

4D Internet Commands 6.8

リファレンス
Windows® and Mac OS™



4D Internet Commands 6.8 リファレンス

Windows® and Mac™ OS

Copyright© 1999 - 2002 4D 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 4D 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 Server、4D、4D ロゴ、4D ロゴ、およびその他の 4D 製品の名称は、4D SA の商標または登録商標です。

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。

第 1 章	4D Internet Commands	9
	はじめに	9
	ソフトウェアの必要条件	11
	インストール	11
	ソフトウェアの必要条件	12
	用語集と用語	13
	引数のフォーマット	17
第 2 章	IC Send Mail	23
	メールの送信	23
	SMTP_SetPrefs	26
	SMTP_GetPrefs	29
	SMTP_QuickSend	30
	SMTP_New	33
	SMTP_Host	34
	SMTP_Send	35
	SMTP_Clear	37
	SMTP_Date	38
	SMTP_From	41
	SMTP_Sender	43
	SMTP_ReplyTo	45
	SMTP_To	47
	SMTP_Cc	48
	SMTP_Bcc	49
	SMTP_InReplyTo	51
	SMTP_References	52
	SMTP_Comments	53
	SMTP_Keywords	55
	SMTP_Encrypted	56
	SMTP_AddHeader	57

	SMTP_Subject	59
	SMTP_Body	61
	SMTP_Attachment	64
	SMTP_Charset	66
第 3 章	IC POP3 Review Mail	69
	メールの受信	69
	POP3_SetPrefs	72
	POP3_GetPrefs	74
	POP3_Login	75
	POP3_VerifyID	77
	POP3_Reset	78
	POP3_Delete	79
	POP3_Logout	80
	POP3_BoxInfo	81
	POP3_MsgInfo	82
	POP3_GetMessage	83
	POP3_MsgLstInfo	84
	POP3_MsgLst	86
	POP3_Download	88
	POP3_UIDToNum	90
	POP3_Charset	91
第 4 章	IC IMAP Review Mail	95
	IMAP4 コマンド	95
	用語	95
	IMAP 接続の概要	95
	IMAP コマンドテーマ	96
	メールボックスの構造	97
	<メッセージ番号>および<ユニーク ID>	97
	推奨	98
	POP3 コマンドと IMAP4 コマンドの比較	99
	IMAP_SetPrefs	101
	IMAP_GetPrefs	102
	IMAP_Login	103
	IMAP_VerifyID	105
	IMAP_Capability	106
	IMAP_ListMBs	107
	メールボックス属性	109

ワイルドカード文字の使用	110
IMAP_SubscribeMB	112
IMAP_GetMBStatus	113
IMAP_SetCurrentMB	115
IMAP_GetCurrentMB	117
IMAP_CloseCurrent MB	118
IMAP_Delete	119
IMAP_MsgInfo	120
IMAP_GetMessage	121
IMAP_MsgLstInfo	123
IMAP_MsgLst	125
IMAP_SetFlags	127
IMAP_GetFlags	130
IMAP_MsgFetch	131
基本データ項目	132
IMAP_Download	139
IMAP_UIDToMsgNum	141
IMAP_MsgNumToUID	142
IMAP_Search	143
IMAP_CopyToMB	148
IMAP_CreateMB	150
IMAP_DeleteMB	151
IMAP_RenameMB	152
IMAP_Logout	153
第 5 章 IC Downloaded Mail	155
メールのダウンロード	155
MSG_SetPrefs	156
MSG_GetPrefs	158
MSG_FindHeader	159
MSG_MessageSize	160
MSG_GetHeaders	162
MSG_GetBody	163
MSG_GetMessage	165
MSG_HasAttach	166
MSG_Extract	167
MSG_Delete	169
MSG_Charset	170

第 6 章	IC File Transfer	173
	ファイル転送	173
	FTP_Progress	174
	FTP_Login	176
	FTP_GetDirList	178
	FTP_ChangeDir	180
	FTP_PrintDir	181
	FTP_GetFileInfo	182
	FTP_VerifyID	183
	FTP_MakeDir	184
	FTP_RemoveDir	185
	FTP_Rename	186
	FTP_Delete	187
	FTP_MacBinary	188
	FTP_Send	190
	FTP_Append	192
	FTP_GetType	193
	FTP_SetType	194
	FTP_System	195
	FTP_Receive	196
	FTP_Logout	198
	FTP_SetPassive	199
	FTP_GetPassive	200
第 7 章	IC TCP/IP	201
	ローレベルルーチン	201
	TCP_Open	202
	TCP_Listen	204
	TCP_Send	206
	TCP_Receive	207
	TCP_SendBLOB	208
	TCP_ReceiveBLOB	209
	TCP_State	210
	TCP_Close	212
第 8 章	IT Internet	213
	特殊なインターネットコマンド	213
	NET_Finger	214

	NET_Ping	216
	NET_Time	218
	NET_NameToAddr	220
	NET_AddrToName	221
	NET_Resolve	222
第 9 章	IT Utilities	2 2 3
	ユーティリティコマンド	223
	IT_MacTCPInit	224
	IT_Platform	225
	IT_Version	226
	IT_TCPversion	227
	IT_MacTCPVer	228
	IT_MyTCPAddr	229
	IT_SetTimeOut	230
	IT_GetTimeOut	232
	IT_ErrorText	233
	IT_Encode	234
	IT_Decode	236
	IT_GetProxy	238
	IT_SetProxy	239
	IT_GetPort	241
	IT_SetPort	242
	IT_PPPConnect	243
	IT_PPPODisconnect	244
	IT_PPPOStatus	245
第 10 章	付録	2 4 7
	付録A：プログラミングのヒント	247
	Case文を介したコマンドの実行	247
	POP3メールまたはIMAPメールの自動返信機能について	248
	付録B：TCPポート番号	249
	ポート番号の選択方法	249
	付録C：エラーコード	251
	付録D：その他の情報	263
	コマンド索引	2 6 5

はじめに

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”という名前でも知られています。ここ数年間でインターネットにアクセスした個人や企業の数が増加したことが示されています。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“ネット上にいる”ということを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 および IMAP 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、あるいは遠隔地のシステムからのファイル転送、FTP ボリューム上のフォルダ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せん。

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 Commands を利用すれば、SMTP サーバ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示します。

- 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）かまたはIMAP（Internet Message Access Protocol）のどちらかのメールサーバ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、POP3コマンドおよびIMAP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ファイルを送受信するためにFTPサーバ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、フォルダ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または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。さらに、TCP_Openコマンドでは、SSL（Secured Socket Layer）プロトコルを使って接続を行うことができます。

以下にいくつかの例を示します。

- データベースをWWWサーバに移管する
- 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
- HTTPを使ってウェブサーバに接続する。

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。

つまり、メールを送信する場合には、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ます。

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。4D Japan Pack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4D Japan Pack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で逐次、補足いたします。

ソフトウェアの必要条件

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4D製品バージョン6.5をインストールすると、このプラグインは自動的に2つの異なった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：

- マシンのアクティブな4Dフォルダにインストールされると、特別な操作をしなくても、すぐに使用可能になります（4Dフォルダの場所に関する詳細は、『4D 6.8 アップグレード』マニュアルを参照してください）。
- 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または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。
- Mac4DX または Win4DX フォルダを、4D アプリケーション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3つのすべて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。
- ・ アクティブな 4D フォルダまたは 4D アプリケーションと同じレベル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または移動する場合、あるいは、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 4D 6.8 製品と独立してインストールする場合は、4D 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。

詳細については、『4D Product Line インストールガイド』を参照してください。

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。

システム

システムの必要条件は、4D 6.8 の条件と同じです。詳細については、『4D Product Line インストールガイド』を参照してください。

4th Dimension (Macintosh および Windows)

- バージョン 6.8 以降の Macintosh 版 4th Dimension
- バージョン 6.8 以降の Windows 版 4th Dimension

Open Transport、Winsock

4D Internet Commands を通しての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Open Transportや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お尋ねください。

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

ほとんどの4D Internet Commandsがドメインネームサーバ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、ネットワーク管理者にお尋ねください。

SMTP メールサーバ

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メールサーバにメッセージを転送するSMTPメールサーバに送り手がアクセスする必要があります。

POP3 サーバ

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

IMAP メールサーバ

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

用語集と用語

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。第1章の「引数のフォーマット」の節で示した用語は、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ドキュメントにおいて定義されています。付録D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.4dus.com」や「ftp.4dus.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 = "4dus.com"

ホスト名(コンピュータ名) = IP アドレス = ip_倍長整数

"www.4dus.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を持っているというわけではありません。また、“jsmith@4dus.com”のような「メールアドレス」は、個人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“4dus.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、そのマシン上で稼動しているPOP3サーバに配信され、“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する、あるいは突きとめ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なりません。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4dus.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、記憶する、あるいは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、元のファイル名に拡張子を追加する点です。以下にいくつかの一般的な拡張子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を覚えておいてください。

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表します。日本語の全角文字はすべて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@4dus.com"
アドレスリスト	テキスト	→ 例: "jsmith@4dus.com,jdupont@4d.fr"または "jsmith@4d.com"+Char(13)+"jdupont@4d.fr"
ローカルパス	テキスト	→ - ドキュメント Mac : "My Hard Drive:4DDB:SalesDB:Report" Win : "C:\MyDrive\4DDB\SalesDB\Report.txt" - ディレクトリ Mac : "My Hard Drive:CoolStuff:" (最後の":"に注意) Win : "C:\MyDrive\CoolStuff\"
ホストパス	テキスト	→ - ドキュメント "/usr/jsmith/reports/salesreport.txt" - ディレクトリ "/usr/jsmith/reports/" (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
imap_ID	倍長整数	→ IMAP接続を開始する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
戻り値	整数	← エラーコード

引数<ホスト名>

引数<ホスト名>は、“dns.4dus.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.4dus.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等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述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なりません。<メールアドレス>は、POP3サーバ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：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。“()”で括られた文字列はすべてコメントです。またメールアドレス中の“<>”で括られた文字があった場合、括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。以下に示した3つの例は、各々受け入れ可能な\$AddressListの値を生成します。

```
$AddressList:="jsmith@4dus.com
$AddressList:="jsmith@4dus.com, scott@4dus.com, marcel@4d.fr"
For ($i;1;Size of Array(aAddresses))
    $AddressList:=$AddressList+aAddresses{$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フォルダ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となります。Macintosh上のフォルダの指定は、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:”のフォルダを参照します。「ファイル名」または「フォルダ名」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もとに決められます。Windows上では、コロン文字ではなく円マーク“¥”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動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、“4D”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、“/4D/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 >、< pop3_ID >、< imap_ID >、< ftp_ID >、< 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、IMAP、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セッションは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
imap_ID	IMAP_Login コマンド	IMAP_Logout コマンド、または IMAP_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

戻り値

すべての4D Internet Commands (**IT_ErrorText** コマンドと **IT_Version** コマンドを除く) は、戻り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: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。

メールの送信

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示します。

- 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します。
- 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します (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します。

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”にいる誰が、“そこ”にいる誰に、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ったら、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用に“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ひとつは、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サーバに結果を配信します。SMTPサーバは、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ことを示しています。

■ 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)
戻り値	整数	← エラーコード

説明

SMTP_SetPrefs コマンドは、SMTP コマンドを使用し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/段落マーカーの終わりで見なす、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。(デフォルト値の) 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、下記の表に従った値を指定することで、送信するメッセージ本文に使用されている文字コード (Body-Content-Type) と、さらにメッセージ本文に適用されるエンコード方法 (Content-Transfer-Encoding) を指定します。例えば"US-ASCII & 7bit" (値2を指定) は現在使用されているメッセージ本文の文字コードがUS ASCII (WindowsやMacintoshで一般的な標準ASCIIコード (0から127) のみを含む) であり、4DICはメッセージ本文を7bitエンコーディングを使用してエンコードすることを示します。**SMTP_SetPrefs** コマンドはエンコードそのものを行わないことに注意してください。

必要であればユーザが管理する必要があります。文字コード変換を行うには、**SMTP_Charset** コマンドの説明を参照してください。

変更されない限りデフォルトでこの値は1であり、SMTP コマンドはメッセージ本文の内容に従って設定を自動的に検出します。

- 1 変更しない
- 0 アプリケーションおよびバイナリ：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
- 2 US-ASCII および7ビット
- 3 US-ASCII およびQuoted - printable
- 4 US-ASCII およびbase64
- 5 ISO-8859-1 およびQuoted - 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) & 7bit
- 11 ISO-2022-KR(韓国) & 7bit
- 12 ISO-2022-CN(中国、simplified、traditional) & 7bit
- 13 HZ-GB-2312(中国、simplified) & 7bit
- 14 Shift-JIS(日本) & base64*

*：4D Internet Commandバージョン6.7より導入。

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ことがあります。これに対する対策は4D Japna Packの日本語ルーチンを使用し、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行の文字数が多い場合、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
Quoted - 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

注：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	整数	←	0=ラインフィードを追加しない 1=ラインフィード追加
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
戻り値	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

SMTP_QuickSend

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
戻り値	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** コマンドをもととした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、「From」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となります。

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：通常subjectは区別的発音符 (é,ö など) 文字を含むべきではありません。このような拡張文字を使用したい場合は**SMTP_SetPrefs** や**SMTP_Charset** コマンドの説明を参照してください。

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。

例題

このコマンドを使用した例を以下に示します。

```
$err:=SMTP_SetPrefs ( 1 ; 10 ; -1 )
$error:=SMTP_Charset ( 1 ; 1 )
$Host:="mail.4d.co.jp"
$ToAddress:="sale@4d.fr"
$FromAddress:="komaba@4d.co.jp"
$Subject:="売上レポート"
$message:="1999年売上レポートを送信します。"
$error:=SMTP_QuickSend($Host;$FromAddress;$ToAddress;$Subject;
                        $message)

If($error=0)
    ALERT("送信に成功しました。")
Else
    ALERT("エラー ; SMTP_QuickSend"+Char(13)+IT_ErrorText($error))
End if
```

参照

SMTP_New

SMTP_New

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
戻り値	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用して **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。

例題

SMTP_Body コマンド、**SMTP_Send** コマンドの例を参照してください。

参照

SMTP_Clear、SMTP_QuickSend、SMTP_Send

SMTP_Host

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
戻り値	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サーバに配信されます。SMTPサーバは、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。

例題

SMTP_Body コマンド、**SMTP_Send** コマンドの例を参照してください。

参照

SMTP_New

SMTP_Send

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
戻り値	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。

例題

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せん。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)
$error:=SMTP_New ($smtp_id)
$error:=SMTP_Host ($smtp_id;"mail.4d.co.jp")
$error:=SMTP_From ($smtp_id;"komaba@4d.co.jp")
$error:=SMTP_ReplyTo ($smtp_id;"support@4d.co.jp")
$error:=AJP_mime_Encode ("広告スペース" ; $Subject)
$error:=SMTP_Subject ($smtp_id;$subject)
For ($i;1;Records in selection([顧客]))
    If ([顧客]販売金額>100000)
        $Body:=<> 徳意様向けドキュメント
    Else
        $Body:=<> 普通向けドキュメント
    End if
    $Body:=Replace string($Body;"<氏名>";[顧客]顧客氏名)+$Body

$error:=SMTP_To ($smtp_id;[顧客]email;1) `To"のヘッダを置き換える
$error:=AJP_Nkf ("-j";$Body;$body)
$error:=SMTP_Body ($smtp_id;$body) `JIS 漢字コード変換
$error:=SMTP_AddHeader ($smtp_id;"Content-Type:";"text/plain; charset
                        ="+<>quote+"iso-2022-jp"+<>quote)
$error:=SMTP_AddHeader ($smtp_id;"Content-Transfer-Encoding:";"7bit")
```

```
$error:=SMTP_Send ($smtp_id)  
NEXT RECORD([顧客])  
End for  
$error:=SMTP_Clear ($smtp_id)
```

参照

SMTP_New、AJP mime Encode、AJP Nkf

SMTP_Clear

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
戻り値	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。

例題

SMTP_Body の例を参照してください。

参照

SMTP_New

SMTP_Date

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	時刻	→	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
戻り値	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：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、下記の表の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-この引数の値は、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せん。上記の説明、あるいは下記の一覧を参照し、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

SMTP_From

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=追加、1=置換、2=削除
戻り値	整数	←	エラーコード

説明

SMTP_From コマンドは、メッセージの「From (差出人)」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、「From」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「From」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。

「From」ヘッダは、必須ではありません。「From」ヘッダは持っていないが、「Sender (送信人)」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。「From」ヘッダにアドレスが指定された場合、「Sender」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、「From」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：「ReplyTo」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「From」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、「From」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を指定した場合は、「From」フィールド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設定を無視して、「From」フィールドを新しい値に設定します。値として2を指定した場合は、「From」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が設定されません。

例題

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「From」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。

```
$From:="syaty@4d.co.jp, senmu@4d.co.jp, kansa@4d.co.jp"  
$error:=SMTP_From ($smtp_id;$From;0)  
$error:=AJP_mime_Encode ("会社の方針変更" ; $Subject)  
$error:=SMTP_Subject ($smtp_id;$subject)  
$error:=SMTP_To ($smtp_id;<> すべての従業員;0)
```

参照

SMTP_New

SMTP_Sender

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、1=置換、2=削除
戻り値	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。「Sender (送信人)」フィールドの内容が「From (差出人)」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、「Sender」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ドレス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「Sender」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、「Sender」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、「Sender」フィールドに新しい値を追加します。値として1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、「Sender」フィールドを新しい値に設定します。値として2を指定した場合は、「Sender」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。

例題

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「From」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。

```
$From:="syaty@4d.co.jp, senmu@4d.co.jp, kansa@4d.co.jp"  
$error:=SMTP_From ($smtp_id;$From;0)  
$error:=SMTP_Sender ($smtp_id;"hisyo@4d.co.jp")  
$error:=AJP_mime_Encode ("会社の方針変更" ; $Subject)  
$error:=SMTP_Subject ($smtp_id;$subject)  
$error:=SMTP_To ($smtp_id;<> すべての従業員;0)
```

参照

SMTP_New

SMTP_ReplyTo

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
戻り値	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「From (差出人)」です。送信するメッセージの「ReplyTo (返信先)」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「From」や「Sender (送信人)」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) を送りたいこともあります。

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<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を指定した場合は、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。

例題

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@4d.co.jp) および人事 (jinji@4d.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。

```
$From:="syatyo@4d.co.jp, senmu@4d.co.jp, kansa@4d.co.jp"  
$error:=SMTP_From ($smtp_id;$From;0)  
$error:=SMTP_Sender ($smtp_id;"hisyo@4d.co.jp")  
$error:=SMTP_ReplyTo ($smtp_id;"hisyo@4d.co.jp, jinji@4d.co.jp")  
$error:=AJP_mime_Encode ("会社の方針変更" ; $Subject)  
$error:=SMTP_Subject ($smtp_id;$subject)  
$error:=SMTP_To ($smtp_id;<> すべての従業員;0)
```

参照

SMTP_New

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
戻り値	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。

例題

SMTP_Body コマンドの例を参照してください。

参照

SMTP_New

SMTP_Cc

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
戻り値	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を指定した場合は、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。

例題

SMTP_Body コマンドの例を参照してください。

参照

SMTP_Bcc、SMTP_New

SMTP_Bcc

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
戻り値	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すみます。

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、「Bcc」ヘッダの追加または削除を示す整数値です。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。

例題

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、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。

```

$error:=SMTP_From ($smtp_id;"sales@4d.co.jp")
$error:=AJP mime Encode ("特別セール！今週限り！"; $Subject)
$error:=SMTP_Subject ($smtp_id;$subject)
$error:=AJP Nkf ("-j";$Body;$body)
$error:=SMTP_Body ($smtp_id;$body)           `JIS 漢字コード変換
For ($i;1;Records in selection([顧客]))
    $error:=SMTP_Bcc ($smtp_id;[顧客]email;0) `メールアドレス追加
    NEXT RECORD([顧客])
End for
$error:=SMTP_Send ($smtp_id)                 `全員にメールを送信
$error:=SMTP_Clear ($smtp_id)

```

参照

SMTP_Cc、SMTP_New

SMTP_InReplyTo

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=置換 (<返信先テキスト>が空でない場合)、 1=置換、2=削除
戻り値	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、本文の始めであるという印です。その後に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<返信先テキスト>を置換するか削除する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、引数<返信先テキスト>フィールドに新しい値を定義し、それまでの定義に上書きします (<返信先テキスト>に空の文字列を渡した場合、それまでのヘッダは保たれます)。値として1を指定した場合は、引数<返信先テキスト>フィールドに新しい値を定義し、それまでの定義に上書きします (<返信先テキスト>に空の文字列を渡した場合、それまでのヘッダは削除されます)。値として2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。<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。

参照

SMTP_New

SMTP_References

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 = 置換 (<リファレンス>が空でない場合)、 1 = 置換、2 = 削除
戻り値	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、本文の始めであるという印です。その後に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>のヘッダを置換するか削除する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>フィールドに新しい値を定義し、それまでの定義に上書きします（<リファレンス>に空の文字列を渡した場合、それまでのヘッダは保たれます）。値として1を指定した場合は、引数<リファレンス>フィールドに新しい値を定義し、それまでの定義に上書きします（<リファレンス>に空の文字列を渡した場合、それまでのヘッダは削除されます）。値として2を指定した場合は、<リファレンス>フィールドからテキストを削除し、<リファレンス>のすべてのテキストを無視します。<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。

参照

SMTP_New

SMTP_Comments

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=置換 (<コメント>が空でない場合)、 1=置換、2=削除
戻り値	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、本文の始めであるという印です。その後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>のヘッダを置換するか削除するかを指定するオプションの整数値です。値として0を指定した場合は、<コメント>フィールドに新しい値を定義し、それまでの定義に上書きします（<コメント>に空の文字列を渡した場合、それまでのヘッダは保たれます）。値として1を指定した場合は、引数<コメント>フィールドに新しい値を定義し、それまでの定義に上書きします（<コメント>に空の文字列を渡した場合、それまでのヘッダは削除されます）。値として2を指定した場合は、<コメント>フィールドからテキストを削除し、<コメント>のすべてのテキストを無視します。<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。

例題

SMTP_Body コマンドの例を参照してください。

参照

SMTP_New

SMTP_Keywords

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=置換 (<キーワード>が空でない場合)、 1=置換、2=削除
戻り値	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、本文の始めであるという印です。その後に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<キーワード>のヘッダを置換するか削除する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、<キーワード>フィールドに新しい値を定義し、それまでの定義に上書きします (<キーワード>に空の文字列を渡した場合、それまでのヘッダは保たれます)。値として1を指定した場合は、引数<キーワード>フィールドに新しい値を定義し、それまでの定義に上書きします (<キーワード>に空の文字列を渡した場合、それまでのヘッダは削除されます)。値として2を指定した場合は、<キーワード>フィールドからテキストを削除し、<キーワード>のすべてのテキストを無視します。<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。

参照

SMTP_New

SMTP_Encrypted

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=置換 (<暗号化>が空でない場合)、 1=置換、2=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、本文の始めであるという印です。その後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<暗号化>のヘッダを置換するか削除する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、<暗号化>フィールドに新しい値を定義し、それまでの定義を上書きします (<暗号化>に空の文字列を渡した場合、それまでのヘッダは保たれます)。値として1を指定した場合は、引数<暗号化>フィールドに新しい値を定義し、それまでの定義を上書きします (<暗号化>に空の文字列を渡した場合、それまでのヘッダは削除されます)。値として2を指定した場合は、<暗号化>フィールドからテキストを削除し、<暗号化>のすべてのテキストを無視します。<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。

参照

SMTP_Body、SMTP_New

SMTP_AddHeader

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
戻り値	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** コマンドを使用し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、本文の始めであるという印です。その後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。

例題

HTMLが埋め込まれたメッセージを送信するためには、メッセージ本文にHTMLタグを挿入し（例<HTML><HEAD>等）、すべての"Content-Type"ヘッダを"text/html;charset=us-ascii"に置き換えます。

```
    If(Substring;($body;1;6)="<HTML>")
        $err:=SMTP_AddHeader($SMTP_ID;"Content-type;";"test/html;
                                charset=us-ascii";1)
    End If
```

参照

POP3_FindHeader、SMTP_New

SMTP_Subject

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=置換 (<件名>が空でない場合)、 1=置換、2=削除
戻り値	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：通常 subject は区別的発音符 (é,ö など) 文字を含むべきではありません。このような拡張文字を使用したい場合は **SMTP_SetPrefs** や **SMTP_Charset** コマンドの説明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、本文の始めであるという印です。その後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。<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<件名>のヘッダを置換するか削除する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、<件名>フィールドに新しい値を定義し、それまでの定義に上書きします（<件名>に空の文字列を渡した場合、それまでのヘッダは保たれます）。値として1を指定した場合は、引数<件名>フィールドに新しい値を定義し、それまでの定義に上書きします（<件名>に空の文字列を渡した場合、それまでのヘッダは削除されます）。値として2を指定した場合は、<件名>フィールドからテキストを削除し、<件名>のすべてのテキストを無視します。<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。

例題

SMTP_Body コマンドの例を参照してください。

参照

SMTP_New、AJP mime Encode

SMTP_Body

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=置換 (<メッセージ本文>が空でない場合)、 1=置換、2=削除
戻り値	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。

警告：通常メッセージ本文は区別的発音符(é,öなど)文字を含むべきではありません。このような拡張文字を使用したい場合は**SMTP_SetPrefs**や**SMTP_Charset**コマンドの説明を参照してください。

引数<削除オプション>は、引数<メッセージ本文>のヘッダを置換するか削除する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、<メッセージ本文>フィールドに新しい値を定義し、それまでの定義を上書きします(<メッセージ本文>に空の文字列を渡した場合、それまでのヘッダは保たれます)。値として1を指定した場合は、引数<メッセージ本文>フィールドに新しい値を定義し、それまでの定義を上書きします(<メッセージ本文>に空の文字列を渡した場合、それまでのヘッダは削除されます)。値として2を指定した場合は、<メッセージ本文>フィールドからテキストを削除し、<メッセージ本文>のすべてのテキストを無視します。<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が設定されません。

例題

次はSMTPに関する完全な使用例です。

```

C_LONGINT($smtp_id)
C_BOOLEAN($SentOK;$OK)
$送信OK:=False `すべ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
$error:=AJP mime Encode (v件名$Subect)
$error:=AJP mime Encode ("4Dを介して送信" ; $comments)
$error:=AJP Nkf ("-j";$Body;$body)

```

Case of

```

\(Not(ERRCHECK ("SMTP_New";SMTP_New ($smtp_id))))
\(Not(ERRCHECK ("SMTP_Host";SMTP_Host ($smtp_id;<>pref_Server))))
\(Not(ERRCHECK ("SMTP_From";SMTP_From ($smtp_id;vFrom))))
\(Not(ERRCHECK ("SMTP_To";SMTP_To ($smtp_id;vTo))))
\(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-4D demo";<>version))))
\(Not(ERRCHECK ("SMTP_Body";SMTP_Body ($smtp_id;$body))))
\(Not(ERRCHECK ("SMTP_Send";SMTP_Send ($smtp_id))))

```

Else

```

$SentOK:=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（\$Command）、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($1;$Command;vErrorText;$s;$scr)
C_LONGINT($2;$error)
$Command:=$1
$error:=$2

```

```

$Result:=True
$cr:=Char(13)
If ($error#0)
    $Result:=False
    If (<>SHOWERRORS) `エラー表示するかどうかを決定するブール値
        vErrorText:=IT_ErrorText($error)
        $s="エラー..."+$cr
        $s=$s+"コマンド："+$Command+$cr
        $s=$s+"エラーコード："+String($error)+$cr
        $s=$s+"説明："+vErrorText
        ALERT($s)
    End if
End if
$0:=$Result

```

参照

SMTP_New、AJP Nkf

SMTP_Attachment

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
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
戻り値	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」ダイアログを表示します。

"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します。

"パス:ファイル名"：「ファイル名」を含んだ完全なパスで指定します。

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用して、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>に負の値を使用し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、SMTP_New、SMTP_Send

SMTP_Charset

SMTP_Charset (ヘッダエンコード;本文文字セット)→整数

引数	タイプ		説明
ヘッダエンコード	整数	→	-1=現在の設定を使用する、0=操作なし、1 = ISO-8859-1 または ISO-2022-JP の場合、指定された文字セットを使用して変換し、拡張文字をエンコードする
本文文字セット	整数	→	-1=現在の設定を使用する、0=操作なし、1 = ISO-8859-1 または ISO-2022-JP の場合、指定された文字セットを使用して変換する
戻り値	整数	←	エラーコード

説明

SMTP_Charset コマンドは、**SMTP_QuickSend** や **SMTP_Send** コマンドを使用した、拡張文字を含むメッセージ送信を自動的にサポートします。このコマンドがコールされていない、あるいは引数に0が渡された場合、バージョン 6.7 以降の 4D Internet Commands はバージョン 6.5.x と同様に動作します。

SMTP_Charset コマンドは、まず **SMTP_SetPrefs** コマンドの引数<本文タイプ>に設定した値をメッセージヘッダまたは本文の変換に適用するか、次に拡張文字を含むヘッダを RFC#1342 に示された "=?ISO-8859-1?Q?Test=E9?=" のシンタックスにエンコードするかどうかを設定します。

このコマンドは特に「件名」や ("=?ISO-8859-1?Q?Test=E9?=<test@n.net>" のような) メールアドレスなどメッセージヘッダに含まれる拡張文字のサポートに有効です。

エンコードはメッセージヘッダに依存し、常に Base64 (件名ヘッダを除く) に設定され、**SMTP_SetPrefs** の引数<本文タイプ>の値にもとづいて以下のように行われます。

- ・ 件名、コメント (非構造的ヘッダ) : 拡張文字を含む場合、文字列すべてがエンコードされます。
- ・ 送信人、宛先、CC、Bcc、返信先、InReplyTo (構造的ヘッダ) :
 - angle 括弧 ("<<",">") に囲まれたすべてのテキストは電子メールアドレスだと解釈され、エンコードされません。
 - SPC<>()@;:;"/?.= のような特別な区切り文字はエンコードされません。
 - 特別な区切り文字で区切られた文字列は、拡張文字が含まれていればエンコードされます。

アドレスの例：

- someone@somewhere はエンコードされません；
- 太郎 <taro@somewhere> は「太郎」だけがエンコードされます。

引数<ヘッダエンコード>は、メッセージを送信する際に、ヘッダの変換とエンコードをどのように扱うかを示します。デフォルト値は0です。

■ -1：現在の設定を使用します。

■ 0：操作しません。

■ 1：

SMTP_SetPrefs コマンドの引数<本文タイプ>がISO-8859-1またはISO-2022-JPに設定されている場合、ヘッダは示された文字セットを使用して変換されます。

示された文字セットが何であれ、拡張文字が含まれている場合は、ヘッダは以下のシンタックス、"?SMTP_SetPrefs コマンドで設定された文字セット?Base64 エンコード?Test=E9?=..." (RFC#1342を参照) を使用してエンコードされます。

例外：件名ヘッダは、必要であればSMTP_SetPrefs コマンドの引数<本文タイプ>で与えられたエンコーディング方法を使ってエンコードされます。

注："X_..."のような拡張ヘッダはUS ASCIIコードのみで記述されていなければなりません。

引数<本文文字セット>は、メッセージを送信する際に、メッセージ本文の文字コード変換をどのように扱うかを示します。デフォルト値は0です。

■ -1：現在の設定を使用します。

■ 0：操作しません。

■ 1：**SMTP_SetPrefs** コマンドの引数<本文タイプ>がISO-8859-1またはISO-2022-JPに設定されている場合、メッセージ本文は示された文字セットを使用して変換されません。

例題

(1) 4D インターネットコマンドバージョン 6.5 を使用した場合：

```
SMTP_SetPrefs (1;1;0)
$err:=SMTP_QuickSend ("mymail.com";"myaddress";"destination";
Mac to ISO ("the yen ¥");Mac to ISO ("the yen Symbol is ¥"))
```

(2) 4D インターネットコマンドバージョン 6.7 を使用した場合

```
SMTP_SetPrefs (1;1;0)
$err:=SMTP_Charset (1;1)
$err:=SMTP_QuickSend ("mymail.com";"myaddress";"destination";"the yen \";
"the yen Symbol is \)
```

- ・件名と本文は ISO-8859-1 文字セットを使用して変換されます
- ・件名は RFC11342 のシンタックスに従ってエンコードされます

参照

POP3_Charset、SMTP_SetPrefs

メールの受信

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する、あるいは取り出すことができます。

POP3に関連するコマンドは第3章の「IC POP3 Review Mail」と第5章の「IC 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。この章ではPOP3サーバから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上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、第5章のコマンドを使用して、その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、再び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いて、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サーバにログインし、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、メッセージ番号3と4のメッセージを削除したと仮定します。このセッションをログアウトすると、今行なった削除処理はコミット（許可）されます。ここで、POP3サーバの中にログを戻すと、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、使用しているPOP3サーバから割り当てられた値です。ただし、POP3サーバ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か。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上に残しておきたいようなソリューションを開発している場合には、いくつか注意する必要があります。

注：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

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)
戻り値	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設定します。

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。

引数<メッセージフォルダ>は、**POP3_Download** コマンドを使用し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。

互換性についての注意（バージョン6.8.1）：引数<ラインフィード除去>および引数<メッセージフォルダ>は、以前はMSG_テーマのコマンドにも適用されていました。しかし、新たに追加されたMSG_SetPrefsコマンドを使用した場合は、そちらの設定が優先されます。

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。

互換性についての注意（バージョン6.8.1）：引数<添付フォルダ>はPOP3_SetPrefsとMSG_SetPrefsのどちらにもあるため、この2つのどちらかのコマンドを使って変更することができます。

しかしながら、互換性の理由により使用されているPOP3_SetPrefsの引数は将来的に使用されなくなるため、MSG_SetPrefs コマンドの使用を強くお勧めします。POP3_SetPrefs コマンドの<添付フォルダ>はオプションであるため、この引数を渡さないようにお勧めします。この推奨内容はPOP3_GetPrefsにも該当します。

参照

MSG_Extract、MSG_GetPrefs、MSG_SetPrefs、POP3_Download、POP3_GetPrefs

POP3_GetPrefs

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)
戻り値	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。

互換性についての注意 (バージョン6.8.1) : POP3_GetPrefs コマンドの引数<添付フォルダ>はオプションであり、使用されなくなるため、この引数を渡さないようにお勧めします。この引数はMSGコマンドでのみ使用されるため、POP3コマンドには影響しない点に留意してください。

参照

MSG_SetPrefs、POP3_SetPrefs

POP3_Login

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
戻り値	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>を使用して、POP3 メールサーバ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。任意のサーバに一度ログインしたら、必要な操作はすべて実行し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コマンド間の呼び出しにおいて、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、4D社の経験から言うと、ほとんどのサーバ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接続を中止したサーバのイベントでは、行な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@4d.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

POP3_VerifyID

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
戻り値	整数	←	エラーコード

説明

POP3サーバは、アドミニストレータ（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サーバと対話する各コマンドは、強制的にこの非稼動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セッションの非稼動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

POP3_Reset

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
戻り値	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：**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、POP3_Login

POP3_Delete

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
戻り値	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：**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

POP3_Logout

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。
戻り値	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

POP3_BoxInfo

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
戻り値	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

POP3_MsgInfo

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	←	サーバ上のメッセージのユニークID
戻り値	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大変有効なリファレンスです。

参照

POP3_Login

POP3_GetMessage

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
戻り値	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 Dimension 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、POP3_SetPrefs、AJP Nkf

POP3_MsgLstInfo

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
戻り値	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実行できた場合は、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が、何らかの問題が起こった場合、<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しまうこともあります。

注：メールのヘッダは拡張文字を含む場合がありますが、**POP3_Charset** コマンドを使用して管理を自動化できます。

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： **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、POP3_MsgLst

POP3_MsgLst

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	文字列/テキスト配列	← 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
戻り値	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。

例題

```
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。

```
aValues{1}{1}..."Mon, 12 Jul 1999 22:19:01 +0900"  
aValues{2}{1}..."takahasi@4d.co.jp"  
aValues{3}{1}..."奥さんに電話してください。"
```

※ ヘッダ中の日本語は 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： **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が、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、POP3_MsgLstInfo

POP3_Download

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
戻り値	整数	←	エラーコード

説明

POP3_Download コマンドは、POP3 サーバ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、このコマンドを使用し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）という名前で作成します。

"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

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
戻り値	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。見つかり、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。

POP3_Charset

POP3_Charset (ヘッダデコード;本文文字セット)→整数

引数	タイプ		説明
ヘッダデコード	整数	→	-1=現在の設定を使用する、0=操作なし、1 = ISO-8859-1 または ISO-2022-JP の場合、MacOS 文字セットを使用して変換し、拡張文字をデコードする
本文文字セット	整数	→	-1=現在の設定を使用する、0=操作なし、1 = ISO-8859-1 または ISO-2022-JP の場合、MacOS 文字セットを使用して変換する
戻り値	整数	←	エラーコード

説明

POP3_Charset は、POP3 や MSG コマンドを使用した、拡張文字を含むメッセージ処理を自動的にサポートします。このコマンドがコールされていない、あるいは引数に0が渡された場合、バージョン6.7以降の4D Internet Commandsはバージョン6.5.xと同様に動作します。

POP3_Charset は、まず拡張文字を含むヘッダのデコードを行うかどうか、次にメッセージ本文とヘッダの文字セットの変換を行うかどうかを設定します。このコマンドは特に「件名」や ("=?ISO-8859-1?Q?Test=E9?= <test@n.net>") のようなメールアドレス等、メッセージヘッダに含まれる拡張文字のサポートに有効です。

引数<ヘッダデコード>は、**POP3_MsgLst** や **MSG_FindHeader** コマンドを実行する際に、ヘッダのデコードと変換をどのように扱うかを示します。デフォルト値は0です。

- ・ -1：現在の設定を使用します。
- ・ 0：操作しません。
- ・ 1：必要に応じてヘッダがデコードされます。デコードされ、さらに明示された文字セットがISO-8859-1またはISO-2022-JPの場合、ヘッダはそれぞれMacOS AsciiコードかShift-JISを使用して変換されます。

互換性についての注意（バージョン6.8.1）：MSG_Charset コマンドを事前に実行していない限り、POP3_Charset は（POP3_MsgLst と同様に）MSG_FindHeader に適用されます。

引数<本文文字セット>は、**MSG_GetBody** コマンドを実行する際に、メッセージ本文の文字コード変換をどのように扱うかを示します。デフォルト値は0です。

- ・ -1 : 現在の設定を使用します。
 - ・ 0 : 操作しない
 - ・ 1 : "Body_Content_Type"文字セットがISO-8859-1またはISO-2022-JPに設定されている場合、メッセージ本文はそれぞれMacOS AsciiコードかShift-JISを使用して変換されます。
- 互換性についての注意（バージョン6.8.1）：MSG_Charsetコマンドを事前に実行していない限り、従来通り、POP3_CharsetはMSG_GetBodyに適用されます。

例題

(1) 4D インターネットコマンドバージョン6.5を使用した場合

```

$Err:=MSG_FindHeader ($msgfile;"From";$from)
$from:=ISO to Mac ($from)
$Err:=MSG_FindHeader ($msgfile;"To";$to)
$to:=ISO to Mac ($to)
$Err:=MSG_FindHeader ($msgfile;"Cc";$cc)
$cc:=ISO to Mac ($cc)
$Err:=MSG_FinHeader ($msgfile;"Subject";$subject)
$subject:=ISO to Mac ($subject)

$Err:=MSG_MessageSize ($msgfile;$HdrSize;$BdySize;$MsgSize)
$Err:=MSG_GetBody ($msgfile;0;$BdySize;$BodyContent)
$BodyContent:=ISO to Mac ($BodyContent)

```

(2) 4D インターネットコマンドバージョン6.7を使用した場合

```

$Err:=POP3_Charset (1;1)
$Err:=MSG_FindHeader ($msgfile;"From";$from)
$Err:=MSG_FindHeader ($msgfile;"to";$to)
$Err:=MSG_FindHeader ($msgfile;"Cc";$cc)
$Err:=MSG_FindHeader ($msgfile;"Subject";$subject)

$Err:=MSG_MessageSize ($msgfile;$HdrSize;$BdySize;$MsgSize)
$Err:=MSG_GetBody ($msgfile;0;$BdySize;$BodyContent)

```

(3) 4D Internet Commandsバージョン6.8を使用した場合

```

$Err:=MSG_Charset (1;1)
$Err:=MSG_FindHeader ($msgfile;"From";$from)
$Err:=MSG_FindHeader ($msgfile;"To";$to)
$Err:=MSG_FindHeader ($msgfile;"Cc";$cc)
$Err:=MSG_FindHeader ($msgfile;"Subject";$subject)

```

\$Err=**MSG_MessageSize** (\$msgfile;\$HdrSize;\$BdySize;\$MsgSize)

\$Err=**MSG_GetBody** (\$msgfile;0;\$BdySize;\$BodyContent)

参照

MSG_Charset、SMTP_Charset

IMAP4 コマンド

一連のIMAPコマンドを使って、データベースからIMAP電子メールサーバ上の電子メールメッセージの呼出しや処理を行い、IMAPサーバから電子メールメッセージを取り出すことができます。IMAPコマンドは、RFC2060によって定義されたインターネットメッセージ・アクセスプロトコルのバージョン4改訂1 (IMAP4rev1) に準拠しています。IMAP4rev1により、“メールボックス”と呼ばれるリモートメッセージフォルダをローカルメールボックスと機能的には似たような方法で管理することができます。

IMAPコマンドを使って行うことができる操作には、メールボックスの作成、削除および名前の変更、新着メッセージの確認、メッセージの永続的な削除、メッセージフラグの付加および削除、メッセージの検索、および、メッセージの選択部分の取り出しがあります。

用語

“接続”とは、ネットワーク接続の開始 (**IMAP_Login**) から選択の終了 (**IMAP_Logout**) までのIMAPクライアント/サーバ間のやりとりの全体を指します。

“セッション”とは、メールボックスが選択 (**IMAP_SetCurrentMB**) されてから選択の終了 (**IMAP_SetCurrentMB**、**IMAP_CloseCurrentMB**) または接続の終了 (**IMAP_Logout**) までのクライアント/サーバ間のやりとりの全体を指します。

IMAP 接続の概要

- TCP通信の初期化：**IT_MacTCPInit** (PPP接続の場合は**IT_MacTCPInit**の前に**IT_PPPOConnect**コマンドを呼出す必要があります)。
- 接続の開始：**IMAP_Login**
- メールボックスの管理：リスト、作成、削除、名前の変更、登録/登録解除およびステータス取得の引数

- カレントワーキング・メールボックスを指定してセッションを開始：**IMAP_SetCurrentMB**
カレントメールボックスを指定すると、その中のメッセージを管理することができます。
- メッセージの管理：リスト、メッセージのダウンロードまたは削除、メッセージフラグのリスト、メッセージフラグの変更、別のメールボックスへのコピー、電子メールのダウンロードを行わない検索および取り出し等。
- カレントメールボックスのメッセージの処理が完了したら、セッションを終了または別のカレントメールボックスを指定して新しいセッションを開始することができます。いずれの場合もIMAPサーバはメッセージを永久的に更新します。例えば、\Deletedフラグを付加すると、IMAPサーバはそのフラグが付加されたすべてのメッセージを削除します。
- 作業が完了したらログアウトしてください。接続の終了：**IMAP_Logout**
- その他の操作：初期設定、ケイパビリティ、接続確認、およびIMAPサーバ上のあらゆる休止自動ログアウトタイマーのリセット。

IMAP コマンドテーマ

IMAPに関連するコマンドについてはIC IMAP Review MailおよびIC Downloaded Mailの2つの節に分かれています。これらのコマンドは電子メールを読む方法の違いによって分かれています。IMAPサーバから電子メールを読む場合は、メッセージ（またはメッセージ情報）を4Dのストラクチャ（変数、フィールド、配列）に読み込むか、またはディスクにダウンロードします。この節では、4D Internet CommandsのIMAPサーバからメッセージを読む機能について説明します。

メッセージの読み出し方法が2通りある理由は、主に大量の情報をダウンロードする動作にはメモリの制約があるためです。例えば、1つのIMessageに5MBの添付ファイルが含まれる場合、それだけですぐにデータベースの記憶容量を超えてしまいます。ピクチャフィールドまたはBLOBフィールドについては、唯一このサイズのものを格納することができる4Dのストラクチャですが、ピクチャやBLOBの呼出しにはクライアント側の煩雑なメモリ条件があるので、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あまり効率的ではありません。こうした問題を解決するために、この節ではIMAPサーバからユーザのハードディスクにメッセージを転送する**IMAP_Download**コマンドについて説明します。

ディスクに読み込みを行った後のローカルファイルの処理コマンドについては“IC Downloaded Mail”の節で説明します。

メールボックスの構造

IMAP メールボックスはフォルダのように取り扱うことができ、その中にファイルやサブフォルダを含めることができます。同様に、メールボックスの中にメッセージやサブメールボックスを含めることもできます。

メールボックスは完全な階層名を使って呼出します。IMAPサーバに応じて各階層レベルは階層セパレータ（セパレータは **IMAP_ListMBs** コマンドで返されます）で区切られます。

セパレータを使って、子メールボックスの作成や指定する階層の上下のレベルの検索を行うことができます。最上レベルの階層ノードの子メールボックスにはすべて同じセパレータ文字を使用します。

注：メッセージはカレントワーキング・メールボックスが選択されている場合 (**IMAP_SetCurrentMB**) にのみ管理することができます。

各アカウントに1つまたは複数のメールボックスを作成することができます。

メールボックス名は大文字と小文字が区別するため、大文字・小文字の違いのみの同じ名前のメールボックスを2つ作成することはできません。

INBOX メールボックスは特別で、各アカウントにあり、受信メッセージの格納に使用されます。INBOX メールボックスはアカウントを開設すると自動的に作成されます。

ユーザはINBOX メールボックスを削除することはできませんが名前を変更することはできます。名前を変更すると新しい空のINBOX メールボックスが直ちに作成されます。INBOX メールボックスの名前については、大文字と小文字が区別されます。

メッセージまたは新着メッセージの合計数のようないくつかのメールボックスの属性については、カレントメールボックスではなくても確認することができます。

<メッセージ番号>および<ユニークID>

IMAP コマンドを使用する際には、最も頻繁に使用される引数、より詳しくはメールボックスの構造である<メッセージ番号>および<ユニークID>についてよく理解しておくことが重要です。<メッセージ番号>は、**IMAP_SetCurrentMB** コマンドが実行される時点のメールボックス内のメッセージの番号です。カレントメールボックスを選択すると、1からメールボックス内の合計メッセージ数までの番号がメールボックス内のメッセージに割り当てられます。番号は最も古いものを1とし、メールボックスにメッセージが受信された順番に割り当てられます。メッセージに割り当てられた番号は、カレントワーキング・メールボックスを選択(**IMAP_SetCurrentMB**)してから終了(**IMAP_CloseCurrentMB**、**IMAP_SetCurrentMB**または**IMAP_Logout**)するまでの間のみ有効です。

メールボックスを終了すると、削除の印が付けられたメッセージはすべて削除されます。

IMAPサーバに再度ログオンすると、メールボックス内のメッセージに再び1から順に番号が振り直されます。例えば、メールボックスに10個メッセージがあり、1から5までの番号のメッセージを削除した場合は、メールボックスを再度開くと前回の6から10のメッセージに対して1から5の番号が振り直されています。

例として、IMAPサーバにログオンし、次のようなメッセージリストを取得するとします。

セッション中に3と4の番号のメッセージを削除するとします。カレントワーキング・メールボックスを終了すると削除の要求が実行されます。サーバに再度ログオンすると、メッセージリストは次のように番号が振り直されています。

<メッセージ番号>はスタティック（固定的）な値ではなく、それぞれのセッションごとに異なります。カレントワーキング・メールボックスが選択された時点のメールボックス内の他のメッセージとの関係で変わります。

これに対して、<ユニークID>は一意の番号であり、IMAPサーバによって厳密な昇順でメッセージに割り当てられます。各メッセージがメールボックスに追加されるたびに、それ以前に追加されたメッセージよりも大きい番号のIDが割り当てられます。残念ながら、IMAPサーバは<ユニークID>をメッセージの一次リファレンスとして使用しません。そのため、IMAPコマンドを使用する際は、<メッセージ番号>をサーバ上のメッセージのリファレンスとして指定する必要があります。メッセージの本文そのものをサーバ上に残したままメッセージ・リファレンスをデータベースに読み込む方法を開発する際には、開発者は注意する必要があります。

推奨

IMAPの醍醐味は相互接続性であり、それを真空内でテストすることは不可能ですので、最終的には“すべてをテストする”ことをお勧めします。したがって、アカウントを取得することができるあらゆるサーバに対してクライアントのテストを行ってください。

詳細は次のサイトをご覧ください。

■ IMAP製品およびサービス：<http://www.imap.org/products.html> IMC

■ MailConnect：<http://www.imc.org/imc-mailconnect>

POP3 コマンドとIMAP4 コマンドの比較

Login	同一	IMAP コマンドには POP 引数はありません。
VerifyID	同一	
Delete	同一	IMAP コマンドは即時に削除します。POP3 コマンドは POP3_Logout の実行を待ってメッセージを永久的に削除します。IMAP_SetFlags を使って \Deleted フラグを付加すると、POP3_Delete コマンドと同じ結果を得ることができます。
Logout	同一	
SetPrefs	同一	IMAP コマンドには引数<添付フォルダ>はなく、POP3 コマンドの<添付フォルダ>はオプションとなります。
GetPrefs	同一	SetPrefs コマンドの<添付フォルダ>に関する注記を参照
MsgLstInfo	同一	
MsgInfo	同一	
MsgLst	同一	
UIDToMsgNum	同一	IMAP コマンドの引数<ユニークID>は倍長整数で、POP3 コマンドの<ユニークID>は文字列です。
Download	同一	
POP3_Reset	直接的な同一コマンドなし	\Deleted フラグを削除するには、\Deleted フラグについての IMAP_Search と IMAP_SetFlags を組み合わせて使う必要があります。
POP3_BoxInfo	直接的な同一コマンドなし	IMAP_SetCurrentMB コマンドと IMAP_MsgLstInfo コマンドを組み合わせて使う必要があります。
IMAP_MsgNumToUID	直接的な同一コマンドなし	
GetMessage	ほぼ同一	IMAP コマンドはより強力になり、<メッセージパート>引数に新たに本文のみを選択することが可能です。
POP3_Charset	相当するコマンドなし	IMAP コマンドでは、自動的に文字コード系の処理を行います。
IMAP_Capability	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_ListMBs	相当するコマンドなし	IMAP プロトコルに特有のコマンド

IMAP_GetMBStatus	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_SetCurrentMB	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_GetCurrentMB	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_CloseCurrentMB	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_CopyToMB	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_SubscribeMB	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_CreateMB	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_DeleteMB	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_RenameMB	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_SetFlags	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_GetFlags	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_Search	相当するコマンドなし	IMAP プロトコルに特有のコマンド
IMAP_MsgFetch	相当するコマンドなし	IMAP プロトコルに特有のコマンド

注意：

- ・ IMAP および POP3 サーバ：IMAP サーバの場合は <メッセージ ID> は倍長整数であるため、同じように入力してはいけません。

- ・ 削除は、POP3 プロトコルと IMAP プロトコルでは必ずしも同じように機能しません。IMAP_Delete の場合は即時にメッセージを削除します。POP3_Delete と同じ結果を得るには IMAP_SetFlags を使って \Deleted フラグを付加し、POP3_Reset と同じ結果を得るには IMAP_SetFlags を使って \Deleted フラグを取り出します。

IMAP_SetPrefs

IMAP_SetPrefs (ラインフィード除去; メッセージフォルダ) → 整数

引数	タイプ		説明
ラインフィード除去	整数	→	0 = ラインフィードを除去しない 1 = ラインフィードを除去、 -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 ("" = 変更なし)
戻り値	整数	←	エラーコード

説明

IMAP_SetPrefs コマンドで、すべての IMAP コマンドの環境設定を行います。

<ラインフィード除去>は、保存したメッセージでのラインフィード文字の処理方法を指定する整数値です。ほとんどの IMAP サーバで、行の終わりを示すのにキャリッジリターンとラインフィード文字が併用されます。Macintosh のアプリケーションでは、行の終わりを示す文字としてキャリッジリターンのみを使用することが好まれます。このオプションの引数を使ってメッセージテキストからラインフィード文字を除去することができます。<ラインフィード除去>が 0 の場合は、IMAP サーバ上に格納されているフォーマットのままでメッセージを取り出します。1 の場合は、取り出されたメッセージからラインフィード文字を除去します。-1 の場合は、前回の設定を使用します。値を指定しない場合はデフォルト値として 1 が設定され、メッセージ内に見つかったラインフィードは自動的に除去されます。

<メッセージフォルダ>は、**IMAP_Download** コマンドを使って取り出したメッセージをデフォルトで格納するフォルダのローカルパス名を示すテキスト値です。

参照：IMAP_Download、IMAP_GetPrefs

IMAP_GetPrefs

IMAP_GetPrefs (ラインフィード除去; メッセージフォルダ) → 整数

引数	タイプ		説明
ラインフィード除去	整数	←	0 = ラインフィードを除去しない 1 = ラインフィードを除去、 -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 ("" = 変更なし)
戻り値	整数	←	エラーコード

説明

IMAP_GetPrefs コマンドはIMAPコマンドの現在の環境設定を返します。

設定内容は引数にリストされる変数に返されます。

<ラインフィード除去>は、ラインフィード除去に関する現在のユーザ設定の内容を返します。

<メッセージフォルダ>は、取り出したメッセージを格納するデフォルトの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：IMAP_SetPrefs

IMAP_Login

IMAP_Login (ホスト名; ユーザ名; パスワード; imap_ID) → 整数

引数	タイプ		説明
ホスト名	文字列	→	IMAP メールサーバのホスト名または 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
imap_ID	倍長整数	←	この IMAP ログインのリファレンス
戻り値	整数	←	エラーコード

説明

IMAP_Login コマンドは、与えられたユーザ名およびパスワードを使って IMAP 電子メールサーバへユーザをログインします。この方法でログインすると、その後の IMAP コマンドに使用することができる接続のリファレンス (**imap_ID**) が与えられます。

接続の終了は、**IMAP_Logout** コマンドを使って行うか、または IMAP サーバ休止タイマーのタイムアウト時に自動的に行われます。

<ホスト名>は、IMAP 電子メールサーバのホスト名または 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、できるだけホスト名を使用することをお勧めします。

<ユーザ名>は、IMAP 電子メールサーバ上でのユーザ名です。<ユーザ名>にはドメインを付ける必要はありません。例えば、“jack@4d.com” というアドレスの<ユーザ名>は “jack” となります。

<パスワード>は、IMAP 電子メールサーバ上での<ユーザ名>のパスワードです。

<imap_ID>には、確立されたばかりの接続のリファレンスとなる倍長整数変数が返されます。4D の変数に戻り値を受け取るにはこの引数を渡す必要があります。その変数は、このセッションに関する動作を実行するその後のすべてのコマンドに使用します。

IMAP_Login が機能しない場合は、<imap_ID>の値は0に設定されます。

▼ 下記は、典型的な接続シーケンスの例です。

```
$ErrorNum:=IMAP_Login (vHost; vUserName; vUserPassword; vImap_ID)
If($ErrorNum =0)
    C_TEXT(vCapability)
    $ErrorNum:=IMAP_Capability (vImap_ID; vCapability))
    ... ` vImap_ID 引数を使用する IMAP コマンド
End if
$ErrorNum:=IMAP_Logout (vImap_ID)
```

参照：IMAP_Logout、IMAP_VerifyID

IMAP_VerifyID

IMAP_VerifyID (imap_ID) → 整数

引数	タイプ		説明
imapID	倍長整数	→	このIMAPログインのリファレンス
		←	0 = 接続はすでに終了
戻り値	整数	←	エラーコード

説明

IMAPサーバは、管理者によって決められた時間内にアクティビティを示さない接続を自動的に終了します。IMAPサーバと相互に作用する各コマンドは、強制的に休止タイマーをリセットします。**IMAP_VerifyID** コマンドは、他のいかなる動作も実行することなく指定されたIMAP接続の休止タイマーをリセットします。これにより、タイムアウトする可能性がある接続をアクティブにしておくことができます。

IMAP_VerifyID コマンドを実行すると、まだ終了されていない接続が確認されます。アクティブの接続がある場合は、このコマンドはIMAPサーバに対して接続のタイムアウトカウンタを0に戻すように指示します。接続がすでに終了している場合は、**IMAP_VerifyID** は適切なエラーおよびIMAP接続に使用できる空きメモリを返し、<imap_ID>に0を返します。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

参照：IMAP_Login

IMAP_Capability

IMAP_Capability (imap_ID; ケイパビリティ) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	このIMAPログインのリファレンス
ケイパビリティ	テキスト	←	IMAPのケイパビリティ
戻り値	整数	←	エラーコード

説明

IMAP_Capability コマンドは、IMAPサーバでサポートされるケイパビリティ名がスペースで区切られたリストを含むテキストエリアを返します。このリストには、サーバがサポートするIMAPのバージョンやオプション機能（IMAP4rev1プロトコルの拡張、改訂または修正等）が含まれます。

IMAP4rev1がケイパビリティのテキストに表示されていなければ、4D Internet Commandsとの整合性は確保されません。

参照：IMAP_Login

IMAP_ListMBs

IMAP_ListMBs (imap_ID; リファレンス名; メールボックス名; メールボックス名配列; メールボックス属性配列; メールボックス階層配列 {;登録メールボックス})
→ 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
リファレンス名	テキスト	→	空の文字列またはメールボックス名 またはメールボックス階層のレベル
メールボックス名	テキスト	→	空の文字列またはメールボックス名 またはワイルドカード
メールボックス名配列	文字列/テキスト配列	←	メールボックス名の配列 (パス名)
メールボックス属性配列	文字列/テキスト配列	←	メールボックス属性の配列
メールボックス階層配列	文字列/テキスト配列	←	階層デリミタの配列
登録メールボックス	整数	→	0 = 使用可能なすべてのユーザメール ボックスをリスト 1 = 登録されているメールボックスのみ をリスト
戻り値	整数	←	エラーコード

説明

IMAP_ListMBs コマンドは、接続したユーザおよび添付の情報に使用可能なメールボックスのリストを返します。このコマンドが機能しない場合は、指定した配列が初期化されています。返されるメールボックスのリストは<リファレンス名>と<メールボックス名>の値の組み合わせに左右されるため、この2つの引数は一緒に考える必要があります。

最後の引数である<登録メールボックス>を1に設定すると、登録されているメールボックス（「**IMAP_SubscribeMB**」の節を参照）のみのリストが返されます。

IMAP_ListMBs の実行に時間が掛かる場合は、多数のメールボックスが走査されているためか、または多数の複雑な階層メールボックスの構造等が原因です。このような場合は、次のような処理を行うことができます。

- **IMAP_ListMBs** にワイルドカード（下記参照）を使用する
- または、引数<登録メールボックス>を1に設定して**IMAP_ListMBs** コマンドを使用し、**IMAP_SubscribeMB** コマンドを使って指定したメールボックスのみをリストさせる。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

<リファレンス名>は、どのメールボックスを探すのかを決める際に引数<メールボックス名>と組み合わせて使うテキスト値です。<リファレンス名>は、Unixシステム上でカレントワーキングディレクトリとして使用するものです。つまり、<メールボックス名>は、<リファレンス名>で指定されるディレクトリにあるファイルと解釈されません。

IMAPの指定では、<リファレンス名>の解釈は“実行依存”となる点に注意してください。ユーザには引数<リファレンス名>を使用しない動作モードを与えることを強くお勧めします。そうすれば、リファレンス引数の使用を実行しない古いサーバでも相互運用することが可能です。

<リファレンス名>が空の文字列である場合は、引数<メールボックス名>のみを使ってメールボックスのリストが行われます。

<リファレンス名>にメールボックス名またはメールボックス階層のレベルが含まれる場合は、引数<メールボックス名>が解釈されるコンテキストを定義するのに使用されます。

注：リファレンス引数には後置階層デリミタ（区切り文字）を使用することを強くお勧めします。これにより、いかなるIMAPサーバが使用される場合でも完全な整合性を確保することができます。

<メールボックス名>は、通常<メールボックス名>が解釈されるコンテキストを定義する引数<リファレンス名>と組み合わせて使用するテキスト値です。

<メールボックス名>が空の文字列である場合は、階層デリミタ（区切り文字）が返されます。

注：引数<リファレンス名>を使用するブレイクアウト機能を実行する場合は、ユーザがメールボックス引数に前置階層デリミタ（区切り文字）を使用するかどうかを選択できるようにしてください。これは、メールボックスの前置階層デリミタ（区切り文字）の処理方法がサーバごとに、また同じサーバ上の異なる電子メール記憶装置間でさえも異なるからです。場合によっては、前置階層デリミタ（区切り文字）は“リファレンス引数の廃棄”を意味することもあり、また一方では、2つの引数が連結され余分な階層デリミタが廃棄されることもあります。

<メールボックス名配列>は使用可能なメールボックス名のリストを受け取ります。

<メールボックス属性配列>は使用可能なメールボックス属性のリストを受け取ります。

メールボックス属性

メールボックス属性は次の4つに定義されます。

- `\NoInferiors` : 子レベルが現在存在せず、作成することもできません。
- `\NoSelect` : この名前を選択可能なメールボックスとして使用することはできません。
- `\Marked` : サーバはメールボックスに“interesting”とマークし、メールボックスには最後の選択以降に追加されたメッセージが含まれます。
- `\Unmarked` : メールボックスには最後の選択以降の追加メッセージは含まれません。

＜メールボックス階層の配列＞は使用可能なメールボックスの階層デリミタ（区切り文字）のリストを受け取ります。

階層デリミタ（区切り文字）は、メールボックス名の階層レベルを区切るのに使用される文字です。階層デリミタを使って、子メールボックスの作成や指定する階層の上下のレベルの検索を行うことができます。最上レベルの階層ノードの子メールボックスにはすべて同じセパレータ文字を使用します。

＜登録メールボックス＞は、“登録されている”メールボックスのみをリストしたい場合に指定することができる整数値です。0の場合はすべての使用可能なユーザメールボックスをリストします。1の場合は登録されているメールボックスのみをリストします。＜登録メールボックス＞は、指定しない場合はデフォルト値として0が設定されるオプションの引数です。

▼ 例

```
IMAP_ListMBS (imap_ID;"4DIC/Work/";"Tes"mbNamesArray;
                mbAttribsArray; mbHierarArray)
```

… “4DIC/Work/Test” メールボックスの使用可能なすべてのメールボックスを返します。

IMAPサーバが意図するように解釈しない場合は、＜リファレンス名＞を使用せずに次のように＜リファレンス名＞と＜メールボックス名＞の値を＜メールボックス名＞に連結してください。

```
IMAP_ListMBS (imap_ID;"";"4DIC/Work/Test"; mbNamesArray;
                mbAttribsArray; mbHierarArray)
```

▼ 例

```
IMAP_ListMBS (imap_ID;"";"";mbNamesArray; mbAttribsArray; mbHierarArray)
```

…階層デリミタ（区切り文字）を返します。

ワイルドカード文字の使用

メールボックスの選択をより簡単にするために、引数<リファレンス名>および引数<メールボックス名>にワイルドカードを使用することができます。下記は現在のワイルドカードの例ですが、ワイルドカードの解釈はIMAPサーバに左右されるため、以下の例が機能しない場合があることに留意してください。機能しない場合は、IMAPサーバのワイルドカードをチェックしてください。

■ 以下の例において“*”は0以上の文字に相当します。

```
IMAP_ListMBs (imap_ID;"*"; mbNamesArray; mbAttribsArray; mbHierarArray)
```

…接続したユーザが使用可能なすべてのメールボックスを返します。

```
IMAP_ListMBs (imap_ID;"*;Work*"; mbNamesArray; mbAttribsArray;
mbHierarArray)
```

…ルート“Work”に一致するすべての使用可能なメールボックスを返します。

■ “%”は“*”に似ていますが、階層デリミタ（区切り文字）には相当しません。“%”ワイルドカードが引数<メールボックス名>の最後の文字である場合は、一致する階層レベルも返されます。これらの階層レベルが選択可能なメールボックスではない場合は、\Noselectメールボックス属性（「メールボックス属性」の節を参照）と共に返されます。

```
IMAP_ListMBs (imap_ID;"*;Work/%"; mbNamesArray; mbAttribsArray;
mbHierarArray)
```

ルート“Work”に一致するすべてのメールボックスに加えて接続したユーザが使用可能な階層レベルを1つ返します。

“%”は、メールボックスの階層レベルをレベルごとに解析する際に有効です。

次のようなメールボックス階層であると仮定します。

INBOX

```
MailboxA
    MailboxAA
    MailboxAB
MailboxB
    MailboxBA
    MailboxBB
MailboxC
    MailboxCA
    MailboxCB
```

IMAP_ListMBs (imap_ID;"";"%"; mbNamesArray; mbAttribsArray; mbHierarArray)

…INBOX、MailboxA、MailboxB および MailboxC を返します。

IMAP_ListMBs (imap_ID;"";"MailboxA%"; mbNamesArray; mbAttribsArray;
mbHierarArray)

…MailboxAA および MailboxAB を返します。

このテクニックを使うことにより、**IMAP_ListMBs** (imap_ID;"";"*";mbNamesArray;
mbAttribsArray; mbHierarArray)に対して長すぎる戻り値が返ることもなくなり、ユーザに
完全な柔軟性を与えることができます。

注：IMAP サーバ自体が走査するレベルの数を制限することがあります。

参照：IMAP_SubscribeMB、IMAP_GetMBStatus

IMAP_SubscribeMB

IMAP_SubscribeMB (imap_ID; メールボックス名; 登録メールボックス) → 整数

引数	タイプ		説明
imap_ID	整数	→	IMAP ログインのリファレンス
メールボックス名	テキスト	→	登録または登録解除するメールボックス名
登録メールボックス	整数	→	0 = 登録解除、1 = 登録
戻り値	整数	←	エラーコード

説明

IMAP_SubscribeMB コマンドを使って、IMAPサーバの“登録”ユーザメールボックスに対して指定したメールボックス名の追加や削除を行うことができます。

この機能を使って、ユーザは通常よく見るメールボックスを登録することにより、使用可能なメールボックスの大量のリストを絞り込んで選択することができます。

これを行うには、**IMAP_ListMBs** コマンドを使用しオプションの引数<登録メールボックス>を1に設定（「**IMAP_ListMBs**」の節を参照）します。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

<メールボックス名>は登録または登録解除するメールボックスの完全な名前です。

登録解除の場合は<登録メールボックス>に0を渡し、登録の場合は1を渡します。

参照： **IMAP_ListMBs**

IMAP_GetMBStatus

IMAP_GetMBStatus (imap_ID; メールボックス名; メッセージ番号; 新着メッセージ番号; 未読メッセージ番号; ユニークID) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
メールボックス名	テキスト	→	メールボックス名
メッセージ番号	倍長整数	←	指定されたメールボックスのメッセージの番号
新着メッセージ番号	倍長整数	←	\Recent フラグが付加されているメッセージの番号
未読メッセージ番号	倍長整数	←	\Seen フラグが付加されているメッセージの番号
ユニークID	倍長整数	←	指定されたメールボックスのユニークID
戻り値	整数	←	エラーコード

説明

IMAP_GetMBStatus コマンドは、`<メールボックス名>`で指定されたメールボックスのステータス引数の値を返します。カレントメールボックス（「**IMAP_SetCurrentMB**」の節を参照）を変更することなく、指定されたメールボックスのいかなるメッセージのステータスにも影響しません（特に、メッセージからの\Recent フラグの脱落を生じません、しかしこれはIMAP4サーバの実装異なります）。このコマンドは、カレントメールボックスを選択解除せずにメールボックスのステータス引数を確認するのに使用することもできます。

このコマンドは次のような場合に特に有効です。

- メールボックスのユニークIDを確認するまたは取り出す場合
- メールボックスのセッションを開始することなく新着および未読のメッセージを確認する場合

カレントメールボックスを使って**IMAP_GetMBStatus** コマンドを呼出さないことを強くお勧めします。問題が生じ、返される情報が必ずしもカレントメールボックスのステータスと一致しない（特に新着の電子メール場合）おそれがあります。

`<imap_ID>`は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

`<メールボックス名>`は、ステータス引数の値を取得したい既存のメールボックスの完全な名前です。

注： **IMAP_ListMBs** コマンドとは異なり、引数<メールボックス名>はワイルドカードを受け付けません。

<メッセージ番号>は、カレントメールボックス内のメッセージの番号を返します（コマンドを呼出す際は0に設定し、エラーの場合は-1を返します）。

<新着メッセージ番号>は、カレントメールボックス内の新着メッセージの番号を返します（コマンドを呼出す際は0に設定し、エラーの場合は-1を返します）。

<未読メッセージ番号>は、カレントメールボックス内の未読メッセージの番号を返します（コマンドを呼出す際は0に設定し、エラーの場合は-1を返します）。

<ユニーク ID>は、メールボックスのユニーク IDの有効性の値を返します（コマンドを呼出す際は0に設定し、エラーの場合は-1を返します）。

IMAP4 プロトコルでは、メールボックス名のみではメールボックスを識別するのに十分ではありません。したがって、ユニーク IDの有効性の値は各メールボックスと関連しています。この識別子は複数の作業を同時進行する際に特に有効です。

例えば、ユニーク IDの有効性の値を確認するだけでメールボックス“A”が“B”に名前を変更されたのか、または削除されたのかを確認することができます。また、この識別子を使ってメールボックス“A”が削除され、別の“A”という名前のメールボックスが作成されたのかを確認することもできます。

参照： **IMAP_ListMBs**、**IMAP_SetCurrentMB**、**IMAP_SetFlags**、**IMAP_GetFlags**

IMAP_SetCurrentMB

IMAP_SetCurrentMB (imap_ID; メールボックス名; メッセージ番号; 新着メッセージ番号; カスタムフラグ; 永久フラグ; ユニークID) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
メールボックス名	テキスト	→	選択するメールボックス名
メッセージ番号	倍長整数	←	カレントメールボックス内のメッセージの番号
新着メッセージ番号	倍長整数	←	\Recent フラグが付加されているメッセージの番号
カスタムフラグ	テキスト	←	メールボックスに現在使用されているフラグのリスト
永久フラグ	テキスト	←	永久的に変更可能なフラグのリスト
ユニークID	倍長整数	←	メールボックスのユニークIDの値
戻り値	整数	←	エラーコード

説明

IMAP_SetCurrentMB を使って、セッションを開始（すなわち、カレントワーキング・メールボックスを選択）し、指定したメールボックスのメッセージを管理することができます。

接続中に複数のセッションを同時に開始することはできません。複数のメールボックスの同時呼出しには並列接続（複数の **IMAP_Login**）が必要です。**IMAP_SetCurrentMB** コマンドは、新しいセッションを開始しようとする自動的に現在のセッションを終了します。しかし、あるメールボックスをカレントに指定しても **IMAP_SetCurrentMB** コマンドが機能しない場合は、カレントに指定されるメールボックスが存在しなくなります。

存在しない<メールボックス名>を使って **IMAP_SetCurrentMB** コマンドを実行することにより、新しいセッションを選択することなくセッションを終了（すなわち、カレントメールボックスを終了）することができます。また、返されたエラーの処理中は、**IMAP_CloseCurrentMB** または **IMAP_Logout** コマンドを実行してセッションを終了することができます。

<imap_ID>は **IMAP_Login** によって開始された接続のリファレンスとなる倍長整数です。

<メールボックス名>は、カレントに指定する既存のメールボックスの完全な名前です。

<メッセージ番号>は、カレントメールボックス内のメッセージの番号を返します（**IMAP_SetCurrentMB** コマンドを呼出す際は0に設定し、エラーの場合は-1を返します）。

<新着メッセージ番号>は、カレントメールボックス内の新着メッセージの番号を返します (**IMAP_SetCurrentMB** コマンドを呼出す際は0に設定し、エラーの場合は-1を返します)。

<カスタムフラグ>は、カレントメールボックス内で使用されているフラグの完全なリストを返します。<永久フラグ>文字列にリストされているフラグのみが変更可能である点に注意してください。

<永久フラグ>は、メールボックス内の永久的に変更することができるメッセージフラグのリスト返します (**IMAP** サーバによって管理される **\Recent** フラグを除く)。(**IMAP_SetCurrentMB** コマンドを呼出す際は空の文字列に設定します)。<永久フラグ>文字列には、特別なフラグ “*” も使うことができる点に留意してください。このフラグをメールボックスに格納することによりキーワードを作成することができます (「**IMAP_SetFlags**」の節を参照)。

<永久フラグ>が空の文字列を返す場合は、引数<カスタムフラグ>にリストされたすべてのフラグが永久的に変更可能であることを意味します。

<ユニークID>は、カレントメールボックスのユニークIDの有効性の値を返します。

あるメールボックスを削除し、それと同じ名前の新しいメールボックスを後日作成する場合にこの識別子は特に有効です。なぜなら、名前が同じであれば、ユニークIDの有効性が異なる限りクライアントはこれが新しいメールボックスであるとは気付かない場合があるからです。

参照： **IMAP_ListMBs**、**IMAP_GetMBStatus**、**IMAP_SetFlags**、**IMAP_GetFlags**、**IMAP_Logout**、**MAP_CloseCurrentMB**

IMAP_GetCurrentMB

IMAP_GetCurrentMB (imap_ID; メールボックス名) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
メールボックス名	テキスト	←	カレントメールボックス名
戻り値	整数	←	エラーコード

説明

IMAP_GetCurrentMB はカレントワーキング・メールボックス名を返します。

<imap_ID> は **IMAP_Login** によって開始された接続のリファレンスとなる倍長整数です。

<メールボックス名> はカレントメールボックスの完全な名前を返します。<メールボックス名> の値が空の文字列である場合は、メールボックスは現在選択されていません。

参照：IMAP_SetCurrentMB、IMAP_CloseCurrentMB

IMAP_CloseCurrent MB

IMAP_CloseCurrentMB (imap_ID) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
戻り値	整数	←	エラーコード

説明

IMAP_CloseCurrentMB を使って、別のメールボックスの選択や **IMAP_Logout** を実行することなくカレントワーキング・メールボックスを終了することができます。**IMAP_CloseCurrentMB** は、\Deleted フラグが付加されているすべてのメッセージを永久的に削除します。

注：IMAPでは、クライアント/サーバモードで同じメールボックスで並行して作業を行うことができます。誰かが同時進行の作業を行い接続を繋いだままにしておくとは定すると、最後に使用されたメールボックスは被選択モードのままになっています。他の誰かがこのメールボックスを使用しようとしても、サーバの実装によってはユーザが“非接続モード”（すなわち、接続はされているがデータを使って作業を……状態）で作業をしていても、有効な情報を得ることや正しく作業を行うことができません。

< imap_ID > は **IMAP_Login** によって開始された接続のリファレンスとなる倍長整数です。

参照：IMAP_SetCurrentMB、IMAP_GetCurrentMB、IMAP_Delete、IMAP_SetFlags

IMAP_Delete

IMAP_Delete (imap_ID; 開始メッセージ; 終了メッセージ) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
開始メッセージ	倍長整数	→	開始メッセージの番号
終了メッセージ	倍長整数	→	終了メッセージの番号
戻り値	整数	←	エラーコード

説明

IMAP_Delete は<開始メッセージ>から<終了メッセージ>までの範囲のメッセージに \Deleted フラグを付加し、\Deleted フラグが付加されたすべてのメッセージを削除します (現在のセッションのために以前に \Deleted フラグが付加されているメッセージを含む)。削除は、接続の終了 (**IMAP_Logout**)、別のカレントメールボックスの選択 (**IMAP_SetCurrentMB**) またはカレントメールボックスの終了 (**IMAP_CloseCurrentMB**) の際に IMAP サーバによって実行されます。

すぐに削除しない場合は、**IMAP_SetFlags** コマンドを使用し、後でメッセージを削除するように \Deleted フラグを付加することができます。

<imap_ID> は **IMAP_Login** によって開始された接続のリファレンスとなる倍長整数です。

<開始メッセージ> は、削除するメッセージの最初のメッセージの番号を示す倍長整数の数値です。

<終了メッセージ> は、削除するメッセージの最後のメッセージの番号を示す倍長整数の数値です。

注：<開始メッセージ>が<終了メッセージ>の値より大きい場合は、**IMAP_Delete**、**IMAP_MsgLstInfo**、**IMAP_MsgLst**、**IMAP_SetFlags**、**IMAP_GetFlags** および **IMAP_CopyToMB** コマンドはエラーを返しません。このような場合は、コマンドは事実上機能しません。

参照： **IMAP_Logout**、**IMAP_SetCurrentMB**、**IMAP_CloseCurrentMB**、**IMAP_SetFlags**

IMAP_MsgInfo

IMAP_MsgInfo (imap_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
imap_ID	倍長整数	→	IMAP ログインのリファレンス
メッセージ番号	倍長整数	→	メッセージの番号
メッセージサイズ	倍長整数	←	メッセージのサイズ
ユニークID	倍長整数	←	サーバ上のメッセージのユニークID
戻り値	整数	←	エラーコード

説明

IMAP_MsgInfo コマンドは、<メッセージ番号>で識別された現在選択されているメールボックスの内のメッセージについての情報を返します。メッセージのサイズおよびそのユニークIDについての情報が返されます。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

<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。<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内のメッセージの位置を表します。ただし、特定の電子メールについて、その<メッセージ番号>はセッションが異なれば同じであるとは限りません。

<メッセージサイズ>には、<メッセージ番号>によって参照されるメッセージのサイズを示す倍長整数値が返されます。

<ユニークID>はサーバ上のメッセージのユニークIDを示す倍長整数変数です。<ユニークID>はIMAP4サーバソフトウェアによってメッセージに割り当てられる値です。この値は、<メッセージ番号>のようにセッションごとに変わるものではありません。<ユニークID>の値は、サーバからデータベースにメッセージがダウンロードされたかどうかを確認するのに有効なリファレンスです。

参照：IMAP_Login、IMAP_SetCurrentMB

IMAP_GetMessage

IMAP_GetMessage (imap_ID; メッセージ番号; オフセット; 文字数; メッセージパート; メッセージテキスト {;既読更新}) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP 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文字のオフセット
文字数	倍長整数	→	返す文字数
メッセージパート	整数	→	0 = メッセージ全体、 1 = ヘッダのみ、2 = 本文のみ
メッセージテキスト	テキスト	←	メッセージのテキスト
既読更新	整数	→	0 = ÅSeen フラグの更新、 1 = 更新しない
戻り値	整数	←	エラーコード

説明

IMAP_GetMessage コマンドは、**IMAP_SetCurrentMB** によって参照されるカレントメールボックス内の<メッセージ番号>によって識別されるメッセージの全テキストを返します。

IMAP_SetPrefs コマンドで特に指定しない限り、メッセージ内のラインフィード文字はすべて除去されます。

IMAP_GetMessage コマンドは、引数<メッセージパート>に応じて、ヘッダ情報を含むメッセージのブロック全体、ヘッダのみ、または本文のみを返します。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

<メッセージ番号>は、メールボックス内の取り出したいメッセージを示す倍長整数値です。この値は、現在選択されているメッセージリスト内のメッセージの位置を表します。ただし、特定の電子メールについて、その<メッセージ番号>はセッションが異なれば同じであるとは限りません。

<オフセット>は、指定した<メッセージパート>のはじめの読み込みを開始する文字の番号を示す倍長整数値です。通常は、この引数には0を渡します。

<文字数>は、<オフセット>の位置からの取り出す文字数を表す倍長整数値です。4Dのテキスト変数の最大文字数が32000バイトに制限されているため、引数<文字数>には32000以下の数値を指定してください。<メッセージパート>のサイズが32Kを超えるメッセージについては、**IMAP_Download** コマンドを使ってディスクに保存する必要があります。

<メッセージパート>は、メッセージの取り出す部分を示します。次のように、0、1または2を渡すことができます。

- 0 = メッセージ全体
- 1 = ヘッダのみ
- 2 = 本文のみ (最初のプレーンテキスト)。

メッセージ全体またはヘッダのみの取り出しの場合は、デコードせずにそのままのテキストを取り出します。それに対して本文のみを取り出す場合は、テキストは自動的にデコードされ、必要に応じて変換されます (デコードおよび変換の規則に関する詳細は、「POP3_Charset」の節を参照)。

<既読更新>は、暗黙的であるかどうかに関わらず、\Seen フラグをメッセージフラグとして付加するかしないかを示す整数値です。この引数はオプションで、値を渡さない場合はデフォルト値が使用されます。

- 0 = \Seen フラグを付加 (デフォルト値)
- 1 = \Seen フラグを付加しない

<メッセージテキスト>は、取り出されたテキストを受け取るテキスト変数です。

参照: [IMAP_SetCurrentMB](#)、[IMAP_Login](#)、[IMAP_Download](#)、[IMAP_SetPrefs](#)

IMAP_MsgLstInfo

IMAP_MsgLstInfo (imap_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
imap_ID	倍長整数	→	IMAP ログインのリファレンス
開始メッセージ	倍長整数	→	開始メッセージの番号
終了メッセージ	倍長整数	→	終了メッセージの番号
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	倍長整数配列	←	ユニークIDの配列
戻り値	整数	←	エラーコード

説明

IMAP_MsgLstInfo コマンドは、カレントワーキング・メールボックス (**IMAP_SetCurrentMB** コマンドで指定) 内のメッセージに関する情報を返します。情報は、1つのメッセージに対応する各要素が3つの配列に返されます。返される情報は、メッセージのサイズおよび番号に関するものです。引数として渡される配列については、どんなサイズでも構いませんが、前もってタイプが宣言されたものである必要があります。**IMAP_MsgLstInfo** コマンドは、取り出したメッセージ番号に対して各配列のサイズをリセットします。

IMAP_MsgLstInfo コマンドは、現在選択されているメッセージリスト内のメッセージ情報の取り出しに失敗しても、エラーの番号を返しません。エラーが発生した場合は、問題があったメッセージの配列に配列要素が作成されません。これに対して、コマンドの各メッセージの読み込みに問題がない場合は、<メッセージ番号配列>に順番に並んだ数値が入ります。問題が起こった場合は、<メッセージ番号配列>に入る数値の順番が途切れていることがあります。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

<開始メッセージ>は、調べるメッセージ範囲の開始メッセージの番号を指定する倍長整数です。メッセージ番号は、カレントワーキング・メールボックス内のすべてのメッセージのリストにおけるメッセージの位置を表す値です。

<終了メッセージ>は、調べるメッセージ範囲の終了メッセージの番号を指定する倍長整数です。メッセージ番号は、カレントワーキング・メールボックス内のすべてのメッセージのリストにおけるメッセージの位置を表す値です。

注：＜開始メッセージ＞が＜終了メッセージ＞よりも大きい番号である場合は、IMAP_Delete、IMAP_MsgLstInfo、IMAP_MsgLst、IMAP_SetFlags、IMAP_GetFlagsおよびIMAP_CopyToMB コマンドはエラーを返しません。このような場合は、コマンドは事実上機能しません。

＜サイズ配列＞は、＜開始メッセージ＞から＜終了メッセージ＞までの間の各メッセージのサイズが返される倍長整数配列です。

＜メッセージ番号配列＞は、＜開始メッセージ＞から＜終了メッセージ＞までの間のメッセージの番号が返される倍長整数配列です。

＜ユニーク ID 配列＞は、＜開始メッセージ＞から＜終了メッセージ＞までの間のメッセージのユニーク ID を返す倍長整数配列です。

参照： IMAP_SetCurrentMB

IMAP_MsgLst

IMAP_MsgLst (imap_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
imap_ID	倍長整数 →	IMAP ログインのリファレンス
開始メッセージ	倍長整数 →	開始メッセージの番号
終了メッセージ	倍長整数 →	終了メッセージの番号
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 ←	ヘッダ値の2次元配列
戻り値	整数 ←	エラーコード

説明

IMAP_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のに使用します。

このコマンドを使って、メッセージリストの特定の列を要求することができます。しかし、このコマンドはヘッダ項目の値のみを返すため、メッセージの本文を取り出すのに使用することはできません。ヘッダの内容は自動的にデコードされ、必要に応じて変換されます (デコードおよび変換の規則に関する詳細は、「POP3_Charset」の節を参照)。

<imap_ID>は **IMAP_Login** によって開始された接続のリファレンスとなる倍長整数です。

<開始メッセージ>は、調べるメッセージ範囲の開始メッセージの番号を指定する倍長整数です。メッセージ番号は、カレントワーキング・メールボックス内のすべてのメッセージのリストにおけるメッセージの位置を表す値です。

<終了メッセージ>は、調べるメッセージ範囲の終了メッセージの番号を指定する倍長整数です。メッセージ番号は、カレントワーキング・メールボックス内のすべてのメッセージのリストにおけるメッセージの位置を表す値です。

注：<開始メッセージ>が<終了メッセージ>よりも大きい番号である場合は、**IMAP_Delete**、**IMAP_MsgLstInfo**、**IMAP_MsgLst**、**IMAP_SetFlags**、**IMAP_GetFlags** および **IMAP_CopyToMB** コマンドはエラーを返しません。このような場合は、コマンドは事実上機能しません。

<ヘッダ配列>は、取り出したい特定の電子メールのヘッダをリストする文字列配列またはテキスト配列です。

<メッセージ番号配列>は、<開始メッセージ>から<終了メッセージ>までの間のメッセージの番号が返される倍長整数配列です。

<ユニーク ID 配列>は、<開始メッセージ>から<終了メッセージ>までの間のメッセージのユニーク ID を返す倍長整数配列です。

<値配列>は<ヘッダ配列>に指定し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には、<値配列>の1次元に一致する配列があります。

▼ 例

```
aHeaders{1}:="Date:"
aHeaders{2}:="From:"
aHeaders{3}:="Subject:"
IMAP_MsgLst (IMAP_ID; vStart; vEnd; aHeaders; aMsgNum; aMsgId; aValues)
aValues{1}{1}="Thu, 19 November 1998 00:24:02 -0800"
aValues{2}{1}="Jack@4d.com"
aValues{3}{1}="奥様に電話してください"
```

エラーは次のように処理されます。

1) 通信に関連するエラーコードのみが返されます。エラー（ネットワーク、シンタックス、サーバ等のエラー）のためにコマンドが作業を完了することができない場合は、適切なエラーコードが返されます。

2) 指定した範囲にメッセージが存在しない場合、またはエラーの場合。

■ そのメッセージの配列要素は作成されません。

■ エラーコードは返されません。

3) メッセージ内の指定したヘッダの一部またはすべてを見つけない場合は、エラーにはなりません。

■ そのメッセージの配列要素は作成されます。

■ <メッセージ番号配列>および<ユニーク ID 配列>の配列要素には適切な値が入りません。

■ メッセージにヘッダが存在しない場合は、配列要素に空の文字列が返されます。

■ エラーコードは返されません。

参照：IMAP_MsgLstInfo

IMAP_SetFlags

IMAP_SetFlags (imap_ID; 開始メッセージ; 終了メッセージ; フラグリスト; 削除オプション) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
開始メッセージ	倍長整数	→	開始メッセージの番号
終了メッセージ	倍長整数	→	終了メッセージの番号
フラグリスト	文字列/テキスト	→	付加または削除するフラグの値
削除オプション	整数	→	1 = フラグの値を付加、 0 = フラグの値を削除
戻り値	整数	←	エラーコード

説明

IMAP_SetFlags コマンドを使って、指定した範囲のメッセージに付加される複数のフラグを一度に付加または削除することができます。

IMAP プロトコルを使って、フラグのリストをメッセージに付けることができます。フラグには、永久フラグとセッションフラグの2種類があります。

永久フラグは、メッセージフラグに対して永久的に付加または削除されます(「**IMAP_SetCurrentMB**」の節を参照)、つまり、永久フラグの変更はその後のセッションにすべて反映されます。それに対して、セッションフラグへの変更は、変更を行ったセッション中にのみ有効です。

現在定義されているシステムフラグは以下の通りです。

- **Seen** : メッセージが読まれたことを示します。
- **Answered** : メッセージに対して返信したことを示します。
- **Flagged** : メッセージに緊急/特別の注意を促す印が付けられていることを示します。
- **Deleted** : **IMAP_Delete**、**IMAP_CloseCurrentMB**、**IMAP_SetCurrentMB** または **IMAP_Logout** コマンドの使用時にメッセージが削除されることを示します。
- **Draft** : メッセージがドラフト (未完成) であることを示します。
- **Recent** : メッセージがこのメールボックスに"最近"届いたことを示します。このセッションが、このメッセージの到着が通知される初めてのセッションである場合に表示され、その後のセッションにおいてはこのメッセージに対して **\Recent** フラグは表示されません。これは IMAP サーバによって管理される永久フラグであり、**IMAP_SetFlags** コマンド等を使ってクライアントが変更することはできません。

IMAPサーバを使ってクライアントが新しい"フラグ"を定義することや、別のIMAPサーバを使って上記以外のフラグを管理することが可能ですが、IMAPサーバの実装によってはできないこともあります。可能な場合に、こうした特別なフラグは"キーワード"と呼ばれ、"\"記号が前に付きません（「**IMAP_SetCurrentMB**」の節を参照）。

注：\Deleted フラグを付加し、**IMAP_SetCurrentMB**、**IMAP_CloseCurrentMB**、**IMAP_Delete** または **IMAP_Logout** コマンドを実行することにより現在選択されているセッションを終了すると、そのフラグが付加されたメッセージは永久的に"削除"されます。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

<開始メッセージ>は、調べるメッセージ範囲の開始メッセージの番号を指定する倍長整数です。メッセージ番号は、カレントワーキング・メールボックス内のすべてのメッセージのリストにおけるメッセージの位置を表す値です。

<終了メッセージ>は、調べるメッセージ範囲の終了メッセージの番号を指定する倍長整数です。メッセージ番号は、カレントワーキング・メールボックス内のすべてのメッセージのリストにおけるメッセージの位置を表す値です。

注：<開始メッセージ>が<終了メッセージ>よりも大きい番号である場合は、**IMAP_Delete**、**IMAP_MsgLstInfo**、**IMAP_MsgLst**、**IMAP_SetFlags**、**IMAP_GetFlags** および **IMAP_CopyToMB** コマンドはエラーを返しません。このような場合は、コマンドは事実上機能しません。

<フラグリスト>には、1つまたは複数のフラグを指定することができます。複数のフラグを指定する場合は、文字列はスペースで区切ったフラグのリストになります。以下のサンプルを参照してください。

<永久フラグ>としてリストされるフラグ（「**IMAP_SetCurrentMB**」の節を参照）のみが適用されます。

<削除オプション>は、引数<フラグリスト>で指定したフラグの削除または付加を指定する整数値です。

■ 0の場合は、<フラグリスト>に指定したフラグが削除されます。

■ 1の場合は、<フラグリスト>に指定したフラグが付加されます。

▼ <開始メッセージ>および<終了メッセージ>で指定したメッセージに対して、前もって付加されているかどうかに関わらず、\Answered および \Draft フラグを付加する場合。

```
msgFlagsName:="\Answered \Draft"
```

```
 ` \Answered および \Draft はスペース (ASCII コード) で区切ります
```

```
IMAP_SetFlags (imap_ID; 開始メッセージ; 終了メッセージ; メッセージフラグ名; 1)
```


- ▼ <開始メッセージ>および<終了メッセージ>で指定したメッセージに対して、前もって付加されているかどうかに関わらず、\Deleted フラグを削除する場合。

```
msgFlagsName:="\Deleted"
```

```
IMAP_SetFlags (imap_ID; 開始メッセージ; 終了メッセージ; メッセージフラグ名; 0)
```

- ▼ <開始メッセージ>および<終了メッセージ>で指定したメッセージに対して、前もって付加されているかどうかに関わらず、\Deleted フラグを付加する場合。

```
msgFlagsName:="\Deleted"
```

```
IMAP_SetFlags (imap_ID; 開始メッセージ; 終了メッセージ; メッセージフラグ名; 1)
```

```
IMAP_CloseCurrentMB (imap_ID)
```

、カレントメールボックスを閉じると、指定されたメッセージは永久的に削除されます。

- ▼ CheckBoxAnswered の値に応じて\Answered フラグを付加する場合。

```
$Error:= IMAP_SetFlags (vimap_ID; $msgNum; $msgNum;"\Answered"; Num  
(CheckBoxAnswered =0))
```

参照：IMAP_GetFlags、IMAP_SetCurrentMB

IMAP_GetFlags

IMAP_GetFlags (imap_ID; 開始メッセージ; 終了メッセージ; メッセージフラグ配列; メッセージ番号配列) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
開始メッセージ	テキスト	→	開始メッセージの番号
終了メッセージ	テキスト	→	終了メッセージの番号
メッセージフラグ配列	文字列/テキスト配列	←	各メッセージのフラグの値
メッセージ番号配列	文字列/テキスト配列	←	メッセージ番号の配列
戻り値	整数	←	エラーコード

説明

IMAP_GetFlags コマンドは、指定されたメッセージに対するフラグのリストを返します。

<imap_ID>は **IMAP_Login** によって開始された接続のリファレンスとなる倍長整数です。

<開始メッセージ>は、調べるメッセージ範囲の開始メッセージの番号を指定する倍長整数です。メッセージ番号は、カレントワーキング・メールボックス内のすべてのメッセージのリストにおけるメッセージの位置を表す値です。

<終了メッセージ>は、調べるメッセージ範囲の終了メッセージの番号を指定する倍長整数です。メッセージ番号は、カレントワーキング・メールボックス内のすべてのメッセージのリストにおけるメッセージの位置を表す値です。

注：<開始メッセージ>が<終了メッセージ>よりも大きい番号である場合は、**IMAP_Delete**、**IMAP_MsgLstInfo**、**IMAP_MsgLst**、**IMAP_SetFlags**、**IMAP_GetFlags** および **IMAP_CopyToMB** コマンドはエラーを返しません。このような場合は、コマンドは事実上機能しません。

<メッセージフラグ配列>は、<開始メッセージ>から<終了メッセージ>までの間の各メッセージ番号の、スペースで区切られたフラグのリストが返される文字列配列またはテキスト配列です。

<メッセージ番号配列>は、<開始メッセージ>から<終了メッセージ>までの間のメッセージの番号が返される倍長整数配列です。

参照： **IMAP_SetFlags**、**IMAP_SetCurrentMB**

IMAP_MsgFetch

IMAP_MsgFetch (imap_ID; メッセージ番号; データ項目; データ項目値) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP 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
データ項目	テキスト	→	取り出すデータ項目
データ項目値	テキスト	←	データ項目の値
戻り値	整数	←	エラーコード

説明

IMAP_MsgFetch コマンドを使って、メッセージをダウンロードすることなく指定したメッセージに対する1つまたは複数の基本データ項目を要求することができます。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

<メッセージ番号>はチェックするメッセージを示す倍長整数値です。この値は、現在選択されているメッセージリスト内のメッセージの位置を表します。ただし、特定の電子メールについて、その<メッセージ番号>はセッションが異なれば同じであるとは限りません。

<データ項目>は、1つまたは複数の取り出したいデータ項目を示すテキスト変数です。複数のデータ項目を指定する場合は、各項目をスペースで区切ります。データ項目には次の2種類があります。

- 基本データ項目：1つの情報のみを取り出します
- マクロデータ項目：一度に複数の基本データ項目を指定する概念から生じ、複数の情報を取り出します。3つのマクロが一般に使用される一連のデータ項目を指定し、そのデータ項目の代わりとして使用することができます。マクロは単独で使用し、その他のマクロやデータ項目と一緒に使用することはできません。

データ項目に関する詳細は、以下の"基本データ項目"および"マクロデータ項目"の節を参照してください。

<データ項目値>は引数<データ項目>の値に応じて、データ項目/データ項目値の1セット、またはデータ項目/データ項目値のセットのリストを返すテキスト変数です。

- データ項目/データ項目値が1セットの場合は、返されるテキストの構造は次のようになります。

データ項目名 + スペース + データ項目値

- データ項目／データ項目値のセットのリストの場合は、返されるテキストの構造は次のようになります。

データ項目名1 + スペース + データ項目値1 + スペース + データ項目名2 + スペース + データ項目値2

- <データ項目値>には、引数<データ項目>に応じて、カッコ付きのリスト、引用符付き文字列または単独の文字列が入ります。

- カッコ付きのリストの場合の構造は次のようになります（例として、**FLAGS**の場合を参照）：（1番目のデータ項目値 + スペース + 2番目のデータ項目値）

カッコ付きのリストのカッコのみが返される場合は、項目値がないことを意味します。しかし、この規則はアドレスのカッコ付きリストにはあてはまりません（**ENVELOPE**を参照）。

- 引用符付き文字列の場合の構造は次のようになります（例として、**INTERNALDATE**の場合を参照）：データ項目名 + スペース + 引用符 + データ項目値 + 引用符

データ項目値に""が返される場合は、空の文字列であることを意味します。

- 引用符が付かない文字列は、整数、倍長整数または数値を表し、構造は次のようになります：データ項目名 + スペース + データ項目値

この場合は、適切なタイプに変換する必要があります（例として、**UID**の場合を参照）。

注：引用符は一般的に、文字列値にスペースやカッコ等の特殊文字が含まれる場合に使用されます。そのため、**IMAP_Fetch** コマンドの結果として生じる文字列を解析する場合は、文字列の中身を処理する際に引用符に注意してください。

基本データ項目

■ INTERNALDATE

IMAPサーバのメッセージの内部の日付と時刻を取り出します。これは"日付"ヘッダが返す日付と時刻ではなく、メッセージをいつ受け取ったのかを示す日付と時刻です。SMTPサーバを通して届けられるメッセージについては、この日付は通常メッセージが最後に届いた日付と時刻を表します。IMAP_Copy コマンド後に送信されるメッセージについては、この日付は通常ソースメッセージの内部の日付と時刻を表します。

INTERNALDATE データ項目値は引用符付き文字列を返します。

例：

```
msgDataItem:="INTERNALDATE"
```

```
$Err: IMAP_MsgFetch (imap_ID; 1; データ項目; データ項目値)
```

データ項目は次の値を返します：INTERNALDATE"17-Jul-2001 15:45:37 +0200"

■ FLAGS

指定したメッセージに付いているフラグのかっこ付きのリストを取り出します。フラグの値はスペースで区切られます。

例：

```
msgDataItem:="FLAGS"
```

```
$Err:= IMAP_MsgFetch (imap_ID; 1; データ項目; データ項目値)
```

指定したメッセージにフラグが付加されていない場合は、メッセージデータ項目は次の値を返します：FLAGS ()

指定したメッセージに\Seen フラグおよび\Answered フラグが付加されている場合は、データ項目は次の値を返します：FLAGS (\Seen \Answered)

■ RFC822.SIZE

RFC-822 フォーマットで表されるメッセージのバイト数を取り出します。データ項目と返される値はスペースで区切られます。引用符が付かない文字列が返される場合は、この文字列を倍長整数値に変換する必要があります (UID の例を参照)。

例：

```
msgDataItem:="RFC822.SIZE"
```

```
$Err:= IMAP_MsgFetch (imap_ID; 1; データ項目; データ項目値)
```

データ項目は次の値を返します：RFC822.SIZE 99599

■ ENVELOPE

指定したメッセージのヘッダ部分を示すかっこ付きのリストを取り出します。サーバはメッセージのヘッダを解析し、必要であれば様々なフィールドをデフォルトにしてこの値を算出します。

ヘッダフィールドは次のような順序で返されます：日付、件名、差出人、送信人、返信先、送信先、カーボンコピー、ブラインドカーボン、返信先テキスト、メッセージ ID。日付、件名、返信先テキストおよびメッセージ ID のフィールドは、次のような引用符付き文字列です：ENVELOPE ("日付" "件名" (差出人) (送信人) (返信先) (送信先) (カーボンコピー) (ブラインドカーボン) "返信先テキスト" "メッセージ ID")

例：

```
msgDataItem:="ENVELOPPE"
```

```
$Err:= IMAP_MsgFetch (imap_ID; 1; データ項目; データ項目値)
```

データ項目は次の値を返します：ENVELOPE ("Tue, 17 Jul 2001 17:26:34 +0200" "Test" (("RSmith" NIL "RSmith" "test")) (("RSmith" NIL "RSmith" "test")) (("RSmith" NIL "RSmith" "test")) (("RSmith" NIL "RSmith" "test")) () () "" "<ee6b33a.-1@Mail.x6foadRIbnm>")

日付:	"Tue, 17 Jul 2001 17:26:34 +0200"	日付ヘッダ
件名:	"Test"	件名ヘッダ
差出人:	(("RSmith" NIL "RSmith" "test"))	アドレス構造
送信人:	(("RSmith" NIL "RSmith" "test"))	アドレス構造
返信先:	(("RSmith" NIL "RSmith" "test"))	アドレス構造
送信先:	(("RSmith" NIL "RSmith" "test"))	アドレス構造
カーボンコピー:	()	カーボンコピーヘッダは未使用
ブラインドカーボン:	()	ブラインドカーボンは未使用
返信先テキスト:	""	返信先テキストヘッダ
メッセージID:	"<ee6b33a-1@Mail.x6foadRlbnm>"	メッセージIDヘッダ

差出人、送信人、返信先、送信先、カーボンコピーおよびブラインドカーボンのフィールドは、アドレス構造のかっこ付きのリストです。アドレスの構造は、電子メールアドレスを示すかっこ付きのリストです。アドレス構造のフィールドは次のような順序になります：個人名、[SMTP] at-domain-list（ソースルート）、メールボックス名、ホスト名。例：(("RSmith" NIL "RSmith" "test"))

([RFC-822]より引用) ホスト名フィールドがNIL（空）の場合は、グループシンタックスはアドレス構造の特別なフォームで示されます。メールボックス名フィールドもNIL（空）の場合は、これはグループの最後を示すマーカです（RFC 822シンタックスではセミコロン）。メールボックス名フィールドがNIL（空）ではない場合は、これはグループの最初を示すマーカであり、メールボックス名フィールドにはグループ名が入ります。

適用できないエンベロープまたはアドレス構造のフィールドはNIL（空）として表示されます。サーバは、必ず"差出人"フィールドから返信先および送信人フィールドをデフォルトにしなければならない点に注意してください。したがって、クライアントにはこれを行う方法がわかりません。

■ BODY:

拡張データは返されませんが、それを除いてBODYSTRUCTUREと同じ情報を返します（「BODYSTRUCTURE」の節を参照）。

例:

```
msgDataItem:="BODY"
```

```
$Err:= IMAP_MsgFetch (imap_ID; 1;データ項目;データ項目値)
```

データ項目は次の値を返します：BODY ("TEXT" "PLAIN" ("CHARSET" "us-ascii") NIL NIL "8BIT" 8 1)

■ BODYSTRUCTURE:

メッセージのMIME本文構造を取り出します。サーバは、メッセージヘッダのMIMEヘッダフィールドおよび本文部分のMIMEヘッダを解析することによりこの値を算出します。このデータ項目は、ダウンロードを行わずにメッセージのコンテンツを走査する際に特に有効です。例えば、各パートのサイズをすばやく確認することや、添付ファイル名のみを確認することができます。

BODYSTRUCTUREは、かっこ付きのリスト、引用符付き文字列、および引用符がつかない文字列を含むかっこ付きのリストを返します。

メッセージのコンテンツに応じて、BODYSTRUCTUREは"非マルチパート"のかっこ付きリストまたは入れ子のリスト("マルチパート"のかっこ付きリスト)を返します。

■ "非マルチパート"のかっこ付きリスト：これは、例えば、非マルチパートの電子メールに似ており、48行2279バイトのシンプルテキストメッセージの本文構造は次のようになります：("TEXT" "PLAIN" ("CHARSET" "us-ascii") NIL NIL "8BIT" 8 1 NIL NIL NIL)

"非マルチパート"のかっこ付きリストの基本フィールドは、次のような順序になります。

本文タイプ

コンテンツのメディアタイプ名を与える文字列 (Content-type: media type e.g. TEXT)

本文サブタイプ

コンテンツのサブタイプ名を与える文字列 (Content-type: sub type e.g. PLAIN)

本文引数

属性/値のセットのかっこ付きのリスト [e.g.

かっこ付きのリスト

("CHARSET" "US-ASCII" "NAME" "cc.diff")において"US-ASCII"は"CHARSET"の値で、"cc.diff"は"NAME"の値です。

本文ID

ある本文を別の本文と区別するためのリファレンスとなるコンテンツのIDを与える文字列です。したがって、"Content-ID"ヘッダフィールドを使って本文にラベルを付けることができます。マルチパート/選択メディアタイプの場合は、コンテンツIDは特別な意味を持ちます。これについては、マルチパート/選択の場合を取り扱っているRFC 2046の節で説明されています。

本文説明

コンテンツの説明を与える文字列

本文エンコーディング

コンテンツの転送エンコーディングを与える文字列 (Content-Transfer-Encoding)

本文サイズ

本文のサイズをバイトで与える数値。これは転送エンコーディング中のサイズであり、デコード後の結果のサイズではないことに注意してください。

- メッセージタイプまたはRFC822サブタイプの本文には、基本フィールドのすぐ後に続いて、エンベロープ構造、本文構造およびカプセル化されたメッセージのテキスト行でのサイズが入ります。

- テキストタイプの本文には、基本フィールドのすぐ後に続いて、テキスト行での本文のサイズが入ります。これはコンテンツの転送エンコーディング中のサイズであり、デコード後の結果のサイズではないことに注意してください。

拡張データは、基本フィールドおよび前のページのタイプ特定フィールドに続きます。拡張データはBODYフェッチで返されることはありませんが、BODYSTRUCTUREフェッチで返されることはあります。

存在する場合は"非マルチパート"のかっこ付きのリストである拡張データは、必ず次のように定義された順序に並びます。

本文MD5

[MD5]で定義された本文MD5の値を与える文字列

本文性質

性質の属性／値のセットのかっこ付きのリストが後に続く、[DISPOSITION]で定義された性質タイプの文字列からなるかっこ付きのリスト

本文言語

[LANGUAGE-TAGS]で定義された本文言語の値を与える文字列またはかっこ付きのリスト

後に続く拡張データは、このバージョンのプロトコルではまだ定義されていませんが、前述の通りマルチパート拡張データの下になります。

例：

```
("TEXT" "PLAIN" ("CHARSET" "US-ASCII") NIL NIL "7BIT" 2279 48 NIL NIL NIL)
```

説明：

```
("bodytype" "bodysubtype" (BodyParameterParenthesizedList) bodyId  
bodyDescription "bodyEncoding" BodySize BodySizeInTextLines  
ExtensionDataBODYmd5 ExtensionDataBodyDisposition ExtensionDa-  
taBodyLanguage)
```

- "マルチパート"のかっこ付きのリスト：これはマルチパートの電子メールの場合で、"非マルチパート"のかっこ付きのリストが入ります。

かっこで囲まれた箇所は複数のパートを示します。かっこ付きのリストの最初の要素は本文のタイプではなく入れ子の本文です。かっこ付きのリストの2番目の要素はマルチパートのサブタイプ（混合、要約、並列、選択等）です。

マルチパートのサブタイプの後には拡張データが続きます。拡張データが存在する場合は必ず次のように定義された順序に並びます。

本文引数

属性／値のセットのかっこ付きのリスト

かっこ付きリスト

本文性質

性質の属性／値のセットのかっこ付きのリストが後に続く、[DISPOSITION]で定義された性質タイプの文字列からなるかっこ付きのリスト

本文言語

[LANGUAGE-TAGS]で定義された本文言語の値を与える文字列またはかっこ付きのリスト

後に続く拡張データはこのバージョンのプロトコルではまだ定義されていません。この拡張データには、0以上のNIL（空）、文字列、数値またはそのようなデータの潜在的に入れ子のかっこ付きリストが入ります。クライアントがBODYSTRUCTUREフェッチを行う実装の場合は、そのような拡張データを受け付ける用意ができています。サーバの実装では、このプロトコルの改訂によって定義されるまでそのような拡張データを送信してはいけません。

例：

```
BODYSTRUCTURE (("TEXT" "PLAIN" ("CHARSET" "us-ascii") NIL NIL "7BIT" 22 1
NIL NIL NIL)("APPLICATION" "BYTE-STREAM" ("NAME" "casta37.jpg" "X-MAC-
TYPE" "4A504547" "X-MAC-CREATOR" "6F676C65") NIL NIL "BASE64" 98642 NIL
("ATTACHMENT" ("FILE-NAME" "casta37.jpg")) NIL) "MIXED" ("BOUNDARY"
"4D_====1385356==") NIL NIL)
```

説明：

```
(("bodytype" "bodysubtype" (BodyParameterParenthesizedList) bodyId
bodyDescription "bodyEncoding" BodySize BodySizeInTextLines
ExtensionDataBODYmd5 ExtensionDataBodyDisposition
ExtensionDataBodyLanguage) ("bodytype" "bodysubtype"
(BodyParameterParenthesizedList) bodyId bodyDescription "bodyEncoding"
BodySize BodySizeInTextLines ExtensionDataBODYmd5
ExtensionDataBodyDisposition ExtensionDataBodyLanguage) "multipartSubtype"
(Extension-DataBodyParameterList) ExtensionDataBodyDisposition
ExtensionDataBodyLanguage))
```

■ UID

メッセージのユニーク ID を表す数値を取り出します。これは IMAP_UIDToMsgNum の実行に相当します。この数値はテキストエリアに返されるため、倍長整数に変換する必要があります。

例：

```
msgDataItem:="UID"
```

```
$Err:= IMAP_MsgFetch (imap_ID; 1; データ項目; データ項目値)
```

データ項目値は次の値を返します：UID 250000186

倍長整数を取り出すには次のように行います。

```
C_LONGINT (vLongint)
```

```
VLongint:= Num ("250000186")
```

マクロデータ項目

■ FAST

次に相当するマクロ：(FLAGS INTERNALDATE RFC822.SIZE)

例：

```
$Err:= IMAP_MsgFetch (imap_ID; メッセージ番号; "FAST"; データ項目値)
```

データ項目値は次の値を返します："FLAGS (\Seen \Answered) INTERNALDATE "17-Jul-2001 15:45:37 +0200" RFC822.SIZE 99599"

■ ALL

次に相当するマクロ：(FLAGS INTERNALDATE RFC822.SIZE ENVELOPE)

■ FULL

次に相当するマクロ：(FLAGS INTERNALDATE RFC822.SIZE ENVELOPE BODY)

参照：IMAP_UIDToMsgNum、IMAP_SetFlags

IMAP_Download

IMAP_Download (imap_ID; メッセージ番号; ヘッダのみ; ファイル名; 既読更新)
→ 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP 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=メッセージ全体、1=ヘッダのみ
ファイル名	テキスト	→	ローカルファイル名
		←	結果として生じるローカルファイル名
既読更新	整数	→	0=Seen フラグを付加、 1=Seen フラグを付加しない
戻り値	整数	←	エラーコード

説明

IMAP_Download コマンドは、ディスクベースのファイルにダウンロードすることにより、IMAPサーバからメッセージを取り出すように設計されています。添付ファイルを含むものやサイズが32Kを超えるIMAPメッセージはすべて、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。添付ファイルは、この方法で取り出されたメッセージからのみ展開することができます。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

<メッセージ番号>は、メールボックス内の取り出したいメッセージを示す倍長整数値です。<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内のメッセージの位置を表します。ただし、特定の電子メールについて、その<メッセージ番号>はセッションが異なれば同じであるとは限りません。

<ヘッダのみ>は、メッセージの全内容を取り出すかまたはヘッダ情報のみを取り出すかを示す整数値です。

<ファイル名>には、ファイル名およびオプションでメッセージを保存したい場所を示すパスが入ります。この値は、次のような3通りの方法で指定することができます。

■ "" = **IMAP_SetPrefs** コマンドで設定したフォルダのファイルを“temp1”（同じ名前のファイルがすでに存在する場合は、使用されていないファイル名が見つかるまで“temp2”、“temp3”…のように順番に試されます）という名前でも保存します。

■ "ファイル名" = **IMAP_SetPrefs** コマンドで設定したフォルダのファイルを<ファイル名>として保存します。

- "パス: ファイル名" = <ファイル名>の名前で指定されたパスにファイルを保存します。

1番目と2番目の指定方法では、**IMAP_SetPrefs** コマンドで指定されたフォルダがない場合は、メッセージはデータベースのストラクチャと同じフォルダ（4D シングルユーザの場合）または4D Client フォルダ（4D Server の場合）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>として渡した変数に返されます。ダウンロードフォルダ内にすでに存在する<ファイル名>で**IMAP_Download** コマンドを呼び出そうとすると、そのファイル名の数字が増していき、ディスクに保存された時点の新しい値が<ファイル名>変数に返されます。

<既読更新>は、暗黙的であるかどうかに関わらず、\Seen フラグをメッセージフラグとして付加するかどうかを示す整数値です。この引数はオプションで、値を渡さない場合はデフォルト値が使用されます。

- 0 = \Seen フラグを付加
- 1 = \Seen フラグを付加しない

値を指定しない場合はデフォルト値として0 が設定され、\Seen フラグが暗黙的に付加されます。

参照：IMAP_SetPrefs、IMAP_GetMessage

IMAP_UIDToMsgNum

IMAP_UIDToMsgNum (imap_ID; メッセージID; メッセージ番号) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
ユニーク ID	倍長整数	→	メッセージのユニーク ID の値
メッセージ番号	倍長整数	←	メッセージ番号
戻り値	整数	←	エラーコード

説明

IMAP_UIDToMsgNum コマンドは、メッセージのユニーク ID の値を <imap_ID> で参照されるカレントメールボックスのメッセージリスト内での現在の <メッセージ番号> に変換します。特定の電子メールメッセージの <メッセージ番号> は電子メールリスト内の他のメッセージとの関係で変わる流動的な値であるため、このコマンドは先の IMAP セッション中に情報が取り出されたメッセージの現在の位置を返します。

<imap_ID> は **IMAP_Login** によって開始された接続のリファレンスとなる倍長整数です。

ユニーク ID は、メッセージの固有の ID が IMAP サーバに置かれることを示している長い整数値です。

<メッセージ番号> には、<ユニーク ID> で識別されるメッセージ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での位置）を示す倍長整数値が返されます。<ユニーク ID> がサーバ上に見つからない場合は、<メッセージ番号> に 0 が返され、エラーは返されません。

参照：IMAP_SetCurrentMB、IMAP_MsgNumToUID

IMAP_MsgNumToUID

IMAP_MsgNumToUID (imap_ID; メッセージ番号; メッセージID) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP 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
ユニーク ID	倍長整数	←	メッセージのユニーク ID の値
戻り値	整数	←	エラーコード

説明

IMAP_MsgNumToUID コマンドは、<imap_ID>で参照されるカレントメールボックスのメッセージリスト内でのメッセージ番号をそのメッセージの現在のユニーク ID の値に変換します。

<imap_ID>は **IMAP_Login** によって開始された接続のリファレンスとなる倍長整数です。

<メッセージ番号>には、<ユニーク ID>で識別されるメッセージ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での位置）を示す倍長整数値が返されます。<ユニーク ID>がサーバ上に見つからない場合は、<メッセージ番号>に0が返され、エラーは返されません。

<ユニーク ID>は、メッセージのIMAPサーバ上での位置を決定するユニーク ID を返す倍長整数値です。

参照：IMAP_SetCurrentMB、IMAP_UIDToMsgNum

IMAP_Search

IMAP_Search (imap_ID; 検索基準; メッセージ番号配列) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインへのリファレンス
検索基準	テキスト	→	検索基準
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
戻り値	整数	←	エラーコード

説明

IMAP_Search コマンドは、カレントメールボックスで与えられた検索基準に一致するメッセージを検索します。〈検索基準〉は1つまたは複数の検索キーで構成されます。〈メッセージ番号配列〉は、検索基準に一致するメッセージに対応するメッセージ通番のリストを返します。

〈imap_ID〉は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

〈検索基準〉は、探す値に関連するかどうかに関わらず、1つまたは複数の検索キー（この節の最後の「承認検索キー」を参照）をリストするテキスト引数です。検索キーは、1つのアトムでも1つまたは複数の検索キーを含むかっこ付きのリストでも可能です。以下はその例です。

検索キー1 = フラグ付き
 検索キー2 = フラグなし
 検索キー3 = フラグ付き ドラフト

注：マッチングにおいて通常大文字と小文字は区別されません。

■ 〈検索基準〉が空の文字列の場合は、次の例のように検索は“すべてを選択”に相当します。

IMAP_Search (imap_ID;"";メッセージ番号配列)

…カレントワーキング・メールボックスのすべてのメッセージを返します。

■ 〈検索基準〉に複数の検索キーが含まれる場合は、返される値はこれらのキーに一致するすべてのメッセージの集合（AND関数）になります。

検索基準 = FLAGGED FROM "SMITH"

…\Flagged フラグが付加されており、スミスさんから送られたすべてのメッセージを返します。

- OR または NOT 演算子を次のように使用することができます。

検索基準 = OR SEEN FLAGGED

…\Seen フラグまたは\Flagged フラグが付加されているすべてのメッセージを返します。

検索基準 = NOT SEEN

…\Seen フラグが付加されていないすべてのメッセージを返します。

検索基準 = HEADER CONTENT-TYPE "MIXED" NOT HEADER CONTENT-TYPE "TEXT"...

…コンテンツタイプヘッダに “Mixed” が含まれ、“Text” が含まれないメッセージを返します。

検索基準 = HEADER CONTENT-TYPE "E" NOT SUBJECT "o" NOT HEADER CONTENT-TYPE "MIXED"

…コンテンツタイプヘッダに “e” が含まれ、件名ヘッダに “o” が含まれず、コンテンツタイプヘッダが “Mixed” ではないメッセージを返します。

検索基準 = OR (ANSWERED SMALLER 400) (HEADER CONTENT-TYPE "E" NOT SUBJECT "o" NOT HEADER CONTENT-TYPE "MIXED")

…1 番目または2 番目のかっこ付きリストに一致するメッセージを返します。

検索基準 = OR ANSWERED SMALLER 400 (HEADER CONTENT-TYPE "E" NOT SUBJECT "o" NOT HEADER CONTENT-TYPE "MIXED")

…\Answered フラグが付加されているまたはサイズが400 バイトよりも小さく、指定のかっこ付きリストに一致するメッセージを返します。

最後の2つの例に関して、1 番目の検索キーのリストのかっこを外すと検索結果が変わる点に注意してください。

- <検索基準>には、オプションの[CHARSET]指定を入れることも可能です。これは、“CHARSET” の後に続けて登録されている[CHARSET] (US ASCII, ISO-8859) を指定します。これが検索基準文字列の文字コードセットを示します。したがって、[CHARSET]指定 (4th Dimension の Mac コマンドから ISO コマンドへを参照) を使用する場合は、検索基準文字列を指定された文字コードセットに変換する必要があります。

デフォルトでは、拡張文字が含まれる場合に4D Internet Commands は引用可／印刷可で検索基準文字列をエンコードします。

検索基準 = CHARSET "ISO-8859" BODY "Help"

…検索基準がiso-8859 文字コードセットを使用し、サーバは必要に応じて検索前に検索基準を変換する必要があることを意味します。

検索値タイプ

検索キーには、値を要求して検索することができます。

■ 日付の値を使用する検索キー

<日付>は、次のようにフォーマットされた文字列にしなければいけません：`date-day+"-"+date-month+"-"+date-year` このフォーマットにおいて、`date-day`は日にち（最大2文字）、`date-month`は月（Jan/Feb/Mar/Apr/May/June/Jul/Aug/Sep/Oct/Dec）、`date-year`は年（最大4文字）を示します。

例：検索条件 = `SENTBEFORE 1-Feb-2000`（日付には特殊文字が含まれることはないため、通常は引用符を付ける必要はありません）

■ 文字列の値を使用する検索キー

<文字列>にはあらゆる文字を入れることができ、引用符を付けて指定します。しかし、スペースのように文字列に特殊文字が含まれない場合は、引用符を付ける必要はありません。文字列に引用符をつけることにより、文字列の値は確実に正しく解釈されます。

例：検索基準 = `FROM "SMITH"`

注：文字列を使用するすべての検索キーについて、文字列がフィールドの部分文字列である場合はメッセージはキーに一致します。また、マッチングにおいて大文字と小文字は区別されません。

■ フィールド名の値を使用する検索キー

<フィールド名>はヘッダフィールドの名前です。

例：検索基準 = `HEADER CONTENT-TYPE "MIXED"`

■ フラグの値を使用する検索キー

<フラグ>には、1つまたは複数のキーワード（標準フラグを含む）をスペースで区切って指定することができます。

例：検索基準 = `KEYWORD \Flagged \Draft`

■ メッセージ設定の値を使用する検索キー

一連のメッセージを識別します。メッセージ通番は、1からメールボックス内のメッセージの合計数までの連続番号です。

コンマで個々の番号を区切り、2つの番号の間の番号を含める場合はコロンで区切ります。

例：

2,4:7,9,12:* は15個のメッセージが含まれるメールボックスの場合には、2、4、5、6、7、9、12、13、14、15を意味します。

検索基準 = 1:5 ANSWERED はメッセージ通番が1番から5番までのメッセージの中から \Answered フラグが付加されているものを検索します。

検索基準 = 2,4 ANSWERED は選択されたメッセージ（メッセージ番号2および4）から \Answered フラグが付加されているものを検索します。

承認検索キー

- ALL：メールボックス内のすべてのメッセージ
- ANSWERED：\Answered フラグが付加されているメッセージ
- UNANSWERED：\Answered フラグが付加されていないメッセージ
- DELETED：\Deleted フラグが付加されているメッセージ
- UNDELETED：\Deleted フラグが付加されていないメッセージ
- DRAFT：\Draft フラグが付加されているメッセージ
- UNRAFT：\Draft フラグが付加されていないメッセージ
- FLAGGED：\Flagged フラグが付加されているメッセージ
- UNFLAGGED：\Flagged フラグが付加されていないメッセージ
- RECENT：\Recent フラグが付加されているメッセージ
- OLD：\Recent フラグが付加されていないメッセージ
- SEEN：\Seen フラグが付加されているメッセージ
- UNSEEN：\Seen フラグが付加されていないメッセージ
- NEW：\Recent フラグが付加されており、\Seen フラグが付加されていないメッセージ。
これは機能的には“(RECENT UNSEEN)”に相当します。
- KEYWORD <フラグ>：指定したキーワードが付加されているメッセージ
- UNKEYWORD <フラグ>：指定したキーワードが付加されていないメッセージ
- BEFORE <日付>：内部日付が指定した日付よりも前のメッセージ
- ON <日付>：内部日付が指定した日付の範囲内であるメッセージ
- SINCE <日付>：内部日付が指定した日付の範囲内またはそれよりも後のメッセージ
- SENTBEFORE <日付>：日付ヘッダが指定した日付よりも前のメッセージ
- SENTON <日付>：日付ヘッダが指定した日付の範囲内であるメッセージ
- SENTSINCE <日付>：日付ヘッダが指定した日付の範囲内またはそれよりも後の
メッセージ

- TO<文字列>：送信先ヘッダに指定した文字列を含むメッセージ
- FROM<文字列>：差出人ヘッダに指定した文字列を含むメッセージ
- CC<文字列>：カーボンコピーヘッダに指定した文字列を含むメッセージ
- BCC<文字列>：ブラインドカーボンヘッダに指定した文字列を含むメッセージ
- SUBJECT<文字列>：件名ヘッダに指定した文字列を含むメッセージ
- BODY<文字列>：メッセージ本文に指定した文字列を含むメッセージ
- TEXT<文字列>：ヘッダまたはメッセージ本文に指定した文字列を含むメッセージ
- HEADER<フィールド名><文字列>：ヘッダに指定したフィールド名を持ち、フィールド本文に指定した文字列を含むメッセージ
- UID<メッセージID>：指定したユニークIDに一致するユニークIDを持つメッセージ
- LARGER<n>：指定したバイト数よりも大きいサイズのメッセージ
- SMALLER<n>：指定したバイト数よりも小さいサイズのメッセージ
- NOT<検索キー>：指定した検索キーに一致しないメッセージ
- OR<検索キー1><検索キー2>：検索キーのどちらかに一致するメッセージ

<メッセージ番号配列>は、検索基準に一致するメッセージに対応するメッセージ通番のリストを返します。

参照：IMAP_SetFlags、IMAP_GetFlags

IMAP_CopyToMB

IMAP_CopyToMB (imap_ID; 開始メッセージ; 終了メッセージ; ターゲットメールボックス名 {; メッセージ削除}) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
開始メッセージ	倍長整数	→	開始メッセージの番号
終了メッセージ	倍長整数	→	終了メッセージの番号
ターゲットメールボックス名	テキスト	→	配置先メールボックス名
メッセージ削除	整数	→	0= ソースメールボックスから削除しない 1= ソースメールボックスから削除する
戻り値	整数	←	エラーコード

説明

<開始メッセージ>から<終了メッセージ>までのメッセージ範囲を与えると、**IMAP_CopyToMB** コマンドは、<ターゲットメールボックス名>で指定した配置先メールボックスの最後に指定したメッセージをコピーします。メッセージのフラグや内部日付は、IMAPサーバの実装に依存しますが、通常は配置先メールボックスで保存されます。

オリジナルのメッセージは、コピー後もソースメールボックスから削除されません。削除したい場合は、次の3つの処理方法のうちの1つを使って行うことができます。

■ **IMAP_Delete** コマンドを使用する

■ オプションの引数<メッセージ削除>を1に設定する

■ **IMAP_SetFlags** (\Deleted)を付加する：セッションを終了するとメッセージが削除されます。

注：引数<メッセージ削除>は強制的に**IMAP_Delete**を実行します。その際、<開始メッセージ>と<終了メッセージ>の間のメッセージおよび\Deletedフラグが付加されているすべてのメッセージが削除されます。

配置先メールボックスが存在しない場合は、エラーが返されます。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

<開始メッセージ>は、コピーするメッセージ範囲の開始メッセージの番号を指定する倍長整数です。メッセージ番号は、<imap_ID>で識別されるメールボックス内のすべてのメッセージのリストにおけるメッセージの位置を表す値です。

<終了メッセージ>は、コピーするメッセージ範囲の終了メッセージの番号を指定する倍長整数です。メッセージ番号は、<imap_ID>で識別されるメールボックス内のすべてのメッセージのリストにおけるメッセージの位置を表す値です。

注：<開始メッセージ>が<終了メッセージ>よりも大きい番号である場合は、IMAP_Delete、IMAP_MsgLstInfo、IMAP_MsgLst、IMAP_SetFlags、IMAP_GetFlagsおよびIMAP_CopyToMB コマンドはエラーを返しません。このような場合は、コマンドは事実上機能しません。

<ターゲットメールボックス名>は、指定したメッセージがコピーされるメールボックスの完全な名前です。

ソースメールボックスからメッセージを削除したい場合は、オプションの引数<メッセージ削除>を設定することにより行うことができます。

- 0 = ソースメールボックスから削除しない（デフォルト値）
- 1 = ソースメールボックスから削除する

<メッセージ削除>を省略すると、デフォルト値が使用されます。

コピーに失敗した場合は、メッセージはソースメールボックスから削除されません。

ユーザがメッセージを削除するアクセス権を持たない場合は、エラーメッセージが生成されます。

参照：IMAP_CreateMB、IMAP_RenameMB、IMAP_ListMBs

IMAP_CreateMB

IMAP_CreateMB (imap_ID; メールボックス名) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
メールボックス名	テキスト	→	作成するメールボックス名
戻り値	整数	←	エラーコード

説明

IMAP_CreateMB コマンドは、与えられた名前で作成するメールボックスを作成します。IMAP サーバの階層セパレータ文字が指定したメールボックス名に含まれる場合は、IMAP サーバは与えられた名前のメールボックスを作成するのに必要な親名を作成します。

例として、“/” が階層セパレータ文字であるサーバ上に “Projects/IMAP/Doc” という名前のメールボックスを作成しようとする、次のようになります。

- “Projects” および “IMAP” というメールボックスがすでに存在する場合は、“Doc” メールボックスのみを作成します。
- “Projects” というメールボックスのみがすでに存在する場合は、“IMAP” および “Doc” メールボックスを作成します。
- どのメールボックスも存在しない場合は、“Projects”、“IMAP” および “Doc” メールボックスを作成します。

<imap_ID>は **IMAP_Login** によって開始された接続のリファレンスとなる倍長整数です。

<メールボックス名>は、作成するメールボックスの完全な名前です（IMAP 導入の命名規則を参照）。

注：INBOX（“このサーバ上のこのユーザにとっての一次メールボックス”として確保される特別なメールボックス名）または既存のメールボックスを指す名前のメールボックスを作成しようとする、エラーになります。

参照：IMAP_RenameMB、IMAP_ListMBs

IMAP_DeleteMB

IMAP_DeleteMB (imap_ID; メールボックス名) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
メールボックス名	テキスト	→	削除するメールボックス名
戻り値	整数	←	エラーコード

説明

IMAP_DeleteMB コマンドは与えられた名前のメールボックスを永久的に削除します。INBOX または存在しないメールボックスを削除しようとするとエラーが発生します。

IMAP_DeleteMB コマンドでは、子メールボックスおよび \Noselect メールボックス属性を持つメールボックスを削除することはできません。

子名を持ち、\Noselect メールボックス属性を持たないメールボックスを削除することは可能です。この場合、メールボックス内のすべてのメッセージは削除され、メールボックスは \Noselect メールボックス属性を取得します。

注：IMAP プロトコルは空ではないメールボックスの削除を保証しませんが、サーバによってはこれを認めるものもあります。この削除処理を試みる場合は、万が一便利な方法が失敗した場合に別の方法を使用する用意をしておく必要があります。また、カレントワーキング・メールボックスを開いている状態で削除しようとせず、必ず先に閉じてください。サーバによってはカレントメールボックスの削除を許可しないものもあります。

<imap_ID> は **IMAP_Login** によって開始された接続のリファレンスとなる倍長整数です。

<メールボックス名> は削除するメールボックスの完全な名前です。

参照：IMAP_Delete、IMAP_SetCurrentMB、IMAP_CloseCurrentMB

IMAP_RenameMB

IMAP_RenameMB (imap_ID; メールボックス名; 新しいメールボックス名) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
メールボックス名	テキスト	→	名前を変更するメールボックスの名前
新しいメールボックス名	テキスト	→	新しいメールボックス名
戻り値	整数	←	エラーコード

説明

IMAP_RenameMB コマンドはメールボックスの名前を変更します。存在しないメールボックス名の変更や、すでに存在するメールボックス名への変更を試みるとエラーが発生します。

注：INBOXの名前を変更することは可能ですが特別な処理が必要です。その方法は、指定した名前の新しいメールボックスにINBOX内のすべてのメッセージを移動し、INBOXを空にします。サーバの実装がINBOXの子名を認める場合でも、名前の変更による子名への影響はありません。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

<メールボックス名>は、名前を変更するメールボックスの完全な名前です（IMAP導入の命名規則を参照）。

<新しいメールボックス名>は、<メールボックス名>を変更するのに使用される完全な名前です。

参照：IMAP_Login、IMAP_CreateMB、IMAP_ListMBs

IMAP_Logout

IMAP_Logout (imap_ID) → 整数

引数	タイプ		説明
imap_ID	倍長整数	→	IMAP ログインのリファレンス
		←	0 = コマンドがログアウトに成功
戻り値	整数	←	エラーコード

説明

IMAP_Logout コマンドは、<imap_ID>変数によって参照される IMAP 接続をログアウトします。IMAP サーバーのログアウトができた場合は、コマンドは現在の<imap_ID>として0を返します。

注：接続を終了すると、現在のセッションは自動的に終了します。

<imap_ID>は**IMAP_Login**によって開始された接続のリファレンスとなる倍長整数です。

参照：IMAP_Login

メールのダウンロード

“MSG_”という接頭辞の付いたコマンドは、第3章で説明した**POP3_Download**コマンドまたは**IMAP_Download**を使用し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。

一度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_SetPrefs

MSG_SetPrefs (ラインフィード除去; 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 = ラインフィードを除去しない、 1 = ラインフィードを除去、 -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 (" " = 変更なし)
添付フォルダ	テキスト	→	添付フォルダのパス (" " = 変更なし)
戻り値	整数	←	エラーコード

説明

MSG_SetPrefs を使ってすべてのMSGコマンドの環境設定を行います。

<ラインフィード除去>は、ダウンロードしたメッセージでのラインフィード文字の処理方法を指定する整数値です。ほとんどのインターネットメッセージで、行の終わりを示すのにキャリッジリターンとラインフィード文字が併用されます。Macintoshのアプリケーションでは、行の終わりを示す文字としてキャリッジリターンのみを使用することが好まれます。このオプションの引数を使ってメッセージテキストからラインフィード文字を除去することができます。0の場合は、メッセージをそのままの状態に残します。1の場合は、メッセージからラインフィード文字を除去します。-1の場合は、環境設定を前回の設定のまま残します。省略時はデフォルト値として1が設定され、メッセージ内に見つかったラインフィードはすべて自動的に除去されます。

<メッセージフォルダ>は、取り出したメッセージをデフォルトで格納するフォルダのローカルパス名を示すテキスト値です。

互換性についての注意 (バージョン6.8.1) : **MSG_SetPrefs** コマンドを使用しない場合は、**POP3_SetPrefs** コマンドを以前に使用していれば、その引数<ラインフィード除去>および引数<メッセージフォルダ>が使用されます。**MSG_SetPrefs** を使用する場合は、**POP3_SetPrefs** の引数は無視されます。

<添付フォルダ>は、**MSG_Extract** コマンドが添付ファイルをメッセージの本文から切り離す場合に、添付フォルダを格納するフォルダのローカルパス名を示すテキスト値です。

互換性についての注意（バージョン6.8.1）：この引数はPOP3_SetPrefsとMSG_SetPrefsのどちらにもあるため、この2つのどちらかのコマンドを使って変更することができます。しかしながら、MSG_SetPrefsコマンドの使用を強くお勧めします。POP3_SetPrefsの引数は互換性の理由により使用されており、将来的に使用されなくなります。POP3_SetPrefsコマンドの引数<添付フォルダ>はオプションであるため、この引数を使用しないようにお勧めします。この推奨内容はPOP3_GetPrefsにも該当します。

参照：MSG_Extract、POP3_Download、POP3_SetPrefs、IMAP_Download、IMAP_SetPrefs

MSG_GetPrefs

MSG_GetPrefs (ラインフィード除去; 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 = キャリッジリターン/ ラインフィードを除去しない、 1 = キャリッジリターン/ ラインフィードを除去
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 (" " = 変更なし)
添付フォルダ	テキスト	←	添付フォルダのパス (" " = 変更なし)
戻り値	整数	←	エラーコード

説明

MSG_GetPrefs コマンドはMSG コマンドの現在の環境設定を返します。

環境設定は引数に渡された変数に返されます。

<ラインフィード除去>は、ラインフィード除去のユーザの現在の環境設定を返します。

<メッセージフォルダ>は、取り出したメッセージを格納するデフォルトのフォルダのローカルパス名を返すテキスト変数です。

<添付フォルダ>は、展開した添付ファイルを格納するデフォルトの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：MSG_SetPrefs、POP3_SetPrefs

MSG_FindHeader

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
戻り値	整数	← エラーコード

説明

POP3_Download コマンドまたは **IMAP_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** コマンドまたは **MSG_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 (互換性についての注意を参照)。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。

互換性についての注意 (バージョン 6.8.1) : **MSG_SetPrefs** コマンドが使用されない場合、**POP3_SetPrefs msgFolder** パラメータは使用されます。**MSG_SetPrefs** が使用される場合、**POP3_SetPrefs msgFolder** パラメータは無視されます。

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、拡張文字を含む場合がありますが、**POP3_Charset** コマンドまたは **MSG_Charset** コマンドを使って処理を自動化することができます。

互換性についての注意 (バージョン 6.8.1) : **MSG_Charset** コマンドを使用しない場合は **POP3_Charset** の引数<本文文字セット>が使用され、**MSG_Charset** を使用する場合は **POP3_Charset** の引数<本文文字セット>は無視されます。

参照

IMAP_Download、**MSG_Charset**、**MSG_SetPrefs**、**POP3_Charset**、**POP3_Download**、**POP3_SetPrefs**

MSG_MessageSize

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)
戻り値	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** コマンドまたは**MSG_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（互換性についての注意を参照）。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せん。

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。

この2つの引数は、**POP3_SetPrefs** コマンドまたは**MSG_SetPrefs** コマンドで設定された引数<ラインフィード除去>を考慮します。

互換性についての注意（バージョン6.8.1）： **MSG_SetPrefs** コマンドを使用しない場合は、**POP3_SetPrefs** コマンドを以前に使用していれば、その引数<メッセージフォルダ>および引数<ラインフィード除去>が使用されます。**MSG_SetPrefs** を使用する場合は、**POP3_SetPrefs** の引数<メッセージフォルダ>および引数<ラインフィード除去>は無視されます。

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

IMAP_Download、MSG_SetPrefs、POP3_Download、POP3_SetPrefs

MSG_GetHeaders

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)
戻り値	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** コマンドまたは **MSG_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 (互換性についての注意を参照)。フォルダが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。この引数は、**POP3_SetPrefs** コマンドまたは **MSG_SetPrefs** コマンドで設定された引数<ラインフィード除去>を考慮します。

互換性についての注意 (バージョン 6.8.1) : **MSG_SetPrefs** コマンドを使用しない場合は **POP3_SetPrefs** コマンドの引数<メッセージフォルダ>が使用され、**MSG_SetPrefs** を使用する場合は **POP3_SetPrefs** の引数<メッセージフォルダ>は無視されます。

参照

MSG_MessageSize、**MSG_SetPrefs**、**POP3_SetPrefs**

MSG_GetBody

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)
戻り値	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** コマンドまたは**MSG_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 (互換性についての注意を参照)。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>は、拡張文字を含む場合がありますが、**POP3_Charset** コマンドまたは**MSG_Charset** コマンドを使って処理を自動化することができます (互換性についての注意を参照)。この引数は、**POP3_SetPrefs** コマンドまたは**MSG_SetPrefs** コマンドで設定された引数<ラインフィード除去>を考慮します (互換性についての注意を参照)。

互換性についての注意（バージョン6.8.1）：

- MSG_SetPrefs コマンドを使用しない場合は、POP3_SetPrefs コマンドの引数<メッセージフォルダ>および引数<ラインフィード除去>が使用されます。MSG_SetPrefs を使用する場合は、POP3_SetPrefs の引数<メッセージフォルダ>および引数<ラインフィード除去>は無視されます。
- MSG_Charset コマンドを使用しない場合は POP3_Charset の引数<本文文字セット>が使用され、MSG_Charset を使用する場合は POP3_Charset の引数<本文文字セット>は無視されます。

参照

MSG_Charset、MSG_SetPrefs、POP3_Charset、POP3_SetPrefs

MSG_GetMessage

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)
戻り値	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** コマンドまたは **MSG_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 (互換性についての注意を参照)。フォルダが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。

互換性についての注意 (バージョン 6.8.1) : **MSG_SetPrefs** コマンドを使用しない場合は **POP3_SetPrefs** コマンドの引数<メッセージフォルダ>が使用され、**MSG_SetPrefs** を使用する場合は **POP3_SetPrefs** の引数<メッセージフォルダ>は無視されます。

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

MSG_SetPrefs、**POP3_SetPrefs**

MSG_HasAttach

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
戻り値	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** コマンドまたは **MSG_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 (互換性についての注意を参照)。フォルダが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。

互換性についての注意 (バージョン 6.8.1) : **MSG_SetPrefs** コマンドを使用しない場合は **POP3_SetPrefs** コマンドの引数<メッセージフォルダ>が使用され、**MSG_SetPrefs** を使用する場合は **POP3_SetPrefs** の引数<メッセージフォルダ>は無視されます。

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

MSG_SetPrefs、**POP3_SetPrefs**

MSG_Extract

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
戻り値	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** コマンドまたは **MSG_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 (互換性についての注意を参照)。フォルダが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** コマンドまたは **MSG_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 (互換性についての注意を参照)。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。

互換性についての注意 (バージョン 6.8.1) : **MSG_SetPrefs** コマンドを使用しない場合は **POP3_SetPrefs** コマンドの引数<メッセージフォルダ>および引数<添付パス>が使用され、**MSG_SetPrefs** を使用する場合は **POP3_SetPrefs** の引数<メッセージフォルダ>および引数<添付パス>は無視されます。

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

MSG_SetPrefs、POP3_SetPrefs

MSG_Delete

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
戻り値	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引数<フォルダ>は考慮されます。

- 引数<フォルダ>が0の場合は、ファイルは **POP3_SetPrefs** コマンドまたは **MSG_SetPrefs** コマンドで指定されたメッセージのフォルダの中に入っています。
- 引数<フォルダ>が1の場合は、ファイルは **POP3_SetPrefs** コマンドまたは **MSG_SetPrefs** コマンドで指定された添付ファイルのフォルダの中に入っています。

どちらの場合も、(メッセージまたは添付)フォルダが **POP3_SetPrefs** コマンドまたは **MSG_SetPrefs** コマンドで指定されなかった場合は、データベース (4D シングルユーザの場合) または 4D Client フォルダ (4D Server の場合) 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。

互換性についての注意 (バージョン 6.8.1) : **MSG_SetPrefs** コマンドを使用しない場合は **POP3_SetPrefs** コマンドの引数<メッセージフォルダ>および引数<添付フォルダ>が使用され、**MSG_SetPrefs** を使用する場合は **POP3_SetPrefs** の引数は無視されます。

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

MSG_SetPrefs、POP3_SetPrefs

MSG_Charset

MSG_Charset (デコードヘッダ; 本文文字コードセット) → 整数

引数	タイプ		説明
デコードヘッダ	整数	→	-1 = 現在の設定を使用、0 = 処理しない、1 = ISO-8859-1 または ISO-2022-JP の場合は MacOS の文字コードセットを使って変換、拡張文字をデコードする
本文文字コードセット	整数	→	-1 = 現在の設定を使用、0 = 処理しない、1 = ISO-8859-1 または ISO-2022-JP の場合は MacOS の文字コードセットを使って変換
戻り値	整数	←	エラーコード

MSG_Charset コマンドを使って、MSG コマンドを使った処理中に拡張文字を含むメッセージを自動的にサポートすることができます。このコマンドを呼出さない場合、または 0 に設定されている場合は、4D Internet Commands バージョン 6.8.1 以降も 6.5.X と同様に動作します。

MSG_Charset を使って、まず拡張文字ヘッダのデコードについて設定し、次にメッセージ本文およびヘッダの文字コードセット変換について決定することができます。

このコマンドは“件名”や電子メールアドレス等のメッセージヘッダに含まれる拡張文字をサポートするのに特に有効です（例えば、“=?ISO-8859-1?Q?Test=E9?= <test@n.net>”のようなアドレスをデコードする際）。

引数<ヘッダデコード>は、**MSG_FindHeader** を実行する際のヘッダのデコードおよび変換の処理について指定します。指定しない場合はデフォルト値として 0 が設定されます。

■ -1：現在の設定を使用

■ 0：処理しない

■ 1：必要に応じてヘッダをデコードする：デコードを行い、指定した文字コードセットが ISO-8859-1 または ISO-2022-JP である場合は、ヘッダはそれぞれ MacOS の ASCII コードまたはシフト-JIS を使って変換されます。

引数<本文文字コードセット>は、**MSG_GetBody** コマンドを実行する際のメッセージ本文の文字コードセット変換について指定します。指定しない場合はデフォルト値として 0 が設定されます。

- -1：現在の設定を使用
- 0：処理しない
- 1：“Body-Content-Type”の文字コードセットをISO-8859-1またはISO-2022-JPに設定すると、メッセージ本文はそれぞれMacOSのASCIIコードまたはシフト-JISを使って変換されます。

▼ 例

(1) 4D Internet Commandsバージョン6.5.xを使用する場合

```

$Err:= MSG_FindHeader ($msgfile;"From";$from)
$from:= ISO to Mac ($from)
$Err:= MSG_FindHeader ($msgfile;"To";$to)
$to:= ISO to Mac ($to)
$Err:= MSG_FindHeader ($msgfile;"Cc";$cc)
$cc:= ISO to Mac ($cc)
$Err:= MSG_FindHeader ($msgfile;"Subject";$subject)
$subject:= ISO to Mac ($subject)
$Err:= MSG_MessageSize ($msgfile;$HdrSize;$BdySize;$msgSize)
$Err:= MSG_GetBody ($msgfile;0;$BdySize;$BodyContent)
$BodyContent:= ISO to Mac ($BodyContent)

```

(2) 4D Internet Commandsバージョン6.8.1またはそれ以降を使用する場合

```

$Err:= MSG_Charset (1;1)
$Err:= MSG_FindHeader ($msgfile;"From";$from)
$Err:= MSG_FindHeader ($msgfile;"To";$to)
$Err:= MSG_FindHeader ($msgfile;"Cc";$cc)
$Err:= MSG_FindHeader ($msgfile;"Subject";$subject)
$Err:= MSG_MessageSize ($msgfile;$HdrSize;$BdySize;$msgSize)
$Err:= MSG_GetBody ($msgfile;0;$BdySize;$BodyContent)

```

参照

SMTP_Charset、POP3_Charset

ファイル転送

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：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サーバソフトウェアは接続しているクライアントに対し、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

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
戻り値	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呼び出している最中に表示されます。**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、あるいは非表示にできるパラメータ（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、サーモメータがサインポール（理髪店の前で回っているポール）のように表示され、ファイルサイズが“unknown（不明）”と表示されます。

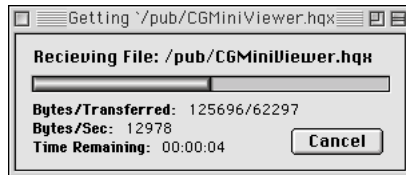
引数<左>は、サーモメータの進捗ウインドウの左側の座標です。<左>が-1の場合、ウインドウは画面上の水平に中央配置されます。

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。



例題

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

Case of

```
\(FTP_Login ("ftp.4dus.com";"anonymous";vEmailID;vFTP_ID;
vFTP_Msg)#0)
\ (FTP_Send (vFTP_ID;"HD:Document:Report(7 X 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

FTP_Login (ホスト名;ユーザ名;パスワード;ftp_ID{;ウエルカムテキスト})→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
ウエルカムテキスト	テキスト	←	FTP ウエルカムテキスト
戻り値	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サーバで接続を開設し、設定済みの<ユーザ名>と<パスワード>を使用し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<ウエルカムテキスト>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エルカムメッセージを持っています。<ウエルカムテキスト>を指定する場合は、<ウエルカムテキスト>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。

例題

```
$OK:=False
```

Case of

```
\(FTP_Login ("ftp.4dus.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

FTP_GetDirList

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
修正日	日付配列	← 修正日リスト
戻り値	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>の中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（ファイルまたはフォルダ上のエイリアス）
5	FIFO特殊ファイル
6	AF_UNIXアドレスファミリーソケット

注：シンボリックリンク（値=4）の場合は、FTPサーバは特定のパス名（エイリアス名+シンボル+ソースファイルまたはフォルダのパス名）を返します。ソースファイルまたはフォルダを呼び出すのにこのパス名を使用しようとすると、エラーが返されます。ソースファイルまたはフォルダのパス名は、記号文字のすぐ後のFTP_GetDirListによって返される文字列から抜き出す必要があります。それ以外の場合には、ファイルまたはフォルダが見つからないためFTP_GetFileInfoのようなコマンドは-10085エラーを返しません。

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、FTP_VerifyID

FTP_ChangeDir

FTP_ChangeDir (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
戻り値	整数	←	エラーコード

説明

FTP_ChangeDir コマンドは、カレントディレクトリを引数<ディレクトリ>に渡されたパスに変更します。

FTP_GetDirList と **FTP_GetFileInfo** コマンドも、カレントディレクトリを変更します。しかし、**FTP_ChangeDir** コマンドの実行はこれらのコマンドよりも速く、引数も少なくてすみます。

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、存在する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。ディレクトリが存在しないか、ユーザが動作を行うためのアクセス権を持っていない場合にはエラーが返されます。この場合カレントディレクトリは変更されません。

例題

この記述は、カレントディレクトリをFTP rootに設定します。

```
$err=FTP_ChangeDir (ftp_ID;"/")
```

参照

FTP__GetDirList、FTP_GetFileInfo、FTP_PrintDir

FTP_PrintDir

FTP_PrintDir (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
戻り値	整数	←	エラーコード

説明

FTP_PrintDir コマンドは、引数<ディレクトリ>にカレントディレクトリを返します。

FTP_GetDirList コマンドも、カレントディレクトリを返します。しかし、**FTP_PrintDir** コマンドの実行はより速く、引数も少なくてすみます。

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、カレントディレクトリを返します。

例題

この例では\$Cwdにカレントディレクトリが返されます。

```
$err:=FTP_PrintDir (ftp_ID;$Cwd)
```

参照

FTP_GetDirList

FTP_GetFileInfo

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
戻り値	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

FTP_VerifyID

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
戻り値	整数	←	エラーコード

説明

FTPサーバは、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サーバと対話する各コマンドは、この非稼動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セッションの非稼動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

FTP_MakeDir

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
戻り値	整数	←	エラーコード

説明

FTP_MakeDir コマンドは、アクセス可能なくディレクトリ>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。

注：**FTP_ChangeDir** コマンドを使用してカレントディレクトリを変更することができます。また、**FTP_PrintDir** コマンドを使用して、いつでもカレントディレクトリを調べることができます。

参照

FTP_GetDirList、FTP_RemoveDir

FTP_RemoveDir

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
戻り値	整数	←	エラーコード

説明

FTP_RemoveDir コマンドは、アクセス可能なく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、FTP_MakeDir

FTP_Rename

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
戻り値	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。

参照

なし

FTP_Delete

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サーバ上にあるファイルのパス名
戻り値	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

FTP_MacBinary

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)
戻り値	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データやバイナリ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ため、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 サーバが MacBinary プロトコルをサポートしているわけではないため、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。

例題

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

FTP_Send

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
戻り値	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。ローカルのドキュメント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>は一旦FTPサーバで受信されると、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。

例題

例題1：

```

$OK:=False
Case of
  \(\FTP_Login ("ftp.4dus.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、FTP_Receive

FTP_Append

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
戻り値	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

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、 “L8” = 論理8ビット
戻り値	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** コマンドを使用してセットされます。

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

FTP_SetType

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
戻り値	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ドキュメント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

FTP_System

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
戻り値	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サーバに関する情報を含んでいます。

参照

なし

FTP_Receive

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
戻り値	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用して、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。

例題

```
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
End if
```

参照

FTP_MacBinary、IT_Decode

FTP_Logout

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
戻り値	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。

例題

```
If (FTP_Login ("ftp.4dus.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

FTP_SetPassive

FTP_SetPassive (ftpID;パッシブモード)→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
パッシブモード	整数	→	0=Activeモード、 1=Passiveモード（デフォルト）
戻り値	整数	←	エラーコード

説明

FTP_SetPassive コマンドは、**FTP_GetDirList**、**FTP_Send**、**FTP_Append**、または**FTP_Receive** コマンドが実行されている間の、FTPサーバとFTPクライアント間のデータストリーム交換モードを設定します。データストリーム交換モードの設定は**FTP_SetPassive** コマンドが実行された後に、これらのコマンドに対して適用されます。

FTPサーバとFTPクライアント間の交換は2つのストリーム、コントロールストリーム（デフォルトポート21）とデータストリーム（デフォルトポート20）にもとづいて行われます。通常FTPサーバはデータ接続を開いて管理するために、「Active」として定義されます。

過去の経緯から、4D Internet Commandsのデフォルトデータストリーム交換モードは、FTPサーバが「Passive」モードで動作していることを前提としています。つまりデータストリームを交換する前ごとに、FTPコマンド"PASV"が送信されます。

しかしFTPサーバによっては「Passive」モードがサポートされない場合や、ファイアウォールがそれを許可しない場合があります。このような場合、「Active」モードをクライアントのデータストリーム交換モードとして設定する必要があります。

注：FTPによる交換に「Active」モードと「Passive」モードどちらが使用されているか、ネットワーク管理者に尋ね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<パッシブモード>に渡される値は、データストリーム交換モードを指定します：

- ・ 値0はFTPサーバが「Active」モードで動作していると指定します。
- ・ 値1はFTPサーバが「Passive」モードで動作していると指定します。

参照

FTP_GetPassive

FTP_GetPassive

FTP_GetPassive (ftp_ID;パッシブモード)→ 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
パッシブモード	整数	←	カレントのデータストリーム交換モード 0=Activeモード、1=Passiveモード
戻り値	整数	←	エラーコード

説明

FTP_GetPassive コマンドはカレントのデータストリーム交換モードを返します。FTP交換モードについての詳細は、**FTP_SetPassive** コマンドの説明を参照してください。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<パッシブモード>は、カレントのデータストリーム交換モードを返します：

- ・ 0が返された場合、FTPサーバは現在「Active」モードで動作するよう設定されています。
- ・ 1が返された場合、FTPサーバは現在「Passive」モードで動作するよう設定されています（デフォルト）。

参照

FTP_SetPassive

ローレベルルーチン

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番目の方法は、**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用して任意の接続をオープンします。**TCP_Open** コマンドでは、安全な接続を行うことができる SSL (セキュア・ソケット・レイヤー) プロトコルを使用することができます。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用し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** コマンドを使用して、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用して、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 RFC の中で参照できます。

TCP_Open

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
セッション設定	整数	→	TCPセッションの設定 0 = 同期 (省略時のデフォルト) 1 = 非同期 2 = SSL使用、同期 3 = SSL使用、非同期
戻り値	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。

<セッション設定>は、TCPセッションの設定を選択することができる整数のオプション引数です。この設定はセッション中に呼び出される各TCPコマンドに適用される点に注意してください。指定しない場合はデフォルト値として0 (同期、非SSL) が設定されます。

SSL (セキュア・ソケット・レイヤー) は安全なTCP通信を行うことができるプロトコルです (詳細およびインストールの必要条件については、4th Dimensionのリファレンスを参照してください)。

非同期 / 同期

非同期モードは、接続プロセスが終了するのを待たずに（リモートホストによる接続が確立されるのを待たずに）直ちに4Dカーネルに制御を返します。非同期モードは、すべてのCPUタイムを4D Internet Commandsに占有させたくない場合に有効です。

同期モードは、接続プロセスが終了した場合にのみ（問題の有無に関わらず）4Dカーネル（他の4Dプロセス）に制御を返します。

- 0 = 同期モード（デフォルトモード、4D Internet Commands の以前のバージョンと同様の動作をします）
- 1 = 非同期モード
- 2 = SSL使用、同期：このTCPセッションのリファレンス（tcp_ID）を使用するすべてのTCPコマンドは同期モードで動作し、SSLプロトコルを使用します。
- 3 = SSL使用、非同期：このTCPセッションのリファレンス（tcp_ID）を使用するすべてのTCPコマンドは非同期モードで動作し、SSLプロトコルを使用します。

注：SSL接続を開始することができない場合（4D Extensions フォルダにSLIライブラリが見つからない場合）に2または3を渡すと、10089エラーが返されることがあります。

例

HTTPを使ってウェブサイト连接到りたい場合は、SLIが正しくインストールされていること、およびポート番号443を使って接続を開始することを確認してください。

```
$vError:=TCP_Open (hostName; 443; tcp_ID;2)
```

参照

IT_TimeOut

TCP_Listen

TCP_Listen (リモートホスト;ローカルポート;リモートポート;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
		←	IPアドレスは空の文字列を含む変数
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
リモートポート	整数	→	接続待機中のポート番号、0=任意
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
戻り値	整数	←	エラーコード

説明

TCP_Listen コマンドは、引数<リモートホスト>上のマシンの引数<リモートポート>で参照されるポートから生成される接続を待機します。このコマンドは、接続が開始されるか、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プロセスに制御を戻しません。これは接続が開始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、このコマンドは他のプロセスの実行を妨げないようにタイムスライスを分け与えます。

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、接続してくるの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。

この引数に空の文字列を渡すと、このコマンドはあらゆるマシンから入ってくる接続を受け入れます。

この引数に空の文字列を含む変数を渡すと、接続されたマシンのIPアドレスを返します。

注：Windowsでは、引数<リモートホスト>へ渡す変数に予めIPアドレスを代入しておくと、“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”という-10049エラーが発生します。結果として、IPアドレスにフィルタをかける必要がある場合は、空の文字列を含む変数を使用する方が有用です。

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<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せん。

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。任意の接続が一度も設定されないと、4th Dimension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を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。

例題

```
C_LONGINT(vTCPID)
C_INTEGER(vStatus)
$err:=TCP_Listen ("";0;49152;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、TCP_State

TCP_Send

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
戻り値	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用して設定されます。

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、TCP_Open

TCP_Receive

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
戻り値	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用して設定されません。

引数<テキスト>は、受け取ったテキストです。TCP パケット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。

例題

```

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

TCP_SendBLOB

TCP_SendBLOB (tcp_ID;送信 BLOB)→整数

引数	タイプ		説明
tcp_ID	整数	→	オープンしたTCPセッションのリファレンス
送信 BLOB	BLOB	→	送信する BLOB
戻り値	整数	←	エラーコード

説明

TCP_SendBLOB コマンドは、引数<tcp_ID>に示されたTCPセッションにデータを送信します。このコマンドは**TCP_Send**と同じ動作を行いますが、テキストではなくBLOBデータを送信するところが違います。これにより、テキストの制限32kを避けることができます。このコマンドを使用して、バイナリオブジェクトを送信することができます。

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用して設定されます。

送信BLOBは、<tcp_ID>によって参照される、TCPセッションに送信されるBLOBです。

プラットフォーム独立性に関する注意：MacintoshとPCプラットフォーム間でBLOBを交換する場合、必要に応じてバイトスワップの処理を行うのは開発者の仕事です。

例題

この例ではTCPセッションにBLOBを送信します。

```
C_BLOB ($Blob_Send)
C_TEXT (v_Txt_Send)
TEXT TO BLOB (v_Txt_Send;$Blob_Send;Text without length;)
$err:=TCP_SendBLOB (v_tcp_ID;$Blob_Send)
```

参照

TCP_Listen、TCP_Open、TCP_Send

TCP_ReceiveBLOB

TCP_ReceiveBLOB (tcp_ID;受信 BLOB)→整数

引数	タイプ		説明
tcp_ID	整数	→	オープンしたTCPセッションのリファレンス
受信 BLOB	BLOB	←	データとして受け取る BLOB
戻り値	整数	←	エラーコード

説明

確立されたTCPセッションへの倍長整数型参照を渡すことで、**TCP_ReceiveBLOB** コマンドは引数<受信 BLOB>にデータパケットを受け取ります。

このコマンドは、**TCP_Receive** コマンドと同じ動作を行います。テキストではなく BLOB にデータを受け取る場所が違います。これによりテキストの制限 32k を避けることができ、バイナリオブジェクトを受け取ることができるようになります。

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用して設定されます。

引数<受信 BLOB>はデータを受け取る BLOB です。TCP パケットを使用してデータを受信する場合、一回の **TCP_ReceiveBLOB** のコールですべてのデータを受け取ることを期待することはできません。**TCP_ReceiveBLOB** は通常、接続状況を確認する、あるいは既知の値をスキャンする Repeat...Until ループの中でコールされます。

例題

この例は、**TCP_ReceiveBLOB** を使用したメソッドの典型的な構造を例示します。

```

C_BLOB ($Blob_Received; $Blob_All)
C_LONGINT ($srcpos; $dstpos)
Repeat
    $Err:=TCP_ReceoveBLOB ($TCP_ID;$Blob_Received)
    $Err:=TCP_State ($TCP_ID;$State)
    $srcpos:=0
    $dstpos:=BLOB size($Blob_All)
    `受信した BLOB をつなぐ
    COPY BLOB ($Blob_Received;$Blob_All;$srcpos;$dstpos;
                BLOB size ($Blob_Received))
Until (($State=0) | ($Err#0))

```

参照

TCP_Listen、TCP_Open、TCP_Receive

TCP_State

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
戻り値	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用して設定されません。

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。

例題

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サーバは自動的に接続をクローズするので、一旦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を続けます。

```
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、TCP_Open

TCP_Close

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
戻り値	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用して設定されます。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、TCP_Open

特殊なインターネットコマンド

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

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
戻り値	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、“**4dus.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。

```
$error:=NET_Finger ("www.4dus.com";"johnt";$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に質問します。例えば、次のコードはユーザ “**johnt@4dus.com**” のリモート**Finger** クエリを実行するためにドメイン名 “**4dus.com**” によって識別されるマシンに質問します。

```
$error:=NET_Finger ("www.4dus.com";"johnt@4dus.com";$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。

参照

なし

NET_Ping

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	→	Ping 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
戻り値	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。Ping 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が。一般的に Ping されたマシンは、自分のユーザに対して Ping を受けたことを通知しません。Ping (ICMP エコー) に対し、返事をして良いと判断します。

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 Ping 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、Ping 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、その保護機能を使用して、マシンが Ping されることを防ぎます)。

引数<ホスト名>は、Ping するホスト名または IP アドレスです。

引数<テキスト>は、Ping 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、Ping 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 Ping 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、Ping 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、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。

注：Windows 95/98 および Millennium では、引数<タイムアウト>は考慮されません。

参照

IT_SetTimeOut

NET_Time

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
戻り値	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：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。このタイムサーバがユーザの場所と地理的に同じ場合であっても、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。

例題

次の例は、“apple.com”のタイムサーバ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 Dimentionの**Time String**関数を使用して変換されます。

```
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 Dimentionの
    `時刻に変換する
End if
```

参照

Time string (4th Dimention コマンド)

NET_NameToAddr

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
戻り値	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>の値は有効な使用方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_AddrToName

NET_AddrToName

NET_AddrToName (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
戻り値	整数	←	エラーコード

説明

NET_AddrToName 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

NET_Resolve

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
戻り値	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。

例題

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。
```

参照

なし

ユーティリティコマンド

この章のコマンドは、4D Internet Commandsの他の章で役に立つ様々なユーティリティを提供します。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読したり、ファイルのエンコードとデコードの手助けを行い、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

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。
戻り値	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つの関数として機能し、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。

ダイヤルアップ方式を使用しない場合は、このコマンドは4th Dimension データベースの On Startup データベース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。このコマンドを使用するとダイヤルアップ接続の開始および初期化が行われるため、PPP や SLIP 等の TCP で利用されるダイヤルアップ方式のユーザは、TCP 接続が必要となるまではこのコマンドの使用を避けたほうがよいでしょう (**IT_PPPStatus** コマンドを参照)。

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

参照

IT_PPPConnect

IT_Platform

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。			
戻り値	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。

例題

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

参照

なし

IT_Version

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。
戻り値	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。

例題

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)
```

参照

なし

IT_TCPversion

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、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
戻り値	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、現在Open Transportのみがサポートされます。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 (旧バージョンのスタック、互換性についての注意を参照)
2	Open Transport
3	WinSock

互換性についての注意：MacTCPは現在ではサポートされません（バージョン6.8より）。したがって、コマンドが1を返すことはありません。

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。

参照

なし

IT_MacTCPVer

互換性についての注意：IT_MacTCPVer コマンドは、Open Transport や WinSock のバージョン情報をも調べることができるより強力な IT_TCPVersion コマンドの登場により廃止されました。

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
戻り値	整数	←	エラーコード

説明

4D Internet Commands バージョン 6.8 より、MacTCP はサポートされません。したがって、使用するプラットフォームや OS に関わらず、引数<バージョンコード>は体系的に 0 を返します。

参照

IT_TCPVersion

IT_MyTCPAddr

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
戻り値	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。

注：Windows 95では、引数<サブネット>は、引数<IPアドレス>で指定されたアドレスのタイプのデフォルトのサブネットマスクを返します。

参照

なし

IT_SetTimeOut

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
戻り値	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

IMAP_Login

IMAP_VerifyID

IMAP_Capability

IMAP_ListMBs

IMAP_SubscribeMB

IMAP_GetMBStatus

IMAP_SetCurrentMB

IMAP_Delete

IMAP_Reset

IMAP_MsgInfo
IMAP_MsgLstInfo
IMAP_GetMessage
IMAP_MsgLst
IMAP_SetFlags
IMAP_GetFlags
IMAP_Search
IMAP_MsgFetch
IMAP_Download
IMAP_CopyToMB
IMAP_CreateMB
IMAP_RenameMB
IMAP_DeleteMB
NET_Finger
NET_Ping
NET_Time

注：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まいります。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

IT_GetTimeOut

IT_GetTimeOut

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
戻り値	整数	←	エラーコード

説明

IT_GetTimeOut コマンドは、前ページの **IT_SetTimeOut** コマンドの説明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

IT_ErrorText

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
戻り値	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 の中で戻り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。

例題

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

```

参照

付録A 「プログラミングのヒント」

IT_Encode

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
戻り値	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>を使用して、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用し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せん。
- (ファイル名を指定せずに) エンコードされたファイルが保存されるフォルダを規定する完全なローカルパスフォルダを指定することができます。この場合のファイル名はエンコーディングモードを定義する接尾辞が付いた元のファイル名です。
- 空の文字列 (この場合は、IT_Encode コマンド) を渡すと、そのドキュメントに独自の名前が付けられ、最初の引数で指定されたファイルと同じフォルダに配置されます。

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）を起こす可能性があるため、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利用するようにしてください。

注：MacOS では、接尾辞が付いた元のファイル名の長さが31バイト以上の場合、-37エラーが返されます。接尾辞を付加した元のファイル名の長さが31バイト以上にならないよう注意してください。

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を使用し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

IT_Decode

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
戻り値	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>を使用して、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用し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。
- 元のファイル名を使ってデコードされたファイルを受け取るフォルダを示す完全なローカルパスフォルダを指定することができます。
- 空の文字列（この場合は、**IT_Decode** コマンド）を渡すと、そのドキュメントに独自の名前が付けられ、最初の引数で指定されたファイルと同じフォルダに配置されます。

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。

注：MacOS では、接尾辞が付い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名がMacintosh ファイルの31バイト制限を越えないように元のファイル名を切り取ります。

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を使用してデコードすると、“%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

FTP_MacBinary、FTP_Receive、IT_Encode

IT_GetProxy

IT_GetProxy (プロトコル;種類;ホスト;ポート;ユーザID)→整数

引数	タイプ		説明
プロトコル	整数	→	1 = FTP、2 = SMTP、3 = POP3、4 = IMAP
種類	整数	←	0 = なし、1 = SOCKS
ホスト	文字列	←	SOCKS プロキシホストのホスト名または IP アドレス
ポート	整数	←	接続に使用するプロキシのポート
ユーザID	テキスト	←	SOCKS のユーザID
戻り値	整数	←	エラーコード

説明

IT_GetProxy コマンドは、引数に与えられたプロトコルについて、4D Internet Commands によってルーティングに使用される設定の現在の値を返します。**IT_SetProxy** コマンドによって事前に設定が変更されていないければ、戻り値はデフォルトの値となります。引数についての詳細は**IT_SetProxy**を参照してください。

<プロトコル>は整数値で、設定を調べるプロトコルを指定します。1はFTPプロトコルを示します。2はSMTPプロトコルを示します。3はPOP3プロトコルを示します。4はIMAPプロトコルを示します。

<種類>はSOCKSプロキシホストが使用されているかいないかを返します。1の場合、指定されたプロトコルのすべてのリクエストが、特定のSOCKSホストを使用します。0の場合、指定されたプロトコルのリクエストはSOCKSホストを使用しません。

<ホスト名>は現在使用されているSOCKSプロキシホストのホスト名またはIPアドレスを返します。

<ポート>は指定されたプロトコルがSOCKSプロキシホストとの通信に使用するポート番号を返します。

<ユーザID>はユーザIDを返します。

参照

IT_SetProxy

IT_SetProxy

IT_SetProxy (プロトコル;種類;ホスト;ポート;ユーザID)→整数

引数	タイプ		説明
プロトコル	整数	→	1 = FTP、2 = SMTP、3 = POP3、4 = IMAP
種類	整数	→	0 = なし、1 = SOCKS
ホスト	文字列	→	SOCKS プロキシホストのホスト名または IP アドレス
ポート	整数	→	接続に使用するプロキシのポート
ユーザID	テキスト	→	SOCKS のユーザID
戻り値	整数	←	エラーコード

説明

IT_SetProxy コマンドは、指定されたプロトコルの接続の確立やリクエストの送信に際し、SOCKS ホスト (SOCKS プロキシ) の使用を可能にします。イントラネットに接続するだけならおそらく SOCKS プロキシを使う必要はないでしょう。しかし、すべては会社のファイアウォールの設定に依存します。**IT_SetProxy** による設定はインタープロセスの範囲を持っていて、すべての 4D プロセスで、指定されたプロトコルの通信にその設定が使用されます。

注：SOCKS は、会社内ネットワークのユーザからのリクエストをインターネットに送信できるようにするために使用される、プロキシサーバのプロトコルです。例えば、ワークステーションがファイアウォールの内側にあり、そこからインターネット上にある情報を得たい場合、リクエストは SOCKS ホストによって受信され、ファイアウォールを通り、そして情報がクライアントアプリケーションに返されます。

<プロトコル>は整数値で、SOCKS プロキシホストの設定を行うプロトコルを指定します。1はFTPプロトコルを示します。2はSMTPプロトコルを示します。3はPOP3プロトコルを示します。4はIMAPプロトコルを示します。

<種類>は指定されたプロトコルのルーティングに際し、SOCKS プロキシホストを使用するかしないかを指定します。1の場合、指定されたプロトコルのすべてのリクエストが、特定のSOCKSホストを使用します。0の場合、指定されたプロトコルのリクエストはSOCKSホストを使用しません。

<ホスト名>はSOCKS プロキシホストのホスト名またはIPアドレスを指定します。

<ポート>は整数で、指定されたプロトコルがSOCKS プロキシホストとの通信に使用するポート番号を指定します。

<ユーザID>はユーザを識別するテキストです。ユーザIDはネットワーク管理者から割り当てられます。<ユーザID>は空のテキスト（""）を指定することができます。

例題

以下のメソッドを使用して、すべてのFTP接続は指定されたSOCKSプロキシホストを経由します。

```
$err=IT_SetProxy (1;1;$proxyAdd;$proxyPort;"") `FTP SOCKS プロキシ  
$err=FTP_Login ("ftp.4dus.com";"anonymous";"dbody@aol.com";$ftpID)  
$err=FTP_GetFileInfo ($ftpID;$vpath;$vsize;$vmodDate)  
$err=FTP_Recieve ($ftpID;$vpath;"";0)  
$err=FTP_Logout ($ftpID)
```

注：メソッドを簡潔にするために、この例ではエラーチェックを行なっていません。

以下のステートメントはFTPの接続でSOCKSプロキシホストの使用を停止します。

```
$err=IT_SetProxy (1;0;$proxyAdd;$proxyPort;"")
```

参照

IT_GetProxy

IT_GetPort

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、4 = IMAP
ポート	整数	←	ポート番号
戻り値	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

付録 B 「TCP ポート番号」、IT_SetPort

IT_SetPort

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、4 = IMAP
ポート	整数	→	ポート番号
戻り値	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

付録B 「TCP ポート番号」、IT_GetPort

IT_PPPConnect

IT_PPPConnect (PPP プロファイル) → 整数

引数	タイプ		説明
PPP プロファイル	文字列	→	ダイヤルアップ名 = Mac OS では空の文字列、Windows では接続先を指定
戻り値	整数	←	エラーコード

IT_PPPConnect は、MacOS では現在のダイヤルアップ接続、Windows では指定したダイヤルアップ接続（引数 < PPP プロファイル >）を開始します。このコマンドは関数として機能し、接続を開始することができない場合は整数値のエラーを返します。

オンラインで機能する一連のインターネットコマンドを実行する必要があるたびにこのコマンドを実行する必要があります。作業完了時には、必ず **IT_PPPDisconnect** を実行して現在の接続を終了する必要があります。

PPP（ポイント・ツー・ポイント・プロトコル）はシリアルインタフェース（一般的には電話回線によってサーバに接続されたパソコン）を使用して2台のコンピュータ間の通信を行うためのプロトコルです。例えば、インターネットサーバ・プロバイダーはユーザにPPP接続を供給し、それによりプロバイダーのサーバはユーザの要求に応えることやその要求をインターネットへ渡すこと、また要求されたインターネットの応答をユーザに転送して戻すことができます。本質的には、プロバイダーはユーザコンピュータのTCP/IPパケットをパッケージ化し、それが実際にインターネット上に届けられるサーバへ転送します。

PPPは、同期・非同期の通信をどちらも処理することが可能なため、通常は以前のデファクトスタンダード（事実上の標準）シリアルライン・インターネットプロトコル（SLIP）よりも好まれます。PPPでは他のユーザと回線を共有することができ、エラー検出機能もありますが、このような機能はSLIPにはありません。選択が可能な場合はPPPが好まれます。

参照：IT_PPPDisconnect、IT_PPPStatus、IT_MacTCPInit

IT_PPPODisconnect

IT_PPPODisconnect {(PPP プロファイル)} → 整数

引数	タイプ		説明
PPP プロファイル	文字列	→	ダイヤルアップ名 = Mac OS では空の文字列、Windows ではオプションで接続先を指定
戻り値	整数	←	エラーコード

説明

IT_PPPODisconnect は、**IT_PPPOConnect** を使って開始した現在のダイヤルアップ接続を終了します。

< PPP プロファイル > は終了するダイヤルアップ接続を指定するテキスト値です。Windows では、複数の PPP 接続が同時に開始されている場合にこの引数は有効です。この引数を使用することにより、ユーザのネットワーク設定に関わらず動作性を高めることができます。

■ Windows の場合

- 1つの接続のみが開始されており、< PPP プロファイル > を渡さない、または空の文字列を渡す場合は、**IT_PPPODisconnect** はその開始されている接続を終了します。
- 複数の接続が開始されており、< PPP プロファイル > を渡さない、または空の文字列を渡す場合は、**IT_PPPODisconnect** はエラーを返し、いかなる接続も終了しません。
- < PPP プロファイル > を渡し、それが有効な場合は、開始されている接続の数に関わらず指定した接続を終了します。

■ MacOS の場合

この引数は考慮されません。

参照： **IT_PPPOConnect**、**IT_PPPOStatus**

IT_PPPStatus

IT_PPPStatus {(PPP プロファイル)} → 整数

引数	タイプ		説明
PPP プロファイル	文字列	→	ダイヤルアップ名 = Mac OS では空の文字列、Windows ではオプションで接続先を指定
戻り値	整数	←	接続されている場合は1、接続中の場合は0、エラーの場合は-1

IT_PPPStatus を使って、**IT_PPPConnect** コマンドまたは手動により開始された接続のステータスを確認することができます。

<PPP プロファイル> は、開始されているどの接続について確認するのかを指定するテキスト値です。Windows ではこの引数はオプションですが、ユーザのネットワーク設定に関わらず動作性を高めるためには有効です。

■ Windows の場合

- <PPP プロファイル> を渡し、それが有効な場合は、指定した接続のステータスを返します。
- <PPP プロファイル> を渡さない、または空の文字列を渡す場合は、**IT_PPPStatus** は次の値を返します。
 - ・複数の接続が開始されている場合は、-1
 - ・1つの接続のみが開始されている場合は、その開始されている接続のステータス

■ MacOS の場合

この引数は考慮されません。

IT_PPPStatus は接続のステータスを示す整数を返します。返される値は次の通りです。

- 1 = 接続されている場合
- 0 = 接続中の場合
- -1 = 接続が失敗した場合、または接続されていない場合

▼ 例

、メソッド GetMessages (このメソッドはプロセスにおいて実行します)

```

If (mPPPConnect($vPPPProfil; 120))
    $vErrCode:= IT_MacTCPInit
    If ($vErrCode=0)
        $vErrCode:= POP3_Login...
        ...
    Else
        ALERT ("Connection failed")
    End if
End if
、メソッド mPPPConnect
C_BOOLEAN ($0) ` 現在接続されている場合は True、接続が失敗した場合は
False を返します
C_TEXT ($1) ` Mac OS の場合は空の文字列、Windows の場合はエントリ名
C_INTEGER ($2) ` タイムアウトの秒数
If (IT_PPPStatus =1)
    $0:=True ` すでに接続されています
Else
    $vTimeoutLength:=$2
    $vTimeout:= False
    $vErr:= IT_PPPConnect ($1)
    If ($vErr=0)
        $vStart:= Current time
        Repeat
            DELAY PROCESS (Current process;30)
            $vStatus:= IT_PPPStatus ($1)
            $vTimeout:=(( Current time -$vStart)>
                $vTimeoutLength)
        Until (($vStatus=1) | $vTimeout) ` 接続されている状態または
            タイムアウト
        If (Not ($vTimeout))
            $0:=True ` 接続されています
        End if
    End if ` ... $Err = 0
End if.
```

参照： IT_PPPDisconnect、IT_PPPConnect

付録 A : プログラミングのヒント

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 Dimension 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れば、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ず、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 Dimension 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 Dimension 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 メールまたはIMAP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。“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。その代わりに、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、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。返信メッセージが一旦作成されると、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。

付録 B : TCP ポート番号

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) プロセスか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。

- 49152 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 コマンドを使用し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>) を参照してください。

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
accessbuilder	888	AccessBuilder

付録 C : エラーコード

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。
10086	プロキシ接続に失敗しました。
10087	標準ファイル I/O エラー
10088	FTP 再接続エラー
10089	SLI.DLL がロードされていません。
10091	IMAP 接続を行うことはできません。
10092	メールボックスが選択されていません。
10093	メッセージパートが無効です。
10094	IMAP LOGIN エラー
10095	IMAP LOGOUT エラー
10096	IMAP CAPABILITY エラー
10097	IMAP SELECT エラー
10098	IMAP FETCH エラー
10099	IMAP PARTIAL エラー
10100	IMAP STORE エラー
10101	IMAP EXPUNGE エラー
10102	IMAP SEARCH エラー
10103	IMAP COPY エラー
10104	IMAP CREATE エラー
10105	IMAP DELETE エラー
10106	IMAP RENAME エラー
10107	IMAP SUBSCRIBE エラー
10108	IMAP UNSUBSCRIBE エラー
10109	IMAP LIST エラー
10110	IMAP LSUB エラー
10111	IMAP STATUS エラー
10112	IMAP CLOSE エラー

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

互換性についての注意：4D Internet Commands バージョン 6.8.1 より、MacTCP はサポートされないため、下記のエラーコードには現在は有効ではないものがいくつか含まれています。

-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はありません。
- 23047 ドメインネームサーバ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ーサーバ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コマンドを使用し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。転送チャネルをクローズ中
- 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 ローカルユーザではありません。＜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コマンドを使用して独自のメール通信プロセス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。転送チャンネルをクローズ中
(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 ローカルユーザではありません。<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)。

付録 D : その他の情報

下記の WWW(World Wide Web)において、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.imap.org>

IMAPプロトコル専用のサイト。このサイトでIMAPプロトコルに関する有効な情報を得ることができます。

F

FTP_Append (ftp_ID;ローカルパス;ホストパス;進捗)→整数	192
FTP_ChangeDir (ftp_ID;ディレクトリ)→整数	180
FTP_Delete (ftp_ID;ホストパス)→整数	187
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	178
FTP_GetFileInfo (ftp_ID;ホストパス;サイズ;タイプ;修正日)→整数	182
FTP_GetPassive (ftp_ID;パッシブモード)→整数	200
FTP_GetType (ftp_ID;FTP モード)→整数	193
FTP_Login (ホスト名;ユーザ名;パスワード;ftp_ID{;ウエルカムテキスト})→整数	176
FTP_Logout (ftp_ID)→整数	198
FTP_MacBinary (ftp_ID;MacBinary モード)→整数	188
FTP_MakeDir (ftp_ID;ディレクトリ)→整数	184
FTP_PrintDir (ftp_ID;ディレクトリ)→整数	181
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	174
FTP_Receive (ftp_ID;ホストパス;ローカルパス;進捗)→整数	196
FTP_RemoveDir (ftp_ID;ディレクトリ)→整数	185
FTP_Rename (ftp_ID;ホストパス;新規パス名)→整数	186
FTP_Send (ftp_ID;ローカルパス;ホストパス;進捗)→整数	190
FTP_SetPassive (ftp_ID;パッシブモード)→整数	199
FTP_SetType (ftp_ID;FTP モード)→整数	194
FTP_System (ftp_ID;システム情報)→整数	195
FTP_VerifyID (ftp_ID)→整数	183

I

IMAP_Capability (imap_ID; ケイパビリティ) → 整数	106
IMAP_CloseCurrentMB (imap_ID) → 整数	118
IMAP_CopyToMB (imap_ID; 開始メッセージ; 終了メッセージ; ターゲットメールボックス名 {; メッセージ削除}) → 整数	148
IMAP_CreateMB (imap_ID; メールボックス名) → 整数	150
IMAP_Delete (imap_ID; 開始メッセージ; 終了メッセージ) → 整数	119
IMAP_DeleteMB (imap_ID; メールボックス名) → 整数	151
IMAP_Download (imap_ID; メッセージ番号; ヘッダのみ; ファイル名; 既読更新) → 整数	139
IMAP_GetCurrentMB (imap_ID; メールボックス名) → 整数	117
IMAP_GetFlags (imap_ID; 開始メッセージ; 終了メッセージ; メッセージフラグ配列; メッセージ番号配列) → 整数	130
IMAP_GetMBStatus (imap_ID; メールボックス名; メッセージ番号; 新着メッセージ番号; 未読メッセージ番号; ユニークID) → 整数	113

IMAP_GetMessage (imap_ID; メッセージ番号; オフセット; 文字数; メッセージパート; メッセージテキスト; 既読更新)	→ 整数	121
IMAP_GetPrefs (ラインフィード除去; メッセージフォルダ)	→ 整数	102
IMAP_ListMBs (imap_ID; リファレンス名; メールボックス名; メールボックス名配列; メールボックス属性配列; メールボックス階層配列 {; 登録メールボックス})	→ 整数	107
IMAP_Login (ホスト名; ユーザ名; パスワード; imap_ID)	→ 整数	103
IMAP_Logout (imap_ID)	→ 整数	153
IMAP_MsgFetch (imap_ID; メッセージ番号; データ項目; データ項目値)	→ 整数	131
IMAP_MsgInfo (imap_ID; メッセージ番号; メッセージサイズ; ユニークID)	→ 整数	120
IMAP_MsgLst (imap_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	125
IMAP_MsgLstInfo (imap_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	123
IMAP_MsgNumToUID (imap_ID; メッセージ番号; メッセージID)	→ 整数	142
IMAP_RenameMB (imap_ID; メールボックス名; 新しいメールボックス名)	→ 整数	152
IMAP_Search (imap_ID; 検索基準; メッセージ番号配列)	→ 整数	143
IMAP_SetCurrentMB (imap_ID; メールボックス名; メッセージ番号; 新着メッセージ番号; カスタムフラグ; 永久フラグ; ユニークID)	→ 整数	115
IMAP_SetFlags (imap_ID; 開始メッセージ; 終了メッセージ; フラグリスト; 削除オプション)	→ 整数	127
IMAP_SetPrefs (ラインフィード除去; メッセージフォルダ)	→ 整数	101
IMAP_SubscribeMB (imap_ID; メールボックス名; 登録メールボックス)	→ 整数	112
IMAP_UIDToMsgNum (imap_ID; メッセージID; メッセージ番号)	→ 整数	141
IMAP_VerifyID (imap_ID)	→ 整数	105
IT_Decode (ファイル名; デコードファイル; デコードモード)	→ 整数	236
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	234
IT_ErrorText (エラー)	→ 文字列	233
IT_GetPort (プロトコル; ポート)	→ 整数	241
IT_GetProxy (プロトコル; 種類; ホスト; ポート; ユーザID)	→ 整数	238
IT_GetTimeOut (タイムアウト)	→ 整数	232
IT_MacTCPInit	→ 整数	224
IT_MacTCPVer (バージョンコード)	→ 整数	228
IT_MyTCPAddr (IPアドレス; サブネット)	→ 整数	229
IT_Platform	→ 整数	225
IT_PPPConnect (PPPプロファイル)	→ 整数	243
IT_PPPDisconnect {(PPPプロファイル)}	→ 整数	244
IT_PPPStatus {(PPPプロファイル)}	→ 整数	245
IT_SetPort (プロトコル; ポート)	→ 整数	242
IT_SetProxy (プロトコル; 種類; ホスト; ポート; ユーザID)	→ 整数	239
IT_SetTimeOut (タイムアウト)	→ 整数	230
IT_TCPVersion (スタックタイプ; スタックバージョン)	→ 整数	227
IT_Version	→ 文字列	226

M

MSG_Charset (デコードヘッダ; 本文文字コードセット)	→ 整数	170
MSG_Delete (ファイル名; フォルダ)	→ 整数	169
MSG_Extract (ファイル名; デコード; 添付パス; 同封リスト)	→ 整数	167

MSG_FindHeader (ファイル名;ヘッダラベル;ヘッダ値)→整数	159
MSG_GetBody (ファイル名;オフセット;文字数;本文テキスト)→整数	163
MSG_GetHeaders (ファイル名;オフセット;文字数;ヘッダテキスト)→整数	162
MSG_GetMessage (ファイル名;オフセット;文字数;生テキスト)→整数	165
MSG_GetPrefs (ラインフィード除去;メッセージフォルダ;添付フォルダ)→整数	158
MSG_HasAttach (ファイル名;添付数)→整数	166
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	160
MSG_SetPrefs (ラインフィード除去;メッセージフォルダ;添付フォルダ)→整数	156

N

NET_AddrToName (IP_倍長整数;ホスト名;IP_アドレス)→整数	221
NET_Finger (ホスト名;検索テキスト;結果)→整数	214
NET_NameToAddr (ホスト名;IP_倍長整数)→整数	220
NET_Ping (ホスト名;テキスト;アクティブ{;タイムアウト})→整数	216
NET_Resolve (ホスト名;IP_ホスト)→整数	222
NET_Time (ホスト名;日付;時刻;オフセット)→整数	218

P

POP3_BoxInfo (pop3_ID;メッセージ数;メッセージサイズ)→整数	81
POP3_Charset (ヘッダデコード;本文文字セット)→整数	91
POP3_Delete (pop3_ID;開始メッセージ;終了メッセージ)→整数	79
POP3_Download (pop3_ID;メッセージ番号;ヘッダのみ;ファイル名)→整数	88
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	83
POP3_GetPrefs (改行文字;メッセージフォルダ;添付フォルダ)→整数	74
POP3_Login (ホスト名;ユーザ名;パスワード;APOP;pop3_ID)→整数	75
POP3_Logout (pop3_ID)→整数	80
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	82
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	86
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	84
POP3_Reset (pop3_ID)→整数	78
POP3_SetPrefs (改行文字;メッセージフォルダ;添付フォルダ)→整数	72
POP3_UIDToNum (pop3_ID;ユニークID;メッセージ番号)→整数	90
POP3_VerifyID (pop3_ID)→整数	77

S

SMTP_AddHeader (smtp_ID;ヘッダ名;ヘッダテキスト{;削除オプション})→整数	57
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	64
SMTP_Bcc (smtp_ID;ブラインドカーボン{;削除オプション})→整数	49
SMTP_Body (smtp_ID;メッセージ本文{;削除オプション})→整数	61
SMTP_Cc (smtp_ID;カーボンコピー{;削除オプション})→整数	48
SMTP_Charset (ヘッダエンコード;本文文字セット)→整数	66
SMTP_Clear (smtp_ID)→整数	37
SMTP_Comments (smtp_ID;コメント{;削除オプション})→整数	53
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	38

SMTP_Encrypted (smtp_ID;暗号化{;削除オプション})→整数	56
SMTP_From (smtp_ID;メッセージ差出人{;削除オプション})→整数	41
SMTP_GetPrefs(ラインフィード;本文のタイプ;行の最大長)→整数	29
SMTP_Host(smtp_ID;ホスト名{;削除オプション})→整数	34
SMTP_InReplyTo (smtp_ID;返信先テキスト{;削除オプション})→整数	51
SMTP_Keywords (smtp_ID;コメント{;削除オプション})→整数	55
SMTP_New(smtp_ID)→整数	33
SMTP_QuickSend(ホスト名;差出人;受取人;件名;メッセージ)→整数	30
SMTP_References (smtp_ID;リファレンス{;削除オプション})→整数	52
SMTP_ReplyTo (smtp_ID;返信先{;削除オプション})→整数	45
SMTP_Send(smtp_ID)→整数	35
SMTP_Sender (smtp_ID;メッセージ送信人{;削除オプション})→整数	43
SMTP_SetPrefs (ラインフィード;本文タイプ;行の最大長)→整数	26
SMTP_Subject (smtp_ID;件名{;削除オプション})→整数	59
SMTP_To (smtp_ID;メッセージ送信先{;削除オプション})→整数	47

T

TCP_Close (tcp_ID)→整数	212
TCP_Listen (リモートホスト;ローカルポート;リモートポート;タイムアウト;tcp_ID)→整数	204
TCP_Open (ホスト名;リモートポート;tcp_ID)→整数	202
TCP_Receive (tcp_ID;テキスト)→整数	207
TCP_ReceiveBLOB (tcp_ID;受信BLOB)→整数	209
TCP_Send (tcp_ID;送信テキスト)→整数	206
TCP_SendBLOB (tcp_ID;送信BLOB)→整数	208
TCP_State (tcp_ID;ステータスコード)→整数	210