

RFC 2812 - Internet Relay Chat: Client Protocol 日本語訳

原文URL : <https://datatracker.ietf.org/doc/html/rfc2812>

タイトル : RFC 2812 - インターネットリレーチャット : クライアントプロトコル

翻訳編集 : 自動生成、ST : **Informational**

Dark Orig

Network Working Group
Request for Comments: 2812
Updates: 1459
Category: Informational

C. Kalt
April 2000

Internet Relay Chat: Client Protocol

インターネットリレーチャット : クライアントプロトコル

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

本文書の位置付け

このメモは、インターネットコミュニティに情報を提供します。いかなる種類のインターネット標準を指定しません。このメモの配布は無制限です。

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

著作権表示

Copyright (c) The Internet Society (2000) 。全著作権所有。

IESG NOTE:

The IRC protocol itself enables several possibilities of transferring data between clients, and just like with other transfer mechanisms like email, the receiver of the data has to be careful about how the data is handled. For more information on security issues with the IRC protocol, see for example <http://www.irchelp.org/irchelp/security/>.

IESGノート :

IRCプロトコル自体は、クライアント間でデータを転送するいくつかの可能性を可能にし、電子メールなどの他の転送メカニズムと同様に、データの受信者はデータの処理方法に注意する必要があります。IRCプロトコルのセキュリティ問題の詳細については、<http://www.irchelp.org/irchelp/security/>を参照してください。

Abstract

The IRC (Internet Relay Chat) protocol is for use with text based conferencing; the simplest client being any socket program capable of connecting to the server.

概要

IRC（インターネットリレーチャット）プロトコルは、テキストベースの会議で使用するためのものです。最も単純なクライアントは、サーバーに接続できるソケットプログラムです。

This document defines the Client Protocol, and assumes that the reader is familiar with the IRC Architecture [IRC-ARCH].

このドキュメントはクライアントプロトコルを定義し、読者がIRCアーキテクチャ[IRC-ARCH]に精通していると想定しています。

Table of Contents

目次

1. Labels	3
1.1 Servers	3
1.2 Clients	3
1.2.1 Users	4
1.2.1.1 Operators	4
1.2.2 Services	4
1.3 Channels	4
2. The IRC Client Specification	5
2.1 Overview	5
2.2 Character codes	5
2.3 Messages	5
2.3.1 Message format in Augmented BNF	6
2.4 Numeric replies	8
2.5 Wildcard expressions	9
3. Message Details	9
3.1 Connection Registration	10
3.1.1 Password message	10
3.1.2 Nick message	10
3.1.3 User message	11
3.1.4 Oper message	12
3.1.5 User mode message	12
3.1.6 Service message	13
3.1.7 Quit	14
3.1.8 Squit	15
3.2 Channel operations	15
3.2.1 Join message	16
3.2.2 Part message	17
3.2.3 Channel mode message	18

3.2.4	Topic message	19
3.2.5	Names message	20
3.2.6	List message	21
3.2.7	Invite message	21
3.2.8	Kick command	22
3.3	Sending messages	23
3.3.1	Private messages	23
3.3.2	Notice	24
3.4	Server queries and commands	25
3.4.1	Motd message	25
3.4.2	Lusers message	25
3.4.3	Version message	26
3.4.4	Stats message	26
3.4.5	Links message	27
3.4.6	Time message	28
3.4.7	Connect message	28
3.4.8	Trace message	29
3.4.9	Admin command	30
3.4.10	Info command	31
3.5	Service Query and Commands	31
3.5.1	Servlist message	31
3.5.2	Squery	32
3.6	User based queries	32
3.6.1	Who query	32
3.6.2	Whois query	33
3.6.3	Whowas	34
3.7	Miscellaneous messages	34
3.7.1	Kill message	35
3.7.2	Ping message	36
3.7.3	Pong message	37
3.7.4	Error	37

4.	Optional features	38
4.1	Away	38
4.2	Rehash message	39
4.3	Die message	39
4.4	Restart message	40
4.5	Summon message	40
4.6	Users	41
4.7	Operwall message	41
4.8	Userhost message	42
4.9	Ison message	42
5.	Replies	43
5.1	Command responses	43
5.2	Error Replies	53
5.3	Reserved numerics	59
6.	Current implementations	60
7.	Current problems	60
7.1	Nicknames	60
7.2	Limitation of wildcards	61
7.3	Security considerations	61
8.	Current support and availability	61
9.	Acknowledgements	61
10.	References	62
11.	Author's Address	62
12.	Full Copyright Statement	63

1. Labels

This section defines the identifiers used for the various components of the IRC protocol.

1.1 Servers

Servers are uniquely identified by their name, which has a maximum length of sixty three (63) characters. See the protocol grammar rules (section 2.3.1) for what may and may not be used in a server name.

1.2 Clients

For each client all servers MUST have the following information: a netwide unique identifier (whose format depends on the type of client) and the server which introduced the client.

1.2.1 Users

1. ラベル

このセクションでは、IRCプロトコルのさまざまなコンポーネントに使用される識別子を定義します。

1.1 サーバー

サーバーは名前で一意に識別されます。この名前は、最大63文字（63）の文字です。サーバー名で利用される可能性のあるもの、および使用されないものについては、プロトコル文法ルール（セクション2.3.1）を参照してください。

1.2 クライアント

各クライアントについて、すべてのサーバーには次の情報が必要です。ネットワイドの一意の識別子（フォーマットはクライアントのタイプに依存します）とクライアントを導入したサーバーです。

1.2.1 ユーザー

Each user is distinguished from other users by a unique nickname having a maximum length of nine (9) characters. See the protocol grammar rules (section 2.3.1) for what may and may not be used in a nickname.

While the maximum length is limited to nine characters, clients SHOULD accept longer strings as they may become used in future evolutions of the protocol.

1.2.1.1 Operators

To allow a reasonable amount of order to be kept within the IRC network, a special class of users (operators) is allowed to perform general maintenance functions on the network.

Although the powers granted to an operator can be considered as 'dangerous', they are nonetheless often necessary.

Operators SHOULD be able to perform basic network tasks such as disconnecting and reconnecting servers as needed. In recognition of this need, the protocol discussed herein provides for operators only to be able to perform such functions. See sections 3.1.8 (SQUIT) and 3.4.7 (CONNECT).

A more controversial power of operators is the ability to remove a user from the connected network by 'force', i.e., operators are able to close the connection between any client and server. The justification for this is very delicate since its abuse is both destructive and annoying, and its benefits close to inexistent. For further details on this type of action, see section 3.7.1 (KILL).

1.2.2 Services

Each service is distinguished from other services by a service name composed of a nickname and a server name. As for users, the nickname has a maximum length of nine (9) characters. See the protocol grammar rules (section 2.3.1) for what may and may not be used in a nickname.

1.3 Channels

Channels names are strings (beginning with a '&', '#', '+' or '!' character) of length up to fifty (50) characters. Apart from the requirement that the first character is either '&', '#', '+' or '!', the only restriction on a channel name is that it SHALL NOT contain any spaces (' '), a control G (^G or ASCII 7), a comma (',').

Space is used as parameter separator and command is used as a list item separator by the protocol). A colon (':') can also be used as a delimiter for the channel mask. Channel names are case insensitive.

See the protocol grammar rules (section 2.3.1) for the exact syntax of a channel name.

Each prefix characterizes a different channel type. The definition of the channel types is not relevant to the client-server protocol and thus it is beyond the scope of this document. More details can be found in "Internet Relay Chat: Channel Management" [IRC-CHAN].

2. The IRC Client Specification

2.1 Overview

The protocol as described herein is for use only with client to server connections when the client registers as a user.

各ユーザーは、最大長さ9文字の一意のニックネームで他のユーザーと区別されます。ニックネームで使用される可能性のあるものとできないものについては、プロトコル文法規則（セクション2.3.1）を参照してください。

最大長は9文字に制限されていますが、クライアントはプロトコルの将来の進化で使用される可能性があるため、長い文字列を受け入れる必要があります。

1.2.1.1 オペレーター

IRCネットワーク内で合理的な量の注文を保持するために、特別なクラスのユーザー（オペレーター）がネットワーク上で一般的なメンテナンス機能を実行できます。オペレーターに付与された権限は「危険」と見なすことができますが、それでも必要です。オペレーターは、必要に応じてサーバーの切断や再接続などの基本的なネットワークタスクを実行できる必要があります。このニーズを認識して、ここで説明するプロトコルは、そのような機能を実行できるようにオペレーターを提供します。セクション3.1.8 (Squit) および3.4.7 (Connect) を参照してください。

より物議を醸す演算子の力は、「強制」によって接続されたネットワークからユーザーを削除する機能です。つまり、オペレーターはクライアントとサーバー間の接続を閉じることができます。この虐待は破壊的で迷惑なものであり、その利点は存在しないため、この正当性は非常に繊細です。このタイプのアクションの詳細については、セクション3.7.1 (Kill) を参照してください。

1.2.2 サービス

各サービスは、ニックネームとサーバー名で構成されるサービス名によって、他のサービスと区別されます。ユーザーに関しては、ニックネームの最大長は9文字です。ニックネームで使用される可能性のあるものとできないものについては、プロトコル文法規則（セクション2.3.1）を参照してください。

1.3 チャネル

チャンネル名は、長さの長さの文字列（'&', '#', '!', '文字'）から50文字までの文字列です。最初の文字が'&', '#', 'または'!'のいずれかであるという要件とは別に、チャンネル名の唯一の制限は、スペース（' '）、コントロールg（^を含めてはならないということです。gまたはascii 7）、コンマ（', '）。スペースはパラメーターセパレーターとして使用され、コマンドはプロトコルによってリストアイテムセパレーターとして使用されます）。コロンの（':'）は、チャンネルマスクの区切り文字としても使用できます。チャンネル名はケースの鈍感です。

チャンネル名の正確な構文については、プロトコル文法ルール（セクション2.3.1）を参照してください。

各プレフィックスは、異なるチャンネルタイプを特徴づけます。チャンネルタイプの定義は、クライアントサーバープロトコルには関係がないため、このドキュメントの範囲を超えています。詳細については、「Internet Relay Chat：Channel Management」[IRC-chan]をご覧ください。

2. IRCクライアントの仕様

2.1 概要

本明細書に記載されているプロトコルは、クライアントがユーザーとして登録されたときにクライアントからサーバーへの接続でのみ使用するためのものです。

2.2 Character codes

No specific character set is specified. The protocol is based on a set of codes which are composed of eight (8) bits, making up an octet. Each message may be composed of any number of these octets; however, some octet values are used for control codes, which act as message delimiters.

Regardless of being an 8-bit protocol, the delimiters and keywords are such that protocol is mostly usable from US-ASCII terminal and a telnet connection.

Because of IRC's Scandinavian origin, the characters {}|^ are considered to be the lower case equivalents of the characters []\~, respectively. This is a critical issue when determining the equivalence of two nicknames or channel names.

2.3 Messages

Servers and clients send each other messages, which may or may not generate a reply. If the message contains a valid command, as described in later sections, the client should expect a reply as specified but it is not advised to wait forever for the reply; client to server and server to server communication is essentially asynchronous by nature.

Each IRC message may consist of up to three main parts: the prefix (OPTIONAL), the command, and the command parameters (maximum of fifteen (15)). The prefix, command, and all parameters are separated by one ASCII space character (0x20) each.

The presence of a prefix is indicated with a single leading ASCII colon character (':', 0x3b), which MUST be the first character of the message itself. There MUST be NO gap (whitespace) between the colon and the prefix. The prefix is used by servers to indicate the true origin of the message. If the prefix is missing from the message, it is assumed to have originated from the connection from which it was received from. Clients SHOULD NOT use a prefix when sending a message; if they use one, the only valid prefix is the registered nickname associated with the client.

The command MUST either be a valid IRC command or a three (3) digit number represented in ASCII text.

IRC messages are always lines of characters terminated with a CR-LF (Carriage Return - Line Feed) pair, and these messages SHALL NOT exceed 512 characters in length, counting all characters including the trailing CR-LF. Thus, there are 510 characters maximum allowed for the command and its parameters. There is no provision for continuation of message lines. See section 6 for more details about current implementations.

2.3.1 Message format in Augmented BNF

The protocol messages must be extracted from the contiguous stream of octets. The current solution is to designate two characters, CR and LF, as message separators. Empty messages are silently ignored, which permits use of the sequence CR-LF between messages without extra problems.

The extracted message is parsed into the components <prefix>, <command> and list of parameters (<params>).

2.2 文字コード

特定の文字セットは指定されていません。プロトコルは、8つのビットで構成される一連のコードに基づいており、オクテットを構成します。各メッセージは、これらのオクテットの任意の数で構成されている場合があります。ただし、いくつかのオクテット値は、メッセージデリミターとして機能する制御コードに使用されます。

8ビットのプロトコルであることに関係なく、デリミターとキーワードは、プロトコルが米国ASCII端末とTelnet接続からほとんど使用可能であるようなものです。

IRCのスカンジナビア起源のため、キャラクター{}|^は、それぞれ文字[]\~の小文字と同等であると考えられています。これは、2つのニックネームまたはチャンネル名の等価性を決定する場合の重要な問題です。

2.3 メッセージ

サーバーとクライアントはお互いのメッセージを送信します。これは、返信を生成する場合と生成できない場合があります。メッセージに後のセクションで説明されているように、メッセージに有効なコマンドが含まれている場合、クライアントは指定されたとおりに返信を期待する必要がありますが、返信を永久に待つことはお勧めしません。クライアントからサーバー、サーバーへの通信は、本質的に非同期ではありません。

各IRCメッセージは、プレフィックス（オプション）、コマンド、およびコマンドパラメーター（最大15（15））の最大3つの主要な部分で構成されている場合があります。接頭辞、コマンド、およびすべてのパラメーターは、それぞれ1つのASCIIスペース文字（0x20）で区切られています。

接頭辞の存在は、単一の主要なASCIIコロン文字（':', 0x3b）で示されています。これは、メッセージ自体の最初の文字でなければなりません。コロンと接頭辞の間にギャップ（空白）はありません。プレフィックスは、メッセージの真の起源を示すためにサーバーによって使用されます。プレフィックスがメッセージから欠落している場合、受信した接続から発信されたと想定されます。クライアントは、メッセージを送信するときにプレフィックスを使用しないでください。それらを使用する場合、唯一の有効なプレフィックスは、クライアントに関連付けられた登録されたニックネームです。

コマンドは、有効なIRCコマンドまたはASCIIテキストで表される3桁の数字でなければなりません。

IRCメッセージは、常にCR-LF（キャリッジリターン - ラインフィード）ペアで終了した文字の行の行であり、これらのメッセージは、Trailing CR-LFを含むすべての文字をカウントする長さ512文字を超えてはなりません。したがって、コマンドとそのパラメーターに許可される最大510文字があります。メッセージ行の継続に関する規定はありません。現在の実装の詳細については、セクション6を参照してください。

2.3.1 拡張BNFのメッセージ形式

プロトコルメッセージは、オクテットの連続的なストリームから抽出する必要があります。現在の解決策は、メッセージセパレーターとして2つの文字、CRとLFを指定することです。空のメッセージは静かに無視されているため、追加の問題なくメッセージ間でシーケンスCR-LFを使用できます。

抽出されたメッセージは、コンポーネント<プレフィックス>、<コマンド>、およびパラメーターのリスト（<params>）に解析

されます。

The Augmented BNF representation for this is:

このための拡張BNF表現は次のとおりです。

message	=	[":" prefix SPACE] command [params] crlf
prefix	=	servername / (nickname [["!" user] "@" host])
command	=	1*letter / 3digit
params	=	*14(SPACE middle) [SPACE ":" trailing] = / 14(SPACE middle) [SPACE [":"] trailing]

nospcrlfcl	=	%x01-09 / %x0B-0C / %x0E-1F / %x21-39 / %x3B-FF ; any octet except NUL, CR, LF, " " and ":"
middle	=	nospcrlfcl *(":" / nospcrlfcl)
trailing	=	*(":" / " " / nospcrlfcl)

SPACE	=	%x20 ; space character
crlf	=	%x0D %x0A ; "carriage return" "linefeed"

NOTES: 1) After extracting the parameter list, all parameters are equal whether matched by <middle> or <trailing>. <trailing> is just a syntactic trick to allow SPACE within the parameter.

2) The NUL (%x00) character is not special in message framing, and basically could end up inside a parameter, but it would cause extra complexities in normal C string handling. Therefore, NUL is not allowed within messages.

Most protocol messages specify additional semantics and syntax for the extracted parameter strings dictated by their position in the list. For example, many server commands will assume that the first parameter after the command is the list of targets, which can be described with:

注：1) パラメーターリストを抽出した後、すべてのパラメーターは<middle>または<trailing>で一致するかどうかにかかわらず等しくなります。<trailing>は、パラメーター内のスペースを許可するための構文的なトリックです。

2) NUL（%x00）の文字はメッセージフレーミングに特別なものではなく、基本的にパラメーター内に終わる可能性があります。通常のCストリングの処理に余分な複雑さを引き起こします。したがって、NULはメッセージ内で許可されていません。

ほとんどのプロトコルメッセージは、リスト内の位置によって決定される抽出されたパラメーター文字列の追加のセマンティクスと構文を指定します。たとえば、多くのサーバーコマンドは、コマンド後の最初のパラメーターがターゲットのリストであると想定します。これは以下で説明できます。

target	=	nickname / server
msgtarget	=	msgto *("," msgto)
msgto	=	channel / (user ["%" host] "@" servername)
msgto	= /	(user "%" host) / targetmask
msgto	= /	nickname / (nickname "!" user "@" host)
channel	=	("#" / "+" / ("!" channelid) / "&") chanstring [":" chanstring]
servername	=	hostname
host	=	hostname / hostaddr
hostname	=	shortname *("." shortname)
shortname	=	(letter / digit) *(letter / digit / "-") *(letter / digit) ; as specified in RFC 1123 [HNAME]
hostaddr	=	ip4addr / ip6addr
ip4addr	=	1*3digit "." 1*3digit "." 1*3digit "." 1*3digit
ip6addr	=	1*hexdigit 7(":" 1*hexdigit)
ip6addr	= /	"0:0:0:0:0:" ("0" / "FFFF") ":" ip4addr
nickname	=	(letter / special) *8(letter / digit / special / "-")
targetmask	=	("\$" / "#") mask ; see details on allowed masks in section 3.3.1
chanstring	=	%x01-07 / %x08-09 / %x0B-0C / %x0E-1F / %x21-2B
chanstring	= /	%x2D-39 / %x3B-FF ; any octet except NUL, BELL, CR, LF, " ", ",", and ":"
channelid	=	5(%x41-5A / digit) ; 5(A-Z / 0-9)

Other parameter syntaxes are:

他のパラメーター構文は次のとおりです。

user	=	1*(%x01-09 / %x0B-0C / %x0E-1F / %x21-3F / %x41-FF) ; any octet except NUL, CR, LF, " " and "@"
key	=	1*23(%x01-05 / %x07-08 / %x0C / %x0E-1F / %x21-7F) ; any 7-bit US_ASCII character, ; except NUL, CR, LF, FF, h/v TABs, and " "
letter	=	%x41-5A / %x61-7A ; A-Z / a-z
digit	=	%x30-39 ; 0-9
hexdigit	=	digit / "A" / "B" / "C" / "D" / "E" / "F"
special	=	%x5B-60 / %x7B-7D ; "[", "]", "\", "`", "_", "^", "{", " ", "]"

NOTES: 1) The <hostaddr> syntax is given here for the sole purpose of indicating the format to follow for IP addresses. This reflects the fact that the only available implementations of this protocol uses TCP/IP as underlying network protocol but is not meant to prevent other protocols to be used.

2) <hostname> has a maximum length of 63 characters. This is a limitation of the protocol as internet hostnames (in particular) can be longer. Such restriction is necessary because IRC messages are limited to 512 characters in length. Clients connecting from a host which name is longer than 63 characters are registered using the host (numeric) address instead of the host name.

3) Some parameters used in the following sections of this documents are not defined here as there is nothing specific about them besides the name that is used for convenience. These parameters follow the general syntax defined for <params>.

2.4 Numeric replies

Most of the messages sent to the server generate a reply of some sort. The most common reply is the numeric reply, used for both errors and normal replies. The numeric reply MUST be sent as one message consisting of the sender prefix, the three-digit numeric, and the target of the reply. A numeric reply is not allowed to originate from a client. In all other respects, a numeric reply is just like a normal message, except that the keyword is made up of 3 numeric digits rather than a string of letters. A list of different replies is supplied in section 5 (Replies).

2.5 Wildcard expressions

When wildcards are allowed in a string, it is referred as a "mask".

For string matching purposes, the protocol allows the use of two special characters: '?' (%x3F) to match one and only one character, and '*' (%x2A) to match any number of any characters. These two characters can be escaped using the character '\' (%x5C).

The Augmented BNF syntax for this is:

mask	=	*(nowild / noesc wilddone / noesc wilddmany)
wilddone	=	%x3F
wilddmany	=	%x2A
nowild	=	%x01-29 / %x2B-3E / %x40-FF ; any octet except NUL, "*", "?"
noesc	=	%x01-5B / %x5D-FF ; any octet except NUL and "\"
matchone	=	%x01-FF ; matches wilddone
matchmany	=	*matchone ; matches wilddmany

Examples:

a?c ; Matches any string of 3 characters in length starting with "a" and ending with "c"

a*c ; Matches any string of at least 2 characters in length starting with "a" and ending with "c"

3. Message Details

注：1) IPアドレスをフォローする形式を示すという唯一の目的のために、<HostAddr>構文はここに示されています。これは、このプロトコルの唯一の利用可能な実装がTCP/IPを基礎となるネットワークプロトコルとして使用しているが、他のプロトコルの使用を防ぐことを意図していないという事実を反映しています。

2) <hostname>の最大長は63文字です。これは、インターネットホスト名（特に）が長くなる可能性があるため、プロトコルの制限です。IRCメッセージの長さは512文字に制限されているため、このような制限が必要です。名前が63文字より長いホストから接続するクライアントは、ホスト名の代わりにホスト（数値）アドレスを使用して登録されます。

3) このドキュメントの以下のセクションで使用されているいくつかのパラメーターは、便利さのために使用される名前以外に具体的なものがないため、ここでは定義されていません。これらのパラメーターは、<params>で定義された一般的な構文に従います。

2.4 数値応答

サーバーに送信されたメッセージのほとんどは、ある種の返信を生成します。最も一般的な応答は、エラーと通常の応答の両方に使用される数値応答です。数値応答は、送信者プレフィックス、3桁の数値、および応答のターゲットで構成される1つのメッセージとして送信する必要があります。数値応答は、クライアントから発生することは許可されていません。他のすべての点で、数値応答は通常のメッセージのようなものですが、キーワードは文字列ではなく3桁で構成されています。異なる返信のリストは、セクション5（返信）に提供されています。

2.5 ワイルドカードの表現

ワイルドカードが文字列で許可される場合、それは「マスク」と呼ばれます。

文字列のマッチング目的では、プロトコルでは、2つの特殊文字を使用できます。「？」（%x3f）1つの文字と1つの文字を一致させ、「*」（%x2a）に任意の数の文字を一致させます。これらの2つのキャラクターは、文字 '\'（%x5c）を使用して逃げることができます。

このための拡張BNF構文は次のとおりです。

例：

交流；「A」で始まり、「C」で終わる長さの3文字の任意の文字列に一致します

交流；「A」から始まり、「C」で終わる長さの少なくとも2文字の任意の文字列に一致します

3. メッセージの詳細

On the following pages there are descriptions of each message recognized by the IRC server and client. All commands described in this section MUST be implemented by any server for this protocol.

Where the reply ERR_NOSUCHSERVER is returned, it means that the target of the message could not be found. The server MUST NOT send any other replies after this error for that command.

The server to which a client is connected is required to parse the complete message, and return any appropriate errors.

If multiple parameters is presented, then each MUST be checked for validity and appropriate responses MUST be sent back to the client. In the case of incorrect messages which use parameter lists with comma as an item separator, a reply MUST be sent for each item.

3.1 Connection Registration

The commands described here are used to register a connection with an IRC server as a user as well as to correctly disconnect.

A "PASS" command is not required for a client connection to be registered, but it MUST precede the latter of the NICK/USER combination (for a user connection) or the SERVICE command (for a service connection). The RECOMMENDED order for a client to register is as follows:

1. Pass message
2. Nick message
2. Service message
3. User message

Upon success, the client will receive an RPL_WELCOME (for users) or RPL_YOURESERVICE (for services) message indicating that the connection is now registered and known to the entire IRC network. The reply message MUST contain the full client identifier upon which it was registered.

3.1.1 Password message

Command: PASS Parameters: <password>

The PASS command is used to set a 'connection password'. The optional password can and MUST be set before any attempt to register the connection is made. Currently this requires that user send a PASS command before sending the NICK/USER combination.

Numeric Replies:

ERR_NEEDMOREPARAMS ERR_ALREADYREGISTRED

Example:

PASS secretpasswordhere

3.1.2 Nick message

Command: NICK Parameters: <nickname>

NICK command is used to give user a nickname or change the existing one.

Numeric Replies:

次のページには、IRCサーバーとクライアントによって認識された各メッセージの説明があります。このセクションで説明されているすべてのコマンドは、このプロトコルのために任意のサーバーによって実装する必要があります。

返信がerr_nosuchserverが返される場合、メッセージのターゲットが見つからなかったことを意味します。このコマンドのこのエラーの後、サーバーは他の返信を送信してはなりません。

クライアントが接続されているサーバーは、完全なメッセージを解析し、適切なエラーを返すために必要です。

複数のパラメーターが提示されている場合、それぞれが有効性をチェックし、適切な応答をクライアントに送り返す必要があります。アイテムセパレーターとしてコンマを使用してパラメーターリストを使用する誤ったメッセージの場合、アイテムごとに返信を送信する必要があります。

3.1 接続登録

ここで説明するコマンドは、IRCサーバーとの接続をユーザーとして登録し、正しく切断するために使用されます。

クライアント接続を登録するには「パス」コマンドは必要ありませんが、ニック/ユーザーの組み合わせ（ユーザー接続の場合）またはサービスコマンド（サービス接続用）の後者の先行する必要があります。クライアントが登録するための推奨注文は次のとおりです。

- 1.渡すメッセージ
- 2.ニックメッセージ
- 2.サービスメッセージ
- 3.ユーザーメッセージ

成功すると、クライアントはRPL_Welcome（ユーザー向け）またはRPL_YourEService（サービス用）メッセージを受け取り、接続が現在登録され、IRCネットワーク全体を既知になっていることを示します。返信メッセージには、登録された完全なクライアント識別子が含まれている必要があります。

3.1.1 パスワードメッセージ

パスコマンドは、「接続パスワード」を設定するために使用されます。オプションのパスワードは、接続を登録しようとする前に設定することができます。現在、これには、ニック/ユーザーの組み合わせを送信する前に、ユーザーがパスコマンドを送信する必要があります。

数値応答：

err_needmoreparams err_alreadyregistred

例：

SecretPasswordhereをパスします

3.1.2 ニックメッセージ

Nick Commandは、ユーザーにニックネームを与えるか、既存のものを変更するために使用されます。

数値応答：

ERR_NONICKNAMEGIVEN ERR_ERRONEUSNICKNAME
ERR_NICKNAMEINUSE ERR_NICKCOLLISION
ERR_UNAVAILRESOURCE ERR_RESTRICTED

Examples:

NICK Wiz ; Introducing new nick "Wiz" if session is still
unregistered, or user changing his nickname to "Wiz"

:WiZ!jto@tolsun.oulu.fi NICK Kilroy ; Server telling that WiZ
changed his nickname to Kilroy.

3.1.3 User message

Command: USER Parameters: <user> <mode> <unused> <realname>
--

The USER command is used at the beginning of connection to
specify the username, hostname and realname of a new user.

The <mode> parameter should be a numeric, and can be used
to automatically set user modes when registering with the
server. This parameter is a bitmask, with only 2 bits having any
signification: if the bit 2 is set, the user mode 'w' will be set and
if the bit 3 is set, the user mode 'i' will be set. (See Section 3.1.5
"User Modes").

The <realname> may contain space characters.

Numeric Replies:

ERR_NEEDMOREPARAMS ERR_ALREADYREGISTERED

Example:

USER guest 0 * :Ronnie Reagan ; User registering themselves
with a username of "guest" and real name "Ronnie Reagan".

USER guest 8 * :Ronnie Reagan ; User registering themselves
with a username of "guest" and real name "Ronnie Reagan",
and asking to be set invisible.

3.1.4 Oper message

Command: OPER Parameters: <name> <password>
--

A normal user uses the OPER command to obtain operator
privileges. The combination of <name> and <password> are
REQUIRED to gain Operator privileges. Upon success, the user
will receive a MODE message (see section 3.1.5) indicating the
new user modes.

Numeric Replies:

ERR_NEEDMOREPARAMS RPL_YOUREOPER
ERR_NOOPERHOST ERR_PASSWDMISMATCH

Example:

OPER foo bar ; Attempt to register as an operator using a
username of "foo" and "bar" as the password.

3.1.5 User mode message

Command: MODE Parameters: <nickname>

err_nonicknamegiven err_erroneusnickname
err_nicknameinuse err_nickcollision err_unavailresource
err_restricted

例：

ニック・ウィズ;セッションがまだ登録されていない場合、また
はユーザーのニックネームを「wiz」に変更した場合、新しいニ
ック「ウィズ」を紹介します

：wiz ! jto@tolsun.oulu.fi nick kilroy;WizがニックネームをKilroy
に変更したことを伝えるサーバー。

3.1.3 ユーザーメッセージ

ユーザーコマンドは、接続の先頭に使用され、新しいユーザー
のユーザー名、ホスト名、およびRealNameを指定します。

<mode>パラメーターは数値である必要があり、サーバーに登録
するときにユーザーモードを自動的に設定するために使用でき
ます。このパラメーターはビットマスクで、2ビットのみが意味
を持ちます。ビット2が設定されている場合、ユーザーモード
「W」が設定され、ビット3が設定されている場合、ユーザーモ
ード「I」が設定されます。（セクション3.1.5「ユーザーモー
ド」を参照）。

<RealName>にはスペース文字が含まれている場合があります。

数値応答：

err_needmoreparams err_alreadyregistred

例：

ユーザーゲスト0*：ロニーレーガン。ユーザーは、「ゲスト」
と本名の「ロニー・レーガン」のユーザー名に登録していま
す。

ユーザーゲスト8*：ロニーレーガン。ユーザーは、「ゲスト」
と本名の「ロニーレーガン」のユーザー名で自分自身を登録
し、目に見えないように設定するように求めます。

3.1.4 オペレーションメッセージ

通常のユーザーは、オペレーターコマンドを使用してオペレー
ターの特権を取得します。オペレーターの特権を獲得するに
は、<name>と<パスワード>の組み合わせが必要です。成功する
と、ユーザーは新しいユーザーモードを示すモードメッセージ
（セクション3.1.5を参照）を受信します。

数値応答：

err_needmoreparams rpl_youreoper err_nooperhost
err_passwdmismatch

例：

オペラフーバー。パスワードとして「foo」と「bar」のユーザ
ー名を使用してオペレーターとして登録しようとしています。

3.1.5 ユーザーモードメッセージ

*(("+" / "-") *("i" / "w" / "o" / "0" / "r"))

The user MODE's are typically changes which affect either how the client is seen by others or what 'extra' messages the client is sent.

A user MODE command MUST only be accepted if both the sender of the message and the nickname given as a parameter are both the same. If no other parameter is given, then the server will return the current settings for the nick.

The available modes are as follows:

a - user is flagged as away;
i - marks a users as invisible;
w - user receives wallops;
r - restricted user connection;
o - operator flag;
0 - local operator flag;
s - marks a user for receipt of server notices.

Additional modes may be available later on.

The flag 'a' SHALL NOT be toggled by the user using the MODE command, instead use of the AWAY command is REQUIRED.

If a user attempts to make themselves an operator using the "+o" or "+O" flag, the attempt SHOULD be ignored as users could bypass the authentication mechanisms of the OPER command. There is no restriction, however, on anyone `deopping' themselves (using "-o" or "-O").

On the other hand, if a user attempts to make themselves unrestricted using the "-r" flag, the attempt SHOULD be ignored. There is no restriction, however, on anyone `deopping' themselves (using "+r"). This flag is typically set by the server upon connection for administrative reasons. While the restrictions imposed are left up to the implementation, it is typical that a restricted user not be allowed to change nicknames, nor make use of the channel operator status on channels.

The flag 's' is obsolete but MAY still be used.

Numeric Replies:

ERR_NEEDMOREPARAMS ERR_USERSDONTMATCH
ERR_UMODEUNKNOWNFLAG RPL_UMODEIS

Examples:

MODE WiZ -w ; Command by WiZ to turn off reception of WALLOPS messages.

MODE Angel +i ; Command from Angel to make herself invisible.

MODE WiZ -o ; WiZ 'deopping' (removing operator status).

3.1.6 Service message

Command: SERVICE
Parameters: <nickname> <reserved> <distribution> <type>
 <reserved> <info>

The SERVICE command to register a new service. Command

ユーザーモードは通常、クライアントが他の人にどのように見られるか、クライアントが送信する「余分な」メッセージのいずれかに影響する変更です。

メッセージの送信者とパラメーターとして指定されたニックネームの両方が同じである場合にのみ、ユーザーモードコマンドを受け入れる必要があります。他のパラメーターが指定されていない場合、サーバーはニックの現在の設定を返します。

使用可能なモードは次のとおりです。

追加のモードが後で利用可能になる場合があります。

フラグ「A」は、モードコマンドを使用してユーザーによって切り替えられないものではなく、代わりにアウェイコマンドの使用が必要です。

ユーザーが「O」または「O」フラグを使用してオペレーターにしようとする場合、ユーザーがオペラコマンドの認証メカニズムをバイパスできるため、試行は無視する必要があります。ただし、自分で「脱いで」（「-o」または「-o」を使用する）人には制限はありません。

一方、ユーザーが「-R」フラグを使用して無制限にしようとする場合、試みは無視されるべきです。ただし、自分で「脱いで」（「r」を使用する）人には制限はありません。このフラグは通常、管理上の理由で接続時にサーバーによって設定されます。課された制限は実装に任されていますが、制限されたユーザーがニックネームを変更することも許可されたり、チャンネル上のチャンネル演算子のステータスを使用したりすることも許可されていないことが典型的です。

Flagの「S」は時代遅れですが、それでも使用される場合があります。

数値応答：

err_needmoreparams err_usersdontmatch
err_umodeunknownflag rpl_umodeis

例：

モードwiz -w;Wallopsメッセージの受信をオフにするようにWizによるコマンド。

モードエンジェル;エンジェルからのコマンドは、自分自身を見えないようにする。

モードwiz -o;wiz 'dropping'（オペレーターステータスの削除）。

3.1.6 サービスメッセージ

新しいサービスを登録するサービスコマンド。コマンドパラメ

parameters specify the service nickname, distribution, type and info of a new service.

The <distribution> parameter is used to specify the visibility of a service. The service may only be known to servers which have a name matching the distribution. For a matching server to have knowledge of the service, the network path between that server and the server on which the service is connected MUST be composed of servers which names all match the mask.

The <type> parameter is currently reserved for future usage.

Numeric Replies:

ERR_ALREADYREGISTERED ERR_NEEDMOREPARAMS
ERR_ERRONEUSNICKNAME RPL_YOURESERVICE
RPL_YOURLHOST RPL_MYINFO

Example:

SERVICE dict * *.fr 0 0 :French Dictionary ; Service registering itself with a name of "dict". This service will only be available on servers which name matches "/*.fr".

3.1.7 Quit

Command: QUIT Parameters: [<Quit Message>]

A client session is terminated with a quit message. The server acknowledges this by sending an ERROR message to the client.

Numeric Replies:

None.

Example:

QUIT :Gone to have lunch ; Preferred message format.

:syрк!kalt@millennium.stealth.net QUIT :Gone to have lunch ; User syрк has quit IRC to have lunch.

3.1.8 Squit

Command: SQUIT Parameters: <server> <comment>
--

The SQUIT command is available only to operators. It is used to disconnect server links. Also servers can generate SQUIT messages on error conditions. A SQUIT message may also target a remote server connection. In this case, the SQUIT message will simply be sent to the remote server without affecting the servers in between the operator and the remote server.

The <comment> SHOULD be supplied by all operators who execute a SQUIT for a remote server. The server ordered to disconnect its peer generates a WALLOPS message with <comment> included, so that other users may be aware of the reason of this action.

Numeric replies:

ERR_NOPRIVILEGES ERR_NOSUCHSERVER
ERR_NEEDMOREPARAMS

ータ新しいサービスのニックネーム、配布、タイプ、情報を指定します。

<distribution>パラメーターは、サービスの可視性を指定するために使用されます。このサービスは、分布に一致する名前のサーバーに対してのみ知られている場合があります。一致するサーバーがサービスの知識を持つために、そのサーバーとサービスが接続されているサーバーの間のネットワークパスは、すべてマスクと一致するサーバーで構成する必要があります。

<type>パラメーターは現在、将来の使用のために予約されています。

数値応答：

err_alreadyregistered err_needmoreparams
err_erroneusnickname rpl_youreservice rpl_yourhost
rpl_myinfo

例：

サービスdict * *.fr 0 0：フランス語辞書; 「dict」の名前で登録するサービス。このサービスは、「/*.fr」と一致する名前が付いたサーバーでのみ使用できます。

3.1.7 終了する

クライアントセッションは、終了メッセージで終了します。サーバーは、クライアントにエラーメッセージを送信することにより、これを認めます。

数値応答：

なし。

例：

終了：昼食をとった。優先メッセージ形式。

：syрк ! kalt@millennium.stealth.net quit：昼食をとりました。ユーザーシルクはIRCをやめて昼食をとっています。

3.1.8 squat

Squitコマンドは、オペレーターのみが利用できます。サーバーリンクを切断するために使用されます。また、サーバーはエラー条件でSquitメッセージを生成できます。Squitメッセージは、リモートサーバー接続をターゲットにする場合があります。この場合、Squitメッセージは、オペレーターとリモートサーバーの間のサーバーに影響を与えることなく、単にリモートサーバーに送信されます。

<コメント>は、リモートサーバーのスQUITを実行するすべてのオペレーターから提供される必要があります。他のユーザーがこのアクションの理由を認識できるように、ピアを切断するように命じられたサーバーは、<コメント>を含むWallopsメッセージを生成します。

数値応答：

err_noprivileges err_nosuchserver err_needmoreparams

Examples:

SQUIT tolsun.oulu.fi :Bad Link ? ; Command to uplink of the server tolson.oulu.fi to terminate its connection with comment "Bad Link".

:Trillian SQUIT cm22.eng.umd.edu :Server out of control ; Command from Trillian from to disconnect "cm22.eng.umd.edu" from the net with comment "Server out of control".

3.2 Channel operations

This group of messages is concerned with manipulating channels, their properties (channel modes), and their contents (typically users). For this reason, these messages SHALL NOT be made available to services.

All of these messages are requests which will or will not be granted by the server. The server MUST send a reply informing the user whether the request was granted, denied or generated an error. When the server grants the request, the message is typically sent back (eventually reformatted) to the user with the prefix set to the user itself.

The rules governing how channels are managed are enforced by the servers. These rules are beyond the scope of this document. More details are found in "Internet Relay Chat: Channel Management" [IRC-CHAN].

3.2.1 Join message

Command: JOIN
Parameters: (<channel> *("," <channel>) [<key> *("," <key>)]) / "0"

The JOIN command is used by a user to request to start listening to the specific channel. Servers MUST be able to parse arguments in the form of a list of target, but SHOULD NOT use lists when sending JOIN messages to clients.

Once a user has joined a channel, he receives information about all commands his server receives affecting the channel. This includes JOIN, MODE, KICK, PART, QUIT and of course PRIVMSG/NOTICE. This allows channel members to keep track of the other channel members, as well as channel modes.

If a JOIN is successful, the user receives a JOIN message as confirmation and is then sent the channel's topic (using RPL_TOPIC) and the list of users who are on the channel (using RPL_NAMREPLY), which MUST include the user joining.

Note that this message accepts a special argument ("0"), which is a special request to leave all channels the user is currently a member of. The server will process this message as if the user had sent a PART command (See Section 3.2.2) for each channel he is a member of.

Numeric Replies:

ERR_NEEDMOREPARAMS ERR_BANNEDFROMCHAN
ERR_INVITEONLYCHAN ERR_BADCHANNELKEY
ERR_CHANNELISFULL ERR_BADCHANMASK

例：

Squit Tolsun.oulu.fi：リンクが悪いですか？;コメント「悪いリンク」との接続を終了するためのサーバーtolson.oulu.fiのアップリンクへのコマンド。

：Trillian Squit CM22.eng.umd.edu：server of of Control;Trillianからのコマンドは、「CM22.eng.umd.edu」をネットから切断し、コメント「サーバー」を制御します。

3.2 チャネル操作

このメッセージのグループは、チャネルの操作、そのプロパティ（チャネルモード）、およびその内容（通常はユーザー）に関係しています。このため、これらのメッセージをサービスで利用できるようにしてはなりません。

これらのメッセージはすべて、サーバーによって許可されるか、許可されないリクエストです。サーバーは、リクエストが許可、拒否、またはエラーの生成をユーザーに通知する返信を送信する必要があります。サーバーがリクエストを付与すると、メッセージは通常、ユーザー自体にプレフィックスを設定してユーザーに送信されます（最終的には再フォーマットされます）。

チャネルの管理方法を管理するルールは、サーバーによって実施されます。これらのルールは、このドキュメントの範囲を超えています。詳細については、「Internet Relay Chat：Channel Management」[IRC-chan]をご覧ください。

3.2.1 メッセージに参加します

Joinコマンドは、特定のチャネルの聴取を開始するように要求するためにユーザーによって使用されます。サーバーは、ターゲットのリストの形式で引数を解析する必要がありますが、参加メッセージをクライアントに送信するときはリストを使用しないでください。

ユーザーがチャネルに参加すると、彼は自分のサーバーがチャネルに影響を与えるすべてのコマンドに関する情報を受け取ります。これには、参加、モード、キック、パート、終了、そしてもちろんprivmsg/nocugeが含まれます。これにより、チャネルメンバーは他のチャネルメンバーとチャネルモードを追跡できます。

結合が成功した場合、ユーザーは確認メッセージを確認し、チャンネルのトピック（RPL_TOPICを使用）とチャンネルにいるユーザーのリスト（RPL_NAMREPLYを使用）を送信します（RPL_NAMREPLYを使用）。

このメッセージは、ユーザーが現在メンバーであるすべてのチャネルを残すための特別なリクエストである特別な引数（"0"）を受け入れることに注意してください。サーバーは、ユーザーがメンバーである各チャネルの部品コマンド（セクション3.2.2を参照）を送信したかのようにこのメッセージを処理します。

数値応答：

err_needmoreparams err_bannedfromchan
err_inviteonlychan err_badchannelkey err_channelisfull
err_badchanmask err_nosuchchannel err_toomanychannels
err_toomanytargets err_unavailresource rpl_topicicic

ERR_NOSUCHCHANNEL ERR_TOOMANYCHANNELS
ERR_TOOMANYTARGETS ERR_UNAVAILRESOURCE
RPL_TOPIC

Examples:

JOIN #foobar ; Command to join channel #foobar.

JOIN &foo fubar ; Command to join channel &foo using key "fubar".

JOIN #foo,&bar fubar ; Command to join channel #foo using key "fubar" and &bar using no key.

JOIN #foo,#bar fubar,foobar ; Command to join channel #foo using key "fubar", and channel #bar using key "foobar".

JOIN #foo,#bar ; Command to join channels #foo and #bar.

JOIN 0 ; Leave all currently joined channels.

:WiZ!jto@tolsun.oulu.fi JOIN #Twilight_zone ; JOIN message from WiZ
on channel #Twilight_zone

3.2.2 Part message

Command: PART
Parameters: <channel> *("," <channel>) [<Part Message>]

The PART command causes the user sending the message to be removed from the list of active members for all given channels listed in the parameter string. If a "Part Message" is given, this will be sent instead of the default message, the nickname. This request is always granted by the server.

Servers MUST be able to parse arguments in the form of a list of target, but SHOULD NOT use lists when sending PART messages to clients.

Numeric Replies:

ERR_NEEDMOREPARAMS ERR_NOSUCHCHANNEL
ERR_NOTONCHANNEL

Examples:

PART #twilight_zone ; Command to leave channel "#twilight_zone"

PART #oz-ops,&group5 ; Command to leave both channels "&group5" and "#oz-ops".

:WiZ!jto@tolsun.oulu.fi PART #playzone :I lost ; User WiZ leaving channel "#playzone" with the message "I lost".

3.2.3 Channel mode message

Command: MODE
Parameters: <channel> *(("-" / "+") *<modes> *<modeparams>)

The MODE command is provided so that users may query and change the characteristics of a channel. For more details on available modes and their uses, see "Internet Relay Chat:

例：

#foobarに参加してください。チャンネル#FOOBARに参加するコマンド。

参加&foofubar;キー「Fubar」を使用してChannel&Fooに参加するコマンド。

#foo、&bar fubarに参加してください。キー「Fubar」を使用してチャンネル#fooに参加するコマンドと、キーを使用しない&bar。

#foo、#bar fubar、foobar;キー「Fubar」を使用してチャンネル#fooに参加するコマンド、キー「Foobar」を使用してチャンネル#barを使用します。

参加#foo、#bar;チャンネル#fooと#barに参加するコマンド。

参加0;現在結合されているすべてのチャンネルを残します。

3.2.2 パーツメッセージ

部品コマンドにより、ユーザーはメッセージを送信し、パラメーター文字列にリストされているすべてのチャンネルのアクティブメンバーのリストから削除されます。「パーツメッセージ」が与えられた場合、これはデフォルトのメッセージ、ニックネームの代わりに送信されます。このリクエストは常にサーバーによって許可されます。

サーバーは、ターゲットのリストの形式で引数を解析する必要がありますが、クライアントにパーツメッセージを送信するときにリストを使用しないでください。

数値応答：

err_needmoreparams err_nosuchchannel err_notonchannel

例：

パート#twilight_zone;チャンネル「#twilight_zone」を離れるコマンド

パート#oz-ops、&group5;両方のチャンネルを残すコマンド"&group5"と「#oz-ops」。

：wiz!jto@tolsun.oulu.fiパート#playzone：私は紛失しました。ユーザーウィズは、「#playzone」を「#playzone」というメッセージ「I Lost」を残します。

3.2.3 チャンネルモードメッセージ

モードコマンドが提供され、ユーザーがチャンネルの特性をクエリして変更できるようにします。利用可能なモードとその用途の詳細については、「インターネットリレーチャット：チャン

Channel Management" [IRC-CHAN]. Note that there is a maximum limit of three (3) changes per command for modes that take a parameter.

Numeric Replies:

```
ERR_NEEDMOREPARAMS ERR_KEYSET
ERR_NOCHANMODES ERR_CHANOPRIVSNEEDED
ERR_USERNOTINCHANNEL ERR_UNKNOWNMODE
RPL_CHANNELMODEIS RPL_BANLIST RPL_ENDOFBANLIST
RPL_EXCEPTLIST RPL_ENDOFEXCEPTLIST RPL_INVITELIST
RPL_ENDOFINVITELIST RPL_UNIQOPIS
```

The following examples are given to help understanding the syntax of the MODE command, but refer to modes defined in "Internet Relay Chat: Channel Management" [IRC-CHAN].

Examples:

MODE #Finnish +iml *!*@*.fi ; Command to make #Finnish channel moderated and 'invite-only' with user with a hostname matching *.fi automatically invited.

MODE #Finnish +o Kilroy ; Command to give 'chanop' privileges to Kilroy on channel #Finnish.

MODE #Finnish +v Wiz ; Command to allow WiZ to speak on #Finnish.

MODE #Fins -s ; Command to remove 'secret' flag from channel #Fins.

MODE #42 +k oulu ; Command to set the channel key to "oulu".

MODE #42 -k oulu ; Command to remove the "oulu" channel key on channel "#42".

MODE #eu-ops +l 10 ; Command to set the limit for the number of users on channel "#eu-ops" to 10.

:WiZ!jto@tolsun.oulu.fi MODE #eu-ops -l ; User "WiZ" removing the limit for the number of users on channel "#eu-ops".

MODE &oulu +b ; Command to list ban masks set for the channel "&oulu".

MODE &oulu +b *!*@* ; Command to prevent all users from joining.

```
MODE &oulu +b *!*@*.edu +e *!*@*.bu.edu
                                ; Command to prevent any user from a
                                hostname matching *.edu from joining,
                                except if matching *.bu.edu
```

```
MODE #bu +be *!*@*.edu *!*@*.bu.edu
                                ; Comment to prevent any user from a
                                hostname matching *.edu from joining,
                                except if matching *.bu.edu
```

MODE #meditation e ; Command to list exception masks set for the channel "#meditation".

MODE #meditation I ; Command to list invitations masks set for the channel "#meditation".

MODE !12345ircd O ; Command to ask who the channel creator for "!12345ircd" is

ネル管理」 [IRC-chan]を参照してください。パラメーターを取得するモードのコマンドごとに3つの変更の最大制限があることに注意してください。

数値応答：

```
err_needmoreparams err_keyset err_nochanmodes
err_chanoprivsneeded err_usernotinchannel err_unmode
rpl_channelmodeis rpl_banlist rpl_endofbanlist rpl_exceptlist
rpl_endofexceptlist rpl_inviteList rpl_endofinviteLististrpis
```

モードコマンドの構文を理解するのに役立つが、「インターネットリレーチャット：チャンネル管理」 [IRC-chan]で定義されているモードを参照するために、次の例が示されています。

例：

モード#finnish imi* ! *!*@*.fi;#finnishチャンネルをモデレートし、ホスト名を一致させる *.fiを自動的に招待したユーザーと「招待のみ」にするコマンド。

モード#finnish o kilroy;チャンネル#finnishでキルロイに「チャノップ」 特権を与えるコマンド。

モード#finnish v wiz;ウィズが#finnishで話すことを許可するコマンド。

モード#fins -s;チャンネル#finsから「秘密」 フラグを削除するコマンド。

モード # 42 K oulu;チャンネルキーを「オル」 に設定するコマンド。

モード # 42 -K oulu;チャンネル「# 42」 の「オウル」 チャンネルキーを削除するコマンド。

モード # eu-ops l 10;チャンネル「# eu-ops」 のユーザー数の制限を10に設定するコマンド。

：wiz ! jto@tolsun.oulu.fiモード # eu -opers -l;ユーザー「wiz」は、チャンネル「eu-ops」 のユーザー数の制限を削除します。

モード&オウルB;チャンネル「&oulu」 に設定されたバンマスクをリストするコマンド。

モード&oulu b* ! *!*@*;すべてのユーザーが参加しないようにするコマンド。

モード#Meditation E;チャンネル「#Meditation」 に設定された例外マスクをリストするコマンド。

モード#meditation i;チャンネル「#Meditation」 に設定された招待状のマスクをリストするコマンド。

モード ! 12345ircd o;「 ! 12345ircd」 のチャンネル作成者が誰であるかを尋ねるコマンド

3.2.4 Topic message

Command: TOPIC
Parameters: <channel> [<topic>]

The TOPIC command is used to change or view the topic of a channel. The topic for channel <channel> is returned if there is no <topic> given. If the <topic> parameter is present, the topic for that channel will be changed, if this action is allowed for the user requesting it. If the <topic> parameter is an empty string, the topic for that channel will be removed.

Numeric Replies:

ERR_NEEDMOREPARAMS ERR_NOTONCHANNEL
RPL_NOTOPIC RPL_TOPIC ERR_CHANOPRIVSNEEDED
ERR_NOCHANMODES

Examples:

:Wiz!jto@tolsun.oulu.fi TOPIC #test :New topic ; User Wiz setting the topic.

TOPIC #test :another topic ; Command to set the topic on #test to "another topic".

TOPIC #test : ; Command to clear the topic on #test.

TOPIC #test ; Command to check the topic for #test.

3.2.5 Names message

Command: NAMES
Parameters: [<channel> *(" ," <channel>) [<target>]]

By using the NAMES command, a user can list all nicknames that are visible to him. For more details on what is visible and what is not, see "Internet Relay Chat: Channel Management" [IRC-CHAN]. The <channel> parameter specifies which channel(s) to return information about. There is no error reply for bad channel names.

If no <channel> parameter is given, a list of all channels and their occupants is returned. At the end of this list, a list of users who are visible but either not on any channel or not on a visible channel are listed as being on `channel' ""

If the <target> parameter is specified, the request is forwarded to that server which will generate the reply.

Wildcards are allowed in the <target> parameter.

Numerics:

ERR_TOOMANYMATCHES ERR_NOSUCHSERVER
RPL_NAMREPLY RPL_ENDOFNAMES

Examples:

NAMES #twilight_zone,#42 ; Command to list visible users on #twilight_zone and #42

NAMES ; Command to list all visible channels and users

3.2.6 List message

3.2.4 トピックメッセージ

トピックコマンドは、チャンネルのトピックを変更または表示するために使用されます。Channel <Channel>のトピックは、<トピック>が与えられていない場合に返されます。<topic>パラメーターが存在する場合、ユーザーがそれを要求するためにこのアクションが許可されている場合、そのチャンネルのトピックが変更されます。<topic>パラメーターが空の文字列である場合、そのチャンネルのトピックが削除されます。

数値応答：

err_needmoreparams err_notonchannel rpl_notopic rpl_topic
err_chanoprivsneeded err_nochanmodes

例：

：wiz！jto@tolsun.oulu.fiトピック#test：新しいトピック;ユーザーWIZがトピックを設定します。

トピック#test：別のトピック。#testでトピックを「別のトピック」に設定するコマンド。

トピック#test：;#Testでトピックをクリアするコマンド。

トピック#test;#testのトピックを確認するコマンド。

3.2.5 名前メッセージ

名前コマンドを使用することにより、ユーザーは自分に表示されるすべてのニックネームをリストできます。目に見えるものとそうでないものの詳細については、「インターネットリレーチャット：チャンネル管理」[IRC-chan]を参照してください。<Channel>パラメーターは、情報を返すチャンネルを指定します。悪いチャンネル名のエラー返信はありません。

<Channel>パラメーターが与えられていない場合、すべてのチャンネルとその居住者のリストが返されます。このリストの最後に、見えるが、どのチャンネルでもないかどうかのいずれかのユーザーのリストが「チャンネル」「*」にあるとリストされています。

<target>パラメーターが指定されている場合、リクエストはそのサーバーに転送され、応答が生成されます。

ワイルドカードは<target>パラメーターで許可されています。

数字：

err_toomanymatches err_nosuchserver rpl_namreply
rpl_endofnames

例：

名前;目に見えるすべてのチャンネルとユーザーをリストするコマンド

3.2.6 リストメッセージをリストします


```
Command: LIST
Parameters: [ <channel> *( "," <channel> ) [ <target> ] ]
```

The list command is used to list channels and their topics. If the <channel> parameter is used, only the status of that channel is displayed.

If the <target> parameter is specified, the request is forwarded to that server which will generate the reply.

Wildcards are allowed in the <target> parameter.

Numeric Replies:

```
ERR_TOOMANYMATCHES ERR_NOSUCHSERVER RPL_LIST
RPL_LISTEND
```

Examples:

LIST ; Command to list all channels.

```
LIST #twilight_zone,#42 ; Command to list channels
                        #twilight_zone and #42
```

3.2.7 Invite message

```
Command: INVITE
Parameters: <nickname> <channel>
```

The INVITE command is used to invite a user to a channel. The parameter <nickname> is the nickname of the person to be invited to the target channel <channel>. There is no requirement that the channel the target user is being invited to must exist or be a valid channel. However, if the channel exists, only members of the channel are allowed to invite other users. When the channel has invite-only flag set, only channel operators may issue INVITE command.

Only the user inviting and the user being invited will receive notification of the invitation. Other channel members are not notified. (This is unlike the MODE changes, and is occasionally the source of trouble for users.)

Numeric Replies:

```
ERR_NEEDMOREPARAMS ERR_NOSUCHNICK
ERR_NOTONCHANNEL ERR_USERONCHANNEL
ERR_CHANOPRIVSNEEDED RPL_INVITING RPL_AWAY
```

Examples:

```
:Angel!wings@irc.org INVITE Wiz #Dust
```

; Message to WiZ when he has been invited by user
Angel to channel #Dust

INVITE Wiz #Twilight_Zone ; Command to invite WiZ to
#Twilight_zone

3.2.8 Kick command

```
Command: KICK
Parameters: <channel> *( "," <channel> ) <user> *( "," <user> )
           [<comment>]
```

The KICK command can be used to request the forced removal of a user from a channel. It causes the <user> to PART from the

リストコマンドは、チャンネルとそのトピックをリストするために使用されます。<Channel>パラメーターが使用されている場合、そのチャンネルのステータスのみが表示されます。

<target>パラメーターが指定されている場合、リクエストはそのサーバーに転送され、応答が生成されます。

ワイルドカードは<target>パラメーターで許可されています。

数値応答：

```
err_toomanymatches err_nosuchserver rpl_list rpl_listend
```

例：

リスト;すべてのチャンネルをリストするコマンド。

3.2.7 メッセージを招待します

Inviteコマンドは、ユーザーをチャンネルに招待するために使用されます。パラメーター<Nickname>は、ターゲットチャンネル<Channel>に招待される人のニックネームです。ターゲットユーザーが招待されているチャンネルが存在するか、有効なチャンネルである必要があるという要件はありません。ただし、チャンネルが存在する場合、チャンネルのメンバーのみが他のユーザーを招待することができます。チャンネルに招待のみのフラグが設定されている場合、チャンネルオペレーターのみが招待コマンドを発行できます。

招待されているユーザーのみが招待されているユーザーのみが、招待状の通知を受け取ります。他のチャンネルメンバーには通知されません。（これはモードの変更とは異なり、場合によってはユーザーにとってトラブルの原因です。）

数値応答：

```
err_needmoreparams err_nosuchnick err_notonchannel
err_useronchannel err_chanoprivsneeded rpl_inviting
rpl_away
```

例：

;彼がユーザーエンジェルからチャンネル#dustに招待されたときのウィズへのメッセージ

wiz #twilight_zoneを招待します。#twilight_zoneにwizを招待するコマンド

3.2.8 キックコマンド

キックコマンドを使用して、チャンネルからユーザーの強制削除を要求できます。<user>は<channe>から力を払って譲渡しま

<channel> by force. For the message to be syntactically correct, there MUST be either one channel parameter and multiple user parameter, or as many channel parameters as there are user parameters. If a "comment" is given, this will be sent instead of the default message, the nickname of the user issuing the KICK.

The server MUST NOT send KICK messages with multiple channels or users to clients. This is necessarily to maintain backward compatibility with old client software.

Numeric Replies:

ERR_NEEDMOREPARAMS ERR_NOSUCHCHANNEL
ERR_BADCHANMASK ERR_CHANOPRIVSNEEDED
ERR_USERNOTINCHANNEL ERR_NOTONCHANNEL

Examples:

KICK &Melbourne Matthew ; Command to kick Matthew from &Melbourne

KICK #Finnish John :Speaking English ; Command to kick John from #Finnish using "Speaking English" as the reason (comment).

```
:WiZ!jto@tolsun.oulu.fi KICK #Finnish John
; KICK message on channel #Finnish
from WiZ to remove John from channel
```

3.3 Sending messages

The main purpose of the IRC protocol is to provide a base for clients to communicate with each other. PRIVMSG, NOTICE and SQUERY (described in Section 3.5 on Service Query and Commands) are the only messages available which actually perform delivery of a text message from one client to another - the rest just make it possible and try to ensure it happens in a reliable and structured manner.

3.3.1 Private messages

```
Command: PRIVMSG
Parameters: <msgtarget> <text to be sent>
```

PRIVMSG is used to send private messages between users, as well as to send messages to channels. <msgtarget> is usually the nickname of the recipient of the message, or a channel name.

The <msgtarget> parameter may also be a host mask (#<mask>) or server mask (\$<mask>). In both cases the server will only send the PRIVMSG to those who have a server or host matching the mask. The mask MUST have at least 1 (one) "." in it and no wildcards following the last ".". This requirement exists to prevent people sending messages to "#*" or "\$*", which would broadcast to all users. Wildcards are the '*' and '?' characters. This extension to the PRIVMSG command is only available to operators.

Numeric Replies:

ERR_NORECIPIENT ERR_NOTEXTTOSEND
ERR_CANNOTSENDTOCHAN ERR_NOTOPLEVEL
ERR_WILDTOPLEVEL ERR_TOOMANYTARGETS

す。メッセージが構文的に正しいためには、1つのチャンネルパラメーターと複数のユーザーパラメーター、またはユーザーパラメーターと同じくらい多くのチャンネルパラメーターが必要です。「コメント」が与えられた場合、これはデフォルトのメッセージの代わりに送信されます。これは、ユーザーがキックを発行するというニックネームです。

サーバーは、複数のチャンネルまたはユーザーを含むキックメッセージをクライアントに送信してはなりません。これは、古いクライアントソフトウェアとの逆方向の互換性を必然的に維持するためです。

数値応答：

err_needmoreparams err_nosuchchannel err_badchanmask
err_chanoprivsneeded err_usernotinchannel
err_notonchannel

例：

キック&メルボルンマシュー;メルボルンからマシューをキックするコマンド

キック#finnish John：英語を話す;「スピーキング英語」を理由（コメント）として使用して、#finnishからジョンをキックするコマンド。

3.3 メッセージの送信

IRCプロトコルの主な目的は、クライアントが相互に通信するためのベースを提供することです。privmsg、通知、およびスクエリー（サービスクエリとコマンドのセクション3.5で説明）は、実際にあるクライアントから別のクライアントにテキストメッセージの配信を実行する唯一のメッセージです。および構造化された方法。

3.3.1 プライベートメッセージ

privmsgは、ユーザー間でプライベートメッセージを送信し、チャンネルにメッセージを送信するために使用されます。<sgtarget>は通常、メッセージの受信者またはチャンネル名のニックネームです。

<sgtarget>パラメーターは、ホストマスク（#<mask>）またはサーバーマスク（\$<mask>）でもあります。どちらの場合も、サーバーはマスクに一致するサーバーまたはホストを持っている人にPrivmsgを送信します。マスクには少なくとも1（1）が必要です。その中で、最後の「。」に続くワイルドカードはありません。この要件は、すべてのユーザーにブロードキャストする「#*」または「\$*」にメッセージを送信するのを防ぐために存在します。ワイルドカードは「*」と「?」です文字。PRIVMSGコマンドへのこの拡張機能は、オペレーターのみが利用できます。

数値応答：

err_norecipient err_notexttosend err_cannotsendtochan
err_notoplevel err_wildtoplevel err_toomanytargets
err_nosuchnick rpl_away

ERR_NOSUCHNICK RPL_AWAY

Examples:

:Angel!wings@irc.org PRIVMSG Wiz :Are you receiving this message ? ; Message from Angel to Wiz.

PRIVMSG Angel :yes I'm receiving it ! ; Command to send a message to Angel.

PRIVMSG jto@tolsun.oulu.fi :Hello ! ; Command to send a message to a user on server tolsun.oulu.fi with username of "jto".

PRIVMSG kalt%millennium.stealth.net@irc.stealth.net :Are you a frog? ; Message to a user on server irc.stealth.net with username of "kalt", and connected from the host millennium.stealth.net.

PRIVMSG kalt%millennium.stealth.net :Do you like cheese? ; Message to a user on the local server with username of "kalt", and connected from the host millennium.stealth.net.

PRIVMSG Wiz!jto@tolsun.oulu.fi :Hello ! ; Message to the user with nickname Wiz who is connected from the host tolsun.oulu.fi and has the username "jto".

PRIVMSG \$*.fi :Server tolsun.oulu.fi rebooting. ; Message to everyone on a server which has a name matching *.fi.

PRIVMSG #*.edu :NSFNet is undergoing work, expect interruptions ; Message to all users who come from a host which has a name matching *.edu.

3.3.2 Notice

Command: NOTICE Parameters: <msgtarget> <text>

The NOTICE command is used similarly to PRIVMSG. The difference between NOTICE and PRIVMSG is that automatic replies MUST NEVER be sent in response to a NOTICE message. This rule applies to servers

too - they MUST NOT send any error reply back to the client on receipt of a notice. The object of this rule is to avoid loops between clients automatically sending something in response to something it received.

This command is available to services as well as users.

This is typically used by services, and automatons (clients with either an AI or other interactive program controlling their actions).

See PRIVMSG for more details on replies and examples.

3.4 Server queries and commands

The server query group of commands has been designed to return information about any server which is connected to the network.

In these queries, where a parameter appears as <target>, wildcard masks are usually valid. For each parameter, however, only one query and set of replies is to be generated. In most

例：

： Angel ! Wings@irc.org Privmsg Wiz： このメッセージを受け取っていますか？ ;天使からウィズへのメッセージ。

privmsgエンジェル： はい、私はそれを受け取ります！ ;エンジェルにメッセージを送信するコマンド。

privmsg jto@tolsun.oulu.fi： こんにちは！ ;「JTO」のユーザー名を使用して、サーバーTolsun.oulu.fiでユーザーにメッセージを送信するコマンド。

privmsg Kalt%Mylenium.stealth.net@irc.stealth.net： あなたはカエルですか？ ;「Kalt」のユーザー名を持つサーバーIRC.Stealth.netでユーザーへのメッセージ、およびホストMillennium.stealth.netから接続されています。

privmsg kalt%millennium.stealth.net： チーズは好きですか？ ;「Kalt」のユーザー名を持つローカルサーバー上のユーザーへのメッセージ、およびホストMillennium.stealth.netから接続されています。

privmsg wiz ! jto@tolsun.oulu.fi： こんにちは！ ;ホストTolsun.oulu.fiから接続されており、ユーザー名「JTO」を持っているニックネームWizでユーザーへのメッセージ。

privmsg \$*。 fi： サーバーtolson.oulu.fiの再起動。 ;*.fiの名前が付いているサーバー上のすべての人へのメッセージ。

privmsg # *。 edu： nsfnetは仕事を受けています。 中断を期待しています。 名前を一致させるホストから来たすべてのユーザーへのメッセージ *.edu。

3.3.2 知らせ

通知コマンドは、Privmsgと同様に使用されます。通知とprivmsgの違いは、通知メッセージに応じて自動応答を送信してはならないということです。このルールはサーバーに適用されます

また、通知を受け取ったときにクライアントにエラー返信を送信してはなりません。このルールの目的は、クライアント間のループを避け、受け取ったものに応じて何かを自動的に送信することです。

このコマンドは、ユーザーだけでなくサービスも利用できます。

これは通常、サービスとオートマトン（AIまたはアクションを制御する他のインタラクティブプログラムのいずれかを持つクライアント）で使用されます。

返信と例の詳細については、privmsgを参照してください。

3.4 サーバークエリとコマンド

コマンドのサーバークエリグループは、ネットワークに接続されているサーバーに関する情報を返すように設計されています。

これらのクエリでは、パラメーターが<target>として表示される場合、ワイルドカードマスクは通常有効です。ただし、各パラメーターについては、1つのクエリと返信セットのみが生成さ

cases, if a nickname is given, it will mean the server to which the user is connected.

These messages typically have little value for services, it is therefore RECOMMENDED to forbid services from using them.

3.4.1 Motd message

Command: MOTD Parameters: [<target>]

The MOTD command is used to get the "Message Of The Day" of the given server, or current server if <target> is omitted.

Wildcards are allowed in the <target> parameter.

Numeric Replies: RPL_MOTDSTART RPL_MOTD
RPL_ENDOFMOTD ERR_NOMOTD

3.4.2 Lusers message

Command: LUSERS Parameters: [<mask> [<target>]]
--

The LUSERS command is used to get statistics about the size of the IRC network. If no parameter is given, the reply will be about the whole net. If a <mask> is specified, then the reply will only

concern the part of the network formed by the servers matching the mask. Finally, if the <target> parameter is specified, the request is forwarded to that server which will generate the reply.

Wildcards are allowed in the <target> parameter.

Numeric Replies:

RPL_USERCLIENT RPL_USEROP RPL_USERUNKOWN
RPL_USERCHANNELS RPL_USERME
ERR_NOSUCHSERVER

3.4.3 Version message

Command: VERSION Parameters: [<target>]
--

The VERSION command is used to query the version of the server program. An optional parameter <target> is used to query the version of the server program which a client is not directly connected to.

Wildcards are allowed in the <target> parameter.

Numeric Replies:

ERR_NOSUCHSERVER RPL_VERSION

Examples:

VERSION tolsun.oulu.fi ; Command to check the version of server "tolsun.oulu.fi".

3.4.4 Stats message

Command: STATS Parameters: [<query> [<target>]]
--

The stats command is used to query statistics of certain server.

れます。ほとんどの場合、ニックネームが指定されている場合、ユーザーが接続されているサーバーを意味します。

これらのメッセージは通常、サービスにほとんど価値がないため、サービスを使用することを禁止することをお勧めします。

3.4.1 MOTDメッセージ

MOTDコマンドは、指定されたサーバーの「その日のメッセージ」を取得するために使用されます。

ワイルドカードは<target>パラメーターで許可されています。

数値応答：rpl_motdstart rpl_motd rpl_endofmotd err_nomotd

3.4.2 ルーザーメッセージ

Lusersコマンドは、IRCネットワークのサイズに関する統計を取得するために使用されます。パラメーターが与えられていない場合、返信はネット全体についてです。<mask>が指定されている場合、返信は

マスクに一致するサーバーによって形成されるネットワークの部分に関係します。最後に、<target>パラメーターが指定されている場合、リクエストはそのサーバーに転送され、応答が生成されます。

ワイルドカードは<target>パラメーターで許可されています。

数値応答：

rpl_userclient rpl_luserop rpl_luserunkown rpl_luserchannels
rpl_luserme err_nosuchserver

3.4.3 バージョンメッセージ

バージョンコマンドは、サーバープログラムのバージョンを照会するために使用されます。オプションのパラメーター<target>は、クライアントが直接接続されていないサーバープログラムのバージョンを照会するために使用されます。

ワイルドカードは<target>パラメーターで許可されています。

数値応答：

err_nosuchserver rpl_version

例：

バージョンtolsun.oulu.fi;server "tolsun.oulu.fi"のバージョンを確認するコマンド。

3.4.4 統計メッセージ

Statsコマンドは、特定のサーバーの統計を照会するために使用

If <query> parameter is omitted, only the end of stats reply is sent back.

A query may be given for any single letter which is only checked by the destination server and is otherwise passed on by intermediate servers, ignored and unaltered.

Wildcards are allowed in the <target> parameter.

Except for the ones below, the list of valid queries is implementation dependent. The standard queries below SHOULD be supported by the server:

l	- returns a list of the server's connections, showing how long each connection has been established and the traffic over that connection in Kbytes and messages for each direction;
m	- returns the usage count for each of commands supported by the server; commands for which the usage count is zero MAY be omitted;
o	- returns a list of configured privileged users, operators;
u	- returns a string showing how long the server has been up.

It is also RECOMMENDED that client and server access configuration be published this way.

Numeric Replies:

ERR_NOSUCHSERVER RPL_STATSLINKINFO
RPL_STATSUPTIME RPL_STATSCOMMANDS
RPL_STATSOLINE RPL_ENDOFSTATS

Examples:

STATS m ; Command to check the command usage for the server you are connected to

3.4.5 Links message

Command: LINKS Parameters: [[<remote server>] <server mask>]

With LINKS, a user can list all servernames, which are known by the server answering the query. The returned list of servers MUST match the mask, or if no mask is given, the full list is returned.

If <remote server> is given in addition to <server mask>, the LINKS command is forwarded to the first server found that matches that name (if any), and that server is then required to answer the query.

Numeric Replies:

ERR_NOSUCHSERVER RPL_LINKS RPL_ENDOFLINKS

Examples:

LINKS *.au ; Command to list all servers which have a name that matches *.au;

LINKS *.edu *.bu.edu ; Command to list servers matching *.bu.edu as seen by the first server matching *.edu.

3.4.6 Time message

Command: TIME Parameters: [<target>]

されます。<Query>パラメーターが省略されている場合、統計の終わりのみが返信されます。

宛先サーバーによってのみチェックされ、それ以外の場合は中間サーバーによって渡され、無視され、変更されていない単一の文字にクエリが与えられる場合があります。

ワイルドカードは<target>パラメーターで許可されています。

以下のものを除き、有効なクエリのリストは実装依存です。以下の標準クエリは、サーバーによってサポートされる必要があります。

また、クライアントとサーバーアクセスの構成をこのように公開することをお勧めします。

数値応答：

err_nosuchserver rpl_statslinkinfo rpl_statsuptime
rpl_statscommands rpl_statsoline rpl_endofstats

例：

統計M;接続されているサーバーのコマンド使用法を確認するコマンド

3.4.5 リンクメッセージ

リンクを使用すると、ユーザーはすべてのservernameをリストできます。これらは、サーバーがクエリに答えることで既知のものです。サーバーの返されたリストはマスクと一致する必要があります。または、マスクが与えられていない場合は、完全なリストが返されます。

<リモートサーバー>が<server mask>に加えて指定されている場合、リンクコマンドは最初のサーバーに転送されます。

数値応答：

err_nosuchserver rpl_links rpl_endoflinks

例：

links *.edu *.bu.edu;最初のサーバーを一致する *.bu.eduマッチング *.bu.eduを一致させるセルバーをリストするコマンド *.edu。

3.4.6 時間メッセージ

The time command is used to query local time from the specified server. If the <target> parameter is not given, the server receiving the command must reply to the query.

Wildcards are allowed in the <target> parameter.

Numeric Replies:

ERR_NOSUCHSERVER RPL_TIME

Examples: TIME tolsun.oulu.fi ; check the time on the server "tolson.oulu.fi"

3.4.7 Connect message

Command: CONNECT
Parameters: <target server> <port> [<remote server>]

The CONNECT command can be used to request a server to try to establish a new connection to another server immediately. CONNECT is a privileged command and SHOULD be available only to IRC Operators. If a <remote server> is given and its mask doesn't match name of the parsing server, the CONNECT attempt is sent to the first match of remote server. Otherwise the CONNECT attempt is made by the server processing the request.

The server receiving a remote CONNECT command SHOULD generate a WALLOPS message describing the source and target of the request.

Numeric Replies:

ERR_NOSUCHSERVER ERR_NOPRIVILEGES
ERR_NEEDMOREPARAMS

Examples:

CONNECT tolsun.oulu.fi 6667 ; Command to attempt to connect local server to tolsun.oulu.fi on port 6667

3.4.8 Trace message

Command: TRACE
Parameters: [<target>]

TRACE command is used to find the route to specific server and information about its peers. Each server that processes this command MUST report to the sender about it. The replies from pass-through links form a chain, which shows route to destination. After sending this reply back, the query MUST be sent to the next server until given <target> server is reached.

TRACE command is used to find the route to specific server. Each server that processes this message MUST tell the sender about it by sending a reply indicating it is a pass-through link, forming a chain of replies. After sending this reply back, it MUST then send the TRACE message to the next server until given server is reached. If the <target> parameter is omitted, it is RECOMMENDED that TRACE command sends a message to the sender telling which servers the local server has direct connection to.

If the destination given by <target> is an actual server, the

Timeコマンドは、指定されたサーバーから現地時間を照会するために使用されます。<target>パラメーターが指定されていない場合、コマンドを受信するサーバーはクエリに返信する必要があります。

ワイルドカードは<target>パラメーターで許可されています。

数値応答：

err_nosuchserver rpl_time

例：Time Tolsun.oulu.fi;サーバーの時間を確認します "tolson.oulu.fi"

3.4.7 メッセージを接続します

Connectコマンドを使用して、サーバーを要求して、すぐに別のサーバーへの新しい接続を確立しようとすることができます。Connectは特権コマンドであり、IRCオペレーターのみが利用できる必要があります。<リモートサーバー>が指定されていて、そのマスクが解析サーバーの名前と一致しない場合、接続試行はリモートサーバーの最初の一致に送信されます。それ以外の場合、接続の試行は、リクエストを処理するサーバーによって行われます。

リモート接続コマンドを受信するサーバーは、リクエストのソースとターゲットを説明するWallopsメッセージを生成する必要があります。

数値応答：

err_nosuchserver err_noprivileges err_needmoreparams

例：

Connect tolsun.oulu.fi 6667;ポート6667でローカルサーバーを tolsun.oulu.fiに接続しようとするコマンド

3.4.8 メッセージをトレースします

Traceコマンドは、特定のサーバーへのルートとそのピアに関する情報を見つけるために使用されます。このコマンドを処理する各サーバーは、それについて送信者に報告する必要があります。パススルーリンクからの返信は、宛先へのルートを示すチェーンを形成します。この返信を送信した後、<Target>サーバーに到達するまでクエリを次のサーバーに送信する必要があります。

TRACEコマンドは、特定のサーバーへのルートを見つけるために使用されます。このメッセージを処理する各サーバーは、それがパススルーリンクであることを示す返信を送信することにより、送信者にそれについて伝える必要があります。この返信を送信した後、指定されたサーバーに到達するまで次のサーバーにトレースメッセージを送信する必要があります。<Target>パラメーターが省略されている場合、TRACEコマンドは、ローカルサーバーが直接接続するサーバーを伝えるメッセージを送信者に送信することをお勧めします。

<target>で与えられた宛先が実際のサーバーである場合、宛先

destination server is REQUIRED to report all servers, services and operators which are connected to it; if the command was issued by an operator, the server MAY also report all users which are connected to it. If the destination given by <target> is a nickname, then only a reply for that nickname is given. If the <target> parameter is omitted, it is RECOMMENDED that the TRACE command is parsed as targeted to the processing server.

Wildcards are allowed in the <target> parameter.

Numeric Replies:

ERR_NOSUCHSERVER

If the TRACE message is destined for another server, all intermediate servers must return a RPL_TRACELINK reply to indicate that the TRACE passed through it and where it is going next.

RPL_TRACELINK

A TRACE reply may be composed of any number of the following numeric replies.

RPL_TRACECONNECTING RPL_TRACEHANDSHAKE
RPL_TRACEUNKNOWN RPL_TRACEOPERATOR
RPL_TRACEUSER RPL_TRACESERVER RPL_TRACESERVICE
RPL_TRACENEWTYPE RPL_TRACECLASS RPL_TRACELOG
RPL_TRACEEND

Examples:

TRACE *.oulu.fi ; TRACE to a server matching *.oulu.fi

3.4.9 Admin command

Command: ADMIN Parameters: [<target>]
--

The admin command is used to find information about the administrator of the given server, or current server if <target> parameter is omitted. Each server MUST have the ability to forward ADMIN messages to other servers.

Wildcards are allowed in the <target> parameter.

Numeric Replies:

ERR_NOSUCHSERVER RPL_ADMINME RPL_ADMINLOC1
RPL_ADMINLOC2 RPL_ADMINEMAIL

Examples:

ADMIN tolsun.oulu.fi ; request an ADMIN reply from tolsun.oulu.fi

ADMIN syrk ; ADMIN request for the server to which the user syrk is connected

3.4.10 Info command

Command: INFO Parameters: [<target>]

The INFO command is REQUIRED to return information describing the server: its version, when it was compiled, the patchlevel, when it was started, and any other miscellaneous information which may be considered to be relevant.

サーバーは、それに接続されているすべてのサーバー、サービス、およびオペレーターを報告する必要があります。コマンドがオペレーターによって発行された場合、サーバーはそれに接続されているすべてのユーザーを報告することもできます。<target>で与えられた宛先がニックネームである場合、そのニックネームに対する返信のみが与えられます。<Target>パラメーターが省略されている場合、TRACEコマンドを処理サーバーのターゲットとして解析することをお勧めします。

ワイルドカードは<target>パラメーターで許可されています。

数値応答：

err_nosuchserver

トレースメッセージが別のサーバー用に運命づけられている場合、すべての中間サーバーがRPL_TRACELINKの返信を返す必要があります。

RPL_TRACELINK

トレース応答は、次の数値応答の任意の数で構成されている場合があります。

RPL_TRACECONNECTING RPL_TRACEHANDSHAKE
RPL_TRACEUNKNOUNT RPL_TRACEOPERATOR
RPL_TRACEUSER RPL_TRACESERVER RPL_TRACESERVICE
RPL_TRACENEWTYPE RPL_TRACECLASS RPL_TRACELOG
RPL_TRACEENDEND

例：

trace *.oulu.fi;*.oulu.fiの一致するサーバーにトレースします

3.4.9 管理者コマンド

adminコマンドは、指定されたサーバーの管理者に関する情報を見つけるために使用されます。<ターゲット>パラメーターが省略されている場合は、現在のサーバーです。各サーバーには、他のサーバーに管理メッセージを転送する機能が必要です。

ワイルドカードは<target>パラメーターで許可されています。

数値応答：

err_nosuchserver rpl_adminme rpl_adminloc1 rpl_adminloc2
rpl_adminemail

例：

admin tolsun.oulu.fi;tolsun.oulu.fiから管理者の返信をリクエストします

管理者シルク;ユーザーシルクが接続されているサーバーの管理者要求

3.4.10 情報コマンド

情報コマンドは、サーバーを説明する情報を返す必要があります。そのバージョンは、コンパイルされたとき、PatchLevel、開始時、および関連すると見なされる他のその他の情報です。

Wildcards are allowed in the <target> parameter.

Numeric Replies:

ERR_NOSUCHSERVER RPL_INFO RPL_ENDOFINFO

Examples:

INFO csd.bu.edu ; request an INFO reply from csd.bu.edu

INFO Angel ; request info from the server that Angel is connected to.

3.5 Service Query and Commands

The service query group of commands has been designed to return information about any service which is connected to the network.

3.5.1 Servlist message

Command: SERVLIST Parameters: [<mask> [<type>]]
--

The SERVLIST command is used to list services currently connected to the network and visible to the user issuing the command. The optional parameters may be used to restrict the result of the query (to matching services names, and services type).

Numeric Replies:

RPL_SERVLIST RPL_SERVLISTEND

3.5.2 Squery

Command: SQUERY Parameters: <servicename> <text>

The SQUERY command is used similarly to PRIVMSG. The only difference is that the recipient MUST be a service. This is the only way for a text message to be delivered to a service.

See PRIVMSG for more details on replies and example.

Examples:

SQUERY irchelp :HELP privmsg ; Message to the service with nickname irchelp.

SQUERY dict@irc.fr :fr2en blaireau ; Message to the service with name dict@irc.fr.

3.6 User based queries

User queries are a group of commands which are primarily concerned with finding details on a particular user or group users. When using wildcards with any of these commands, if they match, they will only return information on users who are 'visible' to you. The visibility of a user is determined as a combination of the user's mode and the common set of channels you are both on.

Although services SHOULD NOT be using this class of message, they are allowed to.

3.6.1 Who query

Command: WHO

ワイルドカードは<target>パラメーターで許可されています。

数値応答：

err_nosuchserver rpl_info rpl_endofinfo

例：

情報csd.bu.edu;csd.bu.eduから情報の返信をリクエストしてください

情報エンジェル;Angelが接続されているサーバーから情報をリクエストします。

3.5 サービスクエリとコマンド

コマンドのサービスクエリグループは、ネットワークに接続されているサービスに関する情報を返すように設計されています。

3.5.1 サーブリストメッセージ

Servlistコマンドは、現在ネットワークに接続されているサービスをリストし、コマンドを発行するユーザーに表示されるサービスをリストするために使用されます。オプションのパラメーターは、クエリの結果を制限するために使用できます（サービス名とサービスの種類を一致させる）。

数値応答：

rpl_servlist rpl_servlistend

3.5.2 スクエリー

Squeryコマンドは、Privmsgと同様に使用されます。唯一の違いは、受信者がサービスでなければならないことです。これは、テキストメッセージがサービスに配信される唯一の方法です。

返信と例の詳細については、privmsgを参照してください。

例：

Squery Irchelp：Privmsgを助けます。ニックネームIrchelpでサービスへのメッセージ。

Squery dict@irc.fr：Fr2en Blaireau;名前dict@irc.frでサービスへのメッセージ。

3.6 ユーザーベースのクエリ

ユーザークエリは、主に特定のユーザーまたはグループユーザーの詳細を見つけることに関係するコマンドのグループです。これらのコマンドのいずれかでワイルドカードを使用する場合、それらが一致する場合、彼らはあなたに「見える」ユーザーに関する情報のみを返します。ユーザーの可視性は、ユーザーモードと、両方ともオンになっているチャンネルの共通セットの組み合わせとして決定されます。

サービスはこのクラスのメッセージを使用するべきではありませんが、許可されています。

3.6.1 誰が質問しますか

Parameters: [<mask> ["o"]]

The WHO command is used by a client to generate a query which returns a list of information which 'matches' the <mask> parameter given by the client. In the absence of the <mask> parameter, all visible (users who aren't invisible (user mode +i) and who don't have a common channel with the requesting client) are listed. The same result can be achieved by using a <mask> of "0" or any wildcard which will end up matching every visible user.

The <mask> passed to WHO is matched against users' host, server, real name and nickname if the channel <mask> cannot be found.

If the "o" parameter is passed only operators are returned according to the <mask> supplied.

Numeric Replies:

ERR_NOSUCHSERVER RPL_WHOREPLY RPL_ENDOFWHO

Examples:

WHO *.fi ; Command to list all users who match against "*.fi".

WHO jto* o ; Command to list all users with a match against "jto*" if they are an operator.

3.6.2 Whois query

Command: WHOIS
Parameters: [<target>] <mask> *("," <mask>)

This command is used to query information about particular user. The server will answer this command with several numeric messages indicating different statuses of each user which matches the mask (if you are entitled to see them). If no wildcard is present in the <mask>, any information about that nick which you are allowed to see is presented.

If the <target> parameter is specified, it sends the query to a specific server. It is useful if you want to know how long the user in question has been idle as only local server (i.e., the server the user is directly connected to) knows that information, while everything else is globally known.

Wildcards are allowed in the <target> parameter.

Numeric Replies:

ERR_NOSUCHSERVER ERR_NONICKNAMEGIVEN
RPL_WHOISUSER RPL_WHOISCHANNELS
RPL_WHOISCHANNELS RPL_WHOISSERVER RPL_AWAY
RPL_WHOISOPERATOR RPL_WHOISIDLE ERR_NOSUCHNICK
RPL_ENDOFWHOIS

Examples:

WHOIS wiz ; return available user information about nick WiZ

WHOIS eff.org trillian ; ask server eff.org for user information about trillian

3.6.3 Whowas

Command: WHOWAS

WHOコマンドは、クライアントが提供する<mask>パラメーターに「一致」する情報のリストを返すクエリを生成するためにクライアントによって使用されます。<mask>パラメーターが存在しない場合、すべて表示されます（目に見えないユーザー（ユーザーモードI）、リクエストクライアントと共通のチャンネルを持っていないユーザー）がリストされています。同じ結果は、「0」の<マスク>または目に見えるすべてのユーザーと一致するワイルドカードを使用することで達成できます。

チャンネル<mask>が見つからない場合、ユーザーのホスト、サーバー、本名、ニックネームと一致する人に渡されました。

「o」パラメーターが渡される場合、提供された<mask>に従ってオペレーターのみが返されます。

数値応答：

err_nosuchserver rpl_whoreply rpl_endofwho

例：

WHO *.fi; 「*.fi」と一致するすべてのユーザーをリストするコマンド。

誰がjto* o;すべてのユーザーがオペレーターである場合、すべてのユーザーを「JTO*」との試合でリストするコマンド。

3.6.2 WHOISクエリ

このコマンドは、特定のユーザーに関する情報を照会するために使用されます。サーバーは、マスクに一致する各ユーザーの異なるステータスを示すいくつかの数値メッセージでこのコマンドに回答します（それらを見る権利がある場合）。<mask>にワイルドカードが存在しない場合、あなたが見ることができるニックに関する情報が提示されます。

<target>パラメーターが指定されている場合、クエリを特定のサーバーに送信します。ローカルサーバーのみ（つまり、ユーザーが直接接続されているサーバー）のみがその情報を知っていますが、他のすべてがグローバルに知られているため、問題のユーザーがどれだけの時間をアイドル状態にしているかを知りたい場合に役立ちます。

ワイルドカードは<target>パラメーターで許可されています。

数値応答：

err_nosuchserver err_nonicknamegiven rpl_whoisuser
rpl_whoischannels rpl_whoischannels rpl_whoisserver
rpl_away rpl_whoisoperator rpl_whoisidle err_nosuchnick
rpl_endofwhois

例：

whois wiz;Nick Wizに関する利用可能なユーザー情報を返します

whois eff.orgトリリアン;Trillianのユーザー情報については、サーバーEff.orgに尋ねてください

3.6.3 誰だった

Parameters: <nickname> *("," <nickname>) [<count> [<target>]]

Whowas asks for information about a nickname which no longer exists. This may either be due to a nickname change or the user leaving IRC. In response to this query, the server searches through its nickname history, looking for any nicks which are lexically the same (no wild card matching here). The history is searched backward, returning the most recent entry first. If there are multiple entries, up to <count> replies will be returned (or all of them if no <count> parameter is given). If a non-positive number is passed as being <count>, then a full search is done.

Wildcards are allowed in the <target> parameter.

Numeric Replies:

ERR_NONICKNAMEGIVEN ERR_WASNOSUCHNICK
RPL_WHOWASUSER RPL_WHOISSERVER
RPL_ENDOFWHOWAS

Examples:

WHOWAS Wiz ; return all information in the nick history about nick "WiZ";

WHOWAS Mermaid 9 ; return at most, the 9 most recent entries in the nick history for "Mermaid";

WHOWAS Trillian 1 *.edu ; return the most recent history for "Trillian" from the first server found to match "*.edu".

3.7 Miscellaneous messages

Messages in this category do not fit into any of the above categories but are nonetheless still a part of and REQUIRED by the protocol.

3.7.1 Kill message

Command: KILL
Parameters: <nickname> <comment>

The KILL command is used to cause a client-server connection to be closed by the server which has the actual connection. Servers generate KILL messages on nickname collisions. It MAY also be available available to users who have the operator status.

Clients which have automatic reconnect algorithms effectively make this command useless since the disconnection is only brief. It does however break the flow of data and can be used to stop large amounts of 'flooding' from abusive users or accidents. Abusive users usually don't care as they will reconnect promptly and resume their abusive behaviour. To prevent this command from being abused, any user may elect to receive KILL messages generated for others to keep an 'eye' on would be trouble spots.

In an arena where nicknames are REQUIRED to be globally unique at all times, KILL messages are sent whenever 'duplicates' are detected (that is an attempt to register two users with the same nickname) in the hope that both of them will disappear and only 1 reappear.

When a client is removed as the result of a KILL message, the server SHOULD add the nickname to the list of unavailable

Whowasは、もはや存在しないニックネームに関する情報を求めます。これは、ニックネームの変更またはユーザーがIRCを離れることによるものかもしれません。このクエリに応じて、サーバーはニックネームの履歴を検索し、字句的に同じニックを探しています（ここではワイルドカードが一致していません）。履歴は後方に検索され、最新のエントリを最初に返します。複数のエントリがある場合、最大<count>の応答が返されます（または、<count>パラメーターが与えられている場合はすべて）。非肯定的な数値が<count>として渡される場合、完全な検索が行われます。

ワイルドカードは<target>パラメーターで許可されています。

数値応答：

err_nonicknamegiven err_wasnosuchnick rpl_whowasuser
rpl_whoisserver rpl_endofwhowas

例：

whowas wiz;ニックの歴史の中でニック「ウィズ」についてのすべての情報を返します。

Whowas Mermaid 9;せいぜい、「人魚」のニック史における最新の9つのエントリを返します。

whowas trillian 1 *.edu;「Trillian」の最新の履歴を、「.edu」に一致する最初のサーバーから返品します。

3.7 その他のメッセージ

このカテゴリのメッセージは、上記のカテゴリのいずれにも適合しませんが、それでもプロトコルの一部と要求されています。

3.7.1 メッセージを殺す

Killコマンドは、実際の接続を持つサーバーによってクライアントサーバー接続を閉じるために使用されます。サーバーは、ニックネームの衝突でキルメッセージを生成します。また、オペレーターのステータスを持っているユーザーが利用できる場合があります。

自動再接続アルゴリズムを持っているクライアントは、切断が短いため、このコマンドを効果的に役に立たなくします。ただし、データの流れを破り、虐待的なユーザーや事故から大量の「洪水」を止めるために使用できます。虐待的なユーザーは通常、迅速に再接続し、虐待的な行動を再開するため、気にしません。このコマンドが乱用されるのを防ぐために、ユーザーは他の人が「目」を維持するために生成されたキルメッセージを受信することを選択することができます。

ニックネームが常にグローバルにユニークである必要があるアリーナでは、「複製」が検出されるたびにキルメッセージが送信されます（同じニックネームで2人のユーザーを登録する試みです）。1が再び現れます。

クライアントがキルメッセージの結果として削除されると、サーバーはクライアントがこの名前をすぐに再利用することを避

nicknames in an attempt to avoid clients to reuse this name immediately which is usually the pattern of abusive behaviour often leading to useless "KILL loops". See the "IRC Server Protocol" document [IRC-SERVER] for more information on this procedure.

The comment given MUST reflect the actual reason for the KILL. For server-generated KILLS it usually is made up of details concerning the origins of the two conflicting nicknames. For users it is left up to them to provide an adequate reason to satisfy others who see it. To prevent/discourage fake KILLS from being generated to hide the identify of the KILLer, the comment also shows a 'kill-path' which is updated by each server it passes through, each prepending its name to the path.

Numeric Replies:

```
ERR_NOPRIVILEGES ERR_NEEDMOREPARAMS
ERR_NOSUCHNICK ERR_CANTKILLSERVER
```

NOTE: It is RECOMMENDED that only Operators be allowed to kill other users with KILL command. This command has been the subject of many controversies over the years, and along with the above recommendation, it is also widely recognized that not even operators should be allowed to kill users on remote servers.

3.7.2 Ping message

Command: PING Parameters: <server1> [<server2>]
--

The PING command is used to test the presence of an active client or server at the other end of the connection. Servers send a PING message at regular intervals if no other activity detected coming from a connection. If a connection fails to respond to a PING message within a set amount of time, that connection is closed. A PING message MAY be sent even if the connection is active.

When a PING message is received, the appropriate PONG message MUST be sent as reply to <server1> (server which sent the PING message out) as soon as possible. If the <server2> parameter is specified, it represents the target of the ping, and the message gets forwarded there.

Numeric Replies:

```
ERR_NOORIGIN ERR_NOSUCHSERVER
```

Examples:

PING tolsun.oulu.fi ; Command to send a PING message to server

PING WiZ tolsun.oulu.fi ; Command from WiZ to send a PING message to server "tolsun.oulu.fi"

PING :irc.funet.fi ; Ping message sent by server "irc.funet.fi"

3.7.3 Pong message

Command: PONG Parameters: <server> [<server2>]

PONG message is a reply to ping message. If parameter <server2> is given, this message MUST be forwarded to given

けようとするために、利用できないニックネームのリストにニックネームを追加する必要があります。ループ」。この手順の詳細については、「IRC Server Protocol」ドキュメント[IRC-Server]を参照してください。

与えられたコメントは、殺害の実際の理由を反映している必要があります。サーバーで生成されたキルの場合、通常、2つの競合するニックネームの起源に関する詳細で構成されています。ユーザーにとって、それを見る他の人を満足させる適切な理由を提供することは彼らに任されています。殺人者の識別を隠すために偽の殺害が生成されるのを防ぐ/阻止するために、コメントは、それぞれが通過する各サーバーによって更新される「キルパス」を示し、それぞれがその名前をパスに準備します。

数値応答：

```
err_noprivileges err_needmoreparams err_nosuchnick
err_cantkillserver
```

注：キルコマンドで他のユーザーを殺すことを許可されることをお勧めします。このコマンドは長年にわたって多くの論争の対象となっており、上記の推奨事項とともに、オペレーターでさえリモートサーバーでユーザーを殺すことを許可されてはならないことも広く認識されています。

3.7.2 pingメッセージ

Pingコマンドは、接続の反対側にアクティブなクライアントまたはサーバーの存在をテストするために使用されます。サーバーは、接続から来る他のアクティビティが検出されない場合は、定期的にpingメッセージを送信します。接続が設定された時間内にpingメッセージに応答できない場合、その接続は閉じられます。接続がアクティブであっても、pingメッセージが送信される場合があります。

pingメッセージが受信された場合、適切なポンメッセージは、できるだけ早く<server1>（pingメッセージを送信したサーバー）への返信として送信する必要があります。<server2>パラメーターが指定されている場合、Pingのターゲットを表し、メッセージがそこに転送されます。

数値応答：

```
err_noorigin err_nosuchserver
```

例：

ping tolsun.oulu.fi;pingメッセージをサーバーに送信するコマンド

ping wiz tolsun.oulu.fi;wizから、pingメッセージをサーバーに送信するコマンド「tolsun.oulu.fi」

ping：irc.funet.fi;サーバー「irc.funet.fi」から送信されたpingメッセージ

3.7.3 ポンメッセージ

PongメッセージはPingメッセージへの返信です。パラメーター<server2>が指定されている場合、このメッセージは指定された

target. The <server> parameter is the name of the entity who has responded to PING message and generated this message.

Numeric Replies:

ERR_NOORIGIN ERR_NOSUCHSERVER

Example:

PONG csd.bu.edu tolsun.oulu.fi ; PONG message from csd.bu.edu to tolsun.oulu.fi

3.7.4 Error

Command: ERROR Parameters: <error message>

The ERROR command is for use by servers when reporting a serious or fatal error to its peers. It may also be sent from one server to another but MUST NOT be accepted from any normal unknown clients.

Only an ERROR message SHOULD be used for reporting errors which occur with a server-to-server link. An ERROR message is sent to the server at the other end (which reports it to appropriate local users and logs) and to appropriate local users and logs. It is not to be passed onto any other servers by a server if it is received from a server.

The ERROR message is also used before terminating a client connection.

When a server sends a received ERROR message to its operators, the message SHOULD be encapsulated inside a NOTICE message, indicating that the client was not responsible for the error.

Numerics:

None.

Examples:

ERROR :Server *.fi already exists ; ERROR message to the other server which caused this error.

NOTICE WiZ :ERROR from csd.bu.edu -- Server *.fi already exists ; Same ERROR message as above but sent to user WiZ on the other server.

4. Optional features

This section describes OPTIONAL messages. They are not required in a working server implementation of the protocol described herein. In the absence of the feature, an error reply message MUST be generated or an unknown command error. If the message is destined for another server to answer then it MUST be passed on (elementary parsing REQUIRED) The allocated numerics for this are listed with the messages below.

From this section, only the USERHOST and ISON messages are available to services.

4.1 Away

Command: AWAY Parameters: [<text>]

ターゲットに転送する必要があります。<server>パラメーターは、pingメッセージに応答し、このメッセージを生成したエンティティの名前です。

数値応答：

err_noorigin err_nosuchserver

例：

pong csd.bu.edu tolsun.oulu.fi;csd.bu.eduからtolson.oulu.fiへのpongメッセージ

3.7.4 エラー

エラーコマンドは、ピアに重大または致命的なエラーを報告する際にサーバーが使用するためです。また、あるサーバーから別のサーバーに送信することもできますが、通常の不明なクライアントから受け入れてはなりません。

サーバーからサーバーへのリンクで発生するエラーの報告には、エラーメッセージのみを使用する必要があります。エラーメッセージが反対側のサーバーに送信され（適切なローカルユーザーとログ）、および適切なローカルユーザーとログに送信されます。サーバーから受信された場合、サーバーによって他のサーバーに渡されないようにします。

エラーメッセージは、クライアント接続を終了する前にも使用されます。

サーバーが受信したエラーメッセージをオペレーターに送信する場合、メッセージは通知メッセージ内にカプセル化され、クライアントがエラーについて責任を負わなかったことを示します。

数字：

なし。

例：

エラー：server *.fiは既に存在します。このエラーを引き起こした他のサーバーへのエラーメッセージ。

wizに注意：csd.bu.eduからのエラー - サーバー *.fi既に存在します。上記と同じエラーメッセージですが、他のサーバーでユーザーWizに送信されます。

4. オプションの機能

このセクションでは、オプションのメッセージについて説明します。本明細書に記載されているプロトコルの動作サーバー実装では必要ありません。機能がない場合、エラー返信メッセージを生成するか、不明なコマンドエラーを生成する必要があります。メッセージが別のサーバーに回答するように運命づけられている場合、それは渡される必要があります（基本解析が必要）。

このセクションからは、ユーザーホストとアイソンメッセージのみがサービスで利用できます。

4.1 あちらへ

With the AWAY command, clients can set an automatic reply string for any PRIVMSG commands directed at them (not to a channel they are on). The server sends an automatic reply to the client sending the PRIVMSG command. The only replying server is the one to which the sending client is connected to.

The AWAY command is used either with one parameter, to set an AWAY message, or with no parameters, to remove the AWAY message.

Because of its high cost (memory and bandwidth wise), the AWAY message SHOULD only be used for client-server communication. A server MAY choose to silently ignore AWAY messages received from other servers. To update the away status of a client across servers, the user mode 'a' SHOULD be used instead. (See Section 3.1.5)

Numeric Replies:

RPL_UNAWAY RPL_NOWAWAY

Example:

AWAY :Gone to lunch. Back in 5 ; Command to set away message to "Gone to lunch. Back in 5".

4.2 Rehash message

Command: REHASH Parameters: None

The rehash command is an administrative command which can be used by an operator to force the server to re-read and process its configuration file.

Numeric Replies:

RPL_REHASHING ERR_NOPRIVILEGES

Example:

REHASH ; message from user with operator status to server asking it to reread its configuration file.

4.3 Die message

Command: DIE Parameters: None

An operator can use the DIE command to shutdown the server. This message is optional since it may be viewed as a risk to allow arbitrary people to connect to a server as an operator and execute this command.

The DIE command MUST always be fully processed by the server to which the sending client is connected and MUST NOT be passed onto other connected servers.

Numeric Replies:

ERR_NOPRIVILEGES

Example:

DIE ; no parameters required.

4.4 Restart message

Command: RESTART Parameters: None

An operator can use the restart command to force the server to

アウェイコマンドを使用すると、クライアントは、それらに向けられたprivmsgコマンドの自動応答文字列を設定できます（それらが使用しているチャンネルではありません）。サーバーは、privmsgコマンドを送信するクライアントに自動返信を送信します。唯一の返信サーバーは、送信クライアントが接続されているサーバーです。

アウェイコマンドは、1つのパラメーターで使用され、アウェイメッセージを設定するか、パラメーターを使用して、アウェイメッセージを削除します。

高コスト（メモリと帯域幅が賢明）のため、アウェイメッセージはクライアントサーバー通信にのみ使用する必要があります。サーバーは、他のサーバーから受信したメッセージを静かに無視することを選択できます。サーバー全体でクライアントのアウェイステータスを更新するには、代わりにユーザーモード「A」を使用する必要があります。（セクション3.1.5を参照）

数値応答：

rpl_unaway rpl_nowaway

例：

離れて：昼食に行った。5に戻ります。「昼食に行った。バックイン5」にメッセージを設定するコマンド。

4.2 メッセージを再ハッシュします

コマンド：パラメーターの再ハッシュ：なし

Rehashコマンドは管理コマンドであり、オペレーターが使用してサーバーに再読み込みとその構成ファイルの処理を強制することができます。

数値応答：

rpl_rehashing err_noprivileges

例：

再ハッシュ;オペレーターステータスを持つユーザーからのメッセージサーバーへのメッセージ構成ファイルを再確認するように要求します。

4.3 メッセージをダイ

コマンド：ダイパラメーター：なし

オペレーターは、DIEコマンドを使用してサーバーをシャットダウンできます。このメッセージは、任意の人々がオペレーターとしてサーバーに接続し、このコマンドを実行できるようにするリスクと見なされる可能性があるため、オプションです。

DIEコマンドは、送信クライアントが接続されているサーバーによって常に完全に処理される必要があり、他の接続されたサーバーに渡されてはなりません。

数値応答：

err_noprivileges

例：

死ぬ;パラメーターは不要です。

4.4 メッセージを再起動します

コマンド：再起動パラメーター：なし

オペレーターは、再起動コマンドを使用して、サーバーに再起

restart itself. This message is optional since it may be viewed as a risk to allow arbitrary people to connect to a server as an operator and execute this command, causing (at least) a disruption to service.

The RESTART command MUST always be fully processed by the server to which the sending client is connected and MUST NOT be passed onto other connected servers.

Numeric Replies:

ERR_NOPRIVILEGES

Example:

RESTART ; no parameters required.

4.5 Summon message

Command: SUMMON
Parameters: <user> [<target> [<channel>]]

The SUMMON command can be used to give users who are on a host running an IRC server a message asking them to please join IRC. This message is only sent if the target server (a) has SUMMON enabled, (b) the user is logged in and (c) the server process can write to the user's tty (or similar).

If no <server> parameter is given it tries to summon <user> from the server the client is connected to is assumed as the target.

If summon is not enabled in a server, it MUST return the ERR_SUMMONDISABLED numeric.

Numeric Replies:

ERR_NORECIPIENT ERR_FILEERROR ERR_NOLOGIN
ERR_NOSUCHSERVER ERR_SUMMONDISABLED
RPL_SUMMONING

Examples:

SUMMON jto ; summon user jto on the server's host

SUMMON jto tolsun.oulu.fi ; summon user jto on the host which a server named "tolsun.oulu.fi" is running.

4.6 Users

Command: USERS
Parameters: [<target>]

The USERS command returns a list of users logged into the server in a format similar to the UNIX commands who(1), rusers(1) and finger(1). If disabled, the correct numeric MUST be returned to indicate this.

Because of the security implications of such a command, it SHOULD be disabled by default in server implementations. Enabling it SHOULD require recompiling the server or some equivalent change rather than simply toggling an option and restarting the server. The procedure to enable this command SHOULD also include suitable large comments.

Numeric Replies:

ERR_NOSUCHSERVER ERR_FILEERROR RPL_USERSSTART

動を強制することができます。このメッセージは、任意の人々がオペレーターとしてサーバーに接続し、このコマンドを実行できるようにするリスクと見なされる可能性があるため、オプションです。

再起動コマンドは、送信クライアントが接続されているサーバーによって常に完全に処理される必要があります、他の接続されたサーバーに渡されてはなりません。

数値応答：

err_noprivileges

例：

再起動 ;パラメーターは不要です。

4.5 メッセージを召喚します

召喚コマンドは、IRCサーバーを実行しているホスト上のユーザーに、IRCに参加するように依頼するメッセージを提供するために使用できます。このメッセージは、ターゲットサーバーが召喚され、（b）ユーザーがログインし、（c）サーバープロセスがユーザーのTTY（または同様）に書き込むことができる場合にのみ送信されます。

NO <server>パラメーターが与えられた場合、サーバーから<ユーザー>を召喚しようとしします。クライアントが接続されていることがターゲットとして想定されます。

召喚がサーバーで有効になっていない場合、err_summondisabled数値を返す必要があります。

数値応答：

err_norecipient err_fileerror err_nologin err_nosuchserver
err_summondisabled rpl_summoning

例：

summon jto;サーバーのホストでユーザーJTOを召喚します

summon jto tolsun.oulu.fi; 「tolsun.oulu.fi」 という名前のサーバーが実行されているホストでユーザーjtoを召喚します。

4.6 ユーザー

ユーザーコマンドは、（1）、Ruser（1）、およびFinger（1）と同様の形式でサーバーにログインしたユーザーのリストを返します。無効にした場合、これを示すために正しい数値を返す必要があります。

このようなコマンドのセキュリティへの影響により、サーバーの実装ではデフォルトで無効にする必要があります。有効にすることで、オプションを切り替えてサーバーを再起動するのではなく、サーバーまたは同等の変更を再コンパイルする必要があります。このコマンドを有効にする手順には、適切な大きなコメントも含める必要があります。

数値応答：

err_nosuchserver err_fileerror rpl_usersstart rpl_users

RPL_USERS RPL_NOUSERS RPL_ENDOFUSERS
ERR_USERSDISABLED

Disabled Reply:

ERR_USERSDISABLED

Example:

USERS eff.org ; request a list of users logged in on server eff.org

4.7 Operwall message

Command: WALLOPS
Parameters: <Text to be sent>

The WALLOPS command is used to send a message to all currently connected users who have set the 'w' user mode for themselves. (See Section 3.1.5 "User modes").

After implementing WALLOPS as a user command it was found that it was often and commonly abused as a means of sending a message to a lot of people. Due to this, it is RECOMMENDED that the implementation of WALLOPS allows and recognizes only servers as the originators of WALLOPS.

Numeric Replies:

ERR_NEEDMOREPARAMS

Example:

:csd.bu.edu WALLOPS :Connect '*.uiuc.edu 6667' from Joshua ; WALLOPS message from csd.bu.edu announcing a CONNECT message it received from Joshua and acted upon.

4.8 Userhost message

Command: USERHOST
Parameters: <nickname> *(SPACE <nickname>)

The USERHOST command takes a list of up to 5 nicknames, each separated by a space character and returns a list of information about each nickname that it found. The returned list has each reply separated by a space.

Numeric Replies:

RPL_USERHOST ERR_NEEDMOREPARAMS

Example:

USERHOST Wiz Michael syrk ; USERHOST request for information on nicks "Wiz", "Michael", and "syrk"

:ircd.stealth.net 302 yournick :syrk=+syrk@millennium.stealth.net ; Reply for user syrk

4.9 Ison message

Command: ISON
Parameters: <nickname> *(SPACE <nickname>)

The ISON command was implemented to provide a quick and efficient means to get a response about whether a given nickname was currently on IRC. ISON only takes one (1) type of parameter: a space-separated list of nicks. For each nickname in the list that is present, the

rpl_nousers rpl_endofusers err_usersdisabled

無効な返信：

err_usersdisabled

例：

ユーザーeff.org;サーバーeff.orgにログインしたユーザーのリストをリクエストしてください

4.7 オペワールメッセージ

Wallopsコマンドは、現在接続されているすべてのユーザーにメッセージを送信するために使用されます。（セクション3.1.5「ユーザーモード」を参照）。

Wallopsをユーザーコマンドとして実装した後、多くの人にメッセージを送信する手段として頻繁に虐待されていることがわかりました。このため、Wallopsの実装により、サーバーのみがWallopsの創始者としてのみ許可および認識されることが推奨されます。

数値応答：

err_needmoreparams

例：

:csd.bu.edu wallops :connect '*.uiuc.edu 6667' from joshua;csd.bu.eduからのWallopsメッセージは、Joshuaから受け取って行動した接続メッセージを発表しました。

4.8 userhostメッセージ

userhostコマンドは、最大5つのニックネームのリストを取得し、それぞれがスペース文字で区切られ、見つけた各ニックネームに関する情報のリストを返します。返されたリストには、各返信がスペースによって区切られています。

数値応答：

rpl_userhost err_needmoreparams

例：

userhost wizマイケルシルク。nicks "wiz"、 "Michael"、 および「syrk」に関する情報のユーザーホストリクエスト

4.9 ISONメッセージ

ISONコマンドは、特定のニックネームが現在IRCにあるかどうかについての応答を取得するための迅速かつ効率的な手段を提供するために実装されました。ISONは、パラメーターの1つのタイプのみを取得します。ニックのスペース分離リストです。存在するリストの各ニックネームについて、

server adds that to its reply string. Thus the reply string may return empty (none of the given nicks are present), an exact copy of the parameter string (all of them present) or any other subset of the set of nicks given in the parameter. The only limit on the number of nicks that may be checked is that the combined length MUST NOT be too large as to cause the server to chop it off so it fits in 512 characters.

ISON is only processed by the server local to the client sending the command and thus not passed onto other servers for further processing.

Numeric Replies:

RPL_ISON ERR_NEEDMOREPARAMS

Example:

ISON phone trillian WiZ jarlek Avalon Angel Monstah syrk ;
Sample ISON request for 7 nicks.

5. Replies

The following is a list of numeric replies which are generated in response to the commands given above. Each numeric is given with its number, name and reply string.

5.1 Command responses

Numerics in the range from 001 to 099 are used for client-server connections only and should never travel between servers. Replies generated in the response to commands are found in the range from 200 to 399.

001	RPL_WELCOME "Welcome to the Internet Relay Network <nick>!<user>@<host>"
002	RPL_YOURHOST "Your host is <servername>, running version <ver>"
003	RPL_CREATED "This server was created <date>"
004	RPL_MYINFO "<servername> <version> <available user modes> <available channel modes>"

- The server sends Replies 001 to 004 to a user upon successful registration.

005	RPL_BOUNCE "Try server <server name>, port <port number>"
-----	--

- Sent by the server to a user to suggest an alternative server. This is often used when the connection is refused because the server is already full.

302	RPL_USERHOST ":*1<reply> *(" " <reply>)"
-----	---

- Reply format used by USERHOST to list replies to the query list. The reply string is composed as follows:

reply = nickname ["*"] "=" ("+" / "-") hostname

The '*' indicates whether the client has registered as an Operator. The '-' or '+' characters represent whether the client has set an AWAY message or not respectively.

303	RPL_ISON
-----	----------

サーバーはそれを返信文字列に追加します。したがって、応答文字列は空に戻る場合があります（指定されたニックは存在しません）、パラメーター文字列の正確なコピー（すべてが存在する）、またはパラメーターに与えられたニックのセットの他のサブセット。チェックされるニックの数の唯一の制限は、512文字に収まるようにサーバーを切り替えるために、組み合わせた長さが大きすぎてはならないことです。

ISONは、クライアントがコマンドを送信するクライアントにローカルサーバーによってのみ処理されるため、さらに処理するために他のサーバーに渡されません。

数値応答：

rpl_ison err_needmoreparams

例：

ISON PHONE TRILLIAN WIZ JARLEK AVALON ANGEL
MONSTAH SYRK;7つのニックのアイソンリクエストをサンプリングします。

5. 返信

以下は、上記のコマンドに応じて生成される数値応答のリストです。各数値には、その数、名前、および返信文字列が付属されています。

5.1 コマンド応答

001から099までの範囲の数字は、クライアントサーバー接続のみに使用され、サーバー間を移動しないでください。コマンドへの応答で生成された返信は、200～399の範囲で見つかります。

- サーバーは、登録が成功したときに、001をユーザーに001から004に送信します。

- サーバーからユーザーに送信されて、代替サーバーを提案します。これは、サーバーがすでにいっぱいであるため、接続が拒否された場合によく使用されます。

- userhostが使用する返信形式でクエリリストへの返信をリストします。返信文字列は次のように構成されています。

「*」は、クライアントがオペレーターとして登録しているかどうかを示します。'- 'または"文字は、クライアントがそれぞれアウェイメッセージを設定しているかどうかを表します。

```
":*1<nick> *( " " <nick> )"

```

- Reply format used by ISON to list replies to the query list.

301 RPL_AWAY "<nick> :<away message>" 305 RPL_UNAWAY
":You are no longer marked as being away" 306
RPL_NOWAWAY ":You have been marked as being away"

- These replies are used with the AWAY command (if allowed).
RPL_AWAY is sent to any client sending a PRIVMSG to a client
which is away. RPL_AWAY is only sent by the server to which
the client is connected. Replies RPL_UNAWAY and
RPL_NOWAWAY are sent when the client removes and sets an
AWAY message.

```

311      RPL_WHOSUSER
          "<nick> <user> <host> * :<real name>"
312      RPL_WHOSSERVER
          "<nick> <server> :<server info>"
313      RPL_WHOSOPERATOR
          "<nick> :is an IRC operator"

```

```

317      RPL_WHOSIDLE
          "<nick> <integer> :seconds idle"
318      RPL_ENDOFWHOIS
          "<nick> :End of WHOIS list"
319      RPL_WHOSCHANNELS
          "<nick> :*( ( "@" / "+" ) <channel> " " )"

```

- Replies 311 - 313, 317 - 319 are all replies generated in
response to a WHOIS message. Given that there are enough
parameters present, the answering server MUST either
formulate a reply out of the above numerics (if the query nick
is found) or return an error reply. The '*' in RPL_WHOSUSER
is there as the literal character and not as a wild card. For
each reply set, only RPL_WHOSCHANNELS may appear more
than once (for long lists of channel names). The '@' and '+'
characters next to the channel name indicate whether a client
is a channel operator or has been granted permission to speak
on a moderated channel. The RPL_ENDOFWHOIS reply is
used to mark the end of processing a WHOIS message.

```

314      RPL_WHOWASUSER
          "<nick> <user> <host> * :<real name>"
369      RPL_ENDOFHOWAS
          "<nick> :End of HOWAS"

```

- When replying to a HOWAS message, a server MUST use
the replies RPL_WHOWASUSER, RPL_WHOSSERVER or
ERR_WASNOSUCHNICK for each nickname in the presented
list. At the end of all reply batches, there MUST be
RPL_ENDOFHOWAS (even if there was only one reply and it
was an error).

321 RPL_LISTSTART Obsolete. Not used.

```

322      RPL_LIST
          "<channel> <# visible> :<topic>"
323      RPL_LISTEND
          ":End of LIST"

```

- Replies RPL_LIST, RPL_LISTEND mark the actual replies with
data and end of the server's response to a LIST command. If
there are no channels available to return, only the end reply
MUST be sent.

```

325      RPL_UNIQOPIS

```

- ISONが使用する返信形式でクエリリストへの返信をリストし
ます。

301 rpl_away "<nick> :<away message>" 305 rpl_unaway " :
あなたはもう離れているとマークされていません" 306
rpl_nowaway " :あなたは離れているとマークされています"

- これらの返信は、アウェイコマンド（許可されている場合）
で使用されます。rpl_awayは、クライアントに任意のクライア
ントに送信されます。RPL_AWAYは、クライアントが接続され
ているサーバーによってのみ送信されます。返信rpl_unawayと
rpl_nowawayは、クライアントが削除してアウェイメッセージ
を設定すると送信されます。

- 返信311-313、317-319は、WHOISメッセージに応じて生成
されたすべての返信です。十分なパラメーターが存在するこ
とを考えると、回答サーバーは、上記の数字（クエリニックが見
つかった場合）からの返信を策定するか、エラー返信を返す必
要があります。rpl_whoisuserの「*」は、ワイルドカードとし
てではなく、文字通りのキャラクターとしてあります。各返信
セットについて、RPL_Whoschannelのみが複数回表示される
場合があります（チャンネル名の長いリストの場合）。チャン
ネル名の隣の「@」と「+」文字は、クライアントがチャン
ネルオペレーターであるか、緩和されたチャンネルで話す許
可を与えられているかを示します。RPL_ENDOFWHOIS応答は、
WHOISメッセージの処理の終了をマークするために使用されま
す。

- howasメッセージに返信する場合、サーバーは、提示され
たリストの各ニックネームに対して、RPL_WHOWASUSER、
RPL_WHOSSERVER、またはERR_WASNOSUCHNICKを返信
する必要があります。すべての返信バッチの最後に、
RPL_ENDOFHOWASが必要です（返信が1つしかなく、エラ
ーでした）。

321 RPL_LISTSTART OBERETE。使用されていない。

- RPL_LIST、RPL_LISTENDマーク実際の返信はデータととも
に、リストコマンドに対するサーバーの応答の終了をマークし
ます。返されるチャンネルがない場合は、最終返信のみを送信
する必要があります。

```
"<channel> <nickname>"
```

```
324      RPL_CHANNELMODEIS
        "<channel> <mode> <mode params>"
```

```
331      RPL_NOTOPIC
        "<channel> :No topic is set"
332      RPL_TOPIC
        "<channel> :<topic>"
```

- When sending a TOPIC message to determine the channel topic, one of two replies is sent. If the topic is set, RPL_TOPIC is sent back else RPL_NOTOPIC.

- チャネルトピックを決定するトピックメッセージを送信すると、2つの返信の1つが送信されます。トピックが設定されている場合、rpl_topicはelse rpl_notopicで送信されます。

```
341      RPL_INVITING
        "<channel> <nick>"
```

- Returned by the server to indicate that the attempted INVITE message was successful and is being passed onto the end client.

- サーバーから返されて、試みられた招待メッセージが成功し、エンドクライアントに渡されていることを示します。

342 RPL_SUMMONING "<user> :Summoning user to IRC"

342 rpl_summoning "<user> : ユーザーをIRCに召喚する"

- Returned by a server answering a SUMMON message to indicate that it is summoning that user.

- 召喚メッセージに応答するサーバーによって返され、そのユーザーを召喚していることを示します。

```
346      RPL_INVITELIST
        "<channel> <invitemask>"
347      RPL_ENDOFINVITELIST
        "<channel> :End of channel invite list"
```

- When listing the 'invitations masks' for a given channel, a server is required to send the list back using the RPL_INVITELIST and RPL_ENDOFINVITELIST messages. A separate RPL_INVITELIST is sent for each active mask. After the masks have been listed (or if none present) a RPL_ENDOFINVITELIST MUST be sent.

- 特定のチャネルの「Invitations Masks」をリストするときには、RPL_INVITELISTおよびRPL_ENDOFINVITELISTメッセージを使用してリストを送り返すためのサーバーが必要です。アクティブマスクごとに別のRPL_INVITELISTが送信されます。マスクがリストされた後（または存在しない場合）RPL_ENDOFINVITELISTを送信する必要があります。

```
348      RPL_EXCEPTLIST
        "<channel> <exceptionmask>"
349      RPL_ENDOFEXCEPTLIST
        "<channel> :End of channel exception list"
```

- When listing the 'exception masks' for a given channel, a server is required to send the list back using the RPL_EXCEPTLIST and RPL_ENDOFEXCEPTLIST messages. A separate RPL_EXCEPTLIST is sent for each active mask. After the masks have been listed (or if none present) a RPL_ENDOFEXCEPTLIST MUST be sent.

- 特定のチャネルの「例外マスク」をリストする場合、サーバーはRPL_EXICTERLISTおよびRPL_ENDOFEXCECTERISTメッセージを使用してリストを返送する必要があります。アクティブマスクごとに別のRPL_EXICTERLISTが送信されます。マスクがリストされた後（または存在しない場合）RPL_ENDOFEXCECTELISTを送信する必要があります。

```
351      RPL_VERSION
        "<version>.<debuglevel> <server> :<comments>"
```

- Reply by the server showing its version details. The <version> is the version of the software being used (including any patchlevel revisions) and the <debuglevel> is used to indicate if the server is running in "debug mode".

- そのバージョンの詳細を表示するサーバーによる返信。<バージョン>は、使用されているソフトウェアのバージョン（PatchLevel Revisionsを含む）であり、<DebugLevel>は、サーバーが「デバッグモード」で実行されているかどうかを示すために使用されます。

The "comments" field may contain any comments about the version or further version details.

「コメント」フィールドには、バージョンまたはその他のバージョンの詳細に関するコメントが含まれる場合があります。

```
352      RPL_WHOREPLY
        "<channel> <user> <host> <server> <nick>
        ( "H" / "G" > [ "*" ] [ ( "@" / "+" ) ]
        :<hopcount> <real name>"
```


315 RPL_ENDOFWHO "<name> :End of WHO list"

- The RPL_WHOREPLY and RPL_ENDOFWHO pair are used to answer a WHO message. The RPL_WHOREPLY is only sent if there is an appropriate match to the WHO query. If there is a list of parameters supplied with a WHO message, a RPL_ENDOFWHO MUST be sent after processing each list item with <name> being the item.

```
353      RPL_NAMREPLY
      "( "=" / "*" / "@" ) <channel>
      :[ "@" / "+" ] <nick> *( " " [ "@" / "+" ] <nick> )
      - "@" is used for secret channels, "*" for private
        channels, and "=" for others (public channels).
```

366 RPL_ENDOFNAMES "<channel> :End of NAMES list"

- To reply to a NAMES message, a reply pair consisting of RPL_NAMREPLY and RPL_ENDOFNAMES is sent by the server back to the client. If there is no channel found as in the query, then only RPL_ENDOFNAMES is

returned. The exception to this is when a NAMES message is sent with no parameters and all visible channels and contents are sent back in a series of RPL_NAMEREPY messages with a RPL_ENDOFNAMES to mark the end.

```
364      RPL_LINKS
      "<mask> <server> :<hopcount> <server info>"
365      RPL_ENDOFLINKS
      "<mask> :End of LINKS list"
```

- In replying to the LINKS message, a server MUST send replies back using the RPL_LINKS numeric and mark the end of the list using an RPL_ENDOFLINKS reply.

```
367      RPL_BANLIST
      "<channel> <banmask>"
368      RPL_ENDOFBANLIST
      "<channel> :End of channel ban list"
```

- When listing the active 'bans' for a given channel, a server is required to send the list back using the RPL_BANLIST and RPL_ENDOFBANLIST messages. A separate RPL_BANLIST is sent for each active banmask. After the banmasks have been listed (or if none present) a RPL_ENDOFBANLIST MUST be sent.

371 RPL_INFO ":<string>" 374 RPL_ENDOFINFO ":End of INFO list"

- A server responding to an INFO message is required to send all its 'info' in a series of RPL_INFO messages with a RPL_ENDOFINFO reply to indicate the end of the replies.

```
375      RPL_MOTDSTART
      ":- <server> Message of the day - "
372      RPL_MOTD
      ":- <text>"
376      RPL_ENDOFMOTD
      ":End of MOTD command"
```

- When responding to the MOTD message and the MOTD file is found, the file is displayed line by line, with each line no longer than 80 characters, using

RPL_MOTD format replies. These MUST be surrounded by a RPL_MOTDSTART (before the RPL_MOTDs) and an RPL_ENDOFMOTD (after).

315 rpl_endofwho "<name> : who list of list"

- RPL_WHOREPLYとRPL_ENDOFWHOペアは、WHOメッセージに答えるために使用されます。RPL_WHOREPLYは、WHOクエリに適切な一致がある場合にのみ送信されます。WHOメッセージで提供されたパラメーターのリストがある場合、各リスト項目を処理した後に<Name>がアイテムであるRPL_ENDOFを送信する必要があります。

366 RPL_ENDOFNAMES "<Channel> : 名前の終了リスト" "

- 名前メッセージに返信するには、RPL_NAMREPLYとRPL_ENDOFNAMESで構成される返信ペアがサーバーからクライアントに送信されます。クエリのようにチャンネルが見つからない場合、rpl_endofnamesのみが

戻ってきた。この例外は、名前メッセージがパラメーターなしで送信され、すべての可視チャンネルとコンテンツが rpl_endofnamesを使用して一連のrpl_namereplyメッセージで送信されて終了をマークする場合です。

- リンクメッセージに返信する際に、サーバーはrpl_links数値を使用して返信を送信し、rpl_endoflinksの返信を使用してリストの最後をマークする必要があります。

- 特定のチャンネルのアクティブな「禁止」をリストする場合、サーバーはRPL_BANLISTおよびRPL_ENDOFBANLISTメッセージを使用してリストを返送する必要があります。アクティブなバンマスクごとに別のRPL_BANLISTが送信されます。バンマスクがリストされた後（または存在しない場合）RPL_ENDOFBANLISTを送信する必要があります。

371 rpl_info " : <string>" 374 rpl_endofinfo " : 情報リストの終わり" "

- 情報メッセージに応答するサーバーは、RPL_ENDOFINFOの応答を使用してすべての「情報」をRPL_INFOメッセージで送信するために、返信の終了を示すために必要です。

- MOTDメッセージとMOTDファイルが見つかった場合、ファイルは行ごとに表示され、各行は80文字以下で、使用して表示されます。

rpl_motd形式の返信。これらは、RPL_MOTDSTART（RPL_MOTDSの前）とRPL_ENDOFMOTD（後）に囲まれている必要があります。

381 RPL_YOUREOPER ":You are now an IRC operator"

- RPL_YOUREOPER is sent back to a client which has just successfully issued an OPER message and gained operator status.

382 RPL_REHASHING "<config file> :Rehashing"

- If the REHASH option is used and an operator sends a REHASH message, an RPL_REHASHING is sent back to the operator.

383 RPL_YOURESERVICE "You are service <servicename>"

- Sent by the server to a service upon successful registration.

381 rpl_youreoper "：あなたは今IRCオペレーターです"

- rpl_youreoperはクライアントに送り返され、オペレーションメッセージを正常に発行し、オペレータースtatusを獲得しました。

382 rpl_rehashing "<config file>：rehashing"

- Rehashオプションが使用され、オペレーターがRehashメッセージを送信すると、RPL_Rehashingがオペレーターに送信されます。

383 rpl_youreservice "you ares service <ServiceName>"

- 登録が成功したときにサーバーからサービスに送信されま
す。

391	RPL_TIME	"<server> :<string showing server's local time>"
-----	----------	--

- When replying to the TIME message, a server MUST send the reply using the RPL_TIME format above. The string showing the time need only contain the correct day and time there. There is no further requirement for the time string.

- The Timeメッセージに返信する場合、サーバーは上記のRPL_TIME形式を使用して返信を送信する必要があります。時間を示す文字列には、そこに正しい日と時間しか含まれていません。時間文字列にはそれ以上の要件はありません。

392	RPL_USERSSTART	":UserID Terminal Host"
393	RPL_USERS	":<username> <ttyline> <hostname>"
394	RPL_ENDOFUSERS	":End of users"
395	RPL_NOUSERS	":Nobody logged in"

- If the USERS message is handled by a server, the replies RPL_USERSTART, RPL_USERS, RPL_ENDOFUSERS and RPL_NOUSERS are used. RPL_USERSSTART MUST be sent first, following by either a sequence of RPL_USERS or a single RPL_NOUSER. Following this is RPL_ENDOFUSERS.

- ユーザーメッセージがサーバーによって処理されると、RPL_USERSTART、RPL_USERS、RPL_ENDOFUSERS、RPL_NOUSERSが使用されます。RPL_USERSSTARTは、一連のrpl_usersまたは単一のrpl_nouserのいずれかに続いて、最初に送信する必要があります。これに続くのはRPL_ENDOFUSERSです。

200	RPL_TRACELINK	"Link <version & debug level> <destination> <next server> V<protocol version> <link uptime in seconds> <backstream sendq> <upstream sendq>"
201	RPL_TRACECONNECTING	"Try. <class> <server>"
202	RPL_TRACEHANDSHAKE	"H.S. <class> <server>"
203	RPL_TRACEUNKNOWN	"???? <class> [<client IP address in dot form>]"
204	RPL_TRACEOPERATOR	"Oper <class> <nick>"
205	RPL_TRACEUSER	"User <class> <nick>"
206	RPL_TRACESERVER	"Serv <class> <int>S <int>C <server> <nick!user!*!*>@<host server> V<protocol version>"
207	RPL_TRACESERVICE	"Service <class> <name> <type> <active type>"
208	RPL_TRACENEWTYPE	"<newtype> 0 <client name>"
209	RPL_TRACECLASS	"Class <class> <count>"
210	RPL_TRACERECONNECT	Unused.
261	RPL_TRACELOG	"File <logfile> <debug level>"
262	RPL_TRACEEND	"<server name> <version & debug level> :End of TRACE"

- The RPL_TRACE* are all returned by the server in response

- RPL_TRACE*はすべて、トレースメッセージに応じてサーバ

to the TRACE message. How many are returned is dependent on the TRACE message and whether it was sent by an operator or not. There is no predefined order for which occurs first. Replies RPL_TRACEUNKNOWN, RPL_TRACECONNECTING and RPL_TRACEHANDSHAKE are all used for connections which have not been fully established and are either unknown, still attempting to connect or in the process of completing the 'server handshake'. RPL_TRACELINK is sent by any server which handles a TRACE message and has to pass it on to another server. The list of RPL_TRACELINKs sent in response to a TRACE command traversing the IRC network should reflect the actual connectivity of the servers themselves along that path.

RPL_TRACENEWTYPE is to be used for any connection which does not fit in the other categories but is being displayed anyway. RPL_TRACEEND is sent to indicate the end of the list.

211	RPL_STATSLINKINFO "<linkname> <sendq> <sent messages> <sent Kbytes> <received messages> <received Kbytes> <time open>"
-----	---

- reports statistics on a connection. <linkname> identifies the particular connection, <sendq> is the amount of data that is queued and waiting to be sent <sent messages> the number of messages sent, and <sent Kbytes> the amount of data sent, in Kbytes. <received messages> and <received Kbytes> are the equivalent of <sent messages> and <sent Kbytes> for received data, respectively. <time open> indicates how long ago the connection was opened, in seconds.

212	RPL_STATSCOMMANDS "<command> <count> <byte count> <remote count>"
-----	--

- reports statistics on commands usage.

219 RPL_ENDOFSTATS "<stats letter> :End of STATS report"

242 RPL_STATSUPTIME ":Server Up %d days
%d:%02d:%02d"

- reports the server uptime.

243	RPL_STATSOLINE "0 <hostmask> * <name>"
-----	---

- reports the allowed hosts from where user may become IRC operators.

221 RPL_UMODEIS "<user mode string>"

- To answer a query about a client's own mode, RPL_UMODEIS is sent back.

234	RPL_SERVLIST "<name> <server> <mask> <type> <hopcount> <info>"
-----	---

235	RPL_SERVLISTEND "<mask> <type> :End of service listing"
-----	--

- When listing services in reply to a SERVLIST message, a server is required to send the list back using the RPL_SERVLIST and RPL_SERVLISTEND messages. A separate

ーによって返されます。返される数は、トレースメッセージとオペレーターによって送信されたかどうか依存しています。最初に発生する事前定義された順序はありません。RPL_TRACEUNKNOUNT、RPL_TRACECONNECTINGおよびRPL_TRACEHANDSHAKEはすべて、完全に確立されておらず、まだ不明であり、まだ接続しようとしているか、「サーバーハンドシェイク」を完了している接続に使用されます。RPL_TRACELINKは、トレースメッセージを処理するサーバーから送信され、別のサーバーに渡す必要があります。IRCネットワークを通過するTRACEコマンドに応じて送信されたRPL_TRACELINKSのリストは、そのパスに沿ったサーバー自体の実際の接続性を反映する必要があります。

RPL_TRACENEWTYPEは、他のカテゴリに適合せず、とにかく表示されている任意の接続に使用されます。RPL_TRACEENDは、リストの終了を示すために送信されます。

- 接続に関する統計を報告します。<linkname>特定の接続を識別し、<sendq>は、送信されたメッセージの数が送信され、送信されたメッセージの数、<送信されたKBYTES>送信されたデータの量をKBYTESで識別します。<受信メッセージ>および<受信kbytes>は、受信したデータに対してそれぞれ<送信メッセージ>および<送信kbytes>に相当します。<タイムオープン>数秒で接続が開かれた時間を示します。

- コマンドの使用に関する統計を報告します。

219 RPL_ENDOFSTATS "<stats letter> : stats ext of Statsレポート"

242 rpl_statsuptime "：サーバーアップ%d日%d：%02d：%02d"

- サーバーの稼働時間を報告します。

- ユーザーがIRCオペレーターになる可能性のあるホストを報告します。

221 rpl_umodeis "<ユーザーモード文字列>"

- クライアント自身のモードに関するクエリに答えるために、rpl_umodeisが返送されます。

- サーブリストメッセージに返信してサービスをリストする場合、サーバーはrpl_servlistとrpl_servlistendメッセージを使用してリストを返送する必要があります。各サービスごとに別の

RPL_SERVLIST is sent for each service. After the services have been listed (or if none present) a RPL_SERVLISTEND MUST be sent.

251	RPL_LUSERCLIENT	" :There are <integer> users and <integer> services on <integer> servers"
252	RPL_LUSEROP	"<integer> :operator(s) online"
253	RPL_LUSERUNKNOWN	"<integer> :unknown connection(s)"
254	RPL_LUSERCHANNELS	"<integer> :channels formed"
255	RPL_LUSERME	" :I have <integer> clients and <integer> servers"

- In processing an LUSERS message, the server sends a set of replies from RPL_LUSERCLIENT, RPL_LUSEROP, RPL_USERUNKNOWN, RPL_LUSERCHANNELS and RPL_LUSERME. When replying, a server MUST send back RPL_LUSERCLIENT and RPL_LUSERME. The other replies are only sent back if a non-zero count is found for them.

256	RPL_ADMINME	"<server> :Administrative info"
257	RPL_ADMINLOC1	" :<admin info>"
258	RPL_ADMINLOC2	" :<admin info>"
259	RPL_ADMINEMAIL	" :<admin info>"

- When replying to an ADMIN message, a server is expected to use replies RPL_ADMINME through to RPL_ADMINEMAIL and provide a text message with each. For RPL_ADMINLOC1 a description of what city, state and country the server is in is expected, followed by details of the institution (RPL_ADMINLOC2)

and finally the administrative contact for the server (an email address here is REQUIRED) in RPL_ADMINEMAIL.

263 RPL_TRYAGAIN "<command> :Please wait a while and try again."

- When a server drops a command without processing it, it MUST use the reply RPL_TRYAGAIN to inform the originating client.

5.2 Error Replies

Error replies are found in the range from 400 to 599.

401	ERR_NOSUCHNICK	"<nickname> :No such nick/channel"
-----	----------------	------------------------------------

- Used to indicate the nickname parameter supplied to a command is currently unused.

402 ERR_NOSUCHSERVER "<server name> :No such server"

- Used to indicate the server name given currently does not exist.

403 ERR_NOSUCHCHANNEL "<channel name> :No such channel"

- Used to indicate the given channel name is invalid.

RPL_Servlistが送信されます。サービスがリストされた後（または存在しない場合）RPL_ServListendを送信する必要があります。

- ルーザーメッセージの処理において、サーバーは rpl_userclient、rpl_userop、rpl_userunknown、rpl_userchannels、rpl_usermeから一連の返信を送信します。返信するときは、サーバーはrpl_userclientとrpl_usermeを送り返す必要があります。他の返信は、ゼロ以外のカウントが見つかった場合にのみ送信されます。

- 管理者メッセージに返信すると、サーバーはRPL_ADMINMEをRPL_ADMINEMAILに使用し、それぞれにテキストメッセージを提供することが期待されます。rpl_adminloc1の場合、サーバーがどの都市、州、国にあるかについての説明が予想され、その後に機関の詳細が続きます（rpl_adminloc2）

そして最後に、RPL_ADMINEMAILのサーバーの管理連絡先（こちらのメールアドレスが必要です）。

263 rpl_tryagain "<コマンド>：しばらく待って、もう一度やり直してください。」

- サーバーが処理せずにコマンドをドロップする場合、REPLY RPL_TRYAGAINを使用して発信元のクライアントに通知する必要があります。

5.2 エラーが返信します

エラー応答は、400～599の範囲で見つかります。

- コマンドに提供されたニックネームパラメーターを示すために使用されています。現在使用されていません。

402 err_nosuchserver "<サーバー名>：そのようなサーバーはありません」

- 現在指定されているサーバー名が存在しないことを示すために使用されます。

403 err_nosuchchannel "<チャンネル名>：そのようなチャンネルはありません"

- 指定されたチャネル名が無効であることを示すために使用されます。

404 ERR_CANNOTSENDTOCHAN "<channel name> :Cannot send to channel"

404 err_cannotsendtochan "<チャンネル名>：チャンネルに送信できません"

- Sent to a user who is either (a) not on a channel which is mode +n or (b) not a chanop (or mode +v) on a channel which has mode +m set or where the user is banned and is trying to send a PRIVMSG message to that channel.

- (a) モードnまたは (b) モードmセットを備えたチャネル上のチャノップ（またはモードV）ではないチャネルではないユーザーに送信されます。そのチャネルへのprivmsgメッセージ。

405 ERR_TOOMANYCHANNELS "<channel name> :You have joined too many channels"

405 err_toomanychannels "<チャンネル名>：あまりにも多くのチャネルに参加しました"

- Sent to a user when they have joined the maximum number of allowed channels and they try to join another channel.

- 許可されたチャネルの最大数に参加し、別のチャネルに参加しようとする場合、ユーザーに送信されます。

406 ERR_WASNOSUCHNICK "<nickname> :There was no such nickname"

406 err_wasnosuchnick "<ニックネーム>：そのようなニックネームはありませんでした"

- Returned by WHOWAS to indicate there is no history information for that nickname.

- そのニックネームには履歴情報がないことを示すためにWhowasによって返されます。

407	ERR_TOOMANYTARGETS "<target> :<error code> recipients. <abort message>"
-----	--

- Returned to a client which is attempting to send a PRIVMSG/NOTICE using the user@host destination format and for a user@host which has several occurrences.

- ユーザー@ホスト宛先形式を使用してprivmsg/通知を送信しようとしているクライアントに戻り、いくつかの発生があるユーザー@ホストの場合。
- Returned to a client which trying to send a PRIVMSG/NOTICE to too many recipients.

- あまりにも多くの受信者にprivmsg/通知を送信しようとするクライアントに戻りました。
- Returned to a client which is attempting to JOIN a safe channel using the shortname when there are more than one such channel.

- そのようなチャネルが複数ある場合、ShortNameを使用してセーフチャネルに参加しようとしているクライアントに戻りました。

408 ERR_NOSUCHSERVICE "<service name> :No such service"

408 err_nosuchservice "<サービス名>：そのようなサービスはありません"

- Returned to a client which is attempting to send a SQUERY to a service which does not exist.

- 存在しないサービスにスクエリーを送ろうとしているクライアントに戻りました。

409 ERR_NOORIGIN " :No origin specified"

409 err_noorigin "：原点は指定されていません"

- PING or PONG message missing the originator parameter.

- PingまたはPongメッセージOriginatorパラメーターがありません。

411	ERR_NORECIPIENT ":No recipient given (<command>)"
412	ERR_NOTEXTTOSEND ":No text to send"
413	ERR_NOTOPLEVEL "<mask> :No toplevel domain specified"
414	ERR_WILDTOPLEVEL "<mask> :Wildcard in toplevel domain"
415	ERR_BADMASK "<mask> :Bad Server/host mask"

- 412 - 415 are returned by PRIVMSG to indicate that the message wasn't delivered for some reason.
ERR_NOTOPLEVEL and ERR_WILDTOPLEVEL are errors that are returned when an invalid use of "PRIVMSG \$<server>" or "PRIVMSG #<host>" is attempted.

- 412-415は、メッセージが何らかの理由で配信されなかったことを示すためにprivmsgによって返されます。err_notoplevelとerr_wildtoplevelは、「privmsg \$ <server>」または「privmsg # <host>」の無効な使用が試行されたときに返されるエラーです。

421 ERR_UNKNOWNCOMMAND "<command> :Unknown command"

421 err_unknowncommand "<command>：不明コマンド"

- Returned to a registered client to indicate that the command sent is unknown by the server.

- 登録されたクライアントに戻り、送信されたコマンドがサーバーによって不明であることを示します。

422 ERR_NOMOTD " :MOTD File is missing"

422 err_nomotd "：motdファイルがありません"

- Server's MOTD file could not be opened by the server.		- サーバーのMOTDファイルは、サーバーによって開くことができませんでした。	
423 ERR_NOADMININFO "<server> :No administrative info available"		423 err_noadmininfo "<server>：管理情報はありません」	
- Returned by a server in response to an ADMIN message when there is an error in finding the appropriate information.		- 適切な情報を見つける際にエラーが発生した場合、管理者メッセージに応じてサーバーによって返されます。	
424	ERR_FILEERROR ":File error doing <file op> on <file>"		
- Generic error message used to report a failed file operation during the processing of a message.		- メッセージの処理中に故障したファイル操作を報告するために使用される一般的なエラーメッセージ。	
431 ERR_NONICKNAMEGIVEN ":No nickname given"		431 err_nonicknamegiven "：ニックネームは与えられていません」	
- Returned when a nickname parameter expected for a command and isn't found.		- コマンドにニックネームパラメーターが期待され、見つからないときに返されます。	
432 ERR_ERRONEUSNICKNAME "<nick> :Erroneous nickname"		432 err_erroneusnickname "<nick>：Errounous nickname"	
- Returned after receiving a NICK message which contains characters which do not fall in the defined set. See section 2.3.1 for details on valid nicknames.		- 定義されたセットに該当しない文字を含むニックメッセージを受け取った後に返されました。有効なニックネームの詳細については、セクション2.3.1を参照してください。	
433 ERR_NICKNAMEINUSE "<nick> :Nickname is already in use"		433 err_nicknameinuse "<nick>：ニックネームはすでに使用されています"	
- Returned when a NICK message is processed that results in an attempt to change to a currently existing nickname.		- ニックメッセージが処理されたときに返され、現在既存のニックネームに変更しようとしています。	
436	ERR_NICKCOLLISION "<nick> :Nickname collision KILL from <user>@<host>"		
- Returned by a server to a client when it detects a nickname collision (registered of a NICK that already exists by another server).		- ニックネームの衝突（別のサーバーによってすでに存在するニックに登録されている）を検出したときに、サーバーによってクライアントに返されます。	
437	ERR_UNAVAILRESOURCE "<nick/channel> :Nick/channel is temporarily unavailable"		
- Returned by a server to a user trying to join a channel currently blocked by the channel delay mechanism.		- サーバーが現在ブロックされているチャンネルに参加しようとするユーザーにサーバーによって返されます。	
- Returned by a server to a user trying to change nickname when the desired nickname is blocked by the nick delay mechanism.		- 必要なニックネームがニック遅延メカニズムによってブロックされたときにニックネームを変更しようとするユーザーにサーバーによって返されます。	
441	ERR_USERNOTINCHANNEL "<nick> <channel> :They aren't on that channel"		
- Returned by the server to indicate that the target user of the command is not on the given channel.		- コマンドのターゲットユーザーが指定されたチャンネル上にないことを示すためにサーバーによって返されます。	
442 ERR_NOTONCHANNEL "<channel> :You're not on that channel"		442 err_notonchannel "<channel>：あなたはそのチャンネルにいません"	
- Returned by the server whenever a client tries to perform a channel affecting command for which the client isn't a member.		- クライアントがクライアントがメンバーではないコマンドに影響を与えるチャンネルを実行しようとするたびにサーバーによって返されます。	
443	ERR_USERONCHANNEL "<user> <channel> :is already on channel"		
- Returned when a client tries to invite a user to a channel		- クライアントがユーザーを既に使用しているチャンネルに招待	

they are already on.

444 ERR_NOLOGIN "<user> :User not logged in"

- Returned by the summon after a SUMMON command for a user was unable to be performed since they were not logged in.

445 ERR_SUMMONDISABLED ":SUMMON has been disabled"

- Returned as a response to the SUMMON command. MUST be returned by any server which doesn't implement it.

446 ERR_USERSDISABLED ":USERS has been disabled"

- Returned as a response to the USERS command. MUST be returned by any server which does not implement it.

451 ERR_NOTREGISTERED ":You have not registered"

- Returned by the server to indicate that the client MUST be registered before the server will allow it to be parsed in detail.

461 ERR_NEEDMOREPARAMS "<command> :Not enough parameters"

- Returned by the server by numerous commands to indicate to the client that it didn't supply enough parameters.

462 ERR_ALREADYREGISTRED ":Unauthorized command (already registered)"

- Returned by the server to any link which tries to change part of the registered details (such as password or user details from second USER message).

463 ERR_NOPERMFORHOST ":Your host isn't among the privileged"

- Returned to a client which attempts to register with a server which does not been setup to allow connections from the host the attempted connection is tried.

464 ERR_PASSWDMISMATCH ":Password incorrect"

- Returned to indicate a failed attempt at registering a connection for which a password was required and was either not given or incorrect.

465 ERR_YOUREBANNEDCREEP ":You are banned from this server"

- Returned after an attempt to connect and register yourself with a server which has been setup to explicitly deny connections to you.

466 ERR_YOUWILLBEBANNED

- Sent by a server to a user to inform that access to the server will soon be denied.

しようとする、返されます。

444 err_nologin "<user> : ユーザーがログインしていない"

- ユーザーの召喚コマンドがログインされていないため実行できなかった後、召喚によって返されました。

445 err_summondisabled " : summonが無効になっている」

- 召喚コマンドへの応答として返されます。実装していないサーバーによって返される必要があります。

446 err_usersdisabled " : ユーザーが無効になっている"

- ユーザーコマンドへの応答として返されます。実装していないサーバーによって返される必要があります。

451 err_notregistered " : 登録していない"

- サーバーによって返され、サーバーが詳細に解析される前にクライアントを登録する必要があることを示します。

461 err_needmoreparams "<コマンド> : 十分なパラメーター"

- 十分なパラメーターを提供していないことをクライアントに示すために、多数のコマンドによってサーバーによって返されます。

462 ERR_ALREADYREGISTRED " : 不正コマンド（既に登録）」

- サーバーによって、登録された詳細の一部を変更しようとするリンク（2番目のユーザーメッセージからのパスワードやユーザーの詳細など）に返されます。

463 err_nopermforhost " : あなたのホストは特権の1つではありません」

- ホストからの接続を許可するようにセットアップされていないサーバーに登録しようとするクライアントに戻りました。

464 err_passwdmismatch " : パスワードが正しくない"

- 返され、パスワードが必要であり、与えられていないか、間違っていない接続を登録する試みが失敗したことを示します。

465 err_youbannedcreep " : このサーバーから禁止されています」

- 接続して、あなたへの接続を明示的に拒否するように設定されたサーバーに登録しようとする試みの後に返されます。

466 err_youwillbanned

- サーバーからユーザーに送信されて、サーバーへのアクセスがまもなく拒否されることを通知します。

467 ERR_KEYSET
"<channel> :Channel key already set"
471 ERR_CHANNELISFULL
"<channel> :Cannot join channel (+l)"
472 ERR_UNKNOWNMODE
"<char> :is unknown mode char to me for <channel>"
473 ERR_INVITEONLYCHAN
"<channel> :Cannot join channel (+i)"
474 ERR_BANNEDFROMCHAN
"<channel> :Cannot join channel (+b)"

475	ERR_BADCHANNELKEY "<channel> :Cannot join channel (+k)"
476	ERR_BADCHANMASK "<channel> :Bad Channel Mask"
477	ERR_NOCHANMODES "<channel> :Channel doesn't support modes"
478	ERR_BANLISTFULL "<channel> <char> :Channel list is full"

481 ERR_NOPRIVILEGES ":Permission Denied- You're not an IRC operator"

- Any command requiring operator privileges to operate MUST return this error to indicate the attempt was unsuccessful.

482 ERR_CHANOPRIVSNEEDED "<channel> :You're not channel operator"

- Any command requiring 'chanop' privileges (such as MODE messages) MUST return this error if the client making the attempt is not a chanop on the specified channel.

483 ERR_CANTKILLSERVER ":You can't kill a server!"

- Any attempts to use the KILL command on a server are to be refused and this error returned directly to the client.

484 ERR_RESTRICTED ":Your connection is restricted!"

- Sent by the server to a user upon connection to indicate the restricted nature of the connection (user mode "+r").

485 ERR_UNIQOPPRIVSNEEDED ":You're not the original channel operator"

- Any MODE requiring "channel creator" privileges MUST return this error if the client making the attempt is not a chanop on the specified channel.

491 ERR_NOOPERHOST ":No O-lines for your host"

- If a client sends an OPER message and the server has not been configured to allow connections from the client's host as an operator, this error MUST be returned.

501 ERR_UMODEUNKNOWNFLAG ":Unknown MODE flag"

- Returned by the server to indicate that a MODE message was sent with a nickname parameter and that the a mode flag sent was not recognized.

502 ERR_USERSDONTMATCH ":Cannot change mode for other users"

- Error sent to any user trying to view or change the user mode for a user other than themselves.

5.3 Reserved numerics

These numerics are not described above since they fall into one of the following categories:

1. no longer in use;
2. reserved for future planned use;
3. in current use but are part of a non-generic 'feature' of the current IRC server.

481 err_noprivileges "：許可が拒否されました - あなたはIRCオペレーターではありません

- オペレーターの権限を操作する必要があるコマンドは、このエラーを返して、試みが失敗したことを示す必要があります。

482 err_chanoprivsneeded "<Channel>：あなたはチャンネルオペレーターではありません"

- 「Chanop」特権（モードメッセージなど）を要求するコマンドは、試行を行うクライアントが指定されたチャンネルのチャノップではない場合、このエラーを返す必要があります。

483 err_cantkillserver "：サーバーを殺すことはできません！"

- サーバーでKillコマンドを使用しようとする試みは拒否され、このエラーはクライアントに直接返されます。

484 err_restricted "：接続が制限されています！"

- 接続時にサーバーからユーザーに送信され、接続の制限された性質（ユーザーモード "r"）を示します。

485 err_uniqopprivsneeded "：あなたは元のチャンネルオペレーターではありません

- 「チャンネル作成者」の特権を必要とするモードは、試行を行うクライアントが指定されたチャンネルのチャノップではない場合、このエラーを返す必要があります。

491 err_nooperhost "：ホストのためのo-lines no"

- クライアントがオペレーションメッセージを送信し、サーバーがクライアントのホストからの接続をオペレーターとして許可するように構成されていない場合、このエラーを返す必要があります。

501 err_umodeunknownflag "：不明モードフラグ"

- サーバーによって返され、モードメッセージがニックネームパラメーターで送信され、送信されたモードフラグが認識されていないことを示します。

502 err_usersdontmatch "：他のユーザーのモードを変更できません"

- ユーザーモードを表示または変更しようとするユーザーに送信されたエラーは、ユーザー以外のユーザーモードを変更しようとしています。

5.3 予約済みの数字

これらの数字は、次のカテゴリのいずれかに分類されるため、上記では説明されていません。

1. もはや使用されていません。
2. 将来の計画的な使用のために予約されています。
3. 現在使用しているが、現在のIRCサーバーの非遺伝子「機能」の一部です。

231 RPL_SERVICEINFO 232 RPL_ENDOFSERVICES 233 RPL_SERVICE 300 RPL_NONE 316 RPL_WHOISCHANOP 361 RPL_KILLDONE 362 RPL_CLOSING 363 RPL_CLOSEEND 373 RPL_INFOSTART 384 RPL_MYPORTIS	231 RPL_SERVICEINFO 232 RPL_ENDOFSERVICES 233 RPL_SERVICE 300 RPL_NONE 316 RPL_WHOISCHANOP 361 RPL_KILLDONE 362 RPL_CLOSISE 363 RPL_CLOSEEND 373 RPL_INFOSTART 384 RPL_MYPORTISIS
213 RPL_STATSCLINE 214 RPL_STATSNLINE 215 RPL_STATSIILINE 216 RPL_STATSKLINE 217 RPL_STATSQLINE 218 RPL_STATSYLINE 240 RPL_STATSVLINE 241 RPL_STATSLLINE 244 RPL_STATSHLINE 244 RPL_STATSSLINE 246 RPL_STATSPING 247 RPL_STATSBLINE 250 RPL_STATSDLINE	213 RPL_STATSCLINE 214 RPL_STATSNLINE 215 RPL_STATSIILINE 216 RPL_STATSKLINE 217 RPL_STATSQLINE 218 RPL_STATSYLINE 240 RPL_STATSVLINE 241 RPL_STATSLINE 244 RPL_STATSHLINE 244 RPL_STATSSLINE246 RPL_STATSPING 247 RPL_STATSSLINE 246 RPL_STATSPING 247 RPL_STATSSSLINE
492 ERR_NOSERVICEHOST	492 err_noservicehost

6. Current implementations

The IRC software, version 2.10 is the only complete implementation of the IRC protocol (client and server). Because of the small amount of changes in the client protocol since the publication of RFC 1459 [IRC], implementations that follow it are likely to be compliant with this protocol or to require a small amount of changes to reach compliance.

7. Current problems

There are a number of recognized problems with the IRC Client Protocol, and more generally with the IRC Server Protocol. In order to preserve backward compatibility with old clients, this protocol has almost not evolved since the publication of RFC 1459 [IRC].

7.1 Nicknames

The idea of the nickname on IRC is very convenient for users to use when talking to each other outside of a channel, but there is only a finite nickname space and being what they are, it's not uncommon for several people to want to use the same nick. If a nickname is chosen by two people using this protocol, either one will not succeed or both will removed by use of a server KILL (See Section 3.7.1).

7.2 Limitation of wildcards

There is no way to escape the escape character "\" (%x5C). While this isn't usually a problem, it makes it impossible to form a mask with a backslash character ("\\") preceding a wildcard.

7.3 Security considerations

Security issues related to this protocol are discussed in the "IRC Server Protocol" [IRC-SERVER] as they are mostly an issue for the server side of the connection.

8. Current support and availability

Mailing lists for IRC related discussion: General discussion: ircd-users@irc.org Protocol development: ircd-dev@irc.org	
Software implementations: ftp://ftp.irc.org/irc/server ftp://ftp.funet.fi/pub/unix/irc ftp://ftp.irc.org/irc/clients	

6. 現在の実装

IRCソフトウェアバージョン2.10は、IRCプロトコル（クライアントとサーバー）の唯一の完全な実装です。RFC 1459 [IRC]の公開以来、クライアントプロトコルの変更が少ないため、それに続く実装は、このプロトコルに準拠している可能性があります。

7. 現在の問題

IRCクライアントプロトコル、より一般的にはIRCサーバープロトコルには多くの認識された問題があります。古いクライアントとの後方互換性を維持するために、このプロトコルはRFC 1459 [IRC]の公開以来ほとんど進化していません。

7.1 ニックネーム

IRCのニックネームのアイデアは、ユーザーがチャンネルの外でお互いに話し合うときに使用するのに非常に便利ですが、有限のニックネームスペースしかあり、それらが何であるかだけで、何人かが同じを使いたいと思うことは珍しくありませんニック。このプロトコルを使用して2人によってニックネームが選択された場合、サーバーキルを使用して成功しないか、両方が削除されます（セクション3.7.1を参照）。

7.2 ワイルドカードの制限

エスケープキャラクター「\」（%x5c）を逃れる方法はありません。これは通常問題ではありませんが、ワイルドカードの前にバックスラッシュ文字（「\」）を持つマスクを形成することが不可能になります。

7.3 セキュリティに関する考慮事項

このプロトコルに関連するセキュリティの問題は、接続のサーバー側の問題であるため、「IRCサーバープロトコル」[IRC-Server]で説明されています。

8. 現在のサポートと可用性

9. Acknowledgements

Parts of this document were copied from the RFC 1459 [IRC] which first formally documented the IRC Protocol. It has also benefited from many rounds of review and comments. In particular, the following people have made significant contributions to this document:

Matthew Green, Michael Neumayer, Volker Paulsen, Kurt Roeckx, Vesa Ruokonen, Magnus Tjernstrom, Stefan Zehl.

10. References

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[ABNF] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[HNAME] Braden, R., "Requirements for Internet Hosts -- Application and Support", STD 3, RFC 1123, October 1989.

[IRC] Oikarinen, J. & D. Reed, "Internet Relay Chat Protocol", RFC 1459, May 1993.

[IRC-ARCH] Kalt, C., "Internet Relay Chat: Architecture", RFC 2810, April 2000.

[IRC-CHAN] Kalt, C., "Internet Relay Chat: Channel Management", RFC 2811, April 2000.

[IRC-SERVER] Kalt, C., "Internet Relay Chat: Server Protocol", RFC 2813, April 2000.

11. Author's Address

Christophe Kalt 99 Teaneck Rd, Apt #117 Ridgefield Park, NJ 07660 USA

EMail: kalt@stealth.net

12. Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL

9. 謝辞

このドキュメントの一部は、最初にIRCプロトコルを正式に文書化したRFC 1459 [IRC]からコピーされました。また、多くのラウンドのレビューとコメントの恩恵を受けています。特に、次の人々がこの文書に多大な貢献をしています。

マシュー・グリーン、マイケル・ノイマイヤー、ヴォルカー・ポールセン、カート・ロックズ、ヴェサ・ルーコネン、マグナス・ツェーンストロム、ステファン・ゼール。

10. 参考文献

[キーワード] Bradner、S。、「要件レベルを示すためにRFCで使用するためのキーワード」、BCP 14、RFC 2119、1997年3月。

[ABNF] Crocker、D。およびP. Overell、「構文仕様のためのBNFの増強：ABNF」、RFC 2234、1997年11月。

[HName] Braden、R。、「インターネットホストの要件 - アプリケーションとサポート」、STD 3、RFC 1123、1989年10月。

[IRC] Oikarinen、J。&D. Reed、「インターネットリレーチャットプロトコル」、RFC 1459、1993年5月。

[IRC-Arch] Kalt、C。、「インターネットリレーチャット：アーキテクチャ」、RFC 2810、2000年4月。

[IRC-chan] Kalt、C。、「インターネットリレーチャット：チャンネル管理」、RFC 2811、2000年4月。

[IRC-Server] Kalt、C。、「インターネットリレーチャット：サーバープロトコル」、RFC 2813、2000年4月。

11. 著者の連絡先

Christophe Kalt 99 Teaneck Rd、Apt # 117 Ridgefield Park、NJ 07660 USA

12. 完全な著作権声明

Copyright (c) The Internet Society (2000)。全著作権所有。

このドキュメントと翻訳は他の人にコピーされて提供される場合があります、それについてコメントまたは説明するか、その実装を支援する派生作品は、いかなる種類の制限なしに、準備、コピー、公開、配布される場合があります。、上記の著作権通知とこの段落がそのようなすべてのコピーとデリバティブ作品に含まれている場合。ただし、このドキュメント自体は、インターネット協会や他のインターネット組織への著作権通知や参照を削除するなど、いかなる方法でも変更できない場合があります。インターネット標準のプロセスに従うか、英語以外の言語に翻訳するために必要な場合に従う必要があります。

上記の限られた許可は永続的であり、インターネット社会またはその後継者または譲受人によって取り消されることはありません。

この文書と本書に含まれる情報は、「現状」に基づいて提供されており、インターネット社会とインターネットエンジニアリングタスクフォースは、ここにある情報の使用が行われないと

WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

いう保証を含むがこれらに限定されないすべての保証を否認します。特定の目的に対する商品性または適合性の権利または黙示的な保証を侵害します。

謝辞

RFCエディター機能の資金は現在、インターネット協会によって提供されています。