

4D Internet Commands

リファレンスマニュアル
Windows[®] および Mac OS[™] 版



4D Internet Commands

リファレンスマニュアル

Copyright©1999 ACI SA
All rights reserved.

このマニュアルに記載されている事項は、将来予告なしに変更されることがあり、いかなる変更に関しても ACI SA は一切の責任を負いかねます。このマニュアルで説明されるソフトウェアは、本製品に同梱の License Agreement（使用許諾契約書）のもとでのみ使用することができます。

ソフトウェアおよびマニュアルの一部または全部を、ライセンス保持者がこの契約条件を許諾した上での個人使用目的以外に、いかなる目的であれ、電子的、機械的、またどのような形であっても、無断で複製、配布することはできません。

4th Dimension, 4D, 4D ロゴ, 4D Server, 4D Write, 4D Draw, 4D Calc, 4D Backup, 4D Insider, 4D Compiler, ACI, ACI ロゴは ACI SA の登録商標であり、Customizer Plus は ACI SA の商標です。

Microsoft と Windows は Microsoft Corporation 社の登録商標です。

Apple, Macintosh, Mac, Power Macintosh, Laser Writer, Image Writer, ResEdit, QuickTime は Apple Computer Inc. の登録商標または商標です。

その他、記載されている会社名、製品名は、各社の登録商標または商標です。

注意

このソフトウェアの使用に際し、本製品に同梱の License Agreement（使用許諾契約書）に同意する必要があります。ソフトウェアを使用する前に、License Agreement を注意深くお読みください。

目次

1	4D Internet Commands	9
	序章.....	11
	ソフトウェアの必要条件.....	13
	用語集と用語	15
	引数のフォーマット.....	18
2	IT Send Mail	21
	メールの送信	23
	SMTP_SetPrefs	25
	SMTP_GetPrefs.....	27
	SMTP_QuickSend	28
	SMTP_New.....	30
	SMTP_Host.....	31
	SMTP_Send.....	32
	SMTP_Clear	33
	SMTP_Date	34
	SMTP_From.....	36
	SMTP_Sender	38
	SMTP_ReplyTo.....	40
	SMTP_To.....	42
	SMTP_Cc.....	43
	SMTP_Bcc.....	44
	SMTP_InReplyTo.....	46
	SMTP_References.....	47
	SMTP_Comments.....	48
	SMTP_Keywords	49
	SMTP_Encrypted.....	50
	SMTP_AddHeader.....	51
	SMTP_Subject.....	53
	SMTP_Body.....	54
	SMTP_Attachment.....	56

3	IT Review Mail	59
	メールの受信	61
	POP3_SetPrefs	63
	POP3_GetPrefs	64
	POP3_Login	65
	POP3_VerifyID	67
	POP3_Reset	68
	POP3_Delete	69
	POP3_Logout	70
	POP3_BoxInfo	71
	POP3_MsgInfo	72
	POP3_GetMessage	73
	POP3_MsgLstInfo	74
	POP3_MsgLst	76
	POP3_Download	78
	POP3_UIDToNum	79
4	IT Downloaded Mail	81
	メールのダウンロード	83
	MSG_FindHeader	84
	MSG_MessageSize	85
	MSG_GetHeaders	86
	MSG_GetBody	87
	MSG_GetMessage	88
	MSG_HasAttach	89
	MSG_Extract	90
	MSG_Delete	91
5	IT File Transfer	93
	ファイル転送	95
	FTP_Progress	96
	FTP_Login	98
	FTP_GetDirList	100
	FTP_GetFileInfo	102

FTP_VerifyID	103
FTP_MakeDir.....	104
FTP_RemoveDir	105
FTP_Rename.....	106
FTP_Delete.....	107
FTP_MacBinary	108
FTP_Send.....	110
FTP_Append.....	112
FTP_GetType	113
FTP_SetType.....	114
FTP_System	115
FTP_Receive	116
FTP_Logout	118

6 IT TCP/IP.....119

ローレベルルーチン	121
TCP_Open.....	122
TCP_Listen	123
TCP_Send	125
TCP_Receive.....	126
TCP_State	127
TCP_Close.....	128

7 IT Internet.....129

特殊なインターネットコマンド	131
NET_Finger.....	132
NET_Ping.....	133
NET_Time.....	134
NET_NameToAddr	136
NET_AddToName.....	137
NET_Resolve.....	138

8	IT Utilities	139
	ユーティリティコマンド	141
	IT_MacTCPInit	142
	IT_Platform	143
	IT_Version	144
	IT_TCPversion	145
	IT_MacTCPver	146
	IT_MyTCPAddr	148
	IT_SetTimeOut	149
	IT_GetTimeOut	150
	IT_ErrorText	151
	IT_Encode	152
	IT_Decode	154
	IT_GetPort	156
	IT_SetPort	157
9	付録.....	159
	付録 A プログラミングのヒント	161
	付録 B TCP ポート番号.....	163
	付録 C エラーコード	165
	付録 D その他の情報	174
	コマンド索引	175

4D Internet Commands は、4th Dimension のユーザに対してローカルエリアネットワーク (LAN) またはワイドエリアネットワーク (WAN) で運用可能な一連の通信ユーティリティを提供します。このようなコネクティビティが爆発的に急増した最大の要因は、一般的には“インターネット”という名前で知られています。ここ数年間でインターネットにアクセスした人や企業の数が増加したことを示しています。インターネットにアクセスする人々が急増したことで、その人々はビジネス社会において“ネット上に居る”ということを日々強く実感するようになってきました。

4D Internet Commands が提供する一連のコマンドは、データベースの開発者にインターネットの鍵となる多くの要素にアクセスする方法を提供します。SMTP コマンドには、リストアップされた人々に対してデータベースからの電子メールの配信を自動化するツールが含まれています。同様に、POP3 コマンドは、データベースに保管されている任意のメールボックスからメールを取り出したり、メールの再送信 / 自動返信またはリモート検索の実行等を行うことができます。FTP コマンドを利用すれば、遠隔地のシステムへの / 遠隔地のシステムからのファイル転送、FTP ボリューム上のディレクトリ内のファイルリストの取得といったことが可能になります。また、TCP コマンドはインターネット関連のタスクを実行可能にするローレベルなツールを開発者に提供します。

SMTP (Simple Mail Transfer Protocol) は、インターネットの至るところで利用されている代表的なメール転送プロトコルです。4D Internet Tool を利用すれば、SMTP サーバーを介して素早くメールを作成して送信することができます。メールの作成および配信は、単一コマンドと同じくらい簡単に実現できます。より複雑なメールの配信が必要な場合は、メッセージヘッダや本文、添付ファイルなどを制御し、配信に反映させることができます。インターネットのメールアドレスは、CompuServe や America Online、eWorld 等といったサービスへの配信も提供されているので、電子メールアカウントで実際の人物に連絡を取ることが可能です。SMTP の一連のコマンドが使用可能なその他の例を以下に挙げます：

- データベースレポートの自動配信
- 自動メール転送データベースの作成
- グループメールリストの管理
- 遠隔データベースの保管と転送 / 更新と同期化

4D Internet Commands には、SMTP コマンドの他にメールメッセージとエンコードされた添付ファイルの取得用に POP3 (Post Office Protocol, バージョン 3) のメールサーバーに接続するためのコマンドが含まれています。一連の SMTP コマンドおよび POP3 コマンドは、複数メッセージの同封に対応した標準の MIME に準拠しているので、バイナリーの添付ファイルを簡単にダウンロードまたは保存することができます。

また、これらのコマンドを使用することにより、Binhex や Base64、AppleSingle、AppleDouble といった別の方法で添付ファイルをエンコードすることもできます。

FTP (File Transfer Protocol) コマンドは、テキストファイルまたはバイナリーファイルを送受信するために FTP サーバーとの通信を簡単に行えるメカニズムを備えています。一連の FTP のコマンドは、データベースの開発者がファイルのある遠隔地のボリュームへの操作可能なインタフェースを作成することや、ディレクトリ内のファイルリストを取得することを可能にします。FTP コマンドは、クライアントのアプリケーションで遠隔地のボリュームを直接マウントすることなく、ドキュメント検索アプリケーションを簡単に使用することができます。

TCP/IP (Transmission Control Protocol/Internet Protocol) は、インターネット上でのデータの送受信に使用される主要なプロトコルです。4D Internet Commands は、未加工の TCP パケットを送信および受信するためのコマンドをいくつか用意しています。TCP のコマンド群は、インターネット通信を構築したり管理する上で必要不可欠なツールを開発者に提供します。

以下にいくつかの例を挙げます：

データベースを WWW サーバーに移管する

telnet のインタフェースを構築する

遠隔地のマシン上でシェルコマンドを実行する

World Wide Web からドキュメントを取得する

オンラインデータベースのデータを検索する

遠隔地のサーバーとデータベースとの同期を取る

Federal Express や UPS のパッケージを検索する

注意：日本語の扱いについて

インターネットメールの文化は欧米で生まれました。そのため、メールを送受信する仕組みは 1 バイト文字コードを前提としたものになっています。このため、インターネットメールで日本語を使用するためには、日本語をなんらかの変換により 1 バイトコードに変換させる仕組みが必要とされます。変換する仕組みをエンコード。逆に元に戻すことをデコードと呼びます。スパイが使用したという暗号化の手法に似ているかもしれませんが。

つまり、メールを送信するときには、メールのエンコードが必要です。そしてメールを受信する場合にはデコードが必要ということです。デコードをしていないメールは、まるで暗号文のように意味不明の英数字の固まりに見えるでしょう。

残念なことに 4D Internet Commands は日本語を解釈しません。日本語のエンコード、デコードの仕組みが組込まれていないからです。

しかし安心してください。添付の ACI Pack-J にはインターネットメールで日本語を扱うために必要なエンコード / デコード ルーチンが用意されています。4D Internet Commands と ACI Pack-J を併用することで、日本語メールの送受信が可能です。

日本語メールの送受信に必要な手順についてはコマンドの説明の所で逐次、補足いたします。

インストール

4D Internet Commands プラグインは、他のプラグインと同じように 4th Dimension に統合されます。

4D Internet Commands は ACI 製品 v6.5 をインストールすると、このプラグインは自動的に 2 つの異なる場所にインストールされ、使用することが可能になります：

- ・現在のシステムフォルダ（Windows 上の Windows¥ACI¥Win4DX または Mac OS 上の システムフォルダ：初期設定 :ACI:Mac4DX）にインストールされると、特別な操作をせずにすぐに使用可能になります。
- ・通常のインストールフォルダにインストールされると、インストール作業とは別に、ユーザはプラグインを移動したり複製することが簡単に可能になります。

このアーキテクチャにより、ユーザは 4D Internet Commands プラグインを、次のインストール方法を選択することができます：

- ・Mac4DX または Win4DX フォルダを、データベースストラクチャと同じフォルダ上に配置する。
- ・現在のアクティブなシステムフォルダ上に残す。

インストールに関する重要事項：

- ・4D Internet Commands が両方の場所にインストールされた場合、データベースストラクチャと同じレベルにインストールされた Mac4DX/Win4DX のみがロードされます。
- ・コンパイル：システムフォルダにインストールされた 4D Internet Commands を使用してコンパイルする場合には、4D Internet Commands プラグインの場所を 4D Compiler に指示する必要があります。
- ・データベースをコピー - したり移動する場合、または、他のマシン上で実行する場合、Win4DX/Mac4DX が同じ場所に配置されていることを確認してください。

4D Internet Commands を ACI 6.5 製品と独立してインストールする場合は、ACI Product Line CD-ROM 上のトップレベルにプラグインファイルがあります。

詳しい情報については、『インストールガイド』をご覧ください。

ソフトウェアの必要条件

以下のリストは、4th Dimension のデータベースで 4D Internet Commands を利用するために必要なソフトウェアの条件を列挙しています。これらすべての条件を満たす必要があるのではなく、使用するコマンドに関する要求のみを満たしていれば問題ありませんが、実際はすべての外部呼び出しは TCP/IP プロトコルを介して通信を行い、TCP/IP プロトコルのスタックはすべての相関関係の設定にまたがる必要条件を考慮する必要があります。

ハードウェア

- 680x0 または PowerPC ネイティブ
- Windows95/98 または WindowsNT が稼動する Intel マシン

4th Dimension (Macintosh および Windows)

- バージョン 6.0 以降の Macintosh 版 4th Dimension
- バージョン 6.0 以降の Windows 版 4th Dimension

MacTCP、Open Transport、Winsock

4D Internet Commands を通してすべてのコマンドは、通信用に TCP/IP プロトコルを使用します。このプラグイン内のコマンドを実行するコンピュータには、TCP/IP ドライバをインストールし、一意の IP アドレスを正しく設定しておく必要があります。ほとんどのオペレーティングシステムでは、Macintosh 上の MacTCP や Open Transport、Windows 上の Winsock のような TCP/IP ドライバをプレインストールしてあります。TCP/IP の設定に関する詳細は、ネットワーク管理者に直接尋ねるか、Network Components for 4D Server のリファレンスマニュアルを参照してください。

注意：バージョン 6.5 から使用する場合、4D Macintosh Network Components は Open Transport でしか動作しません（「Classic」のネットワークは、サポートされていません）。しかしながら、4D Internet Commands は 4D TCP/IP Network Component を使って TCP/IP プロトコルと通信してはいません。したがって、4D Internet Commands は MacTCP ドライバを使用することが可能です。一方、バージョン 6.5 以降の 4D Client またはバージョン 6.5 以降の 4D で Web 機能を使用したい場合は、MacOS 上で Open Transport TCP/IP ドライバを使用する必要があります。

ネットワークアクセス

一連の 4D Internet Commands を使用するためには、TCP/IP をサポートするネットワークにアクセスしなければなりません。

ドメインネームサーバー

ほとんどの 4D Internet Commands がドメインネームサーバーにアクセスすることが必要になります。TCP/IP の設定に関する詳細は、Network Components for 4D Server のリファレンスマニュアルを参照するか、または Apple ガイドを参照してください。

SMTP メールサーバー

SMTP のコマンド群を使用してメールを送信するためには、POP3 メールサーバーにメッセージを転送する SMTP メールサーバーに送り手がアクセスする必要があります。

POP3 サーバー

POP3 コマンドを使用するためには、POP3 メールサーバー上にアカウントを持っている必要があります。

ここでは、このマニュアルを通して使用されている多くのリファレンスを定義します。定義は簡略化され、リファレンスに精通していない人を対象に作成されています。“引数のフォーマット”に関する「用語」のところでは、4D Internet Commands の共通引数のフォーマットについて、詳しい情報を提供しています。

NIC : Network Information Center の略です。インターネットは、その大部分が無秩序なものです。インターネットの使用や成長に対して、中央集権化された権限や管理がありません。しかし、単一の機関によって管理される場合、効果的に遂行するためにはドメイン名や IP アドレスの登録といった基本的な管理上の必要性がいくつか存在します。NIC とは、そのような管理上のタスクの責任を負う人々のグループです。

RFC : Request for Comments の略です。4D Internet Commands のほとんどは、インターネット通信を操作するための基本的な定義に基づいています。インターネットで用いられる基本的な方法論、説明およびプロトコルは、RFC として知られている公式文書において定義されています。付録 E の「その他の情報」では、有益な RFC ドキュメントが掲載されている場所を示した WWW サイトがいくつか紹介されています。ローレベルの TCP ルーチンを介して独自の通信プログラムを作成する人々は、それらに精通していなければならないのですが、このような RFC ドキュメントを参照しなくても済むように構築されたのがこの 4D Internet Commands のパッケージです。

TCP/IP アドレス、ホスト名およびドメイン名 : IP アドレスは、世界中のどこかに存在する特定マシンを参照します。IP アドレスは、ピリオドで区切られた 4 つの数値を含む文字列で書式化されています（例えば、「207.94.15.3」）。アドレスの数字の各部分には、0 から 255 までの値を指定することができます。いくつかの数学的な機能を IP アドレスに適用することにより、このマニュアルの中で「**ip_ 倍長整数**」として参照する倍長整数値の中にそのアドレスの値を組み込むことができます。

サイト（例えば、企業、大学等）がインターネット上にコンピュータを置くには、それらの IP アドレスがネットワーク上の他のマシンと衝突しないような確証がいくつか取られていなければなりません。団体（時には個人）は、ドメイン名を取得するために **NIC** でサイトを登録します。**ドメイン名** は、**ドメインネームシステム** (Domain Name System) (DNS) によってネットワークで使用される数値のアドレス（インターネットプロトコル番号）(Internet Protocol [IP] numbers) に変えることができるインターネットアドレスを簡単に記憶するシステムを提供しています。このシステムにより「www.acius.com」や「ftp.acius.com」といった読みやすいフォーマットになります。

ドメイン名 = "acius.com"

ホスト名 (コンピュータ名) = IP アドレス = ip_ 倍長整数

"www.acius.com" = "207.94.15.3" = -815919357

ホスト名と、それに対応する IP アドレスの関係は、DNS (Domain Name System) として知られているデータベースに格納されています。これらのサーバーは、相互に通信を行って世界中のドメインネームリストの新規データや変更されたデータを交換します。TCP/IP コントロールパネルは、使用しているコンピュータが任意の DNS を “ 特定する ” 方法を提供します。そして、この DNS は使用するすべてのドメイン名のリファレンスを分解します。

すべてのドメインネームサーバーがそれに対応する IP アドレスを持っているということを理解することが重要です。しかし、すべての IP アドレスがそれに対応するドメインネームサーバーを持っているというわけではありません。また、"jsmith@acius.com" のような「メールアドレス」は、個人の特定のコンピュータや IP アドレスを参照しません。メールアドレスは、ドメイン "acius.com" を分解することによって識別された IP アドレスを持つマシンへの配信を指示します。メールは、そのマシン上で稼働している POP3 サーバーに配信され、"jsmith" と名付けられたユーザ用にメールを保持します。

ドメイン名：ドメイン名は、インターネット上でコンピュータを識別したり、突きとめたりするためのアドレス構文です。ドメイン名は、ドメインネームシステム (Domain Name System) (DNS) によって、ネットワークで使用される数値のアドレス (インターネットプロトコル番号) (Internet Protocol [IP] numbers) に変えることができるインターネットアドレスを簡単に記憶するシステムを提供しています。ドメイン名は、階層構造であり、しばしばドメイン名を使用している実体のタイプについての情報を伝えます。ドメイン名は、単にドメイン名の全体のスペースのサブセットであるドメインを表現するラベルです。階層内の同レベルのドメイン名は一意でなければいけません。例えば、階層内の最上レベルには com が唯一存在可能であり、また階層内の次のレベルには acius.com のみが存在可能です。組織名が「企業名」である場合、「企業名.com」というドメイン名を登録することができ、電子メールアドレスは「ユーザ名@企業名.com」とすることができます。また、Web ブラウザで「www.企業名.com」へ行くことにより、顧客が組織の web サイトにアクセスすることもできます。

ドメインネームシステム (Domain Name System) (DNS)：配信されたデータベースの情報は、使用者にとって、記憶したり使用するのが簡単なドメイン名をインターネット上で互いに識別する必要のあるコンピュータを特定するインターネットプロトコル (Internet Protocol) (IP) 番号に変換するために使用されます。地球上でコンピュータを使用して作業を行っている人々は、このデータベースの特定の部分を維持しています。また、データベースの各部分に所持されているデータは、インターネット上のすべてのコンピュータおよびすべてのユーザによって作成されます。DNS は、コンピュータとデータファイル、ソフトウェアおよび共に作業をする人々によって成り立っています。

エンコード：文字の集まりのすべての種類をサポートしているわけではない他のコンピュータシステムに渡ってもファイルが移動できるように、エンコードはファイルを 1 つのフォーマットから他のフォーマットに変換します。エンコードの最も多い共通の形態は、バイナリー 16 進法 (Binhex) 暗号化です。Binhex エンコードは、メッセージに付加する添付ファイル用のデフォルトエンコードオプションです。エンコードは、元のファイルよりも大きい新規ファイルを作成しますが、データフォーク、リソースフォークおよびファインダ情報を添付ファイルとして簡単に送信される文字ファイルに変換します。4D Internet Commands は、Binhex、Base64、AppleSingle、AppleDouble、UUEncode および MacBinary といった最も一般的なエンコード方法をサポートしています。

暗号化：暗号化は、メッセージの内容を故意に解読されにくくするために使用されます。メッセージに関するプライバシー保護の必要性が強まったために、メッセージは PGP のような外部暗号化プログラムを使用して暗号化されます。暗号化されたテキストは、読み取られる前に解読される必要があります。4D Internet Commands では、テキストの暗号化の方法は一切提供していません。

圧縮：圧縮は、ファイルによって占有されるスペースを減らす目的で使用されます。ファイルを圧縮するためには、Stuffit Deluxe(tm)、Compact Pro(tm) または WinZip(tm) といったアプリケーションでの処理を通す必要があります。さらに、これらのファイルは、元のフォーマットに戻すために、アプリケーションを使用して解凍（展開）される必要があります。圧縮用アプリケーションを使用してファイルが圧縮されている場合、これらのアプリケーションで共通しているのは、元のファイル名に接尾語を追加する点です。以下にいくつかの一般的な接尾語とアプリケーションを列挙します。

ファイル名 .SIT – Stuffit アプリケーション

ファイル名 .CPT – Compact Pro アプリケーション

ファイル名 .DD – Disk Doubler アプリケーション

ファイル名 .ZIP – Winzip アプリケーション

ファイル名 .SEA – 自己解凍形式の圧縮ファイル (Self Extracting Archive)。このタイプのファイルは、Macintosh のスタンドアローンアプリケーションであり、解凍用のアプリケーションコードを内包しているため、ユーザがファイルをダブルクリックした際にそれ自身が解凍を行います。自己解凍形式の圧縮ファイル (Self Extracting Archive) は、アプリケーションコードを付加しているためにファイル名 .sit やファイル名 .cpt として作成されているファイルよりも通常は大きくなります。しかし、ユーザが圧縮用のアプリケーションを持っている必要がないので、このオプションはエンドユーザにとって有益なものとなります。

ファイルが最終の目的地まで到達する過程でマシンからマシンへ正確に移動される確証を得るためには、圧縮されたファイルを送信する前に暗号化する必要があるということ覚えておくことは重要です。

文字数、文字位置：コマンド説明中の文字数、文字位置はバイト数をあらわします。日本語の全角文字は全て 2 文字として数えます。

引数のフォーマット

以下の説明では、4D Internet Commands を通して使用される鍵となる引数の意味およびフォーマットについて詳しく述べています。

引数	タイプ	説明
ホスト名	文字列	ホスト名 (例: "www.companyname.com") IP アドレス (例: "204.118.90.2")
ip_ 倍長整数	倍長整数	IP アドレスのリファレンスの倍長整数
メールアドレス	テキスト	例: "jsmith@acius.com" (例題にある "<" ">" は含まない)
アドレスリスト	テキスト	例: "jsmith@acius.com, jdupont@aci.fr"
ローカルパス	テキスト	- ドキュメント Mac: "HD:4DDB:SaleDB:Report" Win: "C:¥myFolder¥4DDB¥SaleDB¥Report.txt"
		- ディレクトリ Mac: "HD:Stuff:" (最後の ":" に注意) Win: "C:¥myFolder ¥ Stuff¥"
ホストパス	テキスト	- ドキュメント "/User/jsmith/Report/SaleReport.txt" - ディレクトリ "/User/jsmith/Report/" (最後の "/" に注意)
tcp_ID	倍長整数	TCP セッションを開くためのリファレンス
smtp_ID	倍長整数	新しいメールメッセージのためのリファレンス
pop3_ID	倍長整数	POP3 セッションを開くためのリファレンス
ftp_ID	倍長整数	FTP セッションを開くためのリファレンス
関数の返す値	整数	エラーコード

引数 < ホスト名 >

引数 < ホスト名 > は、"dns.acius.com" または "204.118.90.2" といったホスト名または IP アドレスです。ホスト名は、ドメインネームシステム (DNS) を介して分析されます。デフォルトのドメインネームシステムおよび 2 番目ドメインネームシステムは、インストールされた TCP/IP ドライバのコントロールパネルの中で一般的な設定が行われます。引数として < ホスト名 > を必要とするすべての 4D Internet Commands で、ホスト名 ("www.acius.com") または IP アドレス ("204.118.90.2") フォーマットの値を受け入れます。リモートサイトでのハードウェアの変更などによって正しい結果が得られない場合があるため、一般的にはアプリケーションへの影響を緩和する「ホスト名」フォーマットの方が使用されます。

引数 < ip_ 倍長整数 >

すべてのホスト名は、IP アドレスに対して前ページで説明された方法で分析することができます。さらに値を一意である倍長整数の値に変換するために、IP アドレスに数学的公式を用いることができます。第 7 章にある NET_NameToAddr コマンドと NET_AddrToName コマンドは、この変換の過程を自動化します。この「倍長整数」の値は、このマニュアルを通して < ip_ 倍長整数 > として参照されます。この「倍長整数」の値は、開発者が直接 TCP 通信を行っているような特別な環境でのみ有効です。また、開発者の中には文字列の対応物に比較されるディスクの空きを節約するためにドメイン名の「倍長整数」の値を格納することを好む人もいます。

引数 < メールアドレス >

「メールアドレス」は、" ユーザ名 @ ドメイン名 " というフォーマットの中でメールの明細を完全に与えています。このマニュアルでは、引数 < メールアドレス > は「**1 つの電子メールアドレス**」を参照します。2 つ以上のアドレスを取得する場合、4D Internet Commands では引数として < アドレスリスト > を指定します。引数が唯一のタイプとして < メールアドレス > を持つ場合、その引数は 1 つのメールアドレスを取得します。< メールアドレス > のフォーマットは、ユーザ名とドメイン名の両方を持った完全なリファレンスでなければいけません。< メールアドレス > は、POP3 サーバーから受け取ったメールで時折見られるコメントを含むことはできません。

```
"Felix Unger" <felix@pristine.com>  
oscar@slobs.com (Oscar Madison)
```

注意：POP3 呼び出しから取得したヘッダ情報を用いて返信先のアドレスを指定する場合、アドレスからすべてのコメントが取り除かなくてはならないことに注意してください。

上記の例では "Felix Unger" および (Oscar Madison) がコメントです。詳しくは、RFC822 に記されています。"()" でくくられた文字列は全てコメントです。またメールアドレス中の "<>" でくくられた文字があった場合、くくられた文字がメールアドレスです。それ以外はコメントです。受信したメールのメールアドレスは、このどちらかの方式でコメント（実名など）がついています。

引数 < アドレスリスト >

引数 < アドレスリスト > は、カンマまたはキャリッジリターンで区切られた < メールアドレス > のフォーマットで **1 つ以上の電子メールアドレス** を含んでいます。テキストフィールドに複数のアドレスを入力または貼り付ける場合には、キャリッジリターン区切りが有効です。以下に挙げた 3 つの例は、各々受け入れ可能な \$ アドレスリストの値を生成します。

```
$ アドレスリスト := "jsmith@acius.com"
```

```
$ アドレスリスト := "jsmith@acius.com, scott@acius.com, marcel@aci.fr"
```

```
For ($i;1;Size of Array (a アドレス ))
```

```
$ アドレスリスト := $ アドレスリスト + a アドレス { $i } + Char (13)
```

```
End For
```

引数<ローカルパス>

引数<ローカルパス>は、ユーザのマシン（MacまたはWindows）上でのファイルまたはディレクトリの場所を指定します。Macintosh上では、フォルダ内のアイテムはコロン（:）で区切られます。例えば「HD」というタイトルが付けられたハードディスク上の「Report」というフォルダ内の「myReport」というファイルは、「HD:Report:myReport」というパス名を持つことになります。Macintosh上のディレクトリの指定は、コロン文字で終了します。例えば、新規のレポートを上記の例と同じフォルダに保存したい場合、「HD:Report:」のディレクトリを参照します。「ファイル名」または「ディレクトリ名」のどちらを指定するかは、コマンドによって呼び出されたコンテキストを基に決められます。Windows上でも、コロン文字の代わりに円マーク“¥”が使用されるという点を除いて同様のフォーマットが使用されます。

引数<ホストパス>

引数<ホストパス>は、Unixのオペレーティングシステム下で稼働しているコンピュータ上のファイルまたはディレクトリの場所を指定します。Unix環境下では、ディレクトリはスラッシュ（"/"）で区切られます。例えば、「ACI」ディレクトリ内の「Report」ディレクトリ内にある「Report.txt」ファイルは、「/ACI/Report/Report.txt」というように指定されます。ディレクトリのパス名は、「/」という文字で終わる必要があります。フルパス名は、ボリュームのルートを表す「/」で始まるということに注意してください。

引数<smtp_ID>、引数<pop3_ID>、引数<ftp_ID>、引数<tcp_ID> : 4D Internet

Commandsの各章では、ほとんどのコマンドにおいてリファレンスは「ID」番号から成り立っています。各一連の通信関数は、倍長整数である「ID」番号によって表される「セッション」を生成します。セッションのオープンに関係する後述のコマンドでは、正しい結果を導くためにこの値を使用します。

各区分（SMTP、POP3、FTP、TCP）で取得した「ID」番号は、別の区分に値として渡すことはできません。例えば、ftp_IDという変数で識別されたFTPセッションをどんな特別な処理であろうとTCPルーチンに渡すことはできません。

セッションのリファレンス

オープン

クローズ

tcp_ID	TCP_Open	コマンドまたは TCP_Listen コマンド TCP_Close コマンド
smtp_ID	SMTP_New	コマンド SMTP_Close コマンド
pop3_ID	POP3_Login	コマンド POP3_Logout コマンドまたは POP3_VerifyID コマンド
ftp_ID	FTP_Login	コマンド FTP_Logout コマンドまたは FTP_VerifyID コマンド

関数の返す値

すべての 4D Internet Commands（IT_ErrorText コマンドと IT_Version コマンドを除く）は、関数の返す値として整数値を返します。この整数は、コマンドが 4D のデータベースに伝達する必要のあるエラーコードも含んでいます。

コマンドの実行が成功した場合は、0 が返されます。そうでなければ、エラーコードが返されます。4D Internet Commands のエラーコードに関する詳細は、「付録 C エラーコード」を参照してください。

注意：ローカルパス、ホストパスに指定する一連のパス名にはできる限り日本語を避けることをお勧めします。なぜなら、日本語のコードの中にはパス名の区切りコードである“/”、“:”、“¥”を含むことが有り、誤動作の原因になることがあるからです。

2

IT Send Mail

SMTP (Simple Mail Transport Protocol) は、インターネット間で使用されるメールの標準プロトコルです。4D Internet Commands を使用することにより、開発者は 1 つのコマンドで単純なメッセージを、また複数のコマンドで複雑なメッセージを構築することができます。SMTP コマンドによって、開発者は「返信用ヘッダ」、「送信人ヘッダ」、「添付ファイル」、「コメント」および「リファレンス」を含むメールメッセージのすべての部分を管理することができます。

4th Dimension と 4D Internet Commands を使用すれば、インターネット上でメッセージや添付ファイルを送信する機能を持った非常に強力なデータベースを作成することができます。一連の SMTP コマンドによってデータベースの質を向上させる方法として、以下にいくつかの例を挙げます：

4th Dimension 内で作成されたレポートやドキュメントの送信を自動化する。

データベースが、特殊な事態を開発者に知らせることを可能にする（例：ON ERR CALL("Mail_Error")）。

データベースから海外の人々へのメールの自動送信を可能にする。

一連の SMTP コマンドには、様々な使用方法があります。（ファイルと添付ファイルの両方を取得している）POP3、FTP および TCP 用のコマンドと組み合わせて使用することにより、SMTP コマンドは 4th Dimension の開発者に 4D のデータベースの通信機能を劇的に増大させるツールを提供できます。

メールメッセージを作成する 2 つの方法

SMTP コマンドの節では、“簡単”および“複雑”な電子メールの送信方法がそれぞれ別々に説明されています。“簡単”な方法では、アドレスやメッセージの送信に必要なすべての引数を受け取るコマンドである SMTP_QuickSend のみを使用します。

世界中で送信されている電子メールの大半は、“ここ”に居る誰が、“そこ”に居る誰に、ある“件名”に関する“メッセージ”を送信したいのか、といった極めて単純な構造をしています。これが紙の手紙であつたら、すべてを書き終えた後に封をして、宛名を書き、配達してもらうために郵便局までその手紙を持っていくことになります。SMTP_QuickSend を用いれば、1 つのコマンドで簡単な電子メールの配信に“差出人”、“受取人”、“件名”および“メッセージの本文”を指定することができます。

しかし、すべてのメール配信がこのような限られた引数で適応できるわけではありません。例えば、コピーが必要な手紙や年次報告書のような同封するべき添付ファイルを別の関係者に送ると仮定してください。このような場合、スタッフが資料と照合し、必要な部数のコピーを作成し、送付状を作成することになるでしょう。4D Internet Commands の SMTP コマンドは、電子メール配信におけるすべての場面を管理することにより電子的な配達を容易にします。カーボンコピー（Carbon Copy）およびブラインドカーボンコピー（Blind Carbon Copy）アドレス指定、複数の添付ファイル、メールヘッダの詳述は、SMTP コマンドの「メッセージ構築」機能を通して処理されます。

メール配信の理解

SMTP コマンドの理解する上で重要な概念の 1 つは、どのメールを受取人に配信するかという方法に関係しています。SMTP コマンドは、メールを各受取人に直接配信するわけではありません。コマンドは、メールの正確な構造およびフォーマットを処理し、SMTP_Host コマンドによって指定された SMTP サーバーに結果を配信します。SMTP サーバーは、組織内のマシンであったり、またはインターネットサービスプロバイダのマシンであったりします。SMTP サーバーは、メールの最適な配信経路を決定し、メール管理者によって定義された設定に基づく配信スケジュールを立てます。

複雑な SMTP メッセージの送信に最低限必要なもの

SMTP コマンドを使用して構築されたメールメッセージの配信を成功させるためには、絶対に不可欠なコマンドが正確に定義されている必要があります。以下のコマンドは、電子メール配信を成功させるための最低限必要な事象を示しています：

SMTP_New コマンド

新規のメッセージ用に、メモリに空きを作り、次に実行するコマンドで使用するリファレンスを返します。

SMTP_Host コマンド

メッセージが配信される先の SMTP サーバーを指定します。

SMTP_From コマンドまたは SMTP_Sender コマンド

少なくとも、どちらかのヘッダにアドレスが 1 つ存在します。

SMTP_To コマンドまたは SMTP_Cc コマンドまたは SMTP_Bcc コマンド

少なくとも、これらのヘッダの内の 1 つにアドレスが 1 つ存在します。

SMTP_Send コマンド

メッセージを送信します。

SMTP_Clear コマンド

メッセージの作成の際に使用されたメモリを消去します。

上記のコマンドのみが実行された場合、メッセージは「件名」の定義やメッセージ本文を含んでいないものを送信します。これでは意味がないので、メッセージを効果的に通信するために追加の詳細情報を指定する必要があります。

SMTP_SetPrefs

SMTP_SetPrefs(ラインフィード ; 本文タイプ ; 行の最大長) 整数

引数	タイプ	説明
ラインフィード	整数	1=[デフォルト値] 追加、0= 追加しない、 -1= 変更しない
本文タイプ	整数	本文 - 内容 - タイプ (1=[デフォルト値] 自動検出、 -1= 変更しない)
行の最大長	倍長整数	行の最大長 (0=[デフォルト値] 自動検出、 -1= 変更しない)
関数の返す値	整数	エラーコード

説明

SMTP_SetPrefs コマンドは、SMTP コマンドを使って送信されるメッセージの環境設定を行います。コマンドは、グローバルスコープを持っていて、SMTP コマンドで作成されたその後の全メッセージに影響を及ぼします。設定可能なオプションは、SMTP サーバーに送信されるメールメッセージのフォーマットに影響を及ぼします。環境設定は、インタープロセススコープを持っており、任意の 4D のプロセスで作成されたメールに影響を及ぼします。

SMTP サーバーは、キャリッジリターン / ラインフィード (CR/LF) という連結した一対の文字で行の終わりを認識します。これは、1 つのキャリッジリターンを行 / 段落マーカの終わりで見なす、ほとんどの Macintosh アプリケーションと異なる点です。

引数 <ラインフィード> は、メールメッセージの本文中に存在するキャリッジリターンの処理方法を指定する整数値です。この引数に 0 を渡すと、メッセージ本文のテキストは原文のまま保持され、開発者に独自のラインフィード付加を管理する権限を与えます。(デフォルト値である) 1 の場合は、すべてのキャリッジターン (CR) をキャリッジターン / ラインフィード (CR/LF) の一対の文字に置き換えます。値として -1 を指定すると、現在の設定値がそのまま保持されます。どのオプションを選択すべきか迷う場合は、デフォルト値である 1 を選択してください。

引数 <本文タイプ> は、送信されるメッセージの本文 - 内容 - タイプを以下の表の値に従って指定します。変更されていない場合、デフォルトの本文タイプは 1 で、SMTP コマンドがメッセージの本文の内容に基づいた適切な設定を自動検出することができます。

- 1 変更しない
- 0 アプリケーションおよびバイナリー : エンコードなし
- 1 デフォルト値 : メッセージ内容に基づいて "US-ASCII および 7 ビット " または "ISO-8859-1 および引用可能 - 印刷可能 " のどちらかを選択する。
- 2 US-ASCII および 7 ビット
- 3 US-ASCII および quotable - printable
- 4 US-ASCII および base64

- 5 ISO-8859-1 および quotable – printable
- 6 ISO-8859-1 および base64
- 7 ISO-8859-1 および 8bit
- 8 ISO-8859-1 および binary
- 9 予約
- 10 ISO-2022-JP (日本)
- 11 ISO-2022-KR (韓国)
- 12 ISO-2022-CN (中国, simplified, traditional)
- 13 HZ-GB-2312 (中国, simplified)

「10:ISO-2022-JP (日本)」、「11: ISO-2022-KR (韓国)」、「12: ISO-2022-CN (中国, simplified, traditional)」サポートについては、メッセージボディは 7 ビットエンコードされている必要があります。コンバージョンは、SMTP_QuicikSend や SMTP_Body が呼び出される前に行われている必要があります。なお、「10:ISO-2022-JP (日本)」への変換には、AJP Nkf コマンドにより変換可能です。

引数<行の最大長>は、メッセージ本文のテキストに対する SMTP 行の最大長を指定します。SMTP コマンドは、行の最大長に達する直前の単語の切れ目にキャリッジリターン / ラインフィードの一对の文字を挿入して、本文のテキストを改行します。値として -1 を指定した場合、現在の値を変更しません。すべての番号を指定することが可能ですが、行の長さは 80 バイト以下にすることをお勧めします。ただし、日本語メッセージを含む場合、漢字の泣き別れ (2 バイトの途中で改行される) が発生し、行端で文字化けを発生させる事があります。これに対する対策は ACI Japna Pack の日本語ルーチンを使用し、あなたのプログラムの中で適度に改行処理をしてください。プログラムが面倒な場合は、<行の最大長>を大きめの値にし、4DIC による自動改行がされないように設定してください。1 行を 80 バイト以内にしたい方が良い理由は、メールを読む相手への考慮です。そして一部のメール処理システム (例えばメーリングリスト) では、1 行の文字数が長い場合、日本語変換に支障をきたし、文字化けする可能性があります。

<行の最大長>のデフォルト値は、0 です。値として 0 を指定することによって、引数<本文タイプ>用の RFC の定義内で指定された推奨値を SMTP コマンドが使用可能になります。<行の最大長>が 0 に設定された場合、以下の表に基づいた改行処理が生じます。

本文タイプ	改行
Base64	76
quotable – printable	76
その他	改行なし

多くのシステムおよびメールプログラムは、行の長さに制限のないメッセージの処理において問題を抱えているので、行の改行処理は強く推奨されています。また、メールが多くのシステムを通して最終目的地に到着すること、また配信パスに沿ったコンピュータがメッセージのフォーマットを処理できない場合にメッセージを拒否するかもしれないことを覚えておいてください。

参照

「SMTP_GetPrefs」コマンド

注意: 日本語メールを送信する場合には、<本文タイプ>に 10 を指定してください。またこの時、本文は AJP Nkf コマンドにより JIS コードに変換する必要があります。

SMTP_GetPrefs

SMTP_GetPrefs(ラインフィード ; 本文のタイプ ; 行の最大長) 整数

引数	タイプ	説明
ラインフィード 加	整数	0= ラインフィードを追加しない、1= ラインフィード追
本文のタイプ	整数	ボディの Content_Type
行の最大長	倍長整数	行の最大長
関数の返す値	整数	エラーコード

説明

SMTP_GetPrefs コマンドは、SMTP の環境設定に割り当てられた現在の設定を返します。SMTP_SetPrefs コマンドが呼び出されることで設定が変更されていなければ、値はデフォルトのままです。引数についての説明は、「SMTP_SetPrefs」コマンドの説明を参照してください。

引数<ラインフィード>は、SMTP コマンドがメールメッセージの本文内にあるキャリッジリターンをどのように処理するかを示している現在の設定値を返します。

引数<本文のタイプ>は、ボディの Content_Type についての現在の設定値を返します。値についての説明は、SMTP_SetPrefs コマンドの引数<本文のタイプ>の箇所を参照してください。

引数<行の最大長>は、メッセージ本文中のテキストの行の最大長についての現在の設定値を返します。

参照

「SMTP_SetPrefs」コマンド

SMTP_QuickSend(ホスト名 ; 差出人 ; 受取人 ; 件名 ; メッセージ) 整数

引数	タイプ	説明
ホスト名	文字列	ホスト名または IP アドレス
差出人	テキスト	メールアドレスまたはアドレスリスト
受取人	テキスト	メールアドレスまたはアドレスリスト
件名	テキスト	メッセージの件名
メッセージ	テキスト	メールメッセージ
関数の返す値	整数	エラーコード

説明

SMTP_QuickSend コマンドを使用することによって、メールのメッセージをコマンド 1 つで構築し送信することができます。メッセージに関してより高度な管理を必要とする場合、またはメッセージがより洗練された性質のものである場合、SMTP_New コマンドを基とした SMTP のコマンド群を使用する必要があります。

引数<ホスト名>は、メッセージ配達のために送信される SMTP サーバーのホスト名または IP アドレスです。

引数<差出人>は、メッセージを送信した人が誰であるのかを示す 1 つ以上のメールアドレスで構成される「アドレスリスト」を含むテキストの値です。メッセージを受け取る人は、「差出人」ヘッダに表示されたすべてのアドレスを見ることができます。

引数<受取人>は、1 つ以上のメールアドレスで構成される「アドレスリスト」を含んでいます。引数<受取人>ヘッダに表示されたすべてのアドレスにメッセージのコピーが送信されます。メッセージを受け取る人は、各々メッセージが配信された他のアドレスを見ることができます。

引数<件名>は、メッセージ本文の詳細な内容について、見出しとして簡潔に記述したテキストの値です。

警告：メッセージの件名にアクセント記号が付いた文字（ 、 等）を含んではいけません。

引数<メッセージ>は、メールメッセージの本文を含んでいるテキストの値です。メッセージの最大サイズは、4th Dimension の変数やフィールドの上限である 32K に制限されています。

注意：<件名>に日本語を使用する場合、あらかじめ MIME 拡張メッセージ・ヘッダ（RFC1522）によるエンコードが必要です。"AJP mime Encode" コマンドが使用できます。

<メッセージ>に日本語が含まれる場合、あらかじめ JIS 漢字コード変換が必要です。
"AJP Nkf" コマンドが使用できます。

例

このコマンドを使用した例を以下に挙げます：

```
$ ホスト := "mail.aci.co.jp"
$ 宛先アドレス := "sale@aci.fr"
$ 送信アドレス := "komaba@aci.co.jp"
$ 件名 := " 売上レポート "
$ メッセージ := "1999 年売上レポートを送信します。"
$ エラー := mySMTP_QuickSend ($ ホスト ;$ 送信アドレス ;$ 宛先アドレス ;$ 件名 ;$ メッセージ )
If ($ エラー = 0)
    ALERT (" 送信に成功しました。")
Else
    ALERT (" エラー ; SMTP_QuickSend"+Char(13)+IT_ErrorText ($ エラー ))
End if
```

‘ 日本語の件名、本文を送信するために漢字コード変換が必要なため mySMTP_QuickSend を用意した
If (False)

```
$0:=mySMTP_QuickSend ($ ホスト ;$ 送信アドレス ;$ 宛先アドレス ;$ 件名 ;$ メッセージ )
‘ 日本語メッセージを送るための SMTP_QuickSend メソッド
End if
C_TEXT ($1;$ ホスト ;$2;$ 送信アドレス ;$3;$ 宛先アドレス )
C_TEXT ($4;$ 件名 ;$5;$ メッセージ )
C_INTEGER ($0;$ エラー )
```

```
$ ホスト := $1
$ 送信アドレス := $2
$ 宛先アドレス := $3
$ エラー := AJP MIME Encode ($4)
$ エラー := AJP Nkf ("j";$5;$ メッセージ )
```

```
$ エラー := SMTP_SetPrefs (1;10;70) ‘ 本文タイプに日本語コードを指定
$ エラー := SMTP_QuickSend ($ ホスト ;$ 送信アドレス ;$ 宛先アドレス ;$ 件名 ;$ メッセージ )
```

```
$0:=$ エラー
```

参照

「SMTP_New」コマンド、AJP mime Encode コマンド、AJP Nkf コマンド

SMTP_New(smtp_ID) 整数

引数	タイプ	説明
smtp_ID	倍長整数	新しいメッセージのリファレンス
関数の返す値	整数	エラーコード

説明

SMTP_New コマンドは、SMTP_QuickSend コマンドが使われている場合以外で SMTP メールを構築するコマンドシーケンスの中で最初に呼び出されるコマンドです。SMTP_New コマンドは、メモリ内に新規のメッセージを作成し、メッセージのリファレンスを倍長整数の変数である引数 < smtp_ID > に返します。このコマンドの次に呼び出される SMTP コマンドは、この < smtp_ID > のリファレンスを使って SMTP_Send コマンドを呼び出す前にメッセージのヘッダや本文の情報を作成します。

SMTP_New コマンドの各呼び出しに対して、それに対応する SMTP_Clear コマンドの呼び出しが必要です。メッセージの送信後に SMTP_Clear コマンドを呼び出し、メッセージの内容が占有していたメモリを解放します。

引数 < smtp_ID > は、作成されたメッセージを参照するための倍長整数です。この ID は、その後の全処理において、このメッセージの参照に使用されます。複数の新規メッセージを開くことも可能であり、各々のメッセージに対して、< smtp_ID > が返されます。この < smtp_ID > の値を用いて、その後の処理において対象となるメッセージを特定することができます。

例

「SMTP_Body」コマンドの例を参照してください。

参照

「SMTP_Clear」コマンド、「SMTP_QuickSend」コマンド、「SMTP_Send」コマンド

SMTP_Host

IT Send Mail
バージョン 6.5

SMTP_Host(smtp_ID; ホスト名 {; 削除オプション }) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
ホスト名	文字列	ホスト名または IP アドレス
削除オプション	整数	0 = 追加または置換、1 = 削除
関数の返す値	整数	エラーコード

説明

SMTP コマンドで作成され、送信されたすべてのメールは、特定の SMTP サーバーで管理される必要があります。4D Internet Commands は、メールを各受取人に直接配信することではなく、SMTP コマンドによって指定された SMTP サーバーに配信されます。SMTP サーバーは、アドレス分解のエラーやメッセージ配信のスケジュールを責任を持って行います。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < ホスト名 > は、メッセージ配達を処理する SMTP サーバーのホスト名または IP アドレスです。

引数 < 削除オプション > は、現在のホストの設定を削除するかどうかを指定するオプションの引数です。値として 0 を指定すると、ホストを < ホスト名 > で指定された値に設定します。値として 1 を指定した場合は、< smtp_ID > で識別されたメッセージ用の「ホストの指定」を削除します。これは、オプション引数であり、指定されていない場合のデフォルト値は 0 です。

例

「SMTP_Body」コマンドの例を参照してください。

参照

「SMTP_New」コマンド

SMTP_Send

IT Send Mail
バージョン 6.5

SMTP_Send(smtp_ID) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
関数の返す値	整数	エラーコード

説明

SMTP_Send コマンドは、引数 < smtp_ID > で参照されるメッセージを送信します。しかし、メモリからデータを消去しません。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

例

次の例では、メッセージが作成され、For ループの外でスタティックな要素が定義されます。そして、[顧客] のテーブルの各レコードについて、メッセージがカスタマイズされ、送信されます。

```
<>quote:=Char(34)
$エラー :=SMTP_New ($smtp_id)
$エラー :=SMTP_Host ($smtp_id;"mail.aci.co.jp")
$エラー :=SMTP_From ($smtp_id;"komaba@aci.co.jp")
$エラー :=SMTP_ReplyTo ($smtp_id;"support@aci.co.jp")
$エラー :=AJP_mime_Encode (" 広告スペース "; $Subject)
$エラー :=SMTP_Subject ($smtp_id;$subject)
For ($i;1;Records in selection ([ 顧客 ]))
  If ([ 顧客 ] 販売金額 > 100000)
    $本文 :=<> 徳意様向け文書
  Else
    $本文 :=<> 普通向け文書
  End if
  $本文 :=Replace string ($ 挨拶文;"< 氏名 >";[ 顧客 ] 顧客氏名 )+$本文

  $エラー :=SMTP_To ($smtp_id;[ 顧客 ]email;1) ' "To" のヘッダーを置き換える
  $エラー :=AJP_Nkf ("-" ;$ 本文 ;$body)
  $エラー :=SMTP_Body ($smtp_id;$body) 'JIS 漢字コード変換
  $エラー :=SMTP_AddHeader ($smtp_id;"Content-Type:";"text/plain; charset="+<>quote+"iso-2022-jp"+<>quote)
  $エラー :=SMTP_AddHeader ($smtp_id;"Content-Transfer-Encoding:";"7bit")
  $エラー :=SMTP_Send ($smtp_id)
NEXT RECORD ([ 顧客 ])
End for
$エラー :=SMTP_Clear ($smtp_id)
```

参照

「SMTP_New」コマンド、AJP mime Encode コマンド、AJP Nkf コマンド

SMTP_Clear

IT Send Mail
バージョン 6.5

SMTP_Clear(smtp_ID) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス 成功した場合は 0
関数の返す値	整数	エラーコード

説明

SMTP_Clear コマンドは、メッセージを除去し、メッセージ作成中に使用したメモリを解放します。
SMTP_New コマンドのすべての呼び出しに対して、それに対応する SMTP_Clear コマンドの呼び出しが必要です。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。
SMTP メッセージが正常にクローズできた場合、SMTP_Clear コマンドは < smtp_ID > の変数に 0 を返します。

例

「SMTP_Body」の例を参照してください。

参照

「SMTP_New」コマンド

SMTP_Date(smtp_ID; メッセージ日付; メッセージ時刻; 時間帯; オフセット; {; 削除オプション}) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
メッセージ日付	日付	このメッセージが作成された日付
メッセージ時刻	倍長整数	このメッセージが作成された時刻
時間帯	整数	地域コード
オフセット	整数	引数<タイムゾーン>に依存した値
削除オプション	整数	0 = 追加 / 置換、1 = 削除
関数の返す値	整数	エラーコード

説明

あるメールに対して、作成した日付、時刻および作成者がメールを発信した地域が与えられている場合、SMTP_Date コマンドは、引数< smtp_ID >の値で指定されたメッセージの日付ヘッダを作成します。コマンドに渡された日付は、メッセージを送信するマシンが設置されている場所の現在日付および時刻です。下記の引数は、特定のフォーマットに従う必要があるため、メッセージの最後を受け取るサーバーは、日付と時刻、時間帯および渡されたオフセットに基づいて日付と時刻を判断します。それから、受取側の日付と時刻をローカルな時間に対応するよう変換します。

注意：メールメッセージが「日付」ヘッダなしで構成されている場合、SMTP サーバーは現在の日付および時刻の設定と共にヘッダを追加します。すべての SMTP メールメッセージは、クライアントアプリケーションまたは SMTP サーバーによって追加された日付ヘッダを含んでいます。

引数< smtp_ID >は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数< メッセージ日付 >は、このメッセージが作成された日付を含んだ 4D の日付です。

引数< メッセージ時刻 >は、このメッセージが作成された時刻を含んだ倍長整数です。4D の時間を倍長整数に変換するには、4D の時間に 0 を加算してください [例 : \$ 時刻 :=(Current Time+0)]。

引数< 時間帯 >は、送信側の時間帯を特定します。このフィールドは次の表に基づいた 0 から 6 までの値を指定することができます。

値として 0 を指定した場合は、「世界標準時間」を基準に加減する時間数を引数< オフセット >に直接指定することができます。

値として 1 を指定することにより、Macintosh の PRAM に基づいたオフセットを送信側のマシンが自動的に追加します。< 時間帯 >が 1 である場合、引数< オフセット >は必要ありません。Macintosh コンピュータの時間帯は、コントロールパネルの「**マップ**」(MacOS 8)または「**日付 & 時刻**」(MacOS 8.5以降)の設定によって決定されます。開発者は、この時間帯の値がデータベースにとって重要な要素である場合には、このオプションを正確に設定するような考慮が必要です。

2 から 5 の値は、アメリカの 4 つの時間帯に対応しています。これらの設定値にそれぞれ対応する <オフセット> は、時間帯が夏時間なのか（引数 <オフセット> =1）そうでないのか（引数 <オフセット> =0）を指定します。

値として 6 を指定した場合、軍隊時間であるということを指定します。<オフセット> は、以下の「軍隊時間」の表によって決定されます。受取人の居る場所の軍隊時間のコードに基づいて対応するオフセットの値（-12 から 12 まで）を使用してください。

オフセット - この引数のあたいは、タイムゾーンの引数にあるコードセットとは独立しています。上記の説明をご覧になるか、下の一覧をご覧になり、この引数にセットする正しい値を見つけてください。

コード	時間帯	オフセット
0	UT からの +/- オフセット	オフセットは、+/- 時間
1	UT からの +/- オフセット	オフセット不使用、Mac の PRAM によってオフセット指定
2	EST - EDT	(0 = EST, 1 = EDT)
3	CST - CDT	(0 = CST, 1 = CDT)
4	MST - MDT	(0 = MST, 1 = MDT)
5	PST - PDT	(0 = PST, 1 = PDT)
6	軍隊時間	以下の表を参照

オフセットの設定値 軍隊時間コード

0	Z
-1 から -9	A から I
-10 から -12	K から M
1 から 12	N から Y

略語の説明

UT	世界標準時間 (Universal Time)
EST	東部標準時間 (Eastern Standard Time)
EDT	東部夏時間 (Eastern Daylight Time)
CST	中部標準時間 (Central Standard Time)
CDT	中部夏時間 (Central Daylight Time)
MST	山地標準時間 (Mountain Standard Time)
MDT	山地夏時間 (Mountain Daylight Time)
PST	太平洋標準時間 (Pacific Standard Time)
PDT	太平洋夏時間 (Pacific Daylight Time)

引数 <削除オプション> に 0 を指定すると、与えられた引数と共に日付ヘッダを追加、または設定されている値を置き換えます。値として 1 を指定した場合は、このフィールドについてのすべての既存の定義が消去されます。それ以外の値は、無視されます。<削除オプション> は、オプション引数であり、指定されていない場合のデフォルト値は 0 です。

参照

「SMTP_New」コマンド

SMTP_From (smtp_ID; メッセージ差出人 ;{; 削除オプション }) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
メッセージ差出人	テキスト	メールアドレスまたはアドレスリスト
削除オプション	整数	0 = 既存のリストに追加 1 = 古い値を新しい値に置換 2 = 指定したアドレスを削除

関数の返す値 整数 エラーコード

説明

SMTP_From コマンドは、メッセージの「差出人 (From)」フィールドに表示される人のメールアドレスを含んでいます。このフィールドのアドレスは、メッセージの作成または権限を与える責任を持つ人のアドレスです。通常、「差出人」ヘッダはメッセージを構成し送信したのが誰であることを示すアドレスを 1 つ含んでいます。しかし「差出人」ヘッダ内で別々に識別される人のグループによってメッセージが構成されるという状況もあり得ます。

「差出人」ヘッダは、必須ではありません。「差出人」ヘッダは持っていないが、「送信人 (Sender)」ヘッダ内で指定されたアドレスを持っているというメッセージを配信することが可能です。「差出人」ヘッダにアドレスが指定された場合、「送信人」ヘッダはオプションとなります。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < メッセージ差出人 > は、1 つ以上のメールアドレスで構成される「アドレスリスト」を含むテキストの値です。メッセージの受取人は、「差出人」のヘッダに表示されたすべてのアドレスを見ることができます。

自動返信における注意事項 : 「返信先」ヘッダが < smtp_ID > で識別されたメッセージに対して定義されていない場合、メッセージへの返信はすべて「差出人」ヘッダで指定された各人に向けて返されます。

引数 < 削除オプション > は、< メッセージ差出人 > に挙げられたアドレスの処理方法を示す整数値です。値として 0 を指定した場合、既に割り当てられている値に新しい値を追加します。値として 1 を指定した場合は、既に割り当てられている値を新しい値で置き換えます。< メッセージ差出人 > が空の文字列である場合は、既存の値は削除され、メッセージからヘッダが削除されます。値として 2 を指定した場合は、指定された値を既存の値から削除します。引数 < 削除オプション > はオプションの引数で、値が指定されていない場合は、デフォルト値として 0 が設定されます。

例

この例では、3 人の人物が会社内のすべての人に配布された会社の方針変更という件名のメッセージを作成し、そのメッセージに対する応答が「差出人」ヘッダに表示されている 3 人の人物の各々に向けて送られます。

```
$ 差出人 := "syatyo@aci.co.jp, senmu@aci.co.jp, kansa@aci.co.jp"
$ エラー :=SMTP_From ($smtp_id;$ 差出人 ;0)
$ エラー :=AJP_mime_Encode (" 会社の方針変更 " ; $Subject)
$ エラー :=SMTP_Subject ($smtp_id;$subject)
$ エラー :=SMTP_To ($smtp_id;<> 全ての従業員 ;0)
```

参照

「SMTP_New」コマンド

SMTP_Sender (smtp_ID; メッセージ送信人 ;{; 削除オプション }) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
メッセージ送信人	テキスト	メールアドレス (1つのみ)
削除オプション	整数	0 = 追加 / 置換、1 = 削除
関数の返す値	整数	エラーコード

説明

SMTP_Sender コマンドは、メッセージを送信する人の電子メールアドレスを追加します。これは、送信人がメッセージの実際の作成者でない場合に使用されます。または、作成者の中で実際にメッセージを送信した人が誰であることを示すために使用されます。「送信人 (Sender)」フィールドの内容が「差出人 (From)」フィールドと同じ場合には、このフィールドを指定する必要はありません。

コンピュータのプログラムがメールメッセージの作成者および送信人である場合、「送信人」ヘッダはコンピュータのプログラムによって管理されているアカウントではなく、プログラム処理の責任者である実際の人物のメールアカウントを参照する必要があります。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < メッセージ送信人 > は、メッセージの「送信人」フィールドに表示される 1 つの「メールアドレス」を含むテキストの値です。このヘッダには、1 つのメールアドレスのみが指定されます。

引数 < 削除オプション > は、「送信人」ヘッダを追加するか削除するかを指定する整数値です。値として 0 を指定した場合、既に割り当てられている設定を無視して、「送信人」フィールドを新しい値に設定します。値として 1 を指定した場合は、「送信人」フィールドに設定されているアドレスを削除し、メールの封筒からヘッダを削除します。 < 削除オプション > はオプションの引数で、値が指定されていない場合は、デフォルト値として 0 が設定されます。

例

この例では、3 人の経営者が秘書によって会社内のすべての人に配布された会社の方針変更という件名のメッセージを作成し、そのメッセージに対する応答を「差出人」ヘッダに表示されている 3 人の人物の各々に向けて送られます。

```
$ 差出人 := "syatyo@aci.co.jp, senmu@aci.co.jp, kansa@aci.co.jp"
$ エラー :=SMTP_From ($smtp_id;$ 差出人 ;0)
$ エラー :=SMTP_Sender ($smtp_id;"hisyo@aci.co.jp")
$ エラー :=AJP_mime_Encode (" 会社の方針変更 " ; $Subject)
$ エラー :=SMTP_Subject ($smtp_id;$subject)
$ エラー :=SMTP_To ($smtp_id;<> 全ての従業員 ;0)
```

参照

「SMTP_New」コマンド

SMTP_ReplyTo (smtp_ID; 返信先 {; 削除オプション}) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
返信先	テキスト	メールアドレスまたはアドレスリスト
削除オプション	整数	0 = 既存のリストに追加 1 = 古い値を新しい値に置換 2 = 指定したアドレスを削除
関数の返す値	整数	エラーコード

説明

SMTP_ReplyTo コマンドを用いることにより、メッセージの返信先を管理することができます。通常、メッセージの返信先は「差出人 (From)」です。送信するメッセージの「返信先 (ReplyTo)」ヘッダを設定することにより、メッセージ応答のデフォルトルートに影響を与えます。

データベース開発者にとって、SMTP_ReplyTo コマンドは自動化されたメールへの応答処理の管理を可能にする強力なツールです。ユーザは、応答を記録するために作成された個別のアカウントである「差出人」や「送信人 (Sender)」のアドレスに表示されている人以外のアドレスに返信 (リプライ) を送りたいかもしれません。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < 返信先 > は、1 つ以上のメールアドレスで構成される「アドレスリスト」を含むテキストの値です。このフィールドのアドレスリストは、メッセージ受取人のメールソフトウェアによって、返信を管理するためのメールアカウントのデフォルト値として使用されます。

引数 < 削除オプション > は、引数 < 返信先 > に挙げられたアドレスの処理方法を示す整数値です。値として 0 を指定した場合、ヘッダに既に割り当てられている値に新しい値を追加します。値として 1 を指定した場合は、既に割り当てられている値を新しい値で置き換えます。< 返信先 > が空の文字列である場合は、既存の値は削除され、メッセージからヘッダが削除されます。値として 2 を指定した場合は、指定された値を既存の値から削除します。< 削除オプション > はオプションの引数で、値が指定されていない場合は、デフォルト値として 0 が設定されます。

例：

この例では、3人の経営者が秘書によって会社内の全ての人に配付された「会社の方針変更」という件名のメッセージを作成します。メッセージへの応答は、秘書（hisyo@aci.co.jp）および人事（jinji@aci.co.jp）に向けて送られるように SMTP_ReplyTo により返信先を指定しています。これにより差出人である3人の経営者には直接返信メールが届かないようになります。

```
$ 差出人 := "syatyo@aci.co.jp, senmu@aci.co.jp, kansa@aci.co.jp"
$ エラー := SMTP_From ($smtp_id; $ 差出人 ;0)
$ エラー := SMTP_Sender ($smtp_id; "hisyo@aci.co.jp")
$ エラー := SMTP_ReplyTo ($smtp_id; "hisyo@aci.co.jp, jinji@aci.co.jp")
$ エラー := AJP_mime_Encode (" 会社の方針変更 " ; $Subject)
$ エラー := SMTP_Subject ($smtp_id; $subject)
$ エラー := SMTP_To ($smtp_id; <> 全ての従業員 ;0)
```

参照

「SMTP_New」コマンド

SMTP_To

SMTP_To (smtp_ID; メッセージ送信先 {; 削除オプション }) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
メッセージ送信先	テキスト	メールアドレスまたはアドレスリスト
削除オプション	整数	0 = 既存のリストに追加 1 = 古い値を新しい値に置換 2 = 指定したアドレスを削除
関数の返す値	整数	エラーコード

説明

SMTP_To コマンドは、メッセージの最初の受取人の指定を追加します。フィールド「To:」にアドレスを指定することは必須ではありませんが、「To:」または「Cc:」、「Bcc:」のいずれかに少なくとも 1 つのアドレスを指定する必要があります。メッセージの受取人は、メールメッセージ内の「To:」ヘッダおよび「Cc:」ヘッダに挙げられたすべてのアドレスを見ることができます。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < メッセージ送信先 > は、1 つ以上のメールアドレスで構成される「アドレスリスト」を含むテキストの値です。

引数 < 削除オプション > は、引数 < メッセージ送信先 > に挙げられたアドレスの処理方法を示す整数値です。値として 0 を指定した場合、ヘッダに既に割り当てられている値に新しい値を追加します。値として 1 を指定した場合は、既に割り当てられている値を新しい値で置き換えます。 < メッセージ送信先 > が空の文字列である場合は、既存の値は削除され、メッセージからヘッダが削除されます。値として 2 を指定した場合は、指定された値を既存の値から削除します。 < 削除オプション > はオプションの引数で、値が指定されていない場合は、デフォルト値として 0 が設定されます。

例

「SMTP_Body」コマンドの例を参照してください。

参照

「SMTP_New」コマンド

SMTP_Cc (smtp_ID; カーボンコピー {; 削除オプション}) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
カーボンコピー	テキスト	メールアドレスまたはアドレスリスト
削除オプション	整数	0 = 既存のリストに追加 1 = 古い値を新しい値に置換 2 = 指定したアドレスを削除
関数の返す値	整数	エラーコード

説明

SMTP_Cc コマンドは、引数 < smtp_ID > で指定されたメッセージのカーボンコピーの受取人の指定を追加します。フィールド「Cc:」にアドレスを指定することは必須ではありませんが、「To:」または「Cc:」、「Bcc:」のいずれかに少なくとも 1 つのアドレスを指定する必要があります。メッセージの受取人は、メールメッセージ内の「To:」ヘッダおよび「Cc:」ヘッダに挙げられたすべてのアドレスを見ることができます。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < カーボンコピー > は、1 つ以上のメールアドレスで構成される「アドレスリスト」を含むテキストの値です。

引数 < 削除オプション > は、引数 < カーボンコピー > に挙げられたアドレスの処理方法を示す整数値です。値として 0 を指定した場合、フィールドに既に割り当てられている値に新しい値を追加します。値として 1 を指定した場合は、既に割り当てられている値を新しい値で置き換えます。 < カーボンコピー > が空の文字列である場合は、既存の値は削除され、メッセージからヘッダが削除されます。値として 2 を指定した場合は、指定された値を既存の値から削除します。 < 削除オプション > はオプションの引数で、値が指定されていない場合は、デフォルト値として 0 が設定されます。

例

「SMTP_Body」コマンドの例を参照してください。

参照

「SMTP_Bcc」コマンド、「SMTP_New」コマンド

SMTP_Bcc (smtp_ID; ブラインドカーボン {; 削除オプション}) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
ブラインドカーボン	テキスト	アドレスリスト
削除オプション	整数	0 = 既存のリストに追加 1 = 古い値を新しい値に置換 2 = 指定したアドレスを削除
関数の返す値	整数	エラーコード

説明

SMTP_Bcc コマンドは、引数 < smtp_ID > で指定されたメッセージにブラインドカーボンコピーの受取人の指定を追加します。フィールド「Bcc:」にアドレスを指定することは必須ではありませんが、「To:」、「Cc:」または「Bcc:」のいずれかに少なくとも 1 つのアドレスを指定する必要があります。

人々のまとめりであるグループに宛ててメールを送信する場合、「アドレスリスト」情報の秘密保持を実現する唯一の方法は、「Bcc」ヘッダにアドレスを挙げることです。「Bcc」ヘッダに挙げられたアドレスは、メッセージのヘッダや本文の一部として送信されることはありません。このアドレスは、「To」ヘッダや「Cc」ヘッダ、「Bcc」ヘッダで指定された受取人は見ることはできません。

「Bcc」の受取人は、「To」や「Cc」の受取人をすべて見ることはできますが、他の「Bcc」の受取人を見ることはできません。多くの受取人のグループにメールを送信する場合、すべての受取人を「Bcc」のヘッダに置くことがあります。これは、メッセージの乱雑で大きなアドレスリストを所持することや、他のアドレスからのアクセスからユーザを保護します。

「Bcc」を使用するもう 1 つの理由は、多くのメールアプリケーションが「全員に返信」の機能を持っていて、この機能はメッセージの返信に「To」および「Cc」に指定されているすべての受取人を追加します。「Bcc」ヘッダにすべての受取人を指定しておくことにより、ユーザは元のメッセージを受け取るすべての人へ返信しなくて済みます。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメッセージを参照するための倍長整数です。

引数 < ブラインドカーボン > は、1 つ以上のメールアドレスで構成される「アドレスリスト」を含むテキストの値です。

引数 < 削除オプション > は、引数 < ブラインドカーボン > に挙げられたアドレスの処理方法を示す整数値です。値として 0 を指定した場合、フィールドに既に割り当てられている値に新しい値を追加します。値として 1 を指定した場合は、既に割り当てられている値を新しい値で置き換えます。< ブラインドカーボン > が空の文字列である場合は、既存の値は削除され、メッセージからヘッダが削除されます。値として 2 を指定した場合は、指定された値を既存の値から削除します。< 削除オプション > はオプションの引数で、値が指定されていない場合は、デフォルト値として 0 が設定されます。

例

次の例では、メッセージが作成され、For ループの外でスタティックな要素が定義されます。そして [皆さん] のテーブルの各レコードについて、ブラインドカーボンコピーのリストにアドレスが追加されます。

```
$ エラー :=SMTP_From ($smtp_id;"sales@aci.co.jp")
$ エラー :=AJP mime Encode (" 特別セール！今週限り！ " ; $Subject)
$ エラー :=SMTP_Subject ($smtp_id;$subject)
$ エラー :=AJP Nkf ("j";$ 本文 ;$body)
$ エラー :=SMTP_Body ($smtp_id;$body) 'JIS 漢字コード変換
For ($i;1;Records in selection ([ 顧客 ]))
  $ エラー :=SMTP_Bcc ($smtp_id;[ 顧客 ]email;0) ' メールアドレス追加
  NEXT RECORD ([ 顧客 ])
End for
$ エラー :=SMTP_Send ($smtp_id) ' 全員にメールを送信
$ エラー :=SMTP_Clear ($smtp_id)
```

参照

「SMTP_Cc」コマンド、「SMTP_New」コマンド

SMTP_InReplyTo (smtp_ID; 返信先テキスト {; 削除オプション }) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
返信先テキスト	テキスト	返信先へ付加するテキスト
削除オプション	整数	0 = 追加 / 置換、1 = 削除
関数の返す値	整数	エラーコード

説明

SMTP_InReplyTo コマンドは、返信メッセージの前の通信を特定します。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < 返信先テキスト > は、メッセージが属している前回の通信を参照するテキストの値です。特別なフォーマットが必要な場合は、RFC#822 を参照してください。

警告：テキストには、ラインフィード（アスキーコード =10）を含んではいけません。ラインフィードは、ヘッダ部分の終わりであり本文の初めの部分であるという印です。その後ろに続くヘッダのアイテムは、本文に含められますが、サーバーまたはクライアントのソフトウェアでは正確に認識されません。ヘッダに関する詳細は、RFC#822 を参照してください。

引数 < 削除オプション > は、引数 < 返信先テキスト > を削除するかどうかを指定するオプションの整数値です。値として 0 を指定した場合、< 返信先テキスト > フィールドにテキストを追加します。値として 1 を指定した場合は、< 返信先テキスト > フィールドからテキストを削除し、< 返信先テキスト > のすべてのテキストを無視します。

参照

「SMTP_New」コマンド

SMTP_References (smtp_ID; リファレンス {; 削除オプション}) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
リファレンス	テキスト	リファレンステキスト
削除オプション	整数	0 = 追加 / 置換、1 = 削除
関数の結果	整数	エラーコード

説明

SMTP_References コマンドは、メッセージが参照する追加の通信を特定します。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < リファレンス > は、リファレンステキストを含むテキストの値です。特別なフォーマットが必要な場合は、RFC#822 を参照してください。

警告：テキストには、ラインフィード（アスキーコード=10）を含んではいけません。ラインフィードは、ヘッダ部分の終わりであり本文の初めの部分であるという印です。その後ろに続くヘッダのアイテムは、本文に含められますが、サーバまたはクライアントのソフトウェアでは正確に認識されません。ヘッダに関する詳細は、RFC#822 を参照してください。

引数 < 削除オプション > は、引数 < リファレンス > のヘッダを削除するかどうかを指定するオプションの整数値です。値として 0 を指定した場合、< リファレンス > フィールドにテキストを追加します。< リファレンス > の前のテキストは、すべて置換されます。値として 1 を指定した場合は、< リファレンス > に割り当てられているテキストに関係なく、< リファレンス > フィールドからすべてのテキストを削除します。

参照

「SMTP_New」コマンド

SMTP_Comments (smtp_ID; コメント {; 削除オプション }) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
コメント	テキスト	コメントテキスト
削除オプション	整数	0 = 追加 / 置換、1 = 削除
関数の結果	整数	エラーコード

説明

SMTP_Comments コマンドを使用することにより、メッセージ本文に手を加えることなくメッセージにテキストのコメントを追加することができます。コメントは、メッセージのヘッダ部分にのみ表示されます。メールリーダーの多くは、メッセージヘッダーの全テキストをユーザに表示することはありません。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < コメント > は、メールヘッダに代入したい情報を含むテキスト値です。

警告： テキストには、ラインフィード（アスキーコード =10）を含んではいけません。ラインフィードは、ヘッダ部分の終わりであり本文の初めの部分であるという印です。その後ろに続くヘッダのアイテムは、本文に含められますが、サーバまたはクライアントのソフトウェアでは正確に認識されません。ヘッダに関する詳細は、RFC#822 を参照してください。

引数 < 削除オプション > は、引数 < コメント > のテキストを設定するか削除するかを指定します。値として 0 を指定した場合、< コメント > のヘッダを定義し、フィールドに既に設定されている値をすべて上書きします。値として 1 を指定した場合は、< コメント > ヘッダを削除し、< コメント > のすべての値を無視します。引数 < 削除オプション > はオプションの引数で、値が指定されていない場合は、デフォルト値として 0 が設定されます。

例

「SMTP_Body」コマンドの例を参照してください。

参照

「SMTP_New」コマンド

SMTP_Keywords (smtp_ID; コメント {; 削除オプション}) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
キーワード	テキスト	キーワードリスト
削除オプション	整数	0 = 追加、1 = 削除
関数の結果	整数	エラーコード

説明

このフィールドは、カンマによって区切られたキーワードまたは語句を含んでいます。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < キーワード > は、キーワードまたはキーワードのリストを含むテキスト値です。特別なフォーマットが必要な場合は、RFC#822 を参照してください。

警告：テキストには、ラインフィード（アスキーコード=10）を含んではいけません。ラインフィードは、ヘッダ部分の終わりであり本文の初めの部分であるという印です。その後ろに続くヘッダのアイテムは、本文に含められますが、サーバまたはクライアントのソフトウェアでは正確に認識されません。ヘッダに関する詳細は、RFC#822 を参照してください。

引数 < 削除オプション > は、現在のキーワードリストを削除するかどうかを指定するオプションの整数値です。値として 0 を指定した場合、既に追加されているテキストに、引数 < キーワード > で指定されたテキストを追加します。値として 1 を指定した場合は、現在の < キーワード > の値を削除し、 < キーワード > の値で置き換えます。この場合、 < キーワード > の値が空の文字列であれば、 < キーワード > のヘッダを完全に削除します。

参照

「SMTP_New」コマンド

SMTP_Encrypted (smtp_ID; 暗号化 {; 削除オプション }) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
暗号化	テキスト	暗号化の方法
削除オプション	整数	0 = 追加 / 置換、1 = 削除

説明

SMTP_Encrypted コマンドによって、メッセージの本文で使用されている暗号化のタイプをユーザが知ることができます。4D Internet Commands では、メールメッセージの暗号化や解読はできません。メッセージ本文の暗号化は、開発者に任されています。(SMTP_Body コマンドによる入力の前に) メッセージ本文を暗号化する方法が取られている場合、このコマンドは使用された暗号化の方法を特定するために必要になります。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < 暗号化 > は、メッセージ本文を暗号化するために使用された暗号化の方法のタイプを指定するためのテキスト値です。暗号化されたヘッダは、受取人であるメールのソフトウェアによって使用され、メッセージ本文を解読するために必要な方法を決定します。特別なフォーマットが必要な場合は、RFC#822 を参照してください。

警告：テキストには、ラインフィード (アスキーコード =10) を含んではいけません。ラインフィードは、ヘッダ部分の終わりであり本文の初めの部分であるという印です。その後ろに続くヘッダのアイテムは、本文に含まれますが、サーバまたはクライアントのソフトウェアでは正確に認識されません。ヘッダに関する詳細は、RFC#822 を参照してください。

引数 < 削除オプション > は、暗号化の方法のテキストを含むヘッダをを削除するかどうかを指定するオプションの整数値です。値として 0 を指定した場合、対応したテキストにヘッダを追加します。引数 < 記号化 > 内の前の値は、新しい値で置き換えられます。値として 1 を指定した場合は、メッセージから「暗号化」のヘッダを削除します。

参照

「SMTP_Body」コマンド、「SMTP_New」コマンド

SMTP_AddHeader (smtp_ID; ヘッダ名 ; ヘッダテキスト {; 削除オプション }) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
ヘッダ名	文字列	ヘッダの名称
ヘッダテキスト	テキスト	ヘッダのテキスト
削除オプション	整数	0 = 追加 1 = すべてのヘッダを引数<ヘッダ名>に置換 2 = 引数<ヘッダ名>で指定したすべてのヘッダを削除
関数の返す値	整数	エラーコード

説明

SMTP_AddHeader コマンドを使用することによって、引数< smtp_ID >で指定されたメッセージにユーザ自身のヘッダを追加することができます。4D Internet Commands では、様々なヘッダを作成するためのコマンドを提供していますが、さらに「ユーザ定義の」ヘッダおよび「拡張」ヘッダというヘッダに関する2つのカテゴリーが追加されています。SMTP_AddHeader コマンドを利用することによって、新規のヘッダのタグおよびデータの両方を追加し、それらを関連付けることができます。

拡張ヘッダ：この拡張ヘッダは、正式に NIC によって認識され、元の SMTP 指定後に定義されます。この広範囲のヘッダは、様々なソフトウェアのアプリケーションの動作に影響を与えるための特有の機能を持っています。拡張ヘッダは、文字「X」から始まってはいけません。

ユーザ定義ヘッダ：SMTP プロトコルによって、ヘッダの定義を作成することができます。想定される拡張ヘッダとの衝突を避けるために、すべてのユーザ定義のヘッダは、文字「X」から始まっている必要があります。ユーザ定義のヘッダは、通信の終わりまでに渡ってデザインを管理する場合に、非常に役に立ちます。

ユーザ定義のヘッダを使用すれば、開発者は POP3 の外部コマンドである POP3_FindHeader コマンドを使って簡単に取り出し可能なデータを格納することができます。例えば、ファイル 01 のフィールド 01 に値を含む、「X-001001」と名付けられたヘッダを作成することができます。無数のヘッダをメッセージに追加することができます。ユーザ定義のヘッダによって、メッセージ本文の中身を解析することなく簡単に引き出すことのできる情報を追加することで、ユーザは適切な情報を見つけることができます。

引数< smtp_ID >は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数<ヘッダ名>は、追加されるヘッダの名前を含む文字列です。

引数<ヘッダテキスト>は、引数<ヘッダ名>によって指定されたフィールドに割り当てる情報を含んだテキストの値です。

警告：テキストには、ラインフィード（アスキーコード =10）を含んではいけません。ラインフィードは、ヘッダ部分の終わりであり本文の初めの部分であるという印です。その後ろに続くヘッダのアイテムは、本文に含められますが、サーバまたはクライアントのソフトウェアでは正確に認識されません。ヘッダに関する詳細は、RFC#822 を参照してください。

引数<削除オプション>は、現在のヘッダを削除するかどうかを指定するオプションの整数値です。値として 0 を指定した場合、メッセージに引数<ヘッダ名>を追加します。値として 1 を指定した場合は、すべてのヘッダを<ヘッダ名>に置き換えます。この場合、<ヘッダ名>が空の文字列であれば、すべてのヘッダを削除します。値として 2 を指定した場合は、<ヘッダ名>という名前のすべてのヘッダを削除します。

例

「SMTP_Body」コマンドの例を参照してください。

参照

POP3_FindHeader コマンド、「SMTP_New」コマンド

SMTP_Subject

IT Send Mail
バージョン 6.5

SMTP_Subject (smtp_ID; 件名 {; 削除オプション}) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
件名	テキスト	メッセージの件名
削除オプション	整数	0 = 追加 / 置換、1 = 削除
関数の結果	整数	エラーコード

説明

SMTP_Subject コマンドは、引数 < smtp_ID > で指定されたメッセージに、メッセージの件名を追加します。既に SMTP_Subject コマンドによって件名が追加されている場合、新しい件名は前の件名を無視します。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < 件名 > は、メッセージの本文によって詳細に渡って取り扱われているトピックを簡潔に記述したテキストの値です。

警告：メッセージの件名は、(、 等のような) アクセント記号の付いた文字を含んではいけません。

警告：< 件名 > には、ラインフィード (アスキーコード=10) を含んではいけません。ラインフィードは、ヘッダ部分の終わりであり本文の初めの部分であるという印です。その後ろに続くヘッダのアイテムは、本文に含まれますが、サーバまたはクライアントのソフトウェアでは正確に認識されません。ヘッダに関する詳細は、RFC#822 を参照してください。< 件名 > に日本語を使用する場合、AJP mime Encode によるエンコードが必要です。

引数 < 削除オプション > は、引数 < smtp_ID > によって指定されたメッセージの件名を削除するかどうかを指定するオプションの整数値です。値として 0 を指定した場合、与えられたテキスト文字列に件名を設定します。値として 1 を指定した場合は、メッセージから「件名」のヘッダを削除し、空白のままにしておきます (引数 < 件名 > に渡されたものすべてを無視します)。

例

「SMTP_Body」コマンドの例を参照してください。

参照

「SMTP_New」コマンド、AJP mime Encode コマンド

SMTP_Body (smtp_ID; メッセージ本文 {; 削除オプション}) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
メッセージ本文	テキスト	メッセージの本文
削除オプション	整数	0 = 追加 / 置換、1 = 削除、2 = 付加
関数の返す値	整数	エラーコード

説明

SMTP_Body コマンドは、引数 < smtp_ID > で指定されたメールメッセージの本文に、引数 < メッセージ本文 > のテキストを割り当てます。引数 < メッセージ本文 > は、テキストの主要ブロックです。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 < メッセージ本文 > は、メッセージの本文を含んでいるテキストの値です。引数 < メッセージ本文 > の容量は、4D のテキストオブジェクトの制限である 32K に制限されています。これは、メールメッセージ自体が 32K に制限されているという意味ではありません。本文が 32K よりも大きい容量を持つようなメールを送信するためには、引数 < 削除オプション > (下記を参照) で「付加」のフラグを使用する必要があります。メールメッセージの本文の実際の容量の制限は、利用できるメモリに依存します。

引数 < 削除オプション > は、引数 < smtp_ID > によって指定されたメッセージの本文を削除するかどうかを指定するオプションの整数値です。値として 0 を指定した場合、与えられたテキスト文字列に本文を設定します。値として 1 を指定した場合は、メッセージから本文を削除し、空白のままにしておきます (引数 < メッセージ本文 > に渡されたものすべてを無視します)。値として 2 を指定した場合は、既に SMTP_Body コマンドを呼び出すことによって送信されているテキストに、< メッセージ本文 > のテキストを付加します。

例

次は SMTP に関する完全な使用例です：

```
C_LONGINT ($smtp_id)
C_BOOLEAN ($送信 OK;$OK)
$送信 OK:=False '全てのコマンドを通して作成されたことを示すためのフラグ
$エラー :=AJP mime Encode (v 件名 $Subect)
$エラー :=AJP mime Encode ("4D を介して送信 " ; $comments)
$エラー :=AJP Nkf ("-j";$本文 ;$body)
```

Case of

```
¥(Not (ERRCHECK ("SMTP_New";SMTP_New ($smtp_id))))
¥(Not (ERRCHECK ("SMTP_Host";SMTP_Host ($smtp_id;<> サーバ番号 ))))
¥(Not (ERRCHECK ("SMTP_From";SMTP_From ($smtp_id;v 差出人 ))))
¥(Not (ERRCHECK ("SMTP_To";SMTP_To ($smtp_id;v 宛先 ))))
¥(Not (ERRCHECK ("SMTP_Cc";SMTP_Cc ($smtp_id;vCC))))
¥(Not (ERRCHECK ("SMTP_Bcc";SMTP_Bcc ($smtp_id;vBcc))))
¥(Not (ERRCHECK ("SMTP_Subject";SMTP_Subject ($smtp_id;$subject))))
¥(Not (ERRCHECK ("SMTP_Comments";SMTP_Comments ($smtp_id;$comments))))
¥(Not (ERRCHECK ("SMTP_AddHeader";SMTP_AddHeader ($smtp_id;"x-ACI demo";<>version))))
¥(Not (ERRCHECK ("SMTP_Body";SMTP_Body ($smtp_id;$body))))
¥(Not (ERRCHECK ("SMTP_Send";SMTP_Send ($smtp_id))))
```

Else

\$ 送信 OK:=True ‘メッセージが構築され、メール送信に成功

End case

If (\$smtp_id#0) ‘メッセージ ID が構築された場合、それをここで削除する

\$OK:=ERRCHECK ("SMTP_Clear";SMTP_Clear (\$smtp_id))

End if

以下に挙げるのは、「ERRCHECK」メソッドのコードです。このメソッドは、2つの引数を受け取ります。1つはコマンドの名称（\$ コマンド） もう1つは（メソッドの引数内のコマンドを実行することによって渡される）エラーの値です。「ERRCHECK」メソッドは、エラーの値が0であるかどうかに対応するブール値を返します。エラーの値が0でない場合は返り値（\$0）には False（偽）が返りますが、0である場合は True（真）が返ります。

```
C_TEXT($1;$ コマンド ;v エラーメッセージ ;$s;$cr)
C_LONGINT($2;$ エラー )
$ コマンド :=$1
$ エラー :=$2
$ 結果 :=True
$cr:=Char(13)
If ($ エラー #0)
$ 結果 :=False
If (<> エラー表示 ) ‘エラー表示するかどうかを決定するブール値
v エラーメッセージ :=IT_ErrorText ($ エラー )
$s:=" エラー ..."+$cr
$s:=$s+" コマンド : "+$ コマンド+$cr
$s:=$s+" エラーコード : "+String($ エラー )+$cr
$s:=$s+" 説明 : "+v エラーメッセージ
ALERT($s)
End if
End if
$0:=$ 結果
```

参照

「SMTP_New」コマンド、AJP Nkf コマンド

SMTP_Attachment (smtp_ID; ファイル名 ; エンコードタイプ {; 削除オプション }) 整数

引数	タイプ	説明
smtp_ID	倍長整数	メッセージのリファレンス
ファイル名	テキスト	添付するファイルの名前
エンコードタイプ	整数	0 = エンコードなし (データフォークのみを送信) ± 1 = BinHex ± 2 = Base64 (データフォークのみを送信) ± 3 = AppleSingle ± 4 = AppleDouble ± 5 = AppleSingle および Base64 ± 6 = AppleDouble および Base64 ± 7 = UUEncode ± 8 = Mac バイナリー
削除オプション	整数	0 = 既存のリストに追加 1 = 指定したすべての添付ファイルを <ファイル名> に置換 2 = この添付ファイルのみを削除
関数の返す値	整数	エラーコード

説明

SMTP_Attachment コマンドは、MIME フォーマットのメッセージにテキストファイルまたはバイナリーファイルを添付する方法を提供しています。このコマンドは、1 つのメールメッセージに複数のドキュメントを添付するために何度も呼び出されます。0 よりも大きい値が引数 <エンコードタイプ> に渡された場合、SMTP_Attachment コマンドはメッセージが送信されると同時にエンコードを実行します。

引数 < smtp_ID > は、SMTP_New コマンドで作成されたメールメッセージを参照するための倍長整数です。

引数 <ファイル名> は、メッセージに添付したいファイルの名前を含みます。この値は、以下の 3 通りの方法によって指定されます：

- ""= 標準の「ファイルを開く」ダイアログを表示する。
- "ファイル名"= データベースのストラクチャと同じディレクトリ内の「ファイル名」というファイルを探す。
- "パス:ファイル名"= 「ファイル名」を含んだ完全なパスで指定する。

引数 <エンコードタイプ> は、メッセージに組み込まれる前にファイル上でどのタイプのエンコードが行われたのかを示す整数値です。バイナリーファイルを添付する場合、エンコードの方法は適切な変換 (BinHex、AppleSingle) を行う必要があります。最も一般的なエンコードの方法は、BinHex です。

引数<エンコードタイプ>に正の値を渡した場合、コマンドはメッセージが送信された時に指定された方法を使って、自動的にファイルをエンコードします。SMTP_Send コマンドが実行されると同時にファイルのエンコードが生じます。ファイルが大きい場合、SMTP_Send コマンドを完了するまでに時間がかかります。同じファイルが何回も送信される場合、かなりの時間を必要とします。このような場合、最善の方法は、IT_Encode コマンドで1度にファイルをエンコードし、<エンコードタイプ>に負の値を使ってメッセージにエンコードしたファイルを添付することです。<エンコードタイプ>に負の値を指定した場合、2回目の新たなエンコードを実行しませんが、メッセージヘッダを添付したファイルのエンコードは正しく設定されます。

引数<削除オプション>は、添付ファイルを処理する方法を指定するオプションの整数値の引数です。値として0を指定した場合、添付ファイルの現在のリストに添付ファイルを追加します。値として1を指定した場合は、すべての添付ファイルを引数<ファイル名>のファイルに置き換えます。<ファイル名>が空の文字列である場合は、すべての添付ファイルを削除します。値として2を指定した場合は、添付ファイルのリストから、<ファイル名>で指定された添付ファイルのみを削除します。

参照

「IT_Encode」コマンド、「SMTP_New」コマンド、「SMTP_Send」コマンド

3

IT Review Mail

一連の POP3 コマンドは、使用しているデータベースの中に POP3 メールサーバーからのメッセージ取り出しを可能にします。4D Internet Commands は MIME に対応しており、複数の添付ファイルを含んでいるメッセージを識別したり取り出すことができます。

POP3 に関連するコマンドは、この第 3 章の “ POP3-IT_Review Mail ” と次の第 4 章の “ POP3-IT Downloaded Mail ” の 2 つの章に分かれています。このコマンドの分け方は、メールを取り込む際の処理方法の違いによるものです。POP3 サーバーからメールを取り込む場合、メッセージ（またはそのメッセージの情報）は 4th Dimension のストラクチャ（変数、フィールド、配列）の中に格納されるか、あるいはディスクにダウンロードされます。この章の “ POP3-IT_Review Mail ” の中では、POP3 サーバーから 4th Dimension の中にメッセージを取り込むための 4D Internet Commands の機能について説明します。

上記の 2 つのメッセージ取り込み方法に必要なものは、数メガバイトの情報をダウンロードできる能力を持つ操作におけるメモリ制約によって生まれます。例えば、5MB の添付ファイルを持っている 1 つのメールメッセージはそのデータベース内に保存できる能力を簡単にオーバーフローさせてしまいます。このサイズを格納できる 4D のストラクチャはピクチャフィールドか BLOB フィールドのどちらかしかありません。しかし、このフォーマットにメッセージや添付ファイルを変換することは、そのピクチャフィールドや BLOB フィールドにアクセスしようとしているクライアント上に膨大なメモリが必要となるためほとんど無意味です。この問題を解決するには、この章にある POP3_Download コマンドを使用します。このコマンドは POP3 サーバーからユーザのローカルディスクにメッセージをダウンロードします。ディスクに任意のファイルをダウンロードしたら、次の第 4 章のコマンドを使って、そのダウンロードしてきたファイルを操作することができます。

一連の POP3 コマンドを使用する場合、パラメータ（引数）として頻繁に使用される <メッセージ番号> と <ユニーク ID> を理解することが重要です。引数 <メッセージ番号> は、POP3_Login コマンドが実行された時点のメールボックス内にあるメッセージの番号です。ログインすると、メールボックス内のメッセージには 1 からそのメールボックス内に入っている項目数までの番号が割り当てられます。割り当てられた番号は、メッセージがメールボックス内に受信された順番を基準としています。つまり、1 番古いメッセージが 1 となります。メッセージに割り当てられた番号は、POP3_Login コマンドでログインした時から POP3_Logout コマンドでログアウトした時までの間だけ有効です。

POP3_Logout コマンドが実行されると同時に、削除用に目印が付けられたメッセージは消去されます。ユーザがサーバーの中にログを戻すと、メールボックス内に現在存在しているメッセージは、ふたたび 1 番から番号が振り直されます。例えば、メールボックスに 10 個のメッセージが入ってて、1 から 5 までの番号のメッセージが削除された場合、6 から 10 までの番号のメッセージに対してユーザが次にそのメールボックスにログインした際に 1 から 5 までの番号が振り直されます。

前ページの例を説明するために、ある POP3 サーバーにあなたがログインし、次のようなメッセージリストを持っていると仮定します。

番号	ユニーク ID	日付	差出人	件名
1	bd573a4dbd573a4d	1 Jul 1999...	danw@acme.com	売上について
2	bd574dc7bd574dc7	1 Jul 1999...	frank@acme.com	サイトライセンス注文
3	bd575f06bd575f06	3 Jul 1999...	joe@acme.com	ランチは誰と一緒に？
4	bd5761d4bd5761d4	4 Jul 1999...	kelly@acme.com	奥さんに電話してください
5	bd577dc7db577dc5	5 Jul 1999...	track@fedex.com	宅配履歴

あるセッション中に、あなたはメッセージ番号 3 と 4 のメッセージを削除したとします。このセッションをログアウトすると、あなたが行った削除処理はコミット（許可）されます。ここで、POP3 サーバーの中にログを戻すと、あなたのメッセージリストは次のように変更されているはずです：

番号	ユニーク ID	日付	差出人	件名
1	bd573a4dbd573a4d	1 Jul 1999...	danw@acme.com	売上について
2	bd574dc7bd574dc7	1 Jul 1999...	frank@acme.com	サイトライセンス注文
3	bd577dc7db577dc5	5 Jul 1998...	track@fedex.com	宅配履歴

引数<メッセージ番号>は、特定のメッセージに関係する固定の値ではなく、あるセッションからセッションがオープンした時点のメールボックス内の別メッセージにその関係が依存するセッションへと変化します。これに対して、引数<ユニーク ID>はサーバーから受け取った際のメッセージに割り当てられた一意の番号です。この番号は、メッセージを受信した日時で算出され、あなたが使用している POP3 サーバーから割り当てられた値です。ただし、POP3 サーバーはメッセージを参照する手段としてこの<ユニーク ID>を使用しません。そのため、POP3 コマンドを通して、POP3 サーバー上にあるメッセージの参照手段として<メッセージ番号>を指定したくなることでしょう。開発を行っている人は、データベースの中にメッセージの参照のみを取り込み、そのメッセージの本文はサーバー上に残しておきたいようなソリューションを開発してる場合には、いくつか注意する必要があるかもしれません。

注意：日本語メールの扱い

受信したメールに日本語が含まれる場合、ほとんどの場合、JIS 漢字コードにエンコードされています。POP3 Get Message、POP3 Download、MSG GetHeaders で Subject を取り出したメッセージはエンコードされた生メッセージであり、そのままでは日本語表示できません。AJP mime Encode および AJP Nkf を使用することで表示可能な日本語に変換することができます。

 POP3_SetPrefs (改行文字 ; メッセージフォルダ ; 添付フォルダ) 整数

引数	タイプ	説明
ラインフィード除去	整数	0 = ラインフィードを取り除かない 1 = ラインフィードを取り除く -1 = 変更なし
メッセージフォルダ	テキスト	メッセージフォルダのパス ("" の場合は変更なし)
添付フォルダ	テキスト	添付フォルダのパス ("" の場合は変更なし)
関数の返す値	整数	エラーコード

説明

POP3_SetPrefs コマンドは、すべての POP3 コマンドの環境設定をセットします。

引数<ラインフィード除去>は、ラインフィードが保存したメッセージの中でどのように処理されるかを示す整数値です。ほとんどの POP3 サーバーは、行の最終を認識するためにキャリッジリターンコードとラインフィードコードを組み合わせて使用しています。Macintosh のアプリケーションでは、行の最終区切り文字にキャリッジリターンのみを好んで使用しています。この引数には、ユーザにメッセージテキストからラインフィードを取り除くかどうかのオプションがあります。<ラインフィード除去>が 0 の場合は、POP3 サーバー上に格納されたままのフォーマットでメッセージを取り出します。1 の場合は、抽出メッセージからラインフィードが取り除かれます。-1 の場合は、前回セットした際の設定がそのまま使用されます。デフォルトの値は 1 で、メッセージ内で見つかったラインフィードは取り除かれます。

注意：

取り出したメッセージを 4D 内部で取り扱う場合、Mac/Windows に関係なく<ラインフィード除去>に 1 を指定してください。これは、4D はプラットフォームに関係なく、内部的な改行コードにキャリッジリターンコードのみを使用するからです。取り出したメッセージをファイルとしてダウンロードする場合には、プラットフォームに合わせて<ラインフィード除去>の値を設定してください。

引数<メッセージフォルダ>は、POP3_Download コマンドを使って取り出したメッセージがデフォルトで格納されるフォルダのローカルパス名を表すテキスト値です。

引数<添付フォルダ>は、MSG_Extract コマンドが添付ファイルとメッセージの本文を切り離す際に添付ファイルが格納されるフォルダのローカルパス名を含んでいるテキスト値です。

参照

「MSG_Extract」コマンド、「POP3_Download」コマンド、「POP3_GetPrefs」コマンド

POP3_GetPrefs (改行文字 ; メッセージフォルダ ; 添付フォルダ) 整数

引数	タイプ	説明
ラインフィード除法	整数	0 = CR/LF を取り除かない、1 = CR/LF を取り除く
メッセージフォルダ	テキスト	メッセージフォルダのパス ("" の場合は変更なし)
添付フォルダ	テキスト	添付フォルダのパス ("" の場合は変更なし)
関数の返す値	整数	エラーコード

説明

POP3_GetPrefs コマンドは、POP3 コマンドの現在の環境設定を返します。この設定内容は、パラメータ (引数) 内にリストされた変数の中に返されます。

引数 < ラインフィード除法 > は、ユーザが現在設定しているラインフィードを取り除くかどうかを示す値を返します。

引数 < メッセージフォルダ > は、取り出したメッセージが格納されるデフォルトフォルダのローカルパス名を返すテキスト変数です。

引数 < 添付フォルダ > は、解凍 (展開) された添付ファイルが格納されるデフォルトフォルダのローカルパス名を返すテキスト変数です。

参照

「POP3_SetPrefs」コマンド

POP3_Login (ホスト名 ; ユーザ名 ; パスワード ; APOP ; pop3_ID) 整数

引数	タイプ	説明
ホスト名	文字列	POP3 メールサーバーのホスト名または IP アドレス
ユーザ名	文字列	ユーザ名
パスワード	文字列	パスワード
aPOP	整数	0 = 通常のログイン、1 = APOP ログイン
pop3_ID	倍長整数	この POP3 ログインのリファレンス
関数の返す値	整数	エラーコード

説明

POP3_Login コマンドは、引数<ユーザ名>と<パスワード>を使って、POP3 メールサーバーの中にユーザをログインさせます。引数<aPOP>が0または指定しなかった場合は、通常のユーザ名とパスワードを入力したログインを実行します。特殊なログインでは、後述のコマンドで参照される引数<pop3_ID>のリファレンスを取得します。

警告：POP3 サーバーは、インタラクティブ（対話）形式でアクセスするように設計されていません。任意のサーバーにいったんログインすると、必要な操作はすべて実行し、そしてできるだけ早くログアウトしなければなりません。POP3_Login コマンドと POP3_Logout コマンドの間の呼び出しにおいて、あなたのプロシージャは任意のユーザインタラクティブ画面内に配置されてはいけません。POP3 サーバーはある一定間隔を過ぎても何のアクションもないセッションの接続を自動的に切断します。POP3 の RFC では、この一定間隔として最小 30 分の時間をサポートしています。しかし、ACI 社の経験から言うと、ほとんどのサーバーがもっと短い時間で何のアクションもないクライアントの接続を切断しています。

POP3 サーバーと対話する各コマンドは、強制的に非稼働時間をリセットします。そのため POP3_Logout コマンドの呼び出しを実行する前にあなたの接続を中止したサーバーのイベントでは、あなたが行った削除処理はロールバックされます。

引数<ホスト名>は、POP3 メールサーバーのホスト名または IP アドレスです。IP アドレスを使用することもできますが、できればホスト名を使用することをお勧めします。

引数<ユーザ名>は、POP3 メールサーバー上に登録しているユーザの名前です。<ユーザ名>はドメインを含んではいけません。例えば、アドレス “jack@aci.com ” における<ユーザ名>は “jack ” だけです。

引数<パスワード>は、POP3 メールサーバー上に登録している<ユーザ名>のパスワードです。

引数 < aPOP > は、ログインする際に APOP メカニズムが使用されたかどうかを示す整数値です。 < aPOP > が 1 の場合は、APOP メカニズムが使用されています。0 の場合は、ユーザ名とパスワードを入力してログインします。デフォルト値は 0 です。

引数 < pop3_ID > には、設定されたばかりのセッションのリファレンスを示す倍長整数値が返されます。この章で後述する POP3 コマンド内で使用されます。

参照

「POP3_Logout」コマンド

POP3_VerifyID

IT Review Mail

バージョン 6.5

POP3_VerifyID (pop3_ID) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス 0 = 接続がすでにクローズしている場合。
関数の返す値	整数	エラーコード

説明

POP3 サーバーは、アドミニストレーター（管理者）によって設定された時間内に何のアクションも起こしていないセッションを自動的に切断します。POP3 サーバーと対話する各コマンドは、強制的にこの非稼働時間をリセットします。POP3_VerifyID コマンドは、何のアクションも実行することなく、指定された POP3 セッションの非稼働時間をリセットします。これにより、ユーザはタイムアウトを起こしているかもしれないセッションがある場合に、そのセッションをアクティブ状態のまま保持することができます。

このコマンドを実行すると、POP3_VerifyID コマンドはまだクローズされていない接続をチェックします。セッションがオープンしたままの場合は、このコマンドは POP3 サーバーに対して、0 に戻ったセッションのタイムアウトカウンターをリセットするように通知します。接続がすでにクローズしている場合は、POP3_VerifyID コマンドはその旨を示すエラーを返して、その POP3 セッションで使用されていたメモリ領域を解放し、引数 < pop3_ID > に 0 を返します。

引数 < pop3_ID > は、POP3_Login コマンドで作成されたオープンセッションを参照するための倍長整数です。

参照

「POP3_Login」コマンド

POP3_Reset

IT Review Mail

バージョン 6.5

POP3_Reset (pop3_ID) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス
関数の返す値	整数	エラーコード

説明

POP3_Reset コマンドは、メッセージ総数をリセットし、現在稼動しているセッション内で削除するための目印が付けられたメッセージの削除を取り消します。

注意：POP3_Reset コマンドは、削除されるメッセージ用のフラグをセットするだけです。POP3 サーバー上のメッセージは、(POP3_Logout コマンドで) 正常にログアウトされた場合にしか削除されません。

引数 < pop3_ID > は、POP3_Login コマンドで作成されたオープンセッションを参照するための倍長整数です。

参照

「POP3_Delete」コマンド、「POP3_Login」コマンド

POP3_Delete

IT Review Mail

バージョン 6.5

POP3_Delete (pop3_ID; 開始メッセージ ; 終了メッセージ) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス
開始メッセージ	倍長整数	開始メッセージ番号
終了メッセージ	倍長整数	終了メッセージ番号
関数の返す値	整数	エラーコード

説明

POP3_Delete コマンドは、引数<開始メッセージ>から引数<終了メッセージ>までの各メッセージに削除用の目印を付けます。メッセージの削除処理は、POP3_Logout コマンドが正常に実行されるまでは行われません。POP3_Logout コマンドを呼び出す前に現在稼動しているセッションが何らかの理由（例えば、タイムアウト、ネットワーク障害等）で切断した場合は、削除用に目印が付けられたメッセージはPOP3 サーバー上にそのまま存在します。

引数< pop3_ID > は、POP3_Login コマンドで作成されたオープンセッションを参照するための倍長整数です。

引数<開始メッセージ> は、削除するメッセージの開始メッセージ番号を示す倍長整数です。

引数<終了メッセージ> は、削除するメッセージの終了メッセージ番号を示す倍長整数です。

注意 : POP3_Delete コマンド、POP3_MsgLstInfo コマンド、POP3_MsgLst コマンドは、引数<開始メッセージ>が<終了メッセージ>よりも大きい番号の場合、エラーを返しません。このような問題が起こるイベントでは、POP3_Delete コマンドは何も行いません。

参照

「POP3_Logout」コマンド

POP3_Logout

IT Review Mail

バージョン 6.5

POP3_Logout (pop3_ID) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス 0 = コマンドがログオフに成功した場合。
関数の返す値	整数	エラーコード

説明

POP3_Logout コマンドは、引数 < pop3_ID > 変数によって参照されるオープンしている POP3 セッションをログアウトします。このコマンドは POP3 サーバーのログオフに成功した場合、現在の < pop3_ID > に 0 が返されます。

POP3 サーバーからのログオフにより、そのセッション内で行われた削除処理をコミットしたいことを POP3 サーバーに通知できます。ログアウトする前に行った削除処理をロールバックしたい場合は、POP3_Logout コマンドを実行する前に POP3_Reset コマンドを使用します。

引数 < pop3_ID > は、POP3_Login コマンドで作成されたオープンセッションを参照するための倍長整数です。

参照

「POP3_Reset」コマンド

POP3_BoxInfo

IT Review Mail

バージョン 6.5

POP3_BoxInfo (pop3_ID; メッセージ数 ; メッセージサイズ) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス
メッセージ数	倍長整数	メッセージ総数
メッセージサイズ	倍長整数	すべてのメッセージの合計サイズ
関数の返す値	整数	エラーコード

説明

POP3_BoxInfo コマンドは、引数 < pop3_ID > で参照されるオープンしているセッションのメールボックスに現在存在しているメッセージの総数および合計サイズの情報を返します。

引数 < pop3_ID > は、POP3_Login コマンドで作成されたオープンセッションを参照するための倍長整数です。

引数 < メッセージ数 > には、メールボックス内のメッセージ総数を示す倍長整数が返されます。

引数 < メッセージサイズ > には、メールボックス内の全メッセージの合計サイズを示す倍長整数が返されます。

参照

「POP3_Login」コマンド

POP3_MsgInfo (pop3_ID; メッセージ番号; メッセージサイズ; ユニーク ID) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス
メッセージ番号	倍長整数	メッセージ番号
メッセージサイズ	倍長整数	メッセージサイズ
ユニーク ID	文字列	サーバー上のメッセージのユニーク ID
関数の返す値	整数	エラーコード

説明

POP3_MsgInfo コマンドは、引数 < pop3_ID > で参照されるオープンしているメールボックスの中に存在する引数 < メッセージ番号 > で示されたメッセージに関する情報を返します。そして、そのメッセージのサイズやそのメッセージのユニーク ID がそれぞれ引数 < メッセージサイズ > と < ユニーク ID > に返されます。

引数 < pop3_ID > は、POP3_Login コマンドで作成されたオープンセッションを参照するための倍長整数です。

引数 < メッセージ番号 > は、情報を取り出したいメールボックス内のメッセージを示す倍長整数値です。この < メッセージ番号 > は、現在選択されているメッセージリストの中にあるメッセージの位置を表します。ただし、セッション間で特定の E-メール項目をそのまま同じように残しておくためにこの < メッセージ番号 > を使用することはできません。

引数 < メッセージサイズ > には、< メッセージ番号 > で参照されるメッセージのサイズを示す倍長整数が返されます。

引数 < ユニーク ID > はサーバー上のメッセージのユニーク ID を表す文字列変数です。この < ユニーク ID > は POP3 サーバソフトウェアによってそのメッセージに割り当てられた値です。この値は、< メッセージ番号 > の場合と同じようにセッション間において変化しません。 < ユニーク ID > は、使用しているデータベースが POP3 サーバから任意のメッセージをダウンロードしたかどうかをチェックするのにとても有効なリファレンスです。

参照

「POP3_Login」コマンド

POP3_GetMessage

IT Review Mail

バージョン 6.5

POP3_GetMessage (pop3_ID; メッセージ番号 ; オフセット ; 文字数 ; メッセージ) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス
メッセージ番号	倍長整数	メッセージ番号
オフセット	倍長整数	取り出しを開始する際の文字のオフセット
文字数	倍長整数	返す文字の数
メッセージ	テキスト	メッセージテキスト
関数の返す値	整数	エラーコード

説明

POP3_GetMessage コマンドは、引数 < pop3_ID > で参照されるオープンしているメールボックスの中に存在する < メッセージ番号 > で示されたメッセージの全テキストを返します。POP3_SetPref コマンドで特に指定がない限り、任意のラインフィールドは取り除かれます。POP3_GetMessage コマンドは、ヘッダ情報も含んだメッセージの全ブロックを返します。

引数 < pop3_ID > は、POP3_Login コマンドで作成されたオープンセッションを参照するための倍長整数です。

引数 < メッセージ番号 > は、情報を取り出したいメールボックス内のメッセージを示す倍長整数値です。この < メッセージ番号 > は、現在選択されているメッセージリストの中にあるメッセージの位置を表します。セッション間で特定の E- メール項目をそのまま同じように残しておくためにこの < メッセージ番号 > を使用することはできません。

引数 < オフセット > は、取り出しを開始するメッセージの開始文字の番号を示す倍長整数値です。ほとんどの環境では、この引数に 0 が渡されます。

引数 < 文字数 > は、 < オフセット > の位置から取り出す文字数を表す倍長整数値です。4th Dimention のテキスト変数またはテキストフィールドの最大サイズは 32K(32767 バイト) なので、この < 文字数 > も 32K 以下の数値を設定する必要があります。全体のサイズが 32K 以上のメッセージは、POP3_Download コマンドを介して取り出す必要があります。

引数 < メッセージ > は、取り出されたテキストを受け取るテキスト変数またはテキストフィールドです。 < メッセージ > が日本語の場合、AJP Nkf コマンドによるエンコードが必要です。

参照

「POP3_Download」コマンド、「POP3_SetPrefs」コマンド、AJP Nkf コマンド

POP3_MsgLstInfo (pop3_ID; 開始メッセージ; 終了メッセージ; サイズ配列; メッセージ番号配列; ユニーク ID 配列) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス
開始メッセージ	倍長整数	開始メッセージ番号
終了メッセージ	倍長整数	終了メッセージ番号
サイズ配列	倍長整数配列	サイズの配列
メッセージ番号配列	倍長整数配列	メッセージ番号の配列
ユニーク ID 配列	文字列 / テキスト配列	ユニーク ID の配列
関数の返す値	整数	エラーコード

説明

POP3_MsgLstInfo コマンドは、メールボックスの中にあるメッセージに関する情報を返します。この情報は、1つのメッセージに対応している配列の各要素を示す3つの配列の中に返されます。つまり、各メッセージのサイズ、メッセージ番号、メッセージのユニーク ID に関する情報が返されます。引数として渡される配列は、前もってタイプおよびサイズの定義が行われている必要があります。POP3_MsgLstInfo コマンドは、各配列サイズを取り出したメッセージの数にリセットします。

POP3_MsgLstInfo コマンドは、現在選択されているメッセージリストの中にあるメッセージの情報取得に失敗しても、エラー番号を返しません。エラーが発生した場合、そのエラーメッセージに対して配列の中に作成される配列要素は何もありません。これに対して、各メッセージの情報取得がうまく行った場合は、引数<メッセージ番号配列>に順番に並んだ数値が入っているはずです。何らかの問題が起こったら、<メッセージ番号配列>には順番が飛び飛びになった番号が入っているかもしれません。

引数< pop3_ID > は、POP3_Login コマンドで作成されたオープンセッションを参照するための倍長整数です。

引数< 開始メッセージ番号 > は、検査されるメッセージ範囲の開始メッセージ番号を示す倍長整数です。このメッセージ番号は、引数< pop3_ID > で識別されるメールボックスのすべてのメッセージリストの中にあるメッセージの位置を表している値です。

引数< 終了メッセージ番号 > は、検査されるメッセージ範囲の終了メッセージ番号を示す倍長整数です。このメッセージ番号は、引数< pop3_ID > で識別されるメールボックスのすべてのメッセージリストの中にあるメッセージの位置を表している値です。

引数< サイズ配列 > には、< 開始メッセージ番号 > と < 終了メッセージ番号 > の間にある各メッセージのサイズを含んだ倍長整数配列が返されます。

引数< メッセージ番号配列 > には、< 開始メッセージ番号 > と < 終了メッセージ番号 > の間にあるメッセージ番号を含んだ倍長整数配列が返されます。

引数<ユニーク ID 配列>には、<開始メッセージ番号>と<終了メッセージ番号>の間にあるユニーク ID を含んだ文字型またはテキスト型の配列が返されます。

注意 : POP3_Delete コマンド、POP3_MsgLstInfo コマンド、POP3_MsgLst コマンドは、引数<開始メッセージ>が<終了メッセージ>よりも大きい番号の場合、エラーを返しません。このような問題が起こるイベントでは、POP3_Delete コマンドは何も行いません。

参照

「POP3_MsgInfo」コマンド、「POP3_MsgLst」コマンド

POP3_MsgLst (pop3_ID; 開始メッセージ; 終了メッセージ; ヘッダ配列; メッセージ番号配列; ユニーク ID 配列; 値配列) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス
開始メッセージ	倍長整数	開始メッセージ番号
終了メッセージ	倍長整数	終了メッセージ番号
ヘッダ配列	文字列 / テキスト配列	取り出すヘッダの配列
メッセージ番号配列	倍長整数配列	メッセージ番号の配列
ユニーク ID 配列	倍長整数配列	ユニーク ID の配列
値配列	2 次元の文字列 / テキスト配列	ヘッダの値の 2 次元配列
関数の返す値	整数	エラーコード

説明

POP3_MsgLst コマンドは、メールボックスコンテンツの特定の情報を取得するために使用されます。引数 <ヘッダ配列> は、取り出したい特定のメールヘッダを一覧にした文字列配列またはテキスト配列です。引数 <値配列> は <ヘッダ配列> 内で指定された各ヘッダに関するデータを受け取る 2 次元配列です。要求された各ヘッダは、1 次元配列 <値配列> の中に対応している配列を持っています。

このコマンドを使用することにより、ユーザはメッセージリストの特定カラムを要求することができます。このコマンドはヘッダ項目の値のみを返すだけであり、メッセージの本文を取り出すために使用することはできません。

例 :

```
aHeaders{1}:="Date:"
aHeaders{2}:="From:"
aHeaders{3}:="Subject:"
POP3_MsgLst (<>POP3_ID;vStart;vEnd;aHeaders;aMsgNum;aUIDs;aValues)
```

上記実行後、配列に以下のように値がセットされます。

```
aHeaders{1}{1}..."Mon, 12 Jul 1999 22:19:01 +0900"
aHeaders{2}{1}..."takahasi@aci.co.jp"
aHeaders{3}{1}..." 奥さんに電話してください。 "
```

ヘッダー中の日本語は RFC1522 によりエンコードされているため、**AJP_Nkf** によりデコードする必要があります。

エラーは次のように処理されます :

1) 通信に関するエラーが返されるだけです。このコマンドが何らかのエラー（例えば、ネットワーク障害、シンタックスエラー、サーバーエラー等）のためにタスクを正常に終了することができなかった場合には、そのエラー原因を示すエラーコードが返されます。

2) メッセージリストの指定範囲内にメッセージが存在しないか、またはエラーがある場合：

— そのメッセージ用に作成される配列要素は何もありません。

— 返されるエラーコードは何もありません。

3) あるメッセージ内に指定されたヘッダの一部またはすべてを配置できない場合、任意のエラーを構築しません：

— そのメッセージ用に作成される配列要素は何もありません。

— メッセージ番号およびユニーク ID の配列要素は、それぞれにあった値を含んでいます。

— メッセージに存在しない各ヘッダにおいて、空の文字列がその配列要素の中に返されます。

— 返されるエラーコードは何もありません。

注意：POP3_Delete コマンド、POP3_MsgLstInfo コマンド、POP3_MsgLst コマンドは、引数<開始メッセージ>が<終了メッセージ>よりも大きい番号の場合、エラーを返しません。このような問題が起こるイベントでは、POP3_Delete コマンドは何も行いません。

参照

「POP3_MsgInfo」コマンド、「POP3_MsgLstInfo」コマンド

POP3_Download (pop3_ID; メッセージ番号; ヘッダのみ; ファイル名) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス
メッセージ番号	倍長整数	メッセージ番号
ヘッダのみ	整数	0 = メッセージの全内容、1 = ヘッダのみ
ファイル名	テキスト	ローカルファイル名

関数の返す値 整数 エラーコード

説明

POP3_Download コマンドは、POP3 サーバーから任意メッセージを取り出すために設計されています。そして、ディスクベースのファイルにそれをダウンロードします。添付ファイルを含んでいる POP3 メッセージや全部で 32K 以上になるメッセージは、このコマンドを使ってダウンロードする必要があります。メッセージに添付しているファイルは、この方法で取り出されたメッセージからしか解凍（展開）することはできません。

引数 < pop3_ID > は、POP3_Login コマンドで作成されたオープンセッションを参照するための倍長整数です。

引数 < メッセージ番号 > は、情報を取り出したいメールボックス内のメッセージを示す倍長整数値です。この < メッセージ番号 > は、現在選択されているメッセージリストの中にあるメッセージの位置を表します。ただし、セッション間で特定の E-メール項目をそのまま同じように残しておくためにこの < メッセージ番号 > を使用することはできません。

引数 < ヘッダのみ > は、メッセージの全内容またはヘッダ情報のみのどちらを取り出すかを示した整数値です。

引数 < ファイル名 > は、保存するメッセージ場所を示すパスとファイル名を含んでいます。この値には、次の 3 つの異なる方法を指定することができます：

"" この保存方法は POP3_SetPref コマンドでセットしたフォルダ内のファイルを "temp1" (すでに同じ名前のファイルが存在する場合、ファイル名は "temp2"、"temp3" というように使用されていないファイル名が見つかるまでこの命名規則が順番に繰り返される) という名前で保存します。

"ファイル名" この保存方法は POP3_SetPref コマンドでセットしたフォルダ内のファイルを引数 < ファイル名 > に変えて保存します。

"パス: ファイル名" この保存方法は引数 < ファイル名 > で指定されたパス内のファイルに保存します。

1 番目と 2 番目の指定方法では、POP3_SetPref コマンドによって指定されたフォルダがない場合、メッセージはそのデータベースのストラクチャファイルと同じフォルダに保存されます。ファイルがディスクに保存されると、ファイルの最終的な名前が引数 < ファイル名 > として渡された変数に返されます。ダウンロードフォルダの中にすでに存在する < ファイル名 > で POP3_Download コマンドを呼び出そうとすると、その名前は順番にインクリメントされず、ディスクに保存される新しい値は、< ファイル名 > 変数に返されます。

参照

「POP3_SetPrefs」コマンド

POP3_UIDToNum

POP3_UIDToNum (pop3_ID; ユニーク ID; メッセージ番号) 整数

引数	タイプ	説明
pop3_ID	倍長整数	POP3 ログインのリファレンス
ユニーク ID	文字列	サーバー上のメッセージのユニーク ID
メッセージ番号	倍長整数	メッセージ番号
関数の返す値	整数	エラーコード

説明

POP3_UIDToNum コマンドは、あるメッセージのユニーク ID 値を引数 < pop3_ID > で参照されるメールボックスのメッセージリストの中に存在する < メッセージ番号 > で現在選択されているメッセージのユニーク ID に変換します。ある特定メールメッセージの < メッセージ番号 > はそのメールリスト内の他の項目に関係した流動的な値なので、このコマンドは前回の POP3 セッション内で取り出された情報メッセージの現在位置を返します。

引数 < pop3_ID > は、POP3_Login コマンドで作成されたオープンセッションを参照するための倍長整数です。

引数 < ユニーク ID > は、POP3 サーバー上にあるメッセージのユニーク ID を含んでいる文字列値です。このコマンドは < pop3_ID > で参照されるアカウントのメッセージヘッダ内のこの値を検索します。見つかると、そのメッセージのリスト内の現在位置が < メッセージ番号 > に返されます。

引数 < メッセージ番号 > は、 < pop3_ID > で識別される項目の現在のメッセージ番号（現在選択されているメッセージリスト内のそのメッセージの位置）を含んだ倍長整数値です。 < ユニーク ID > がそのサーバー上で見つからない場合は、 < メッセージ番号 > に 0 が返され、エラーも返されません。

4

IT Downloaded Mail

“MSG_”という接頭辞の付いた POP3 コマンドは、第 3 章の中で説明した POP3_Download コマンドを使ってローカルファイルに保存されたメールメッセージを操作することができます。この章で説明するコマンドは MIME に完全対応しているので、4D Internet Commands は添付ファイルの抽出またはデコード方法を提供します。MIME に関する詳細は、RFC#1521、RFC#1522、RFC#2045 を参照してください。

いったんメッセージがローカルファイルにダウンロードされると、この章のコマンドはそのドキュメントを操作するための様々な機能を提供します。これらのコマンドは、メッセージのある部分における情報取得やメッセージ本文とヘッダ情報の切り離し等を行うことができます。また、既存ドキュメントの削除だけでなくそのメッセージに添付しているファイルの検出および抽出を行うこともできます。

MSG_FindHeader (ファイル名 ; ヘッダラベル ; ヘッダ値) 整数

引数	タイプ	説明
ファイル名	テキスト	ファイル名 (メッセージフォルダのデフォルトパス)
ヘッダラベル	文字列	ヘッダラベル (" From(差出人)", "To(受取人)", "Subject(件名)" 等)
ヘッダ値	テキスト	値
関数の返す値	整数	エラーコード

説明

POP3_Download コマンドによってディスクに取り出されたメッセージドキュメントである引数<ファイル名>を元にして、MSG_FindHeader コマンドは引数<ヘッダラベル>であるヘッダ部を検索して、フィールドに割り当てられた値を引数<ヘッダ値>の中に返します。

引数<ファイル名>は、ファイルの名前またはヘッダ情報を取り出すファイルの完全なパス名です。ファイル名しか指定しなかった場合は、POP3_SetPrefs コマンドで設定したフォルダがデフォルトパスとなります。フォルダが POP3_SetPrefs コマンドで指定されなかった場合は、データベースのストラクチャが入っているフォルダがデフォルトパスとなります。

引数<ヘッダラベル>は、ヘッダラベルの名前を含んだ文字列です。<ヘッダラベル>は “ From: ”、 “ To: ”、 “ Subject: ” 等のユーザが定義したヘッダを参照することができます。

引数<ヘッダ値>は、指定したヘッダフィールドに割り当てられた値が返されるテキスト変数またはテキストフィールドです。

参照

「POP3_Download」コマンド、「POP3_SetPrefs」コマンド

MSG_MessageSize (ファイル名 ; ヘッダサイズ ; 本文サイズ ; メッセージサイズ) 整数

引数	タイプ	説明
ファイル名	テキスト	ファイル名 (メッセージフォルダのデフォルトパス)
ヘッダサイズ	倍長整数	ヘッダサイズ (環境設定が ON の場合は改行文字を除 外)
本文サイズ	倍長整数	本文サイズ (環境設定が ON の場合は改行文字を除外)
メッセージサイズ	倍長整数	メッセージ全体またはファイルサイズ (環境設定は無 視)
関数の返す値	整数	エラーコード

説明

POP3_Download コマンドによってディスクに取り出されたメッセージドキュメントである <ファイル名> を元にして、MSG_FindHeader コマンドはそのメッセージの様々な部分に関するサイズを返します。

引数 <ファイル名> は、ファイルの名前またはメッセージ情報を返すファイルの完全なパス名です。ファイル名しか指定しなかった場合は、POP3_SetPrefs コマンドで設定したフォルダがデフォルトパスとなります。フォルダが POP3_SetPrefs コマンドで指定されなかった場合は、データベースのストラクチャが入っているフォルダがデフォルトパスとなります。

引数 <ヘッダサイズ> には、ヘッダのサイズを含んでいる倍長整数型の変数が返されます。

引数 <本文サイズ> には、本文のサイズを含んでいる倍長整数型の変数が返されます。

引数 <メッセージサイズ> には、メッセージのサイズを含んでいる倍長整数型の変数が返されます。

参照

「POP3_Download」コマンド、「POP3_SetPrefs」コマンド

MSG_GetHeaders (ファイル名 ; オフセット ; 文字数 ; ヘッダテキスト) 整数

引数	タイプ	説明
ファイル名	テキスト	ファイル名 (メッセージフォルダのデフォルトパス)
オフセット	倍長整数	ヘッダの中に開始するオフセット (0 = ヘッダの開始)
文字数	倍長整数	文字数
ヘッダテキスト	テキスト	ヘッダテキスト (環境設定が ON の場合はラインフィードを除外)
関数の返す値	整数	エラーコード

説明

MSG_GetHeaders コマンドは、メッセージの全ヘッダ部分のテキストを返します。POP3 メッセージのヘッダ部分は、メッセージの開始位置から 2 つの連続するキャリッジリターン / ラインフィードが最初に現れる箇所までのテキストとして定義されます。

引数 <ファイル名> は、ファイルの名前またはヘッダ情報を取り出すファイルの完全なパス名です。ファイル名しか指定しなかった場合は、POP3_SetPrefs コマンドで設定したフォルダがデフォルトパスとなります。フォルダが POP3_SetPrefs コマンドで指定されなかった場合は、データベースのストラクチャが入っているフォルダがデフォルトパスとなります。

引数 <オフセット> は、取り出しを開始するヘッダ情報の内部にある文字位置です。

引数 <文字数> は、返される文字の数です。ヘッダ部の長さは、MSG_MessageSize コマンドで調べることができます。

引数 <ヘッダテキスト> は、ヘッダのテキストを受け取ります。

参照

「MSG_MessageSize」コマンド、「POP3_SetPrefs」コマンド

MSG_GetBody (ファイル名 ; オフセット ; 文字数 ; 本文テキスト) 整数

引数	タイプ	説明
ファイル名	テキスト	ファイル名 (メッセージフォルダのデフォルトパス)
オフセット	倍長整数	メッセージ本文の中に開始するオフセット (0 = ヘッダの開始)
文字数	倍長整数	文字数
本文テキスト	テキスト	本文テキスト (環境設定が ON の場合は改行文字を除外)
関数の返す値	整数	エラーコード

説明

MSG_GetBody コマンドは、メッセージのテキストのみを返します。添付ファイルは含まれません。また、すべての MIME ヘッダも除外されます。

引数<ファイル名>は、ファイルの名前またはメッセージ本文を取り出すファイルの完全なパス名です。ファイル名しか指定しなかった場合は、POP3_SetPrefs コマンドで設定したフォルダがデフォルトパスとなります。フォルダが POP3_SetPrefs コマンドで指定されなかった場合は、データベースのストラクチャが入っているフォルダがデフォルトパスとなります。

引数<オフセット>は、取り出しを開始するメッセージ本文の内部にある文字位置です。

引数<文字数>は、返される文字の数です。

引数<本文テキスト>は、メッセージ本文のテキストを受け取ります。<本文テキスト>に日本語を含む場合、AJP Nkf コマンドによるデコードが必要です。

参照

「POP3_SetPrefs」コマンド

MSG_GetMessage (ファイル名 ; オフセット ; 文字数 ; 生テキスト) 整数

引数	タイプ	説明
ファイル名	テキスト	ファイル名 (メッセージフォルダのデフォルトパス)
オフセット	倍長整数	メッセージ行の中に開始するオフセット (0 = ファイルの開始)
文字数	倍長整数	文字数
生テキスト	テキスト	生のテキスト (環境設定は無視)
関数の返す値	整数	エラーコード

説明

MSG_GetMessage コマンドは、添付ファイルの有無に関係なくメッセージの生テキストを返します。MIME ヘッダは除外されません。

引数 < ファイル名 > は、ファイルの名前またはメッセージ本文を取り出すファイルの完全なパス名です。ファイル名しか指定しなかった場合は、POP3_SetPrefs コマンドで設定したフォルダがデフォルトパスとなります。フォルダが POP3_SetPrefs コマンドで指定されなかった場合は、データベースのストラクチャが入っているフォルダがデフォルトパスとなります。

引数 < オフセット > は、取り出しを開始するメッセージ本文の内部にある文字位置です。

引数 < 文字数 > は、返される文字の数です。

引数 < 生テキスト > は、メッセージのテキストを返します。ラインフィード文字を除外する環境設定は無視され、メッセージ本文の中に組み込まれた添付ファイルの除外も無視します。

参照

「POP3_SetPrefs」コマンド

MSG_HasAttach (ファイル名 ; 添付数) 整数

引数	タイプ	説明
ファイル名	テキスト	ファイル名 (メッセージフォルダのデフォルトパス)
添付数	倍長整数	添付数
関数の返す値	整数	エラーコード

説明

ファイルが添付されている場合、MSG_HasAttach コマンドは、その添付ファイルの数を引数 <添付数> に返します。添付ファイルは、非テキストの MIME 同封物です。メッセージにファイルが添付されていない場合は、<添付数> に 0 が返されます。

引数 <ファイル名> は、ファイルの名前または添付ファイルをチェックするファイルの完全なパス名です。ファイル名しか指定しなかった場合は、POP3_SetPrefs コマンドで設定したフォルダがデフォルトパスとなります。フォルダが POP3_SetPrefs コマンドで指定されなかった場合は、データベースのストラクチャが入っているフォルダがデフォルトパスとなります。

引数 <添付数> には、引数 <ファイル名> に添付されているファイル数を指定する整数値が返されます。

参照

「POP3_SetPrefs」コマンド

MSG_Extract (ファイル名 ; デコード ; 添付パス ; 同封リスト) 整数

引数	タイプ	説明
ファイル名	テキスト	ファイル名 (メッセージフォルダのデフォルトパス)
デコード	倍長整数	0 = デコードなし、1 = 可能な場合はデコードあり
添付パス	テキスト	フォルダパス (添付フォルダのデフォルトパス)
同封リスト	文字列 / テキスト配列	同封ファイル名 / フォルダパス
関数の返す値	整数	エラーコード

説明

MSG_Extract コマンドは、すべての添付ファイルを取り出し、それを添付フォルダの中に配置します。

引数 <ファイル名> は、ファイルの名前またはヘッダ情報を取り出すファイルの完全なパス名です。ファイル名しか指定しなかった場合は、POP3_SetPrefs コマンドで設定したフォルダがデフォルトパスとなります。フォルダが POP3_SetPrefs コマンドで指定されなかった場合は、データベースのストラクチャが入っているフォルダがデフォルトパスとなります。

引数 <デコード> は、添付ファイルをデコードするかどうかを指定するための整数値です。 <デコード> が 0 の場合、添付ファイルをデコードしません。1 の場合は、添付ファイルが Binhex、AppleSingle、AppleDouble または Base64 のいずれかでエンコードされている場合にそのファイルをデコードします。

引数 <添付パス> は、添付ファイルが保存されるフォルダパスです。フォルダパスが指定されていないと、そのファイルは、POP3_SetPrefs コマンドで指定された添付ファイルフォルダの中に保存されます。POP3_SetPrefs コマンドで指定された添付ファイルフォルダが存在しない場合は、添付ファイルはデータベースストラクチャと同じフォルダの中に保存されます。

引数 <同封リスト> には、各添付ファイルのファイル名を含んだテキスト配列または文字列配列が返されます。そのファイル名はフルパス名でなく各配列要素の中に返されるだけです。文字列配列を使用している場合、ファイル名の長さは最大 31 バイトなので、少なくともその配列のサイズは 31 バイトにする必要があります。

参照

「POP3_SetPrefs」コマンド

MSG_Delete (ファイル名 ; フォルダ) 整数

引数	タイプ	説明
ファイル名	テキスト	ファイル名 (メッセージフォルダのデフォルトパス)
フォルダ	整数	0 = メッセージのフォルダ、1 = 添付ファイルのフォルダ
関数の返す値	整数	エラーコード

説明

MSG_Delete コマンドは、ローカルファイルを削除します。

引数<ファイル名>は、ファイルの名前または削除するファイルの完全なパス名です。ファイル名しか指定しなかった場合は、次のような条件によってフォルダのデフォルトパスが異なります：引数<フォルダ>がヌルでない(0 または 1) 場合は、そのデフォルトパスは POP3_SetPrefs コマンドで指定されたフォルダを使用します。(メッセージまたは添付) フォルダが POP3_SetPrefs コマンドで指定されなかった場合は、データベースのストラクチャが入っているフォルダがデフォルトパスとなります。引数<フォルダ>がヌルの場合もデータベースのストラクチャが入っているフォルダがデフォルトパスとなります。

引数<フォルダ>は、引数<ファイル名>が削除するファイルの完全なパス名を設定していない場合に指定される整数値です。<フォルダ>には、<ファイル名>で名付けられたファイルが入っているフォルダを指定します。<フォルダ>が 0 の場合は、削除されるファイルはそのメッセージのフォルダの中に入っています。1 の場合は、削除されるファイルはその添付ファイルのフォルダの中に入っています。

警告：MSG_Delete コマンドは、引数に渡された “**任意**” のファイルを削除します。このコマンドの使用については、十分注意してください。

参照

「POP3_SetPrefs」コマンド

5

IT File Transfer

ファイル転送プロトコル (FTP) は、あるコンピュータから別のコンピュータにファイルやアプリケーションを転送するために使用される代表的なプロトコルです。FTP “ サイト ” は、FTP サーバーソフトウェアが稼働している世界中のコンピュータです。ファイル転送プロトコルは、異種システムにおけるファイルの交換方法を提供します。様々なプラットフォーム上のクライアントアプリケーションは、テキストファイルやバイナリーファイルをアップロードまたはダウンロードするために FTP サーバーの中にすべてのログを取ることができます。4D Internet Commands 内の FTP ルーチンは、4th Dimension データベースの内部に FTP クライアントを作成するためのツールを開発者に提供します。

注意：FTP コマンドでパス名を指定する場合、たとえ FTP ホストが FTP サーバーソフトウェアが稼働している Macintosh であることを知っている場合でも、常に UNIX ディレクトリとして FTP サイト上のファイル格納場所を取り扱う必要があります。プラットフォームが何であろうとも、FTP サーバーソフトウェアは接続しているクライアントに対してあなたの UNIX パス名をそのドキュメントを供給する必要のあるフォーマットに内部変換します。

FTP_Progress (左 ; 上 ; ウインドウタイトル ; サーモテキスト ; キャンセル) 整数

引数	タイプ	説明
左	整数	ウインドウの左座標
上	整数	ウインドウの上座標
ウインドウタイトル	文字列	サーモメーターのウインドウタイトル
サーモテキスト	文字列	サーモメーター上のテキスト
キャンセル	文字列	キャンセルボタンのテキスト
関数の返す値	整数	エラーコード

説明

FTP_Progress コマンドは、FTP の進捗インジケータのウインドウ座標およびダイアログボックス内のテキストを定義します。進捗インジケータは、FTP_Send コマンドまたは FTP_Receive コマンドのどちらかを呼び出している最中に現れます。FTP_Progress コマンドは処理がどれだけ進んでいるかを示す進捗ウインドウ自体は表示しません。単にウインドウが送受信コマンドで表示される際のウインドウの特性を定義するだけです。FTP_Send と FTP_Receive の両コマンドは、進捗ウインドウを表示したり、隠したりできるパラメータ（引数）を持っています。

進捗ウインドウはファイル転送が完了した時点で自動的に閉じます。何らかの理由で送受信するファイルサイズが不明の場合、サーモメータが理髪店の前にある紅白の柱みたいに表示され、ファイルサイズが “ unknown (不明) ” と表示されます。

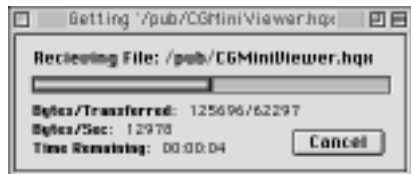
引数 < 左 > は、サーモメータの進捗ウインドウの左側の座標です。 < 左 > が -1 の場合、ウインドウは画面上の水平に中央配置されます。

引数 < 上 > は、サーモメータの進捗ウインドウの上側の座標です。 < 上 > が -1 の場合、ウインドウは画面上の垂直に中央配置されます。

引数 < ウインドウタイトル > は、サーモメータの進捗ウインドウのタイトルです。次ページの図では、ウインドウタイトルが “ Getting: /pub/CGMiniViewer.hqx ” になっています。 < ウインドウタイトル > が空の文字列の場合、ウインドウにはタイトルはありません。

引数 < サーモテキスト > は、サーモメータの進捗状況を示すテキストです。 < サーモテキスト > が “ * ” の場合、テキストはデフォルトを使用します。次ページの図では、 < サーモテキスト > は “ Receiving File: /pub/CGMiniViewer.hqx ” です。 < ウインドウタイトル > が空の文字列の場合、ウインドウにはタイトルはありません。サーモメータのデフォルトテキストは、ホストによって送られる転送処理の状況を示すテキストです。このテキストは、接続の転送処理状況によっていろいろと変化します。

引数<キャンセル>は、「キャンセル」ボタンのテキストです。<キャンセル>が空の文字列の場合、「キャンセル」ボタンは隠されます。<キャンセル>が“*”の場合、テキストはデフォルトテキストの“Cancel (キャンセル)”となります。



例：

```
$Error:=FTP_Progress (-1;-1;"FTP ファイル転送 ","*","*")
```

Case of

```
¥ (FTP_Login ("ftp.acius.com";"anonymous";vEmailID;vFTP_ID;vFTP_Msg)#0)
```

```
¥(FTP_Send (vFTP_ID;"HD:Document:Report(7 × 13)";"/pub/reports";1)#0)
```

```
¥ (FTP_Logout (vFTP_ID)#0)
```

Else

```
$OK:=True ‘すべてのコマンドはエラーなしで実行される
```

End case

FTP_Login (ホスト名 ; ユーザ名 ; パスワード ; ftp_ID { ; ウェルカムテキスト }) 整数

引数	タイプ	説明
ホスト名	文字列	ホスト名または IP アドレス
ユーザ名	文字列	ユーザ名
パスワード	文字列	パスワード
ftp_ID	倍長整数	新規 FTP セッションのリファレンス
ウェルカムテキスト	テキスト	FTP ウェルカムテキスト

関数の返す値	整数	エラーコード
--------	----	--------

説明

FTP_Login コマンドは、引数<ホスト名>のFTP サーバーで接続を開設し、設定済みの<ユーザ名>と<パスワード>を使ってシステムの中にログインします。

引数<ホスト名>は、ホスト名またはリモートシステムの IP アドレスです。

引数<ユーザ名>は、FTP サーバーで認証されるユーザアカウントの名前です。ほとんどの FTP サーバーが “anonymous (匿名)” ユーザを介してゲストがアクセスすることをサポートしています。あるユーザが匿名でログインしている場合、そのユーザの E- メールアドレスをそのユーザのパスワードとして設定するのが慣例となっています。

引数<パスワード>は、システム上の<ユーザ名>用のパスワードです。

引数<ftp_ID>には、新しくオープンしたセッションを示す倍長整数値が返されます。この値は、この章で後述する FTP コマンド内で使用されます。<ftp_ID>は、返される結果を受け入れるために 4th Dimention の変数またはフィールドに渡す必要があります。

引数<ウェルカムテキスト>は、ユーザがシステムの中にログインした際に返されるテキストを含んだオプション引数です。ほとんどの FTP サイトは、ログインと同時に表示されるウェルカムメッセージを持っています。<ウェルカムテキスト>を指定する場合は、<ウェルカムテキスト>は返される結果を受け入れるために 4th Dimention の変数またはフィールドに渡す必要があります。

例：

\$OK:=False

Case of

 ¥(**FTP_Login** ("ftp.acius.com";"anonymous";"dbody@aol.com";vFTP_ID;vFTP_Msg)#0)

 ¥(**FTP_Progress** (-1;-1;" 進捗ウインドウ "; " ファイル取得中 ...";"*)#0)

 ¥(**FTP_Send** (vFTP_ID;"HD:Document:Report";"/pub/ reports";1)#0)

 ¥(**FTP_Logout** (vFTP_ID)#0)

Else

\$OK:=True ‘すべてのコマンドはエラーなしで実行される

End case

参照

「FTP_Logout」コマンド

FTP_GetDirList (ftp_ID; ディレクトリ; オブジェクト名; サイズ; タイプ; 修正日) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
ディレクトリ	テキスト	UNIX ディレクトリのパス名 現在のディレクトリ
オブジェクト名	文字列 / テキスト配列	名前一覧
サイズ	倍長整数配列	サイズ一覧
タイプ	整数配列	タイプ一覧 0 = 標準ファイル 1 = ディレクトリ 2 = ブロックタイプの特権ファイル 3 = 文字タイプの特権ファイル 4 = シンボリックリンク 5 = FIFO 特権ファイル 6 = AF_UNIX アドレスファミリーソケット
修正日	日付配列	修正日リスト
関数の返す値	整数	エラーコード

説明

FTP_GetDirList コマンドは、引数 < ftp_ID > で示された FTP セッションの < ディレクトリ > 内にあるオブジェクトの一覧を取り出します。 < ディレクトリ > 項目であるオブジェクト名、サイズ、タイプ、および修正日に関する情報が、それぞれの 4 つの配列の中に返されます。FTP サイトへの接続は、前もって FTP_Login コマンドでオープンされ、そのまま有効である必要があります。FTP_GetDirList コマンドは、ユーザが現在作業しているディレクトリを指定されたパスに変更して、引数 < ディレクトリ > に返されます。

引数 < ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数 < ディレクトリ > は、FTP ディレクトリを参照する “ホストパス” の書式を持ったテキスト値です。“カレントディレクトリ” が結果として返されるので、4th Dimension の変数やフィールドがこの < ディレクトリ > に渡される必要があります。通常、 < ディレクトリ > に返される値は、 < ディレクトリ > に渡された値と同じものです。ただし、ディレクトリの変更が（アクセスの制限などで）うまく行かない場合もあります。このような場合、 < ディレクトリ > はそのセッションのカレントディレクトリの “ホストパス” を保持します。

< ディレクトリ > に空の文字列が渡されると、配列の中にカレントディレクトリのファイル一覧が返され、 < ディレクトリ > の中にカレントディレクトリのホストパスが返されます。

引数 < オブジェクト名 > は、指定した < ディレクトリ > 内にある各オブジェクトの名前を保持するための文字列配列またはテキスト配列です。

引数<サイズ>は、<ディレクトリ>の中にあるオブジェクトのサイズを保持するための倍長整数配列です。

引数<タイプ>は、<ディレクトリ>の中にある各オブジェクトのタイプを保持するための整数配列です。次の表はこの値を示したものです：

値	ファイルタイプ
0	標準ファイル
1	ディレクトリ
2	ブロックタイプの特殊ファイル
3	文字タイプの特殊ファイル
4	シンボリックリンク
5	FIFO 特殊ファイル
6	AF_UNIX アドレスファミリーソケット

引数<修正日>は、<ディレクトリ>の中にある各オブジェクトの最新修正日を保持するための 4th Dimension の日付配列です。

参照

「FTP_Login」コマンド、「FTP_VerifyID」コマンド

FTP_GetFileInfo (ftp_ID; ホストパス; サイズ; タイプ; 修正日) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
ホストパス	テキスト	ファイルのパス名
サイズ	倍長整数	ファイルのサイズ
修正日	日付配列	修正日リスト
関数の返す値	整数	エラーコード

説明

FTP_GetFileInfo コマンドは、引数<ホストパス>の書式のファイルパス名を指定して、そのファイルのサイズと最新修正日を返します。

引数< ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数<ホストパス> は、ファイル情報を返すドキュメントのテキストパスです。

引数<サイズ> は、<ホストパス>で識別されるファイルのサイズを保持するための倍長整数型の変数またはフィールドです。

引数<修正日> は、ファイルの最新修正日を保持するための日付型の変数またはフィールドです。

参照

「FTP_GetDirList」コマンド

FTP_VerifyID

IT File Transfer

バージョン 6.5

FTP_VerifyID (ftp_ID) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス 0 = 接続がすでにクローズしている場合
関数の返す値	整数	エラーコード

説明

FTP サーバーは、そのアドミニストレータ（管理者）によって設定されたある一定間隔を過ぎても何のアクションもないアカウントの接続を切断します。FTP サーバーと対話する各コマンドは、この非稼働時間を強制的にリセットします。FTP_VerifyID コマンドは、現在の状況やディレクトリを変更することなく、指定された FTP セッションの非稼働時間をリセットします。これにより、ユーザはタイムアウトを起こしているかもしれないセッションがある場合、そのセッションをアクティブ状態のまま保持することができます。

このコマンドを実行すると、FTP_VerifyID コマンドはまだクローズされていない接続をチェックします。セッションがオープンしたままの場合は、このコマンドは FTP サーバーに対して、0 に戻ったセッションのタイムアウトカウンターをリセットするように通知します。接続がすでにクローズしている場合は、FTP_VerifyID コマンドはその旨を示すエラーを返して、その FTP セッションで使用されていたメモリ領域を解放し、引数 < FTP_ID > に 0 を返します。

引数 < FTP_ID > は、FTP_Login コマンドで開設されたセッションのリファレンスを表す倍長整数です。

参照

「FTP_Login」コマンド

FTP_MakeDir (ftp_ID; ディレクトリ) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
ディレクトリ	テキスト	UNIX ディレクトリのパス名
関数の返す値	整数	エラーコード

説明

FTP_MakeDir コマンドは、アクセス可能な<ディレクトリ>を指定して、現在作業を行っているディレクトリ（カレントディレクトリ）の内部に新しいディレクトリを作成します。FTP_GetDirList コマンドを呼び出すことにより、このカレントディレクトリを指定したディレクトリに変更できます。この操作を行うためのアクセス権限を持っていない場合は、エラーが返されます。

引数 < ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数 < ディレクトリ > は、FTP ディレクトリを参照する “ ホストパス ” の書式を持ったテキスト値です。この < ディレクトリ > の値は、完全なパス名か、あるいは単にフォルダ名だけです。省略された形式が用いられている場合は、そのディレクトリはカレントディレクトリの内部に作成されます。 < ディレクトリ > の名前にはスペース文字を含めないことをお勧めします。

参照

「FTP_GetDirList」コマンド、「FTP_RemoveDir」コマンド

FTP_RemoveDir

IT File Transfer

バージョン 6.5

FTP_RemoveDir (ftp_ID; ディレクトリ) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
ディレクトリ	テキスト	UNIX ディレクトリのパス名
関数の返す値	整数	エラーコード

説明

FTP_RemoveDir コマンドは、アクセス可能な<ディレクトリ>を指定して、任意のフォルダディレクトリを削除します。FTP_GetDirList コマンドを呼び出すことにより、このカレントディレクトリを指定したディレクトリに変更できます。この操作を行うためのアクセス権限を持っていない場合は、エラーが返されます。また、何らかの項目を含んでいるディレクトリを削除しようとする、おそらく結果的にセキュリティエラーになります。

引数 < ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数<ディレクトリ>は、既に存在している FTP ディレクトリを参照する“ホストパス”の書式をしたテキスト値です。この<ディレクトリ>の値は、完全なパス名か、あるいは単にフォルダ名だけです。省略された形式が用いられている場合は、指定されたディレクトリはカレントディレクトリの内部に存在しなければなりません。

参照

「FTP_GetDirList」コマンド、「FTP_MakeDir」コマンド

FTP_Rename (ftp_ID; ホストパス ; 新規パス名) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
ホストパス	テキスト	FTP サーバー上にあるファイルのパス名
新規パス名	テキスト	新しいファイル名
関数の返す値	整数	エラーコード

説明

FTP_Rename コマンドは、< ホストパス > の書式を持つファイルのパス名を指定して、リモート FTP サーバー上で指定されたファイルの名前を変更します。この操作を行うためのアクセス権限を持っていない場合は、エラーが返されます。

引数 < ftp_ID > は、FTP_Login コマンドで開設された FTP セッションのリファレンスを表す倍長整数です。

引数 < ホストパス > は、名前が変更されるファイルのテキストパスです。この < ホストパス > の値は、完全なパス名か、あるいは単にファイル名だけです。省略された形式が用いられている場合は、指定されたファイルはカレントディレクトリの内部に存在しなければなりません。

引数 < 新規パス名 > には、リモートファイルの新しい名前が入っています。

FTP_Delete

IT File Transfer

バージョン 6.5

FTP_Delete (ftp_ID; ホストパス) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
ホストパス	テキスト	FTP サーバー上にあるファイルのパス名
関数の返す値	整数	エラーコード

説明

FTP_Delete コマンドは、<ホストパス>の書式を持つファイルのパス名を指定して、リモート FTP サーバーから指定されたファイルを削除します。この操作を行うためのアクセス権限を持っていない場合は、エラーが返されます。

引数 < ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数 <ホストパス> は、名前が変更されるファイルのテキストパスです。この <ホストパス> の値は、完全なパス名か、あるいは単にファイル名だけです。省略された形式が用いられている場合は、指定されたファイルはカレントディレクトリの内部に存在しなければなりません。

参照

「FTP_RemoveDir」コマンド

FTP_MacBinary (ftp_ID;MacBinary モード) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
MacBinary モード	整数	-1 = 現在の設定を取得、1 = 使用可、0 = 使用不可 現在の設定 (-1 が渡された場合)
関数の返す値	整数	エラーコード

説明

FTP_MacBinary コマンドは、FTP_Send コマンドと FTP_Receive コマンドを用いて FTP 転送を行う際の MacBinary モードを使用可 / 使用不可にします。FTP_MacBinary コマンドは、引数 < ftp_ID > で識別される現在の FTP セッションを指定して、引数 < MacBinary モード > に渡された値に従って、MacBinary 転送をオン / オフします。

MacBinary プロトコルは、データフォークとリソースフォークを含んでいるバイナリーデータやバイナリーファイルの転送を容易にするために Macintosh の FTP クライアントとサーバーでしばしば使用されます。

Windows ユーザの注意点 : Windows 環境でも FTP 転送にこの MacBinary プロトコルを使用することはできますが、PC マシン上で MacBinary ファイルをデコードすることはほとんど無意味であることに注目する必要があります。Intel 搭載のマシンは、データフォークとリソースフォークの両方が含まれているファイルを格納することはできません。このようなファイルフォーマットは PC プラットフォームには適さないため、リソースフォークを含む Macintosh ファイルがエンコードしていないフォーマットで保存されている場合、おそらく不正ファイルとなるでしょう。

引数 < ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数 < MacBinary モード > は、MacBinary 転送をオンまたはオフにすることを示す整数です。FTP_MacBinary コマンドは変更が試みられた後で MacBinary 転送の状態を返すことが可能なので、この値を変数で渡す必要があります。 < MacBinary モード > が 1 の場合、MacBinary は使用可能です。0 の場合は使用不可です。-1 の場合は、MacBinary 転送の現在の状態 (1 または 0) が返されます。すべての FTP サーバーが MacBinary プロトコルをサポートしているわけではないので、MacBinary を使用可能にすることを試すことは可能ですが、おそらく FTP_MacBinary コマンドは MacBinary が使用できなかったことを示す 0 を返します。

例：

次の例は、FTP ファイルを受信する前に MacBinary プロトコルを使用可能にします。ファイルが正常に MacBinary で受信されると、MacBinary ファイルはそのオリジナルフォーマットでデコードされ、その MacBinary ファイルは削除されます。

```
vLocalFile:=""
vUseMacBin:=1
  ' ファイルのダウンロードを MacBinary で行う。
$error:=FTP_MacBinary (vFTP_ID;vUseMacBin)
$error:=FTP_Receive (vFTP_ID;"MyApplication";vLocalFile;cbShowTherm)
If ($error=0) & (vUseMacBin=1) ' 受信が OK の場合、ファイルは MacBinary フォーマットになる。
  vDecodePath:=""
  If (IT_Decode (vLocalFile;vDecodePath;8)=0) 'MacBinary のデコード
    ' 元のファイルのデコードがうまく行くと、MacBinary ファイルは削除される。
    DELETE DOCUMENT (vLocalFile)
  End if
End if
```

参照

「IT_Decode」コマンド

FTP_Send (ftp_ID; ローカルパス ; ホストパス ; 進捗) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
ローカルパス	テキスト	送信するドキュメントのパス名
ホストパス	テキスト	送信先のドキュメントのパス名
進捗	整数	1 = 「進捗インジケータ」表示、0 = 「進捗インジケータ」表示しない
関数の返す値	整数	エラーコード

説明

オープンしている FTP セッションの参照番号および送信するドキュメントのパス名とその送信先のパス名を指定して、FTP_Send コマンドはリモートマシンにドキュメントを送信します。FTP_Send コマンドは、FTP ファイルにステータスエラーが発生したかどうかの結果を示すエラーコードを即座に返します。

引数 < ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数 < ローカルパス > は、送信されるドキュメントのパスです。 < ローカルパス > が空の文字列の場合、標準の「ファイルを開く」ダイアログボックスが表示されます。 < ローカルパス > がパスのないファイル名の場合は、FTP_Send コマンドはそのファイルに対するデータベースのストラクチャファイルが入っているフォルダの中を捜しに行きます。ローカルの文書にすべてのパスが付いている場合は、ディレクトリを使用しているプラットフォームに合ったデリミタ（ディレクトリ区切り文字）で区切る必要があります。これに関する詳細は、第 1 章の「用語集と用語」の節を参照してください。

引数 < ホストパス > は、ファイル名を含んでいるドキュメントの送信先のパスです。 < ホストパス > はいったん FTP サーバーで受信されると、その要求されたファイル名を表します。 < ホストパス > が空の文字列の場合、ユーザはディスクから任意のファイルを選択することができます。また、その選択されたファイルの名前が使用されている場合も < ホストパス > は空の文字列になります。

引数 < ホストパス > には、完全なパス名または単にファイル名のみ指定することもできます。完全なパス名を指定した場合は、その指定されたファイルは < ホストパス > で示されたディレクトリの中に配置されます。ファイル名のみが指定されるか、または空の文字列がそのファイルセレクションの中で使用される場合は、ファイルは FTP_GetDirList コマンド（現在作業を行っているディレクトリ [CWD]）で導き出された最終のディレクトリに送られます。

ファイルまたはパス名が正しく区切られていない場合、FTP_Send コマンドはエラーを返します。ユーザがそのディレクトリにファイルを送るための十分な権限を持っていない場合も、エラーが返されます。UNIX のドキュメントにすべてのパスが付いている場合は、パスをスラッシュ (/) で区切る必要があります。これに関する詳細は、第 1 章の「用語集と用語」の節を参照してください。

引数<進捗>は、「進捗インジケータ」を表示するかどうかを示す整数値です。<進捗状況>が1の場合は、「進捗インジケータ」を表示します。0の場合は、「進捗インジケータ」を表示しません。

例：

例題 1

```
$OK:=False
```

Case of

```
  ¥(FTP_Login ("ftp.acius.com";"anonymous";vEmailID;vFTP_ID;vFTP_Msg)#0)
```

```
  ¥(FTP_Progress (-1;-1;" 進捗ウインドウ ";" ファイル取得中 ...";" キャンセル")#0)
```

```
  ¥(FTP_Send (vFTP_ID;"HD:Document:Report";"/pub/reports/" ;1)#0)
```

```
  ¥(FTP_Logout (vFTP_ID)#0)
```

Else

```
  $OK:=True ‘すべてのコマンドはエラーなしで実行される
```

End case

例題 2

```
$error:=FTP_Send (vFTP_ID;"";"";1)
```

参照

「FTP_Progress」コマンド、「FTP_Receive」コマンド

FTP_Append (ftp_ID; ローカルパス ; ホストパス ; 進捗) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
ローカルパス	テキスト	送信するドキュメントのパス名
ホストパス	テキスト	送信先のドキュメントのパス名
進捗	整数	1 = 「進捗インジケータ」表示、0 = 「進捗インジケータ」表示しない
関数の返す値	整数	エラーコード

説明

FTP_Append コマンドは、引数<ホストパス>で識別される既存ファイルの最後に送られているデータを追加するという 1 つの例外を除いて FTP_Send コマンドとまったく同じ動作を行います。このコマンドの主要な機能は、すでに存在しているテキストファイルの最後にデータを追加することです。

参照

「FTP_Send」コマンド

FTP_GetType

FTP_GetType (ftp_ID;FTP モード) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
FTP モード	文字列	"A" = ASCII、"I" = イメージ、"L 8" = 論理 8 ビット
関数の返す値	整数	エラーコード

説明

FTP_GetType コマンドは、現在の FTP 転送モードに関する情報を返します。転送モードは、FTP_SetType コマンドを使ってセットされます。

引数 < ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数 < FTP モード > は、現在の FTP 転送モードを示すコードを返します。

参照

「FTP_SetType」コマンド

FTP_SetType

IT File Transfer
バージョン 6.5

FTP_SetType (ftp_ID;FTP モード) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
FTP モード	文字列	"A" = ASCII、"I" = イメージ、"L 8" = 論理 8 ビット
関数の返す値	整数	エラーコード

説明

FTP_SetType コマンドは、送受信操作中に使用される FTP 転送モードを変更するために使用されます。通常、デフォルトの設定を変更する必要はありません。しかし、様々なプラットフォームや FTP 実行の違いから FTP の転送タイプに関するモードを変更する必要があるかもしれません。特に標準的なテキスト文書を転送する場合には、そのテキストファイルを正しく転送するために ASCII モードに切り替える必要があるかもしれません。

引数 < ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数 < FTP モード > は、送受信操作に用いる任意の FTP 転送モードを示すコードを含んでいる必要があります。

参照

「FTP_GetType」コマンド

FTP_System (ftp_ID; システム情報) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
システム情報	文字列	システム情報
関数の返す値	整数	エラーコード

説明

FTP_System コマンドは、引数 <システム情報> の中に FTP サーバーソフトウェアに関する情報を取得します。

引数 < ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数 <システム情報> は、FTP サーバーに関する情報を含んでいます。

FTP_Receive (ftp_ID; ホストパス ; ローカルパス ; 進捗) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス
ホストパス	テキスト	受信するドキュメントのパス名
ローカルパス	テキスト	受信先のドキュメントのパス名 返されるファイルパス名 (" " が渡される場合)
進捗	整数	1 = 「進捗インジケータ」表示、0 = 「進捗インジケータ」表示しない
関数の返す値	整数	エラーコード

説明

FTP_Receive コマンドは、引数<ホストパス>で参照されるパスからファイル転送プロトコルを使って、ファイルを受け取ります。ファイルが受信先にすでに存在している場合は、エラーコード -48 を返します。

引数< ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

引数<ホストパス>は、受信されるドキュメントのパスを示すテキスト値です。<ホストパス>が完全なパスを持っていない場合は、FTP_Login コマンドはエラーを返します。UNIX のドキュメントにすべてのパスが付いている場合は、パスをスラッシュ (/) で区切る必要があります。これに関する詳細は、第 1 章の「用語集と用語」の節を参照してください。

引数<ローカルパス>は、ドキュメントの受信先パスを示すテキスト値です。<ローカルパス>が空の文字列の場合、標準の「ファイルを開く」ダイアログボックスが表示され、そのダイアログボックスで選択したファイルパス名が<ローカルパス>変数の中に返されます。<ローカルパス>がファイル名しか含んでいない場合は、FTP_Send コマンドはそのファイルに対するデータベースのストラクチャファイルが入っているフォルダの中に保存されます。ローカルのドキュメントにすべてのパスが付いている場合は、使用しているプラットフォームに合ったデリミタ（ディレクトリ区切り文字）で区切る必要があります。これに関する詳細は、第 1 章の「用語集と用語」の節を参照してください。

引数<進捗>は、「進捗インジケータ」を表示するかどうかを示す整数値です。<進捗状況>が 1 の場合は、「進捗インジケータ」を表示します。0 の場合は、「進捗インジケータ」を表示しません。

例

```
vLocalFile:=""  
vUseMacBin:=1  
    ' ダウンロード時に MacBinary に切り替える  
$Error:=FTP_MacBinary (vFTP_ID;vUseMacBin)  
$Error:=FTP_Receive (vFTP_ID;"CGMiniViewer.hqx";vLocalFile;cbShowTherm)  
If ($Error=0) & (vUseMacBin=1)  
    vDecodePath:=""  
    If (IT_Decode (vLocalFile;vDecodePath;8)=0) 'MacBinary デコード  
        DELETE DOCUMENT (vLocalFile) ' デコードがうまく行った場合は、そのドキュメントを削除する。  
    End if  
End if
```

参照

「FTP_MacBinary」コマンド、「IT_Decode」コマンド

FTP_Logout

IT File Transfer
バージョン 6.5

FTP_Logout (ftp_ID) 整数

引数	タイプ	説明
ftp_ID	倍長整数	FTP ログインのリファレンス 0 = セッションが正常にクローズした場合
関数の返す値	整数	エラーコード

説明

オープンしている FTP セッションの参照番号を指定して、FTP_Logout コマンドはサーバーからの接続を切断し、そのセッションで使用されていたメモリを解放します。このコマンドはセッションが正常にクローズした場合、引数 < ftp_ID > の中に 0 を返します。

引数 < ftp_ID > は、FTP_Login コマンドで作成された FTP セッションを参照するための倍長整数です。

例

```
If (FTP_Login ("ftp.acius.com";"anonymous";vEmailID;vFTP_ID;vFTP_Msg)=1)
    $error:=FTP_Send (vFTP_ID;"HD:Document:Report";"/pub/reports";1)
    $error:=FTP_Logout (vFTP_ID)
End if
```

参照

「FTP_Login」コマンド

6

IT TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) は、インターネット上においてデータを送信するのに最もよく使用されているプロトコルです。4D Internet Commands に組み込まれている TCP コマンドにより開発者は TCP セッションを設定し、このセッション経由で TCP パケットを送受信することができます。

TCP 接続を設定する方法には、2通りあります。1 番目方法は、TCP_Open コマンドを実行する方法です。この方法は、特定ポート上で指定されたドメインを使って任意の接続をオープンします。接続をオープンさせるもう 1 つの方法は、TCP_Listen コマンドを実行する方法です。このコマンドも特定ポート上で指定されたドメインを使って任意の接続をオープンし、入ってくる接続を待機します。接続が設定されたかどうかを調べる最も良い方法は、TCP_Listen コマンドが完了した後、TCP_State コマンドを使って、セッションの状態をチェックすることです。返されるステータスコードは、そのセッションの現在の状態に対応します。ここから、TCP_Open コマンドで設定された接続を使って、TCP パケットを送受信することができます。

ローレベルの TCP/IP コマンドは、通信プロトコルに関する高度な知識を必要とします。そのため、これらのルーチンを使用する開発者は、自分が利用しようとしているプロトコルに対して十分に理解している必要があります。ポート番号に割り当てられた TCP/IP、通信プロトコル、アドレス処理の必要条件等に関する情報は、その RFCs の中で参照できます。

TCP_Open

IT TCP/IP
バージョン 6.5

TCP_Open (ホスト名 ; リモートポート ; tcp_ID) 整数

引数	タイプ	説明
ホスト名	文字列	ホスト名または IP アドレス
リモートポート	整数	(0 = 任意) に接続するリモートポート
tcp_ID	倍長整数	この TCP セッションのリファレンス
関数の返す値	整数	エラーコード

説明

TCP_Open コマンドは、ドメインの出でいく TCP 接続を初期化します。

TCP_Open コマンドは、(0 でなければ) 引数<リモートポート>で参照されるポート上で、引数<ホスト名>で参照されるリモート TCP の接続を初期化します。倍長整数の値が、引数< tcp_ID > に返されます。この値はセッションを参照する次のすべての TCP 呼び出しによって使用されます。TCP_Open コマンドは、引数< tcp_ID > で識別されるセッションによって受信されるデータがない場合、30 秒のタイムアウトがセットされます。すべてのコマンドのデフォルトタイムアウト値は、IT_TimeOut コマンドで変更できます。

引数<ホスト名>は、接続をオープンしているマシンのホスト名または IP アドレスです。

引数<リモートポート>は、引数<ホスト名>で示されたマシン上の TCP ポートを指します。

引数< tcp_ID > は、オープンされたセッションのリファレンスを示す倍長整数です。この値は、この章で後述する TCP コマンド内で使用されます。

参照

IT_TimeOut コマンド

TCP_Listen

TCP_Listen (リモートホスト ; リモートポート ; ローカルポート ; タイムアウト ;tcp_ID) 整数

引数	タイプ	説明
リモートホスト	文字列	ホスト名または IP アドレス
リモートポート	整数	接続待機中のポート番号、0 = 任意
ローカルポート	整数	ローカルポート番号、0 = 使用するために未使用ポートを見つける (0 が渡された場合) 使用済みローカルポート番号
タイムアウト	整数	待機する秒数、0 = 永久に待機
tcp_ID	倍長整数	この TCP セッションのリファレンス
関数の返す値	整数	エラーコード

説明

TCP_Listen コマンドは、引数<リモートポート>で参照されるポート上の引数<リモートホスト>で参照されるマシンから生成された接続を待機します。このコマンドは、任意の接続が設定されるか、または引数<タイムアウト>にセットされた時間が経過するまではメソッドを呼び出している 4th Dimention に制御を戻しません。これは任意の接続が設定されるまでデータベースをロックしているかのように見えるかもしれませんが、このコマンドは実行中の他の 4th Dimention プロセスと密接に関連しています。このコマンドは、すでに実行中の他の 4th Dimention プロセス時間をスライス (分割) します。

ほとんどの開発者は、(特に引数<タイムアウト>に永久待機するように指定した場合は) 自分自身の 4D プロセスの中に生成されたメソッドからこの呼び出しを実行します。

引数<リモートホスト>は、接続を待っているマシンのホスト名または IP アドレスです。この引数に空の文字列が渡されると、このコマンドは任意のマシンから入ってくる接続を受け入れます。

引数<リモートポート>は、入ってくる接続用に待機しているポート番号を示します。この引数に 0 の値が渡されると、このコマンドは任意のポート番号上の接続を受け入れます。

引数<ローカルポート>は、通信に使用したいローカルマシン上のポート番号です。この引数に 0 の値が渡されると、このコマンドは未使用のポート番号を見つけて、その番号をこの引数に渡します。

引数<タイムアウト>は、入ってくる接続を待っている秒数を指定します。この引数に 0 の値が渡されると、このコマンドは入ってくる接続を永久に待ち続けます。任意の接続が 1 度も設定されないと、4th Dimention のプロセス呼び出しに制御が 1 度も戻されないのので、この引数に 0 を渡す場合は十分に注意してください。特にシングルプロセスのデータベース内では決してこの引数に 0 は渡さないでください。

引数< tcp_ID > は、オープンされたセッションのリファレンスを示す倍長整数です。この値は、この章で後述する TCP コマンド内で使用されます。

例：

```
C_LONGINT (vTCPID)
C_INTEGER (vStatus)
$err:=TCP_Listen ("";0;0;30;vTCPID)
$err:=TCP_State (vTCPID;vStatus)
If (vStatus=8) ‘ 接続が設定される
    DoSomething
    $err:=TCP_Close (vTCPID)
End if
```

参照

「TCP_Open」コマンド、「TCP_State」コマンド

TCP_Send

IT TCP/IP
バージョン 6.5

TCP_Send (tcp_ID; 送信テキスト) 整数

引数	タイプ	説明
tcp_ID	倍長整数	オープンした TCP セッションのリファレンス
送信テキスト	テキスト	送信テキスト
関数の返す値	整数	エラーコード

説明

TCP_Send コマンドは、引数 < TCP_ID > で指定された TCP セッションのデータを送信します。

引数 < tcp_ID > は、オープンしたセッションを参照するための倍長整数値です。この値は、TCP_Open コマンドまたは TCP_Listen コマンドのどちらかを使って設定されます。

引数 < 送信テキスト > は、< TCP_ID > で参照される TCP セッションに送信されるテキスト値です。

参照

「TCP_Listen」コマンド、「TCP_Open」コマンド

TCP_Receive (tcp_ID; テキスト) 整数

引数	タイプ	説明
tcp_ID	倍長整数	オープンした TCP セッションのリファレンス
テキスト	テキスト	受け取ったテキスト
関数の返す値	整数	エラーコード

説明

TCP_Receive コマンドは、作成された TCP セッションを参照するための倍長整数を取得し、引数 <テキスト> の中にデータの packets を受け取ります。

引数 < tcp_ID > は、オープンしたセッションを参照するための倍長整数値です。この値は、TCP_Open コマンドまたは TCP_Listen コマンドのどちらかを使って設定されます。

引数 <テキスト> は、受け取ったテキストです。TCP packets 経由でデータを受け取ると、1 つの TCP_Receive コマンド呼び出しによって受信しているデータのすべてをカウントできません。TCP_Receive コマンドは、通常、連続して接続ステータスをチェックしたり既存値をスキャンするために用いる Repeat ループの中で呼び出されます。

例 :

```
C_LONGINT ($tcp_id)
C_TEXT ($webpage;$buffer)
C_INTEGER (vState;$error)
$webpage:=""
vState:=0
Repeat
    $error:=TCP_Receive ($tcp_id;$buffer)
    $error:=TCP_State ($tcp_id;vState)
    $webpage:=$webpage+$buffer
Until ((vState=0) | ($error#0)) ' ホストが接続をクローズするか、エラーが発生するまで
```

参照

「TCP_Send」コマンド

TCP_State (tcp_ID; ステータスコード) 整数

引数	タイプ	説明
tcp_ID	倍長整数	オープンした TCP セッションのリファレンス
ステータスコード	整数	TCP ステータスコード
関数の返す値	整数	エラーコード

説明

TCP_State コマンドは、特定の TCP 接続のステータスを返します。

引数 < tcp_ID > は、オープンしたセッションを参照するための倍長整数値です。この値は、TCP_Open コマンドまたは TCP_Listen コマンドのどちらかを使って設定されます。

引数 < ステータスコード > は、整数タイプの変数が返されます。これは、下記のステータスコードの 1 つに対応します。

- 0 接続がクローズされた。
- 2 入ってくる接続の待機中。
- 8 接続が設定された。

例

次の例は、有効な TCP 接続がすでに設定され、変数「\$tcp_id」に代入された倍長整数の値で識別されていることを前提としています。この例では、情報ページを要求している web サーバーに任意のコマンドが送られ、ループに入ってその結果を受け取ります。web サーバーは自動的に接続をクローズするので、いったんこの動作が実行されると、この例ではその接続が終了またはエラーが発生するまで受信続けます。

```
C_LONGINT ($tcp_id)
C_INTEGER (vState;$err)
C_TEXT ($command;$buffer;$response)
If (TCP_Send ($tcp_id;$command)=0)
  vState:=0
  Repeat
    $err:=TCP_Receive ($tcp_id;$buffer)
    $err:=TCP_State ($tcp_id;vState)
    $response:=$response+$buffer
  Until ((vState=0) | ($err#0))
End if
```

参照

「TCP_Listen」コマンド、「TCP_Open」コマンド

TCP_Close

IT TCP/IP
バージョン 6.5

TCP_Close (tcp_ID) 整数

引数	タイプ	説明
tcp_ID	倍長整数	オープンした TCP セッションのリファレンス 0= セッションが正常にクローズした場合
関数の返す値	整数	エラーコード

説明

TCP_Close コマンドは、引数 < tcp_ID > で参照された TCP セッションをクローズします。TCP セッションがクローズされていない場合、TCP セッションで利用可能な 64 のリファレンスの 1 つを占有します。オープンしている 64 のセッションがクローズされていないと、ユーザは別のセッションをオープンすることはできなくなります。

引数 < tcp_ID > は、オープンしたセッションを参照するための倍長整数値です。この値は、TCP_Open コマンドまたは TCP_Listen コマンドのどちらかを使って設定されます。このコマンドは、セッションが正常にクローズした場合は、引数 < tcp_ID > に 0 が返されます。

参照

「TCP_Listen」コマンド、「TCP_Open」コマンド

7

IT Internet

この章のコマンドは、インターネット上で共通のタスクを実行するために使用されます。この章に含まれるコマンドは、任意マシンを “Ping” および “Finger” するためのもので、タイムサーバーから時刻を入手したり、ドメイン名や IP アドレスを分解したり、あるいはドメイン名や IP アドレスを整数タイプの値に変換します。これらのコマンドは、しばしば他の 4D Internet Commands と一緒に使用されます。

NET_Finger (ホスト名 ; 検索テキスト ; 結果) 整数

引数	タイプ	説明
ホスト名	文字列	ホスト名または IP アドレス
検索テキスト	文字列	検索文字列
結果	テキスト	検索結果
関数の返す値	整数	エラーコード

説明

NET_Finger コマンドは、検索するマシンの IP アドレスとマシン上のユーザアカウントの名前を指定して、引数 < 結果 > の中にその検索結果を返します。UNIX の Finger コマンドは、“.plan ” および “.project ” ファイルの内部で設定するためにユーザが選択する追加情報だけでなく、ユーザが最後にログインした時刻に関する情報も返すように設計されています。

Finger 検索に対して 2 つの異なる方法を指定することができます。Finger 検索は、ユーザマシンで直接実行できます。例えば、“ acius.com ” の “ johnt ” に関する情報を入手したい場合は、次のような検索を実行することができます :

```
$error:=NET_Finger ("www.acius.com";"johnt";$fingertext)
```

また同じ Finger 検索を間接的に実行することもできます。間接検索はあなたのクエリを実行する finger コマンドをサポートするリモートサーバーに質問します。例えば、次のコードはユーザ “ johnt@acius.com ” のリモート finger クエリを実行するためにドメイン名 “ acius.com ” によって識別されるマシンに質問します。

```
$error:=NET_Finger ("www.acius.com";"johnt@acius.com";$fingertext)
```

上記の両ケースで返されるメイン情報は同じものでなければなりません、返される検索結果の中には多少異なるものがあるかもしれません。異なるマシンでは、Finger コマンドを実行する際に設定されるオプションが異なり、その検索結果も多少異なるかもしれません。また、直接 Finger コマンドと追加ラインフィードをしばしば含む検索を行う間接 Finger コマンドでの結果は異なる書式でフォーマットされるかもしれません。

引数 < ホスト名 > は、ユーザマシンのホスト名または IP アドレスです。引数 < 検索文字列 > で識別されるマシンは、アカウントを持っています。

引数 < 検索テキスト > は、任意の finger サーバー上を検索するテキストか、あるいはマシン名または IP アドレスのいずれかです。 < 検索テキスト > がユーザ名の場合、このコマンドは < 検索テキスト > 用のサーバー上にあるユーザ名のディレクトリを通して検索します。 < 検索テキスト > がマシン名や IP アドレスの場合、このコマンドは指定したマシンの < ホスト名 > 内にある finger サーバーを通して finger リクエストを送信します。

引数 < 結果 > には、検索結果を含んだテキストが返されます。

NET_Ping

IT Internet
バージョン 6.5

NET_Ping (ホスト名 ; テキスト ; アクティブ { ; タイムアウト }) 整数

引数	タイプ	説明
ホスト名	文字列	ホスト名または IP アドレス
テキスト	テキスト	ピング中に送信するテキスト
アクティブ	整数	1 = アクティブ、2 = タイムアウト / 非アクティブ
タイムアウト	整数	待機する秒数、0 = IT_SetTimeout コマンドを使用
関数の返す値	整数	エラーコード

説明

NET_Ping コマンドは、IP アドレスが現在アクティブになっているかどうかをチェックするためにリモートの IP アドレスを照会するためのメカニズムを提供します。ピングされたマシンが 2 つのサイトが機能している TCP/IP プロトコルとネットワークの間で現在動作していれば、“アクティブ”のステータスが返されるはずです。一般的にピングされたマシンはユーザにアクティブになっていることを通知しません。そのマシンがピング (ICMP エコー) リクエストに応答できたと推測します。

NET_Ping コマンドは、ホスト名または IP アドレスのどちらかで指定されたマシンをピングします。ネットワーク経由でアクセス可能な IP アドレスを持つマシンは、ピングされます。これは、エンドユーザマシンも含まれます。[“ファイアオール” として知られているセキュリティシステムは、その保護機能を使って、マシンがピングされることを防ぎます。]

引数<ホスト名>は、ピングするホスト名または IP アドレスです。

引数<テキスト>は、ピング中に送信するテキストです。この引数は、Ping コマンドが実行される際に送信中の TCP パケットのサイズに影響を与えるためだけに存在します。

引数<アクティブ>には、ピングしたマシンのステータスを表す整数値が返されます。<アクティブ>に 1 が返された場合は、マシンはアクティブ状態です。0 が返された場合は、マシンが非アクティブ状態か、あるいは応答を受信する前にピングがタイムアウトになったかのどちらかです。

引数<タイムアウト>は、ピングを待っている秒数を指定します。この引数はオプションであり、もし省略すると、デフォルトとして 0 がセットされます。この引数が 0 の場合、応答が IT_SetTimeout コマンドで指定された秒数で受信されないと、このコマンドはタイムアウトを行います。

参照

「IT_SetTimeout」コマンド

NET_Time (ホスト名 ; 日付 ; 時刻 ; オフセット) 整数

引数	タイプ	説明
ホスト名	文字列	ホスト名または IP アドレス
日付	日付	日付
時刻	倍長整数	午前 0 時から経過した秒数で表された時刻
オフセット	整数	オフセット時間
関数の返す値	整数	エラーコード

説明

NET_Time コマンドは、インターネットタイムサーバーのホスト名または IP アドレスを指定し、マシンから現在の日付と時刻を取得して、ユーザのローカル時刻を変換するために必要なオフセットに適用します。

注意 : このコマンドは、コンピュータのインターネット時計に何の影響も及ぼしません。

引数<ホスト名>は、インターネットタイムサーバーのホスト名または IP アドレスです。

引数<日付>には、4th Dimention の日付が返されます。この日付は引数<オフセット>が適用された後の日付です。

引数<時刻>には、倍長整数の値が返されます。この値は、引数<オフセット>が適用された後の時刻です。また、この値は<日付>の午前 0 時から経過した秒数で表します。次ページにあるこの値を 4th Dimention の時刻変数に変換するメソッドの例を参照してください。

引数<オフセット>は、元となる時刻の値から加算または減算する時間数です。インターネットタイムサーバーは、世界標準時刻（グリニッジ時刻）でその値を表します。このタイムサーバーがユーザの場所と地理的に同じ場合にあっても、ユーザのローカル時刻と世界標準時刻の間の誤差を修正するためにこの<オフセット>を提供したくなるでしょう。

例

次の例は、“apple.com”のタイムサーバーから世界標準時刻を入手します。そして、このコマンドはオフセットとして設定した7時間を差し引いた日付と時刻の値を返します。(時刻は倍長整数の値で表されます。この値は下記のように4th DimensionのTime String関数を使って変換されます。)

C_DATE (vNetDate)

C_LONGINT (vNetTime)

C_TIME (vTime)

C_INTEGER (vOffset)

If (ERRCHECK ("NET_Time";**NET_Time** ("www.apple.com"; vNetDate; vNetTime; -7)))

 vTime:=**Time** (**Time string** (vNetTime)) ‘ 倍長整数の時刻を4th Dimensionの時刻に変換する

End if

参照

Time string 関数 (4th Dimension コマンド)

NET_NameToAddr (ホスト名 ;IP_ 倍長整数) 整数

引数	タイプ	説明
ホスト名	文字列	ホスト名または IP アドレス
IP_ 倍長整数	倍長整数	アドレスを参照する倍長整数の値
関数の返す値	整数	エラーコード

説明

NET_NameToAddr コマンドは、ホスト名または IP アドレスを指定し、そのアドレスを参照する一意の倍長整数の値を返します。

引数<ホスト名>は、ホスト名または IP アドレスです。

引数< IP_ 倍長整数>は、<ホスト名>内で指定された IP アドレスを表す倍長整数の値です。すべての IP アドレスの文字列は、符号化された倍長整数の値に変換されます。

< IP_ 倍長整数>の値は有効な使用法を持っているわけではありませんが、開発者の中にはデータストレージの面からこのコマンドが IP アドレスをよりコンパクトな倍長整数フォーマットに変換するのに効果的であることに気付くかもしれません。

参照

「NET_AddToName」コマンド

NET_AddToName

IT Internet
バージョン 6.5

NET_AddToName (IP_ 倍長整数 ; ホスト名 ; IP_ アドレス) 整数

引数	タイプ	説明
IP_ 倍長整数	倍長整数	アドレスを参照する倍長整数の値
ホスト名	文字列	ホスト名または IP アドレス
IP_ アドレス	文字列	IP アドレス
関数の返す値	整数	エラーコード

説明

NET_AddToName コマンドは、ホスト名を参照する倍長整数の値を指定し、そのホストの名前と IP アドレスを返します。

引数 < IP_ 倍長整数 > は、IP アドレスを参照する倍長整数の値です。

引数 < ホスト名 > には、ホストの名前を含んだ文字列が返されます。ホスト名が分解されていない場合、< IP_ 倍長整数 > は空の文字列を返しますが、エラーは何も返しません。

引数 < IP_ アドレス > には、IP アドレスを含んだ文字列が返されます。

参照

「NET_NameToAddr」コマンド

NET_Resolve (ホスト名 ;IP_ ホスト) 整数

引数	タイプ	説明
ホスト名	文字列	ホスト名または IP アドレス
IP_ ホスト	文字列	< ホスト名 > で指定された値の反対を返す
関数の返す値	整数	エラーコード

説明

NET_Resolve コマンドは、1 番目の引数でホスト名を指定し、2 番目の引数に IP アドレスを返します。1 番目の引数である< ホスト名 > に IP アドレスが渡されると、2 番目の引数である< IP_ ホスト > には、そのマシン用に登録されたホスト名が返されます。

引数< ホスト名 > は、IP アドレスまたはホスト名のどちらかを含んだ文字列です。

引数< IP_ ホスト > は、1 番目の引数がホスト名を含んでいると、この引数にはホストの IP アドレスが返されます。逆に 1 番目の引数が IP アドレスを含んでいると、この引数にはホスト名が返されます。

例

次の例は、IP アドレスを取得するために NET_Resolve コマンドの 1 番目の引数にホスト名 “www.netcom.com” を渡します。もう 1 つの例は、登録されたホスト名を取得するために 1 番目の引数に IP アドレスを渡します。

```
C_BOOLEAN ($ERR)
C_STRING (80;$Resolved) ' 任意のサイズを持つ文字列またはテキストの値
$ERR:=ERRCHECK ("NET_Resolve ";NET_Resolve ("www.netcom.com";$Resolved))
'$Resolved は、IP アドレス '192.100.81.100' を返す。
$ERR:=ERRCHECK ("NET_Resolve ";NET_Resolve ($Resolved;$Resolved))
'$Resolved は、ホスト名 'www.netcom.com' を返す。
```

8 IT Utilities

この章のコマンドは、4D Internet Commands の他の章で有効なユーティリティをたくさん提供します。これらのコマンドの大部分は、開発者がユーザマシンの操作環境やソフトウェアのバージョンおよびマシンの状態や IP アドレスを決定する際の手助けをします。

またこの章の別のコマンドは、開発者がエラーコードを解釈したり、ファイルのエンコードとデコードの手助けを行い、そしてすべての章にあるコマンドの大部分でデフォルトのタイムアウト値に影響を与えます。

IT_MacTCPInit 整数

引数	タイプ	説明
----	-----	----

このコマンドには、引数はありません。

関数の返す値	整数	エラーコード
--------	----	--------

説明

IT_MacTCPInit コマンドは、4D Internet Commands と一緒に使用する TCP ドライバを開きます。このコマンドは、1 つのファンクションとして機能し、TCP ドライバが開かれていない場合に整数値のエラーを返します。

このコマンドは、4th Dimension データベースの On Startup Database メソッド内で使用することをお勧めします。このコマンドは、1 回しか実行されません。PPP や SLIP と同じように TCP で利用されるダイアルアップ方式のユーザは、TCP 接続がこのコマンドの使用を必要とするまではこのコマンドの使用を延期したくなるかもしれません。なぜなら、このコマンドが 4D Internet Commands にユーザのダイアルアップ接続を開いて初期化させるからです。

注意：このコマンドは、どの 4D Internet Commands のコマンドが実行される前に実行されるべきです。

IT_Platform 整数

引数	タイプ	説明
----	-----	----

このコマンドには、引数はありません。

関数の返す値	整数	プラットフォームタイプ (0=68K コード、1=PPC コード、2=Windows)
--------	----	---

説明

IT_Platform 関数は、4D Internet Commands のコードが現在どのプラットフォームで実行されているかを示す整数値を返します。この関数は、68K コードで実行している場合は 0 を返します。同様に、PPC ネイティブコードで実行している場合は 1、Windows 上で実行している場合は 2 を返します。

例

```
C_BOOLEAN (<>ITnative)
<>ITnative := (IT_Platform =1)
```

IT_Version 文字列

引数	タイプ	説明
----	-----	----

このコマンドには、引数はありません。

関数の返す値	文字列	バージョン文字列
--------	-----	----------

説明

IT_Version 関数は、4D Internet Commands のバージョン番号を示す文字列値を返します。

例

次の例は、現在使用中の 4D Internet Commands のバージョンが何なのかを示す「アラート」ダイアログボックスを表示します。

ALERT ("4D Internet Commands バージョン : "+IT_Version)

IT_TCPversion (スタックタイプ ; スタックバージョン) 整数

引数	タイプ	説明
スタックタイプ	整数	0= なし、1=MacTCP、2=Open Transport、3=WinSock
スタックバージョン	テキスト	TCP スタックのバージョン
関数の返す値	整数	エラーコード

説明

IT_TCPversion コマンドは、4D Internet Commands で現在使用中の TCP スタックタイプに関する情報を返します。このスタックタイプは、プラットフォームによって異なります。Macintosh では、MacTCP と Open Transport の両方をサポートしています。一方、Windows では WinSock TCP スタックがサポートされています。

引数 <スタックタイプ> は、現在使用中の TCP スタックタイプを表す整数値を返します。次の表は、TCP スタックでサポートされているタイプを返す整数値を示しています；

コード TCP スタック

0	なし
1	MacTCP
2	Open Transport
3	WinSock

引数 <スタックバージョン> は、現在使用中の TCP スタックのバージョン番号を表すテキスト値を返し、引数 <スタックタイプ> によって特定されます。

IT_MacTCPver (バージョンコード) 整数

引数	タイプ	説明
バージョンコード	整数	インストールされた MacTCP のバージョンコード
関数の返す値	整数	エラーコード

説明

IT_MacTCPver コマンドは、MacTCP、Open Transport、WinSock のバージョン情報を調べることができる IT_TCPversion コマンドの登場により廃止されました。

このコマンドは、ユーザのコンピュータ上にインストールされている MacTCP のバージョンを示す第 1 引数の中に整数値を返します。返された整数は、Apple Gestalt Manager から生成され、特定の MacTCP バージョンに直接対応してはなりません。このコマンドは、あるユーザが他の多くのコマンドにアクセスするために必要最低限のソフトウェアで実行してるかどうかを調べるのに有効です。

引数 <バージョンコード> - Gestalt Manager は下記の MacTCP バージョンに対応する次の値を返します。アップル社がオペレーティングシステムや MacTCP をアップグレードする度に、返される値が追加されます。

バージョンコード

0=MacTCP ドライバはオープンしていない

1=バージョン 1.1

2=バージョン 1.1,1

3=バージョン 2.0

このコマンドの最も有効な使用法は、4D Internet Commands 内で他のコマンドを使用するために必要なソフトウェア (MacTCP) を持っているかどうかをチェックするユーザシステムのテストに用いる場合です。

例

次の関数は、ユーザが正規の MacTCP バージョンをインストールしているかどうかを示すインタープロセス変数に True または False をセットする Startup で呼び出されます。

```
'メソッド: VERIFY_TCP ( {> バージョン} ): ブール [MacTCP インストール済]

C_INTEGER ($ERR)
C_BOOLEAN ($TCP_OK)
C_LONGINT (vTCPversion)
$ERR:=IT_MacTCPVer (vTCPversion)
$TCP_OK:=((($ERR=0)&(vTCPversion#0)) 'TRUE = ( エラーなし & MacTCP の正規バージョン )

Case of
  ¥(Not ($TCP_OK) & (<>SHOWERRORS)) ' 何らかのエラー
    ALERT ("MacTCP をオープンすることができません。インストールした MacTCP を調べて下さい。")
  ¥(($TCP_OK) & (Count parameters =1)) ' すべて OK の場合、バージョンを返す。
    LIST TO ARRAY ("MacTCP Versions";$aTCPversion)
    If (vTCPversion<=Size of array ($aTCPversion))
      $1>:=$aTCPversion{vTCPversion}
    Else
      $1>:=" 不明 "
    End if ' (vTCPversion<=Size of array($aTCPversion))
End case
$0:=$TCP_OK ' バージョンチェックが OK の場合は、True を返す。
```

IT_MyTCPAddr (IP アドレス ; サブネット) 整数

引数	タイプ	説明
IP アドレス	文字列	ユーザマシンの IP アドレス
サブネット	文字列	IP 形式のサブネットマスク
関数の返す値	整数	エラーコード

説明

IT_MyTCPAddr コマンドは、このコマンドを実行するマシンの IP アドレスを返します。

引数 < IP アドレス > は、IP アドレスを含んだ文字列です。

引数 < サブネット > は、IP アドレスのサブネットマスクを含んだ文字列です。

IT_SetTimeout (タイムアウト) 整数

引数	タイプ	説明
タイムアウト	整数	0 から 127 秒までのタイムアウト時間
関数の返す値	整数	エラーコード

説明

IT_SetTimeout コマンドは、タイムアウト時間を整数値の秒でセットします。この値は、0 から 127 秒までで指定します。

引数 <タイムアウト> は、秒単位で現在セットされているタイムアウト時間です。次のコマンドは、この IT_SetTimeout コマンドによって影響を受けます：

TCP_Open
FTP_Login
FTP_Send
FTP_Receive
SMTP_QuickSend
SMTP_Send
POP3_Login
POP3_BoxInfo
POP3_Delete
POP3_Reset
POP3_MsgInfo
POP3_MsgLstInfo
POP3_GetMessage
POP3_MsgLst
POP3_Download
POP3_VerifyID
POP3_UIDToNum
NET_Finger
NET_Ping
NET_Time

注意： TCP_Listen コマンドの引数 <タイムアウト> に 0 をセットすると、無期限に接続を待機していません。このコマンド使用後は、タイムアウトに別の値を再設定することを忘れないでください。そうしないと、0 は通常デフォルトとして用いられることを意味します。また、タイムアウトの値は “TCP/IP タイムアウト” および “ 応答タイムアウトまでの待機用 ” に使用されます。タイムアウトに 0 をセットした場合、これは任意の応答に対する待機時間としては決して十分な時間とは言えません。

参照 w

「IT_GetTimeOut」コマンド

IT_GetTimeOut (タイムアウト) 整数

引数	タイプ	説明
タイムアウト	整数	タイムアウト秒
関数の返す値	整数	エラーコード

説明

IT_GetTimeOut コマンドは、前ページの IT_SetTimeOut コマンドの節でリストされたコマンド群の中で現在セットされているタイムアウトの値を返します。

引数 <タイムアウト> は、秒単位で現在セットされているタイムアウト時間です。

参照

「IT_SetTimeOut」コマンド

IT_ErrorText (エラー) 文字列

引数	タイプ	説明
エラー	整数	別のコマンドから返されるエラーコード
関数の返す値	文字列	エラーメッセージ

説明

IT_ErrorText コマンドは、唯一の引数として整数タイプのエラーコードを取得し、そのエラーの内容を文字列 / テキストで返します。このコマンドが、4D Internet Commands の中で関数の返す値が整数タイプでない数少ないコマンドの 1 つであることを覚えておいてください。

引数 <エラー> は、整数タイプのエラーコードです。

例

次はエラー原因を説明する警告メッセージを表示する ErrorCheck メソッドの例です。

```
'メソッド : ERRCHECK (" コマンド名 "; エラーコード ) -> True/False
C_TEXT (vErrorMsg)
$Command:=$1
$error:=$2
$Result:=True
If ($Error#0)
    $Result:=False
    vErrorMsg:=IT_ErrorText ($Error)
    ALERT (" エラー -- "+Char (13)+" コマンド : "+$Command+Char (13)+" エラーコード : "
        +String ($Error)+Char (13)+" エラーメッセージ "+vErrorMsg)
End if
$0:=$Result
```

IT_Encode (ファイル名 ; エンコードファイル ; エンコードモード) 整数

引数	タイプ	説明
ファイル名	テキスト	ファイルのローカルパス
エンコードファイル	テキスト	ローカルパスファイル 返されるエンコードファイルのパス
エンコードモード	整数	1 = BinHex 2 = Base64 (データフォークのみ) 3 = AppleSingle 4 = AppleDouble 5 = AppleSingle と Base64 6 = AppleDouble と Base64 7 = UUEncode 8 = MacBinary
関数の返す値	整数	エラーコード

説明

IT_Encode コマンドは、指定された引数<エンコードモード>を使って、任意のファイルをエンコードします。指定されたファイルは変更されずにエンコードされたコピーが作成されます。作成されるエンコードファイルの名前は、元のファイル名の後にエンコード形式を指定するために追加された接尾辞を付加した名前です。例えば、BinHex でエンコードした場合は、接尾辞として「.bhx」が追加されます。同様に Base64 でエンコードした場合は、接尾辞として「.b64」、AppleSingle の場合は「.as」が追加されます。

引数<ファイル名>は、エンコードしたいファイルの完全なパスを取得します。この引数に空の文字列が渡された場合は、ユーザはファイルを選択するためのダイアログを使って指定することになります。

引数<エンコードファイル>には、エンコードファイルの名前と場所を規定する完全なローカルパスファイルが渡されます。指定しないと、IT_Encode コマンドはそのドキュメントに対して独自の名前を付け、データベースストラクチャと同じディレクトリに配置します。指定の有無に関係なく、エンコードされたドキュメントの最終的なパス名は、この引数内に返されます。指定したディレクトリの中でファイル名のコンフリクト（衝突）を起こすかもしれないので、このコマンドの中に渡されたオリジナルの値ではなく、エンコードされたファイルの本当のリファレンスとして返された値の方を常に信用するようにしてください。

注意：接尾辞を付加した元のファイル名の長さが 31 バイト以上の場合、このコマンドは新しいファイルの名前が Macintosh ファイルの 31 バイト制限に引っ掛からないように元のファイル名を切り取ります。

引数<エンコードモード>は、ファイルに用いるエンコード形式を指定します。デフォルトの値は、BinHex エンコード形式を表す 1 です。下記にエンコード形式を示します：

コード エンコード形式

- | | |
|---|----------------------|
| 1 | BinHex |
| 2 | Base64 (データフォークのみ) |
| 3 | AppleSingle |
| 4 | AppleDouble |
| 5 | AppleSingle と Base64 |
| 6 | AppleDouble と Base64 |
| 7 | UUEncode |
| 8 | MacBinary |

<エンコードモード>の“4”と“6”の AppleDouble を使ってエンコードすると、“%ファイル名”と“ファイル名”と名付けられた 2 つのファイルが作成されます。

参照

「IT_Decode」コマンド

IT_Decode (ファイル名 ; デコードファイル ; デコードモード) 整数

引数	タイプ	説明
ファイル名	テキスト	ファイルのローカルパス
デコードファイル	テキスト	ローカルパスファイル 返されるデコードファイルのパス
デコードモード	整数	1 = BinHex 2 = Base64 (データフォークのみ) 3 = AppleSingle 4 = AppleDouble 5 = AppleSingle と Base64 6 = AppleDouble と Base64 7 = UUEncode 8 = MacBinary
関数の返す値	整数	エラーコード

説明

IT_Decode コマンドは、指定された引数<デコードモード>を使って、任意のファイルをデコードします。指定されたファイルは変更されずにデコードされたコピーが作成されます。

引数<ファイル名>は、デコードしたいファイルの完全なパスを取得します。この引数に空の文字列が渡された場合は、ユーザはファイルを選択するためのダイアログを使って指定することになります。

引数<デコードファイル>には、デコードファイルの名前と場所を規定する完全なローカルパスファイルが渡されます。指定しないと、IT_Decode コマンドはそのドキュメントに対して独自の名前を付け、データベースストラクチャと同じディレクトリに配置されます。指定の有無に関係なく、デコードされたドキュメントの最終的なパス名は、この引数内に返されます。

引数<デコードモード>は、ファイルに用いるデコード形式を指定します。デフォルトの値は、BinHex デコード形式を表す 1 です。下記にデコード形式を示します：

コード デコード形式

- 1 BinHex
- 2 Base64 (データフォークのみ)
- 3 AppleSingle
- 4 AppleDouble
- 5 AppleSingle と Base64
- 6 AppleDouble と Base64
- 7 UUEncode
- 8 MacBinary

<デコードモード>の“4”と“6”の AppleDouble を使ってデコードすると、“%ファイル名”と“ファイル名”と名付けられた2つのファイルが作成されます。

参照

「IT_Encode」コマンド

IT_GetPort

IT Utilities
バージョン 6.5

IT_GetPort (プロトコル ; ポート) 整数

引数	タイプ	説明
プロトコル	整数	1 = FTP、2 = SMTP、3 = POP3
ポート	整数	ポート番号
関数の返す値	整数	エラーコード

説明

IT_GetPort コマンドは、引数<プロトコル>で指定したプロトコルを取得し、そのプロトコルに関係する 4D Internet Commands で使用されている現在のポート番号を返します。

参照

「IT_SetPort」コマンド

IT_SetPort

IT Utilities
バージョン 6.5

IT_SetPort (プロトコル ; ポート) 整数

引数	タイプ	説明
プロトコル	整数	1 = FTP、2 = SMTP、3 = POP3
ポート	整数	ポート番号
関数の返す値	整数	エラーコード

説明

IT_SetPort コマンドは、引数<プロトコル>で指定したプロトコルを取得し、そのプロトコルの今後のすべてのやり取りを引数<ポート>で指定したポート番号に指示します。

参照

「IT_GetPort」コマンド

Case 文を介したコマンドの実行

このマニュアルや例題データベースの中で紹介されているプログラミングの例は、多くの開発者にはあまり馴染みのないものばかりかもしれません。この例題のほとんどは、4th Dimention の Case 文の中で改造された形で一連のコマンド群を実行しています。

4D Internet Commands 内のほとんどのコマンドは、コマンドの完全なシーケンスが正常終了するように実行されることを前提としています。そのため、コマンドシーケンス内のある部分でエラーが起こると、そのパスで処理されているプロセスを中止しなければならなくなり、そのせいで If 文の階層がだんだんと深くなり、作業を困難にしていきます：

```
If (SMTP_New ($smtp_id)=0)
  If (SMTP_Host ($smtp_id;<>ref_Server)=0)
    If (SMTP_From ($smtp_id;vFrom)=0)
      If (SMTP_To ($smtp_id;vTo)=0)
        さらにさらに階層が深くなっていく ...
      End if
    End if
  End if
End if
```

この問題を解決する 1 つの方法として、4th Dimention の Case 文を用いる方法があります。Case 文の各項目は、その返す値が True か False かどうかを調べるために 4th Dimention によって実行されます。Case 文のすべての要素が False を返した場合は、その Case 文チェックの各要素がそれぞれ実行されます。上記で説明したことは、次のコードで実行することができます：

```
$SentOK:=False ‘すべての呼び出しを通して処理が実行されたどうかを示すフラグ
Case of
  ¥(SMTP_New ($smtp_id)#0)
  ¥(SMTP_Host ($smtp_id,pref_Server)#0)
  ¥(SMTP_From ($smtp_id;vFrom)#0)
  ¥(SMTP_To ($smtp_id;vTo)#0)
  ¥(SMTP_Subject ($smtp_id;vSubject)#0)
  ¥(SMTP_Body ($smtp_id;vMessage)#0)
  ¥(SMTP_Send ($smtp_id)#0)
Else
  $SentOK:=True ‘メッセージが作成されて、正常にメールが送られた場合
End case
If ($smtp_id#0) ‘メッセージが作成された場合は、すぐにそのメッセージをクリアする必要がある
  $OK:=SMTP_Clear ($smtp_id)
End if
```

前ページの例では、すべての 4D Internet Commands は正常に終了すると、エラーコード 0 を返します。4th Dimension の各 Case 文を評価するには、その返される値を取得するために個々の呼び出しを実際に行う必要があります。各 Case 文の項目は 0 でないリターン結果を比較するので、4th Dimension はそのコマンドの 1 つがエラーになるまでどの項目が中止されたか気付きません。各コマンドが正常に実行されると、4th Dimension はメッセージが作成されて正常に送信されたことを示す “\$SentOK” フラグがセットされる Else 条件を実行するために下の処理を行います。

POP3 メール の自動返信機能について

使用しているデータベースの中にそのデータベースが受け取ったメールをユーザが “返信 (リプライ)” できるメールシステムを組み込む予定がある場合、その返信メッセージのフィールドに埋め込む方法としていくつかの提案があります。次の提案は、RFC822 を元に説明されています：

“送信人 (Sender)” フィールドに一覧表示されているアドレスは、先頭メッセージの配信中に起こった問題の通知を受け取らなければなりません。もし “送信人” フィールドに何も入っていない場合は、通知は “差出人 (From)” フィールドに一覧表示されているアドレスに返信する必要があります。“送信人 (Sender)” メールアドレスは、単にメール配信中の問題に関係している返信を送信するだけで、メッセージのトピックに関係する返信は送信しません。

“送信人 (Sender)” アドレスは、メッセージの自動リプライに絶対に使用してはいけません。その代わり、そのメッセージは下記の状況によって “返信 (Reply-To)” フィールドまたは “差出人 (From)” フィールドのどちらかで使用できます。

“返信 (Reply-To)” フィールドが存在し 1 つまたは複数のメールアドレスを含んでいる場合は、任意の返信はそのリスト内の人に直接送られます。“返信 (Reply-To)” ヘッダ内のアドレスは “差出人 (From)” ヘッダ内にリスト表示されているアドレスを上書きします。ただし、“返信 (Reply-To)” フィールドが存在せずに “差出人 (From)” フィールドが存在する場合は、返信は “差出人 (From)” ヘッダ内に示されたメールボックスに送信されます。

これらの提案は、“返信 (リプライ)” のような動作のメールアドレス処理がプログラムで処理される場合の一例に過ぎません。いったん、返信メッセージが作成されると、エンドユーザはそのメッセージを送信する前にこれらのデフォルトをいつでも上書きすることができます。

ポート番号の選択方法

- ・ 0 から 1023 (一般的なポート): 一般的なポートは、I.A.N.A (Internet Assigned Numbers Authority) によって割り当てられ、多くのシステムではシステム (もしくはルート) プロセスが特権のあるユーザによって実行されるプログラムでのみ使用することが可能です。
 - 20, 21 FTP;
 - 23 TELNET;
 - 25 SMTP
 - 37 NTP;
 - 80, 8080 HTTP;
- ・ 1024 から 49151 (予約ポート): 予約ポートは I.A.N.A によって指定され、多くのシステムでは、通常のユーザプロセスや通常のユーザ (ルーター、特定のアプリケーション ...) によって使用することが可能です。
- ・ 49512 から 65535 (ダイナミック / プライベートポート): ダイナミック / プライベートポートは、使用可能です。

TCP/IP コマンドを使ってデータベースの同期をとりたい場合は、49151 より上のポート番号を使用しなければなりません。

さらに詳しい情報については、I.A.N.A. の Web サイト (<http://www.iana.org>) をご覧ください。

TCP ポート番号

daytime	13	Daytime
qotd	17	Quote of the Day
ftp-data	20	File Transfer [Default Data]
ftp	21	File Transfer [Control]
telnet	23	Telnet
smtp	25	Simple Mail Transfer
time	37	Time
nicname	43	Who Is
domain	53	Domain Name Server
sql*net	66	Oracle SQL*NET
gopher	70	Gopher
finger	79	Finger
http	80	World Wide Web HTTP
poppassd	106	Password Server
rtelnet	107	Remote Telnet Service
pop2	109	Post Office Protocol - Version 2
pop3	110	Post Office Protocol - Version 3
sunrpc	111	SUN Remote Procedure Call
auth	113	Authentication Service
sftp	115	Simple File Transfer Protocol
sqlserv	118	SQL Services

nntp	119	Network News Transfer Protocol
ntp	123	Network Time Protocol
pwdgen	129	Password Generator Protocol
imap2	143	Interactive Mail Access Protocol v2
news	144	NewS
sql-net	150	SQL-NET
multiplex	171	Network Innovations Multiplex
cl/1	172	Network Innovations CL/1
at-rtmp	201	AppleTalk Routing Maintenance
at-nbp	202	AppleTalk Name Binding
at-3	203	AppleTalk Unused
at-echo	204	AppleTalk Echo
at-5	205	AppleTalk Unused
at-zis	206	AppleTalk Zone Information
at-7	207	AppleTalk Unused
at-8	208	AppleTalk Unused
ipx	213	IPX
netware-ip	396	Novell Netware over IP
timbuktu	407	Timbuktu
conferenc	531	chat
netnews	532	readnews
netwall	533	for emergency broadcasts
uucp	540	uucpd
uucp-rlogin	541	uucp-rlogin
whoami	565	whoami
ipcserver	600	Sun IPC server
phonebook	767	phone
accessbyilder	888	AccessBuilder

付録 C エラーコード

付録

バージョン 6.5

IT_ErrorText コマンドと IT_Version コマンドを除くその他の 4D Internet Commands は、その実行結果として整数値を返します。この整数値は、コマンドが 4th Dimension データベースに通知したいエラーコードです。任意のコマンドが正常に実行されると、0 が返されます。エラーコードが示す内容は、そのエラーが発生した値の範囲によって決定されます。次の表は、エラーが生成される範囲を示しています：

エラーコード	エラー生成源
エラー < 0	オペレーションシステムエラー
0	エラーなし
エラー > 10000	4D Internet Commands エラー

4D Internet Commands エラーコード

ある操作中にエラーが発生すると、次の表の数値が返されます：

10000	ユーザがダイアログボックス操作または処理を取り消しました。
10001	実行不可能なインターネットコマンドです。
10002	配列タイプが無効です。
10003	使用可能なリファレンス (TCP、SMTP、POP3 等) がありません。
10004	リファレンスが無効です。
10005	SMTP_Send コマンド内で用いるホストが必要です。
10006	SMTP_Send コマンド内で用いる差出人が必要です。
10007	SMTP_Send コマンド内で用いる受取人が必要です。
10008	すでにログインされています。
10009	POP3 接続を行うことはできません。
10010	POP3 USER エラー
10011	POP3 PASS エラー
10012	POP3 QUIT エラー
10013	POP3 STAT エラー
10014	POP3 LIST エラー
10015	POP3 UIDL エラー
10016	POP3 DELE エラー
10017	POP3 RSET エラー
10018	メッセージ番号が間違っています。
10019	文字オフセットが間違っています。
10020	文字数が間違っています。
10021	POP3 RETR エラー
10022	フィールドがメールヘッダ内で見つかりませんでした。
10023	添付ファイルが見つかりませんでした。
10024	BinHex 処理中にエラーが起きました。
10025	BinHex チェックサムエラー。
10026	インターネットコマンドを使用できません。おそらく MacTCP がインストールされていません。

10027 接続が存在しません。
10028 32K バイトを超えました。
10029 POP3 NOOP エラー
10030 POP3 セッションがサーバーによってクローズされました。
10031 POP3 APOP エラー
10032 レスポンス（応答）が不明または無効です。
10033 SMTP 421 - 転送チャネルをクローズしています。サービスを利用できません。
10034 SMTP 450 - リクエストしたメール処理は行われません：メールボックスを利用できません。
10035 SMTP 452 - リクエストした処理は中止されました：処理中にローカルエラーが起きました。
10036 SMTP 500 - リクエストした処理は行われません：システムストレージが足りません。
10037 SMTP 501 - シンタックスエラー。コマンドを認識できません。
10038 SMTP 502 - シンタックスエラー。パラメータ（引数）が正しくありません。
10039 SMTP 503 - コマンドのシーケンスが間違っています。
10041 SMTP 504 - コマンド引数をインプリメントできません。
10042 SMTP 550 - リクエストした処理は行われません：メールボックスを利用できません。
10043 SMTP 551 - ローカルユーザではありません。＜転送 - パス＞を試してみてください。
10044 SMTP 552 - リクエストしたメール処理は中止されました：ストレージ割り当てが足りません。
10045 SMTP 553 - リクエストした処理は行われません：メールボックス名が許可されていません。
10046 SMTP 554 - トランザクションに失敗しました。
10047 FTP 421 - コントロール接続をクローズしています。サービスを利用できません。
10048 FTP 425 - データ接続をオープンできません。
10049 FTP 426 - 接続がクローズしました。転送が中止されました。
10050 FTP 450 - リクエストしたファイル処理は行われません：ファイルを使用できません（例えば、ファイルビジー）。
10051 FTP 451 - リクエストした処理は中止されました：処理中にローカルエラーが起きました。
10052 FTP 452 - リクエストした処理は行われません：システム内のストレージ空間が足りません。
10053 FTP 500 - シンタックスエラー。コマンドを認識できません。
10054 FTP 501 - シンタックスエラー。パラメータ（引数）が正しくありません。
10055 FTP 502 - コマンドをインプリメントできません。
10056 FTP 503 - コマンドのシーケンスが間違っています。
10057 FTP 504 - コマンドがその引数をインプリメントできません。
10058 FTP 530 - ログインされていません。
10059 FTP 532 - ファイルを保存するためのアカウントが必要です。
10060 FTP 550 - リクエストした処理は行われません：ファイルを使用できません（例えば、ファイルが見つからない、アクセスがない等）。
10061 FTP 551 - リクエストした処理は中止されました：ページタイプが不明です。
10062 FTP 552 - リクエストしたファイル処理は行われません：（カレントディレクトリまたはデータセット用の）ストレージ割り当てが足りません。
10063 FTP 553 - リクエストした処理は行われません：ファイル名が許可されていません。
10064 指定したタイムアウト時間内に何の応答もありません。
10065 FTP ファイルではありません。
10066 Base64 を処理中にエラーが起きました。
10067 AppleSingle を処理中にエラーが起きました。
10068 引用符の付いたプリント中にエラーが起きました。
10069 FTP セッションがサーバーによってクローズされました。
10070 FTP ディレクトリではありません。
10071 TCP セッションがサーバーによってクローズされました。

10072 エンコードタイプが間違っています。
 10073 デコードタイプが間違っています。
 10074 非同期 DNR 呼び出しが正常に実行できませんでした。
 10075 非同期 OpenTransport 呼び出しが正常に実行できませんでした。
 10076 OpenTransport のバインドに失敗しました。
 10077 OpenTransport の接続に失敗しました。
 10078 MacTCP ストリームが最大に達しました。
 10079 uuencode を処理中にエラーが起きました。
 10080 ICMP ライブラリをロードできません。
 10081 Macbinary を処理中にエラーが起きました。
 10082 Macbinary チェックサムエラー
 10083 ファイルをオープンできませんでした。
 10084 受信した FTP 情報がありません。
 10085 受信した FTP 情報が不明です。

Open Transport エラーコード

-3211 Open Transport のメモリが足りません。
 -3201 Open Transport が見つかりません。
 -3216 Open Transport の複製が見つかりました。
 -3150 不正なアドレスが指定されました。
 -3151 不正なオプションが指定されました。
 -3152 アクセス許可がありません。
 -3153 不正のプロバイダリファレンスです。
 -3154 指定されたアドレスがありません。
 -3155 間違ったステータスで呼び出しが実行されました。
 -3156 指定シーケンスが存在しません。
 -3157 システムエラーが起きました。
 -3158 イベントが発生しました - LOOK() 呼び出し
 -3159 不正なデータ量が指定されました。
 -3160 受け渡したバッファの大きさが十分ではありません。
 -3161 プロバイダで制御できる範囲を超えました。
 -3162 読み込みデータがありません。
 -3163 使用可能な接続切断指示がありません。
 -3164 使用可能な単位データエラー指示がありません。
 -3165 不正なフラグ値が指定されました。
 -3166 使用可能な順序立ったリリース指示がありません。
 -3167 コマンドが指定されていません。
 -3168 ステータスを変更中です - 後で再度試してみてください。
 -3169 不正なストラクチャタイプが OTAlloc に対して要求されました。
 -3170 不正なエンドポイント名が指定されました。
 -3171 A Bind to an in-use addr with qlen > 0
 -3172 リクエストしたアドレスはすでに使用中です。
 -3173 接続待機の保留中のため受け入れに失敗しました。
 -3174 互換性のないエンドポイント上で受け入れを試みました。
 -3175 kOTResQLenErr
 -3176 kOTResAddressLenErr
 -3177 kOTQFullErr

-3178 明記されていないプロバイダエラーが起きました。
-3179 割り込み時の同期呼び出しです。
-3180 コマンドが取り消されました。
-3200 アクセス許可がありません。
-3201 そのようなファイルまたはディレクトリはありません。
-3202 そのようなリソースはありません。
-3203 システムサービスが中止されました。
-3204 I/O エラー
-3205 そのようなデバイスまたはアドレスはありません。
-3208 ファイル番号が間違っています。
-3210 後で再度操作を試みてください。
-3211 空きスペースが十分ではありません。
-3212 アクセス許可がありません。
-3213 アドレスが間違っています。
-3215 デバイスまたはリソースがビジーです。
-3216 ファイルが存在します。
-3218 そのようなデバイスははありません。
-3221 引数が間違っています。
-3224 キャラクタデバイスではありません。
-3231 パイプが壊れています。
-3233 ストリーム用のメッセージサイズが大きすぎます。
-3234 呼び出しが妨害されたため、中止されました。
-3236 kEALREADYErr
-3237 非ソケット上でのソケット操作です。
-3238 指定先のアドレスが必要です。
-3239 メッセージが長すぎます。
-3240 ソケット用のプロトコルが間違っています。
-3241 プロトコルが使用できません。
-3242 サポートされていないプロトコルです。
-3243 サポートされていないソケットタイプです。
-3244 ソケット上でサポートされていない操作です。
-3247 アドレスがすでに使用中です。
-3248 リクエストしたアドレスを割り当てることはできません。
-3249 ネットワークがダウンしています。
-3250 ネットワークがまだ届いていません。
-3251 ネットワークがリセット時に接続を切断しました。
-3252 ソフトウェアが接続中止を起こしました。
-3253 接続がピア to ピアネットワークによってリセットされました。
-3254 使用可能なバッファ空間がありません。
-3255 ソケットがすでに接続されています。
-3256 ソケットが接続されていません。
-3257 ソケットがシャットダウンした後は送信できません。
-3258 リファレンスが多すぎます：接続できません。
-3259 接続がタイムアウトしました。
-3260 接続が拒否されました。
-3263 ホストがダウンしています。
-3264 ホストのルートがありません。

-3269	kEPROTOErr
-3270	kETIMEErr
-3271	kENOSRErr
-3272	kEBADMSGErr
-3273	kECANCELErr
-3274	kENOSTRErr
-3275	kENODATAErr
-3276	kEINPROGRESSErr
-3277	kESRCHErr
-3278	kENOMSGErr
-3279	kOTClientNotInittedErr
-3280	kOTPortHasDiedErr
-3281	kOTPortWasEjectedErr
-3282	kOTBadConfigurationErr
-3283	kOTConfigurationChangedErr
-3284	kOTUserRequestedErr
-3285	kOTPortLostConnection

MacTCP エラーおよびその他のエラーコード

-33	ディスクに書き込むことはできません。ファイルディレクトリがいっぱいです。
-34	ディスクに書き込むことはできません。ディスクがいっぱいです。
-35	そのようなボリュームはありません。
-36	I/O エラー。
-37	ファイル名またはボリューム名が間違っています。
-38	読み込みまたは書き込み用のファイルをオープンできません。
-39	ファイルから読み込むことができません。ファイルの最後に達しました。
-42	たくさんのファイルがオープンされているため処理を続行できません。
-43	ファイルが見つかりません。
-44	ボリュームがハードウェアレベルでロックされました。
-45	ファイルがロックされています。
-46	ボリュームがソフトウェアレベルでロックされました。
-47	別のアプリケーションで 1 つまたは複数のファイルがオープンされています。
-48	この名前のファイルがすでに存在しています。
-49	ファイルが書き込み許可ありですすでにオープンしています。
-54	オープンしているファイルでアクセス許可エラーが起きました。
-57	Macintosh のボリューム名ではありません。
-59	ファイル名を変更しようとした際にエラーが起きました。
-61	書き込み許可エラー。
-108	メモリ量が十分ではありません。
-120	ディレクトリが見つかりません。
-23000	ローカルネットワークハンドラを初期化することができません。
-23001	間違って手動によるアドレス設定が行われました。
-23002	環境設定されたリソースが見つかりません。
-23003	MacTCP をロードするためのアプリケーションヒープに十分な空間がありません。
-23004	サーバーからのアドレス取得またはアドレスを取得する際のエラーがすでに他のマシンで使用されています。

- 23005 TCPClose コマンドがすでに発行されているため、この接続で送るデータはこれ以上ありません。
- 23006 WDS で表される合計データサイズが 0 か 65535 バイト以上のどちらかでした。
- 23007 TCP または UDP ストリームがすでにオープンしている接続を持っています。
- 23008 この TCP ストリームはオープンしている接続を持っていません。
- 23009 最大の TCP または UDP ストリームがすでにオープンしています。
- 23010 指定した TCP または UDP ストリームがオープンしていません。
- 23011 オープンしているストリームがすでにこの受信バッファエリアで使用されています。
- 23012 TCP 接続が切断されました。
- 23013 受信バッファエリアのポインターが 0 です。
- 23014 RDS または WDS バッファが無効です。
- 23015 接続が途中で失敗しました。
- 23016 指定したコマンドの操作が指定時間内で正常に終了しませんでした。
- 23017 ストリームがすでにこのローカル UDP ポートを使用しているか、または TCP 接続がローカル IP アドレスと TCP ポートの間、および指定したリモート IP アドレスと TCP ポートの間にすでに存在します。
- 23032 パケットはフラグメンテーションなしに送るには大きすぎます。断片化フラグをセットしないでください。
- 23033 ディスティネーションホストがアドレスの分割要求に応じることができません。
- 23035 icmp エコーパケットが指定されたタイムアウト時間内に応答しませんでした。
- 23036 送信時にこのパケットを断片化するために使用できる内部ドライババッファが十分ではありません。
- 23037 パケットのルーティングを管理するために使用できるゲートウェイがありません。
- 23041 ホスト名フィールドがシンタックスエラーです。アドレスがドット表記で割り当てられ、IP アドレスのシンタックスに準拠していません。
- 23042 指定したドメイン名がキャッシュ内で見つかりませんでした。ドメイン名分割ツールがドメイン名サーバーに現在問い合わせをし、コールバックプロシージャ内にその回答を返します。
- 23043 アドレス分割ツールがアドレスを見つけるために使用されなければならない場合にそのアドレス翻訳に渡されるプロシージャの結果がありません。
- 23044 ネームサーバーが指定した名前文字列を見つけることができませんでした。
- 23045 このドメイン名は存在しません。
- 23046 応答しているネームサーバーはありません。
- 23047 ドメインネームサーバーがエラーを返しました。
- 23048 必要な DNR クエリを発行したり、DNR キャッシュを作成するために使用できるメモリが足りません。

WinSock エラーコード

- 10004 呼び出しブロックが取り消されました。
- 10013 アクセス許可がありません。
- 10014 アドレスが間違っています。
- 10022 引数が無効です。
- 10024 使用可能なソケットがありません。
- 10035 非ブロックソケットが阻止されました。
- 10036 ブロック関数を処理している際に、不正な WinSock 関数が呼び出されました。
- 10037 すでに終了している非同期操作を取り消そうとしました。

- 10038 指定したソケット記述子がこのアプリケーションでは有効ではありません。
- 10039 ディスティネーションアドレスが必要でしたが、その関数に指定されませんでした。
- 10040 データグラムがバッファに対して大きすぎます。
- 10041 指定したプロトコルがその呼び出しの他の引数と一致しません。
- 10042 プロトコルオプションが不明か、あるいは無効です。
- 10043 指定したプロトコルが WinSock インプリメントでサポートされていません。
- 10044 指定したソケットタイプが特定アドレスファミリーでサポートされていません。
- 10045 ソケットが指定した操作をサポートしていません。
- 10046 プロトコルがサポートされていません。
- 10047 指定したアドレスファミリーが WinSock インプリメントでサポートされていないか、または特定ソケットで使用できません。
- 10048 指定したアドレスはすでに使用中です。
- 10049 指定したアドレスはローカルマシンから利用できません。
- 10050 ネットワークサブシステムで問題が発生しました。
- 10051 ネットワークがこの時点でこのホストから届いていません。
- 10052 接続が切断されました。リセットする必要があります。
- 10053 タイムアウトまたは何らかのエラーのため接続が中止されました。
- 10054 接続がリモートホストによってリセットされました。
- 10055 WinSock インプリメントがバッファ空間の範囲外か、あるいはアプリケーションによって API 呼び出しに与えられた空間がリクエストした情報を保持するには小さすぎます。
- 10056 指定したソケットはすでに接続しています。
- 10057 指定したソケットは接続されていません。
- 10058 リクエストした機能性を持ったソケットがシャットダウンしました。
- 10060 接続が設定される前に接続の試みがタイムアウトしました。
- 10061 接続の試みが強制的に拒絶されました。
- 10091 ネットワークサブシステムが通信用にまだ準備されていません。
- 10092 WinSock DLL が、リクエストした WinSock プロトコルのバージョンをサポートしていません。
- 10093 WinSock が初期化されていません。
- 11001 任意の権限によって確認されるので要求したデータベース情報は存在しません。
- 11002 リクエストした情報が見つからなかったため、その回答は信頼できません。
- 11003 回復不能のエラーが発生しました。
- 11004 指定した名前は有効ですが、リクエストしたタイプの情報はデータベース内にはありません。

SMTP RFC 値

次の項目は、任意の外部コマンドによって返されるエラーコードではありません。下記はクライアント - サーバー通信中に様々なステータスをやり取りするために設定された SMTP プロトコルの応答コードです。ローレベル TCP コマンドを使って独自のメール通信プロシージャを作成する際にこのリストが有効であることに開発者は気付くでしょう。

- 211 システムステータス、またはシステムヘルプリプライ。
- 214 ヘルプメッセージ (受信者の使用方法や標準でないコマンドの意味に関する情報、このリプライは人間のユーザにのみ有効です)。
- 220 <ドメイン> サービス準備完了。
- 221 転送チャネルの <ドメイン> サービスをクローズ中。
- 250 リクエストしたメール操作は OK。完了。

251 ローカルでないユーザが<転送パス>に転送されている。
 354 < CRLF >、< CRLF >でメールの開始と終了を入力する。
 421 <ドメイン>サービスが利用できない。転送チャンネルをクローズ中（これは、サービスが
 450 リクエストしたメール処理は行われません：メールボックスを利用できません。（例えば、メール
 ボックスビジーのため）
 451 リクエストした処理は中止されました：処理中にローカルエラーが起きました。
 452 リクエストした処理は行われません：システムストレージが足りません。
 500 シンタックスエラー。コマンドを認識できません。（これは、コマンド行が長過ぎるようなエラー
 を含んでいる可能性があります）
 501 シンタックスエラー。パラメータ（引数）が正しくありません。
 502 コマンドをインプリメントできません。
 503 コマンドシーケンスが間違っています。
 550 リクエストした処理は行われません：メールボックスを利用できません。（例えば、メールボック
 スが見つからない、アクセスがない）
 551 ローカルユーザでありません。<転送パス>を試してみてください。
 552 リクエストしたメール処理は中止されました：ストレージ割り当てが足りません。
 553 リクエストした処理は行われません：メールボックス名が許可されていません。例えば、メール
 ボックスのシンタックスが正しくない）
 554 トランザクションに失敗しました。

FTP RFC 値

次の項目は、任意の外部コマンドによって返されるエラーコードではありません。下記はクライアント - サー
 バー通信中に様々なステータスをやり取りするために設定された FTP プロトコルの応答コードです。ローレベ
 ル TCP コマンドを使って独自のメール通信プロシージャを作成する際にこのリストが有効であることに開発者
 は気付くでしょう。

110 Restart marker reply. In this case, the text is exact and not left to the particular
 implementation; it must read:
 MARK yyyy = mmmm
 Where yyyy is User-process data stream marker, and mmmm server's equivalent
 marker (note the spaces between markers and "=").
 120 nnn 分内にサービス準備完了。
 125 データ接続がすでにオープンされています；転送開始中。
 150 ファイルステータス OK；データ接続をまさにオープンしようとしています。
 200 コマンド OK。
 202 コマンドをインプリメントできません。このサイトでは不要です。
 211 システムステータス、またはシステムヘルプ応答。
 212 ディレクトリスステータス。
 213 ファイルステータス。
 214 ヘルプメッセージ（受信者の使用方法や標準でないコマンドの意味に関する情報、この応答は人間
 のユーザにのみ有効です）。
 215 NAME システムタイプ。NAME は割り当てられた番号ドキュメント内にあるリスト上の公式なシ
 ステム名です。
 220 新規ユーザ用のサービス準備完了。
 221 接続を制御するサービスをクローズ中。必要なら、ログアウトする。
 225 データ接続をオープンします；処理中の転送はありません。

226 データ接続をクローズ中。要求したファイル操作が成功しました（例えば、ファイル転送やファイル中止）。

227 受動モード（h1,h2,h3,h4,p1,p2）を登録中。

230 ユーザログイン中。

250 リクエストしたメール操作は OK。完了。

257 "PATHNAME" が作成されました。

331 ユーザ名 OK。パスワードが必要です。

332 ログイン用のアカウントが必要です。

350 リクエストしたファイル操作がこの後の情報を保留中。

421 <ドメイン> サービスが利用できない。転送チャネルをクローズ中（これは、サービスがシャットダウンしなければならないことを知っている場合に任意のコマンドに返信される）。

425 データ接続をオープンできません。

426 接続がクローズされました；転送が中止されました。

450 リクエストしたメール処理は行われません：メールボックスを利用できません。（例えば、メールボックスビジーのため）

451 リクエストした処理は中止されました：処理中にローカルエラーが起きました。

452 リクエストした処理は行われません：システムストレージが足りません。

500 シンタックスエラー。コマンドを認識できません。（これは、コマンド行が長過ぎるようなエラーを含んでいる可能性があります）

501 シンタックスエラー。パラメータ（引数）が正しくありません。

502 コマンドをインプリメントできません。

503 コマンドシーケンスが間違っています。

504 その引数用のコマンドをインプリメントできません。

530 ログインしていません。

532 ファイルを保存するためのアカウントが必要です。

550 リクエストした処理は行われません：メールボックスを利用できません。（例えば、メールボックスが見つからない、アクセスがない）

551 ローカルユーザでありません。＜転送－パス＞を試してみてください。

552 リクエストしたメール処理は中止されました：ストレージ割り当てが足りません。

553 リクエストした処理は行われません：メールボックス名が許可されていません。例えば、メールボックスのシンタックスが正しくない）

下記の WWW(World Wide Web) にてインターネットプロトコルに関する有益な情報を取得することができます。Web ドキュメントは、Netscape、Internet Explorer 等のプログラムを介してアクセスすることができます。

<http://rs.internic.net/>

ドメイン名とは何か、あるいはドメイン名に登録する際に必要なものは何かを理解することができます。

<http://www.ietf.org/>

Internet Engineering Task Force (IETF) のサイト。

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

RFC とは何かを理解することができます。RFC に関するサイト。

<ftp://ftp.isi.edu/in-notes/rfc821.txt> /

SMTP(Simple Mail Transfer Protocol) -- RFC 821。

<http://www.w3c.org/>

WWW(World Wide Web) に関する全般的なことを知りたい場合。

<http://www.con.wesleyan.edu/~triemer/network/docservs.html>

TCP/IP ポート番号の全リストが掲載されたサイト。ローレベルの TCP ルーチンを使って作業したい場合は、このサイトはとても有益です。

コマンド索引

F

FTP_Append	112
FTP_Delete	107
FTP_GetDirList	100
FTP_GetFileInfo	102
FTP_GetType	113
FTP_Login	98
FTP_Logout	118
FTP_MacBinary	108
FTP_MakeDir	104
FTP_Progress	96
FTP_Receive	116
FTP_RemoveDir 1	05
FTP_Rename	106
FTP_Send	110
FTP_SetType	114
FTP_System	115
FTP_VerifyID	103

I

IT_Decode	154
IT_Encode	152
IT_ErrorText	151
IT_GetPort	156
IT_GetTimeOut	150
IT_MacTCPInit	142
IT_MacTCPVer	146
IT_MyTCPAddr	148
IT_Platform	143
IT_SetPort	157
IT_SetTimeOut	149
IT_TCPversion	145
IT_Version	144

M

MSG_Delete	91
MSG_Extract	90
MSG_FindHeader	84
MSG_GetBody	87
MSG_GetHeaders	86
MSG_GetMessage	88
MSG_HasAttach	89
MSG_MessageSize	85

N

NET_AddrToName	137
NET_Finger	132
NET_NameToAddr	136
NET_Ping	133
NET_Resolve	138
NET_Time	134

P

POP3_BoxInfo	71
POP3_Delete	69
POP3_Download	78
POP3_GetMessage	73
POP3_GetPrefs	64
POP3_Login	65
POP3_Logout	70
POP3_MsgInfo	72
POP3_MsgLst	76
POP3_MsgLstInfo	74
POP3_Reset	68
POP3_SetPrefs	63
POP3_UIDToNum	79
POP3_VerifyID	67

S

SMTP_AddHeader	51
SMTP_Attachment	56
SMTP_Bcc	44
SMTP_Body	54
SMTP_Cc	43
SMTP_Clear	33
SMTP_Comments	48
SMTP_Date	34
SMTP_Encrypted	50
SMTP_From	36
SMTP_GetPrefs	27
SMTP_Host	31
SMTP_InReplyTo	46
SMTP_Keywords	49
SMTP_New	30
SMTP_QuickSend	28
SMTP_References	47
SMTP_ReplyTo	40
SMTP_Send	32
SMTP_Sender	38
SMTP_SetPrefs	25
SMTP_Subject	53
SMTP_To	42

T

TCP_Close	128
TCP_Listen	123
TCP_Open	122
TCP_Receive	126
TCP_Send	125
TCP_State	127