扱い方に注意が必要です。

(その1)では、Date: および Message-ID: に記述がない場合は SMTP サーバ(正確には MTA(Message Transfer Agent))が追加することを述べましたが、送信者のメールソフトで追加することも当然可能です。ただしその場合には以下のフォーマットに従う必要があります。

Date : (例) Fri, 9 Mar 2001 10:17:20 +0900

曜日

日付

月

年

時刻

タイムゾーン

Message-ID : (例) <20010308164742.BB00.WILL@will-ltd.co.jp>

一意に識別するためならば形式は問わないこととなっている。通常は日時+α+ユーザー名となっていることが多い。

ドメイン名

Message-ID: の利用方法はいろいろありますが、代表的なものとしては送信したメールとそのメールに対して返信されてきたメールとの紐付け(スレッド)を行うことです。返信時には **In-Reply-To**: というメールヘッダに、返信の元となるメールの **Message-ID**: を記述することとなっています。これにより返信されたメールを受け取った送信者はメールの紐付け(スレッド)を行うことができます。

Message-ID: は一般的には SMTP サーバが付加します。この場合、そのままでは送信者のメールソフト上のメールに **Message-ID**: が残りません。そのため、紐付けを行う場合には、CC(Carbon Copy): や BCC(Blind Carbon Copy): に自分のアドレスを追加し、**Message-ID**: を送信者のメールソフトに渡す必要があります。このような手間を省くため、メールソフトで **Message-ID**: を追加している例も多く見られます。

なお、**Message-ID**: は一部形式が統一されていないこともあり、一意であることが保証されているものではありません。

Date: について、弊社ではクライアントマシンの時刻よりもサーバマシンの時刻を信頼し、SMTP サーバによって追加するものという立場を取っております。

しかしながら、Date: フィールドの正しい定義は「メッセージが転送できるようになった」時刻を表すものであり、「メッセージを転送した」時刻ではありません。メッセージを MTA が受け取った時刻(メッセージを転送した時刻)は **Received:** フィールドにも記されるため、Date: はクライアントの時刻によって付加すべきであるともいえます。

ダイアルアップなどの環境ではメールの作成後に送信ボタンを押してから、しばらく SMTP サーバに転送しないこともあり、クライアントの時刻を基準とすると Date: がかなり過去のものとなってしまうこともあります。

サーバまたはクライアントの時刻のどちらを適切と考えるかはそのメール送受信機能を追加するソフトの目的に依存するものと思われます。MAILLOCX はどちらの場合でも実現することが可能です。

白紙ページ

Winsock

エラーコード一覧

白紙ページ

SMTP メソッドや POP メソッドを発行した際に発生する Winsock エラーコードの発生までのおよその時間と原因

番号	発生までの時間	エラーメッセージの意味及び考えられる原因
10060	45	接続タイムアウト
		存在しない IP アドレスを指定した
		ルータが故障している
		ハブが故障している
		サーバがダウンしている
		ネットワークケーブルが抜けている
11001	0	ホスト名が見つからない
		ホスト名が正しくない、例えば xxxx.will-ltd.co.jp など
		DNS の設定が間違っている
		DNS の指定が間違っている
10065	45	hosts テーブルが間違っている
		ホストへの到達経路が存在しない
		ルーティングの設定が正しくない
10061	1	プライベートアドレスをインターネットで指定した
		接続が拒否された
11004	0	IP アドレスで指定したマシンは存在するがポート番号の指定が間違えているか、サービスが停止している
		データがない
		ポート番号に指定した文字が正しくない。 例えばポート番号に“SMTP”と書いた

送受信中に発生するエラーコード

番号	発生までの時間	エラーメッセージの意味及び考えられる原因
10054	180	相手が接続を強制的に切った
		送信に対してサーバが応答しない
		ケーブルが抜けた
		ハブが故障した
		ルータが故障した
		ルーターがビジー

白紙ページ

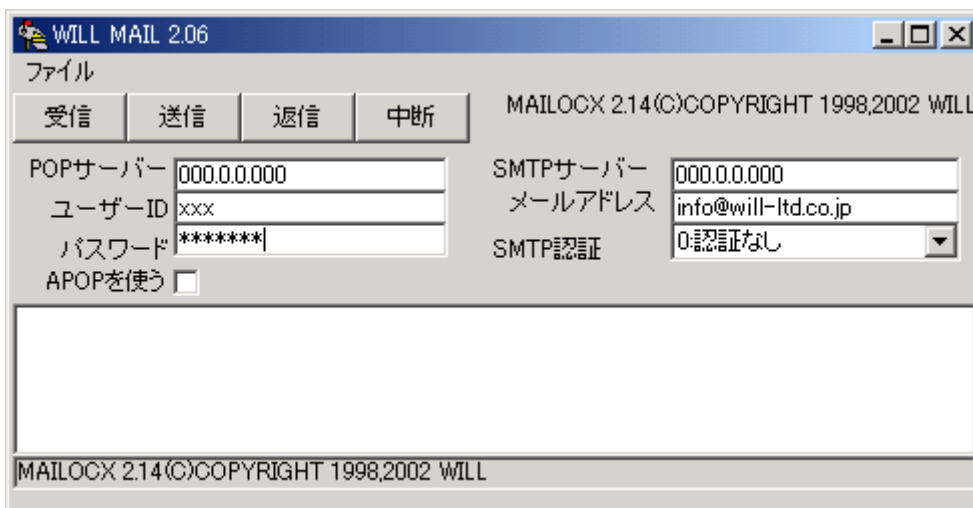
サンプル

サンプル

白紙ページ

WillMAIL

WillMail は、SMTP と POP を利用して電子メールを送受信するプログラムです。日本語や、添付ファイルに対応しているので、このサンプルを参考にすれば、本格的メールクライアントを簡単に作れます。

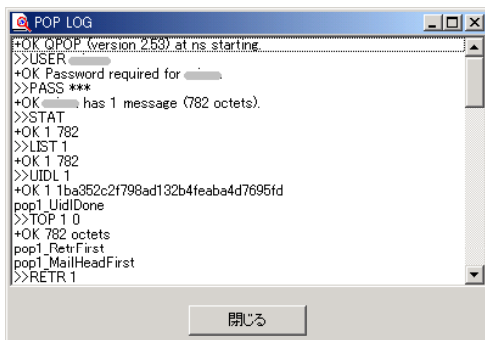


■メールの環境設定

1. POP サーバーと SMTP の IP アドレスあるいはドメインネームを指定してください。
(例、000.0.0.000 あるいは xxx.yyy.co.jp)
2. ユーザーID、パスワードとメールアドレスを指定してください。
3. SMTP 認証を行う場合、認証方法をリストボックスの中から選択してください。
4. APOP を利用しする場合、チェックボックスにチェックを付けてください。

《POP ログ》

ツールバーの「ファイル」→「ログ」を選択すると、POP サーバのログが表示されます。



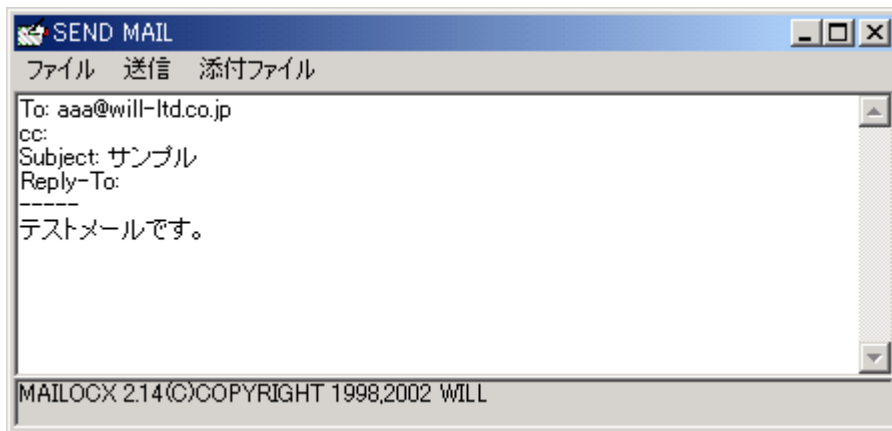
■メールの受信

受信ボタンを押すと、メールのチェックが行われます。新着メールの発信者とタイトルは一覧で表示されます。目的のメールをダブルクリックするとメールが開きます。



■メールの送信・返信

送信ボタンを押すと、SEND MAIL ウィンドウが開きます。



以下の必要な情報を記入し、送信メニューを押すと、メールが送信されます。返信するには、メール一覧を選択して返信ボタンを押すか、または、メールを開き、返信ボタンを押します。

《ヘッダ情報》

コマンド	内容	省略	書式(注 1)
To:	送り先の情報	不可	user user@xxx.co.jp <user@xxx.co.jp> USER <user@xxx.co.jp> <user@xxx.co.jp>(COMMENT)(注 2)
cc:	送り先の情報	可	同上 (注 3)

Bcc:	送り先の情報	可	同上（注 3）
Reply-To:	返信先の情報	可	同上（注 3）
Subject:	メールタイトル	可	自由（日本語対応）

《本文》

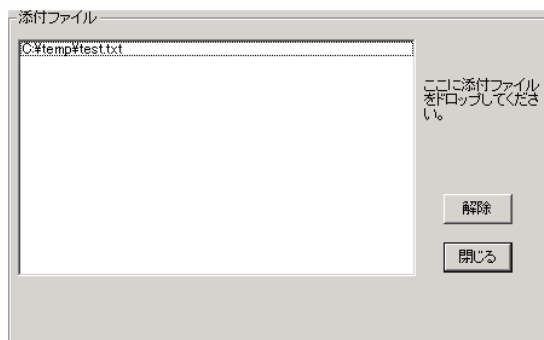
“-----” 以下は本文を書くスペースです。日本語に対応しています。

SMTP プロトコルの場合、二つの改行コードで、メールのヘッダ部分と本文部分を分けます。本プログラムは送信するとき、“-----”を二つの改行に変更して、ヘッダ部分と本文部分を分けます。

注 1	ヘッダ部分のコマンドを複数行で指定する場合は、先頭にタブ([Ctrl][Tab]キー)一つ以上、あるいは、スペース二つ以上を入れてください。
注 2	複数の受取人を指定する場合、“,”でメールアドレス区切ってください。 例:STAFF,INFO
注 3	メール受信時に表示される情報です。送信時では無効となっています。

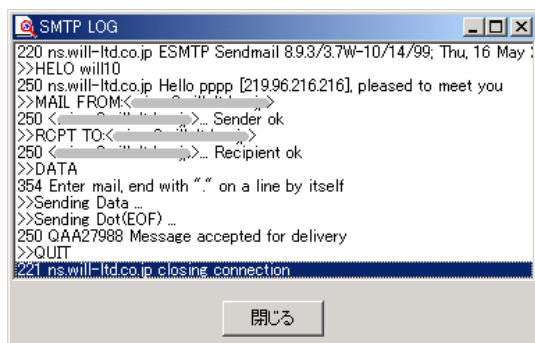
《添付ファイル》

ツールバーの「添付ファイル」を選択すると以下の画面が表示されます。添付するファイルをドロップしてください。



《SMTP ログ》

ツールバーの「ファイル」→「ログ」を選択すると、SMTP サーバのログが表示されます。



SMTP_VC5

VC で作成したメール送信のサンプルです。(ファイル添付可能)

セーフアレイ(SafeArray)を用いて複数の送信先リストを smtp メソッドに指示する方法を示しています。

〈使用方法〉

1.メールサーバの設定

メールサーバの IP アドレスを入力して下さい。

2.送信元ユーザーの設定

送信元のメールアドレスを入力して下さい。入力せずに送信すると、「NO FROM ADDRESS」というエラーメッセージが表示されます。

3.送信先の設定

・「送信先リスト」にあるメールアドレス宛てにメールを送信します。

・「送信先ユーザー」にメールアドレスを入力し、「追加」ボタンを押して「送信先リスト」に追加します。

・リストにあるアドレスを削除する場合は、該当するメールアドレスをダブルクリックして「送信先」に戻してから削除します。

4.表題

メールの表題を入力します。何も入力せず送信すると、「(none title)」と入力されます。

5.本文

送信するメールの本文を入力して下さい。

6.添付ファイル

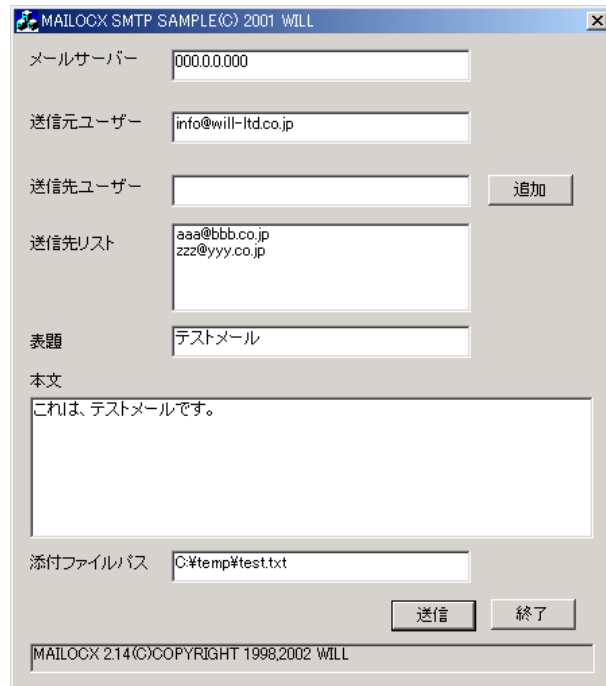
メールに添付するファイルを、フルパスで指定して下さい。

7.メール送信

「送信」ボタンを押すとメールを送信します。

8.終了

「終了」ボタンを押すと画面を閉じ終了します



SimpleDirectMail

MAILOCXを利用して、ダイレクトメールを送るサンプルです。Excel、Access のデータからメールアドレスを取得して、1通ずつ送信先の名前を挿入して送信します。

<ご注意>

送信先リストに ACCESS を選択し、送信を行うと次のようなエラーが表示されます。

「データベースの形式 '[mail.mdb があるフォルダ]¥mail.mdb' を認識できません。」

これは、ACCESS のバージョンが Access2000 Service Release 1(SR-1) の時に DAO のバージョンによりエラーとなります。

(<http://support.microsoft.com/default.aspx?scid=kb;JA;q272622>)

このようなエラーが表示される場合は、SimpleDirectMail.vbp を開き「プロジェクト」-「参照設定」で表示される参照設定画面で、「MicrosoftDAO 3.51 Object Library」のチェックをはずし、「Microsoft DAO 3.6 ObjectLibrary」にチェックをつけて SimpleDirectMail.exe の再作成を行って下さい。

<使用方法>

各項目を設定します

● SMTP サーバ名

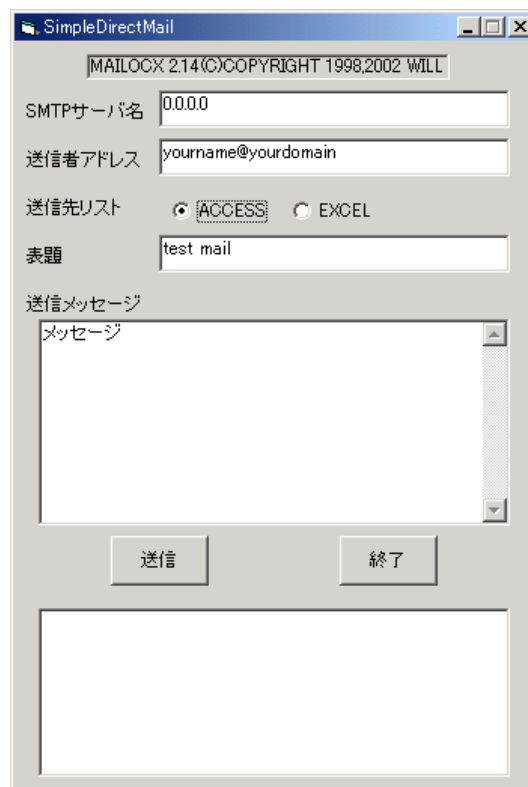
SMTP サーバの IP アドレスを指定して下さい。

● 送信者アドレス

送信者のメールアドレスを指定してください。
返信アドレス、エラー返信アドレスも、このメールアドレスになります。

● 送信先リスト

送信先リストのデータファイルを選択します。
¥SimpleDirectMail のフォルダーに付属のサンプルデータ(Excel、Access)があります。それぞれ名前とメールアドレスを入力して下さい。(初期設定のメールアドレスは、すべてエラーになります。)



サンプル

▼ Excel(初期状態)

	A	B	C
1	no	name	address
2	1	一太郎	name@domain
3	2	二太郎	name@domain
4	3	三太郎	name@domain

▼ Access(初期状態)

	no	name	address
1		一郎	name@domain
10		十郎	name@domain
11		士郎	name@domain

● 表題

送信メールの表題(件名)を入力します。

● 送信メッセージ

送信メールの本文を入力します。この本文の最初に、データの「name」に指定されている宛先がそれぞれ表示されます。

送信者 :	██████████
日時 :	13年 3月 30日 金曜日 午後 02:50
宛先 :	二太郎
件名 :	test mail

二太郎様へ
メッセージ

● メールを送信

「送信」をクリックして、ダイレクトメールを送信します。下部のリストボックスに送信ログが表示されます。

● 終了

「終了」をクリックして、SimpleDirectMail を終了します。

索引

<A>

Abend メソッド.....	40
AngleBracketType プロパティ.....	88
APOP プロパティ.....	34
AUTH プロパティ.....	85

<C>

Charset プロパティ.....	92
Copyright プロパティ(POP).....	31
Copyright プロパティ(SMTP).....	81

<D>

DecodeMimeHeadInMailBody プロパティ.....	32
DeleteOnError プロパティ.....	35
Done イベント(POP).....	45
Done イベント(SMTP).....	101

<F>

Fold78 プロパティ.....	91
-------------------	----

<H>

Hostname プロパティ.....	84
---------------------	----

<K>

KanjiIn プロパティ.....	89
KanjiOut プロパティ.....	90

<M>

MailHeadDone イベント.....	57
MailHeadFirst イベント.....	55
MailHeadNext イベント.....	56
MimeBody イベント.....	105
MimeDataDone イベント.....	64
MimeDataFirst イベント.....	62
MimeDataNext イベント.....	63
MimeHeadDone イベント.....	61

サンプル

MimeHeadFirst イベント	59
MimeHeadNext イベント	60
MimeHead イベント	104
MimeMultiPart プロパティ	82
MoreData イベント	103
MultiPartClose イベント	65
MultiPartOpen イベント	58
<P>	
PASS プロパティ	87
POP メソッド	39
Port プロパティ (POP)	33
Port プロパティ (SMTP)	83
<R>	
RetrDone イベント	66
RetrFirst イベント	53
RetrNext イベント	54
<S>	
Smtplib メソッド	97
Stat イベント	47
<T>	
TopDone イベント	52
TopFirst イベント	50
TopNext イベント	51
Trace イベント (POP)	46
Trace イベント (SMTP)	102
<U>	
UidDone イベント	49
UidFirst イベント	48
USER プロパティ	86

MAILOCX マニュアル

1998年4月21日	初版	第1版
1998年10月1日		第2版
1998年12月3日		第3版
1999年7月1日		第4版
2001年1月18日		第5版
2003年3月10日		第6版

発行所 株式会社ウィル

住所 神奈川県横浜市保土ヶ谷区西久保町 15

グランディシンヤ 302

〒240-0022

TEL: 045-338-3525

FAX: 045-338-3526

Mail-Address: info@will-ltd.co.jp

URL: <http://www.will-ltd.co.jp/>

発行者 小川 史彦

本紙の内容を許可なく複写、転載、データファイル化することを禁じます。
本紙の内容に関するご質問は、上記のメールアドレス宛にお問い合わせください。
