Network Working Group
Internet-Draft
Expires: July 5, 2004

E. Brocklesby

January 5, 2004

IRC RPL_ISUPPORT Numeric Definition
draft-brocklesby-irc-isupport-03

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on July 5, 2004.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This memo presents a way for the server to unobtrusively advertise the ways in which it differs from the Internet Relay Chat (IRC) specification defined in RFC1459. It is a primary goal to implement this in a way which is completely backwards-compatible with the original protocol, and as much as possible with current non-standard implementations of the ISUPPORT numeric.

Table of Contents

1. Introduction 3
1.1 Terminology 3
1.2 Changes to previous version 3
1.3 Motivation 3
1.4 Notes on examples 4
2. Protocol outline 4
3. Currently defined parameters 7
3.1 CASEMAPPING 7
3.2 CHANLIMIT 7
3.3 CHANMODES 8
3.4 CHANNELLEN 9
3.5 CHANTYPES 9
3.6 EXCEPTS 9
3.7 IDCHAN 10
3.8 INVEX 10
3.9 KICKLEN 10
3.10 MAXLIST 11
3.11 MODES 11
3.12 NETWORK 12
3.13 NICKLEN 12
3.14 PREFIX 13
3.15 SAFELIST 13
3.16 STATUSMSG 14
3.17 STD 14
3.18 TARGMAX 15
3.19 TOPICLEN 15
4. Differences to existing implementations 16
4.1 PREFIX parameter without value 16
4.2 EXCEPTS and INVEX value argument 16
4.3 STATUSMSG / WALLCHOPS 16
4.4 Conflicts with RFC 2812 16
4.5 Default value for CASEMAPPING 17
4.6 CHANLIMIT / MAXCHANNELS 17
4.7 MAXLIST / MAXBANS 17
4.8 CHARSET 17
4.9 TARGMAX / MAXTARGETS 17
5. Security Considerations 18
Normative References 18
Informative references 18
Author’s Address 18
A. Default ISUPPORT values 18
B. Acknowledgements 19
Intellectual Property and Copyright Statements 20

1. Introduction

1.1 Terminology

- o Original IRC protocol: The original IRC protocol as described in RFC 1459 [5].
- o The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].
- o The ABNF syntax used in this document is defined in RFC 2234 [2]
- o The term "character" is this document is used to mean an octet, as defined in RFC 1459 [5], section 2.2.

1.2 Changes to previous version

[Note: To be removed by the RFC editor prior to publication.]

The follow significant changes were made from version 02 to version 03 of this document:

- o The semantics of MAXCHANNELS were changed and it was renamed to CHANLIMIT.
- o The semantics of CHIDLEN were changed and it was renamed to IDCHAN.
- o The description of the protocol was clarified significantly, as were several parameter definitions. Several recommendations of what the client should do when faced with protocol violations by the server were also removed.
- o Several implications or recommendations of client or server behaviour were changed into requirements or removed entirely.
- o In particular, the server is now required to send STD as the first parameter upon client registration, followed by all defined parameters which have no default value.
- o The specification of a parameter’s value was changed to allow "\xHH" sequences, to represent spaces and other characters. This feature is considered experimental and comments are particular appreciated.
- o Delivery of 005 numerics from remote servers is now explicitly prohibited.
- o The CHARSET parameter was found to be unworkable and has been entirely removed.
- o The TARGMAX parameter was added.
- o "X" and "X=" were made identical.

1.3 Motivation

Since the publication of RFC 1459 [5] in 1993, a number of changes

and extensions have been made to the IRC protocol. This has led to a problem whereby clients are unable to correctly interpret some server replies, because the reply, channel mode, and so on may have different meanings on different implementations of the IRC server. It is also difficult for the client to ascertain which protocol extensions may be available on a specific server.

A de facto standard has emerged in the community, originally implemented by the Undernet's IRC server software based on the 005 numeric from DALnet's IRC server, which allows the server to advertise to the client upon connection which protocol extensions it supports. This reply, termed RPL_ISUPPORT, uses the non-standard numeric 005.

Unfortunately, since there is no standard document describing the ISUPPORT numeric, differences have emerged between implementations in IRC server software; it is believed that this reduces the potential usefulness of the feature. This memo attempts to standardise the format and content of the ISUPPORT numeric in an extensible way, such that IRC clients can use the information provided to the maximum extent.

1.4 Notes on examples

Several examples of protocol replies are given throughout this document. These are intended only for clarification of the protocol; in the case of a discrepancy between the example and the formal specification, the specification is always preferred.

2. Protocol outline

The ISUPPORT numeric consists of a series of parameters, each of which maps to a protocol extension supported by the IRC server. A parameter may have an associated value, typically a numeric or string value, which provides additional information on the extension.

The format of the ISUPPORT numeric is the same as other server numeric replies currently used. A client which does not understand the numeric may ignore it; however, it is recommended that IRC clients understand ISUPPORT, in order to allow users the full benefit of features implemented by the IRC server.

The ABNF grammar for the numeric is as follows:

```
isupport = ":" servername SP "005" SP nickname SP
           1*13( token SP ) ":are supported by this server"

token = *1"-" parameter / parameter *1( "=" value )
parameter = 1*20letter
value = *letpun
letter = ALPHA / DIGIT
punct = %d33-47 / %d58-64 / %d91-96 / %d123-126
letpun = letter / punct
```

The format of the postfix descriptive text is not fixed, and may be any string subject to the requirements of RFC 1459 [5] regarding numeric replies. Servername and nickname are as defined in RFC 1459 [5].

The "servername" MUST be the name of the server to which the client is connected; the 005 numeric is never sent remotely across the network. As with other local numerics, when delivered remotely it MUST be converted into a 105 numeric before delivery to the client.

A token is of the form "PARAMETER[=VALUE]" or "-PARAMETER". The forms "X" and "X=" are identical; they both define that the parameter is present but has no value. The server SHOULD send "X", not "X="; this is the normalised form.

The server MUST send the parameter in upper-case text; unless otherwise stated, the parameter’s value is case sensitive.

The parameter’s value may contain sequences of the form "\xHH", where HH is a two-digit hexadecimal number. Each such sequence is considered identical to the equivalent octet after parsing of the reply into separate tokens has occurred.

[Example: X=A\x20B defines one token, "X", with the value "A B", rather than two tokens "X" and "B".]
[Note: The literal string "\x" must therefore be encoded as "\x5Cx".]

If the server has not advertised a CHARSET parameter, it MUST not use such sequences with a value outside those permitted by the above ABNF grammar, with the exception of "\x20"; if it has advertised CHARSET, then it may in addition send any printable character defined in that encoding. Characters in multibyte encodings such as UTF-8 should be sent as a series of \x sequences.

RFC 1459 [5] defines a maximum of 15 parameters to any reply,

including the nickname and the text; therefore, only 13 capabilities are possible per reply.

In order to allow flexibility in the protocol, and future expansion, the server may send more than one ISUPPORT reply per connection. It is RECOMMENDED that consecutive ISUPPORT replies are sent adjacent to each other. The client MUST support receiving multiple ISUPPORT replies, and merge them to produce the final list of supported protocol extensions. It is RECOMMENDED that the server attempt to send 13 tokens per line before sending multiple replies.

On connection to the server, all parameters are assumed to be equal to their default values, if any. Unless later changed by the server, this default value persists throughout the connection. Except as explicitly stated in its definition, a parameter SHOULD NOT be sent unless it changes this default value. The server MUST send an 005 reply after client registration but before any further client commands are processed in order to resolve any ambiguities in parameters with no default value.

The form "-PARAMETER" is used to negate a previously specified parameter; that is, revert to the behaviour that would occur if the parameter had not been specified. This is intended to allow servers to change their capabilities without disconnecting clients. Both parameters with and without a value argument may be negated; however, the value argument is not supplied. It is not required to negate a parameter in order to change its value, the server should merely re-advertise the parameter with the new value.

The server may negate tokens which have not been previously advertised to the client; in this case, the client should ignore the negation.

The server may not advertise and negate the same parameter, nor advertise the same parameter with different value specifiers, in the same ISUPPORT numeric reply. However, the server is free to advertise or negate the same parameters in separate replies.

The server MUST NOT negate a parameter which does not have a meaningful default value.

[Note: Implementations often change the value of a particular parameter upon certain events, such as a successful OPER command from a client. It is important that any relevant parameters be (re)advertised when this occurs.]

3. Currently defined parameters

A number of parameters are currently used in the IRC community, and it is believed to be beneficial to standardise these. They are listed below, with relevant information.

[Note: It is intended and expected that future documents will update and extend the set of defined parameters; this is not meant to be an exhaustive list.]

3.1 CASEMAPPING

o CASEMAPPING=string

The CASEMAPPING parameter allows the server to specify which method it uses to compare equality of case-insensitive strings. Possible values are:

o "ascii": The ASCII characters 97 to 122 (decimal) are defined as the lower-case characters of ASCII 65 to 90 (decimal). No other character equivalency is defined.
o "rfc1459": The ASCII characters 97 to 126 (decimal) are defined as the lower-case characters of ASCII 65 to 94 (decimal). No other character equivalency is defined.
o "strict-rfc1459": The ASCII characters 97 to 125 (decimal) are defined as the lower-case characters of ASCII 65 to 93 (decimal). No other character equivalency is defined.

[Note: The only difference between "rfc1459" and "strict-rfc1459" is that the characters "~" and "^" are not considered equivalent in the "strict-rfc1459" encoding. This is believed to be an mistake in the specification of character equivalency in RFC 1459 [5]; the majority of IRC server implementations known to the author treat these characters as equivalent (however, see Section 4.5).]

The CASEMAPPING token requires a value.

The default value for CASEMAPPING is "rfc1459". While this differs from the historical definition in RFC 1459 [5], it is believed to reflect current IRC server implementations, and is as such more useful.

3.2 CHANLIMIT

o CHANLIMIT=pfx:num[,pfx:num,...]

This parameter specifies the maximum number of channels that a client

may join. The value is a series of "pfx:num" pairs, where 'pfx' refers to one or more channel prefix characters (as specified in CHANTYPES), and 'num' indicates how many of these types of channel the client may join in total. If there is no limit to the number of certain channel type(s) a client may join, the limit should be specified as the empty string, for example "#:".

[Example: CHANLIMIT=#+:10,&: indicates that a client may join up to 10 '#' and '+' channels (for example, 7 '#' channels and 3 '+' channels), and any number of '&' channels.]

Clients on either this server or a remote server may be on more than this number of channels; this parameter is only intended for information on how many channels the client it is advertised to may join.

There is no default value for the CHANLIMIT token.

The CHANLIMIT token requires a value.

3.3 CHANMODES

- CHANMODES=A,B,C,D

The CHANMODES token specifies the modes that may be set on a channel. These modes are split into four categories, as follows:

- Type A: Modes that add or remove an address to or from a list. These modes always take a parameter when sent by the server to a client; when sent by a client, they may be specified without a parameter, which requests the server to display the current contents of the corresponding list on the channel to the client.
- Type B: Modes that change a setting on the channel. These modes always take a parameter.
- Type C: Modes that change a setting on the channel. These modes take a parameter only when set; the parameter is absent when the mode is removed both in the client's and server's MODE command.
- Type D: Modes that change a setting on the channel. These modes never take a parameter.

If the server sends any additional types after these 4, the client MUST ignore them; this is intended to allow future extension of this token.

The IRC server MUST NOT list modes in CHANMODES which are also present in the PREFIX parameter; however, for completeness, modes described in PREFIX may be treated as type B modes.

If the server does not support any modes corresponding to a particular type, it should advertise that type as the empty string.

[Example: A server supporting no channel modes would advertise "CHANMODES=,,,".]
[Example: CHANMODES=b,k,l,imnpst]

The CHANMODES token requires a value.

There is no default value for the CHANMODES token.

3.4 CHANNELLEN

o CHANNELLEN=number

The CHANNELLEN parameter specifies the maximum length of the name of a channel that may be created by a client. The server may make known to the client a channel with a name longer than that specified in this value -- that is, the client must not depend on a channel's name never being longer than this.

The CHANNELLEN token does not require a value; if none is given, channel names are not limited in length. If a value is given, it must be numeric.

The default value for CHANNELLEN is 200; this corresponds to RFC 1459 [5].

3.5 CHANTYPES

o CHANTYPES=chars

The CHANTYPES parameter specifies the valid characters to begin a channel name.

[Example: CHANTYPES=+#& defines that channels names may begin with either +, #, or &; for example, #mychannel.]

The default value for CHANTYPES is "CHANTYPES=#&", which corresponds to RFC 1459 [5]. It SHOULD NOT be specified if the server supports exactly these channel types.

The CHANMODES parameter does not require a value; if none is given, the server does not support any channel types.

3.6 EXCEPTS

o EXCEPTS[=modechar]

The EXCEPTS parameter indicates that the server supports "ban exceptions" (channel mode +e), as defined in RFC 2811 [3], section 4.3.1. The optional value argument to EXCEPTS indicates which channel mode is used for ban exceptions. If the token is specified with no value, it is assumed that mode +e is used.

The default value for EXCEPTS is that channel exceptions are not supported.

3.7 IDCHAN

o IDCHAN=pfx:num[,pfx:num,...]

The IDCHAN parameter indicates the existence of "safe" channels as described in RFC 2811 [3], and the length of the "id" portion of those channel names.

Each mode:num pair indicates one or more channel name prefixes which corresponds to a "safe" channel, and the length of the ID portion of those channels' name.

[Example: IDCHAN=!:5 means the client should expect IDs which are 5 characters in length on "!" channels; for example "!JNB4Sircd", where "JNB4S" is the ID and "ircd" is the channel's short name.]

The IDCHAN token requires a value.

The default value for IDCHAN is no value; that is, there are no "safe" channel types.

3.8 INVEX

o INVEX[=modechar]

The INVEX parameter indicates that the server supports "invite exceptions", as defined in RFC 2811 [3], section 4.3.2. The optional value argument to INVEX indicates which channel mode is used for invite exceptions. If the token is specified with no value, it is assumed that mode +I is used.

The default value for INVEX is that channel invite exceptions are not available.

3.9 KICKLEN

o KICKLEN=number

The KICKLEN parameter specifies the maximum length of a KICK message that a client may use. Note that it only specifies the length the client should send to the server -- the server may send KICK messages with a length longer than this value.

The KICKLEN token does not require a value; if none is given, KICK messages are not limited in length. If a value is given, it must be numeric.

There is no default value for the KICKLEN token.

3.10 MAXLIST

o MAXLIST=mode:num[,mode:num,...]

This parameter specifies the maximum numbers of ’list modes’ (type A modes in CHANMODES) that a client may set on a channel at one time. Note that this MUST only be interpreted as applying to new modes which are set by clients -- it should not be used to infer the maximum length of any mode lists returned by the server.

The parameter is a series of mode-number pairs, each of which specifies one or more type A modes, along with the maximum size of the associated list for those modes. Modes which are specified in the same pair share the same maximum size.

[Example: Given "b:25,eI:50", it would be possible to set up to 25 "+b" modes, and up to 50 of a combination of "+e" and "+I" modes, e.g. 30 "+e" and 20 "+I" modes, making up a total of 50.]
[Example: MAXLIST=b:25 indicates that 25 bans may be set on a channel at one time.]

The MAXLIST token requires a value.

There is no default value for the MAXLIST token.

3.11 MODES

o MODES=number

This parameter specifies the maximum number of "variable" modes which may by set on a channel by a single MODE command from a client. A "variable" mode is defined as being type A, B and C modes as defined for CHANMODES, and channel modes specified in the PREFIX parameter.

[Example: MODES=3 indicates that 3 modes may be set with a MODE command.]

The value of MODES does not limit the number of modes in a MODE command which is sent from the server to the client; the client MUST NOT rely on this being the case.

The default value for the MODES parameter is 3, which corresponds to RFC 1459 [5].

The MODES token does not require a value; if none is given, the number of modes which may be set in one command is not limited. If a value is given, it must be numeric.

3.12 NETWORK

o NETWORK=name

The NETWORK parameter defines the name of the IRC network that the client is connected to.

[Example: NETWORK=EFnet indicates that the client is connected to the EFnet IRC network.]

Note that this parameter is intended only for user display purposes; the client SHOULD NOT assume further capabilities or features of the IRC server based on the value of the NETWORK parameter.

The NETWORK token requires a value.

The default value of the NETWORK token is no value; that is, the network does not have a name specified.

3.13 NICKLEN

o NICKLEN=number

This parameter specifies the maximum nickname length that the client may use in a nickname.

[Example: NICKLEN=9 indicates that clients may have nicknames up to 9 characters in length.]

This parameter does not restrict the length of any nicknames other clients on the network may use.

The NICKLEN token requires a numeric value.

The default value for NICKLEN is 9, which corresponds to RFC 1459 [5].

3.14 PREFIX

o PREFIX=[(modes)prefixes]

The PREFIX parameter specifies a list of channel status flags (the "modes" section) that clients may have on channels, followed by a mapping to the equivalent channel status flags ("prefixes"), which are used in NAMES and WHO replies. There is a one to one mapping between each mode and prefix.

The order of the modes is from that which gives most privileges on the channel, to that which gives the least.

[Example: (ab)&* maps the channel mode 'a' to the channel status flag '&', and channel mode 'b' to the channel status flag '*'.]

[Example: PREFIX=(ohv)@%+ maps channel mode 'o' to status '@', 'h' to status '%', and 'v' to status +.

The default value for PREFIX is "PREFIX=(ov)@+", which corresponds to RFC 1459 [5]. It SHOULD NOT be specified if the server provides only these modes. If a server provides ANY additional status flags, it MUST also provide (ov)@+ (assuming they are applicable to the server). The PREFIX parameter may be advertised with a null value specifier; this indicates that no prefixes are supported by the IRC server.

Note that PREFIX does NOT specify whether or not the server sends multiple prefix characters for a user in NAMES replies.

3.15 SAFELIST

o SAFELIST

The SAFELIST parameter indicates that the client may request a "LIST" command from the server, without being disconnected due to the large amount of data generated by the command.

The SAFELIST token must not be specified with a value.

The default value for the SAFELIST token is none; that is, the client may not safely request a LIST command.

3.16 STATUSMSG

o STATUSMSG=string

The server supports a method of sending a NOTICE message to only those people on a channel with the specified status. This is done via a NOTICE command, with the channel prefixed by the desired status flag as the target.

[Example: NOTICE @#channel :Hi there]

The server should deliver the message to all users on the specified channel with equal or higher status on the channel as the status flag indicates.

[Example: STATUSMSG=@+ indicates that "@#channel" and "+#channel" would be valid targets. A message to "+#channel" would deliver the message to all users with voice and channel operator privileges on #channel, assuming that the server supported the PREFIX value (ov)@+.]

The required value argument to STATUSMSG indicates which prefixes (from the PREFIX parameter) are valid status values for use in NOTICE commands.

The server MUST NOT advertise a character in STATUSMSG which is also present in CHANTYPES.

The STATUSMSG token requires a value.

The default value of the STATUSMSG token is none; that is, the server does not support this form of messaging.

3.17 STD

o STD=version[,version[,...]]

The STD parameter indicates which form(s) of the ISUPPORT numeric are used by the server. Currently, one only possible value is defined; that is "rfcnnnn", which refers to this document.

[Note: To be changed by the RFC Editor before publication.]

The STD parameter is intended to be extensible, so that if later standards emerge which update this document, the server may be able to advertise this. The "version" string is free-form subject to the requirements in section ABNF, however, protocol updates defined in RFCs should be named "rfcxxxx", where "xxxx" is the relevant RFC

number.

A server may support any number of STD versions. However, new version strings MUST NOT be added unless there is an ambiguity between two tokens defined with different meanings in two different standards. It is expected that most new features may be advertised simply by additional parameters, in which case a new version string is not required.

The STD token MUST be the first token advertised by the server upon connection.

The STD token requires a value.

The default value for the STD parameter is none; that is, no standardised ISUPPORT is available.

3.18 TARGMAX

o TARGMAX=[cmd:lim,cmd:lim...]

The TARGMAX parameter specifies the maximum number of targets allowable for commands which accept multiple targets. It consists of a series of cmd:lim pairs, where each command ’cmd’ allows up to ’lim’ targets (generally either channels or nicks). In the case of the KICK command, the limit indicates the maximum number of (user, channel) pairs which may be specified in any one KICK command.

[Example: TARGMAX=PRIVMSG:4,NOTICE:3 would allow "PRIVMSG A,B,C,D :message" and "NOTICE A,B,C :message", but not "PRIVMSG A,B,C,D,E :message" or "NOTICE A,B,C,D :message".]

If no argument is given for a particular command (e.g. "WHOIS:"), that command does not have a limit on the number of targets.

The server MUST specify all commands available to the user which support multiple targets.

The default value of TARGMAX is that no commands allow multiple targets. If this is the case, the server SHOULD NOT specify "TARGMAX=".

3.19 TOPICLEN

o TOPICLEN=number

The TOPICLEN parameter specifies the maximum length of the topic specified in the TOPIC command for a channel. Note that it only

specifies the length of topic that may be set -- the server is free to return topics longer than this length to the client.

The TOPICLEN token does not require a value; if none is given, the length of channel topics is not limited.

The TOPICLEN token requires a numeric value.

4. Differences to existing implementations

A number of differences exist between the ISUPPORT defined in this document and traditional implementations of the ISUPPORT numeric.

4.1 PREFIX parameter without value

The PREFIX parameter is traditionally not sent without a value parameter; indeed, the author is not aware of any IRC server implementations where this would be appropriate. However, it is believed that this support is desired to allow extra flexibility, while retaining compatibility with traditional PREFIX implementations.

4.2 EXCEPTS and INVEX value argument

EXCEPTS and INVEX traditionally take no argument -- while they indicate presence of these features on the server, they do not specify the channel mode which is associated with these features. It is believed that the argument value described here provides extra flexibility while retaining backwards compatibility.

4.3 STATUSMSG / WALLCHOPS

The STATUSMSG parameter replaces the traditional WALLCHOPS parameter used by some current implementations. It is believed that the name STATUSMSG better reflects the functionality; since the argument to STATUSMSG is not optional, it would break backwards compatibility to use the name WALLCHOPS. It was not considered beneficial to allow a STATUSMSG flag without a value.

4.4 Conflicts with RFC 2812

RFC 2812 [4], section 5.1, defines a numeric reply "RPL_BOUNCE", with the associated number "005". While this conflicts with the ISUPPORT numeric, it is considered that ISUPPORT has received much more widespread support, and is the de facto standard for use of the 005 numeric. The only server implementation known to use this numeric has now changed it to 010.

RFC2812 is an Informational RFC and does not not specify an Internet standard.

4.5 Default value for CASEMAPPING

The default value for CASEMAPPING ("rfc1459") was chosen because it reflects the prevailing implementations of the IRC server software currently in use. While some IRC servers have moved to the "ascii" case mapping, those known to the author indicate this via CASEMAPPING=ascii; therefore this is not believed to introduce any compatibility problems.

4.6 CHANLIMIT / MAXCHANNELS

CHANLIMIT replaces the traditional MAXCHANNELS parameter. MAXCHANNELS did not specify which types of channel(s) the limit applied to; many server implementation did not apply the limit to server-local "&" channels, for example. The token was renamed from MAXCHANNELS to CHANLIMIT to prevent confusion.

4.7 MAXLIST / MAXBANS

MAXLIST replaces the traditional MAXBANS token. MAXBANS was considered non-useful, because of its ambiguous meaning; two of the largest IRC networks, for example, could not agree whether "MAXBANS=x" was equivalent to "MAXLIST=beI:x" or "MAXLIST=b:x,e:x,I:x". MAXLIST is also considerably more flexible; to standardise either of the described behaviours for MAXBANS would leave some IRC servers unable to accurately describe their capabilities.

4.8 CHARSET

The traditional CHARSET parameter has been entirely removed. It was found to be unworkable; a correct specification could not be devised to represent its meaning across implementations. Several other methods to implement the same functionality are under discussion but are outside the scope of this document.

4.9 TARGMAX / MAXTARGETS

Traditional implementations use MAXTARGETS instead of TARGMAX. However, TARGMAX allowed several commands to be specified (such as WHOIS and KICK) whereas MAXTARGETS only applies to channels. TARGMAX is also more extendable to cope with future changes.

5. Security Considerations

This memo does not raise any security considerations.

Normative References

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

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

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

[4] Kalt, C., "Internet Relay Chat: Client Protocol", RFC 2812, April 2000.

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

Informative references

[6] Roeckx, K., "The 005 numeric", September 2002.

Author’s Address

Edward Brocklesby
57 Williamson Way
Oxford, Oxon OX4 4TU
GB

EMail: ejb@goth.net

Appendix A. Default ISUPPORT values

As an aid to implementation, a standard ISUPPORT reply with all values which may be assumed to be at their defaults upon connection is supplied here (lines broken due to formatting requirements).

```
:irc.example.com 005 nickname :CASEMAPPING=rfc1459 CHANNELLEN=200
    CHANTYPES=#& MODES=3 NICKLEN=9 PREFIX=(ov)@+ TARGMAX
```

In addition, the server must provide values for the following parameters: CHANLIMIT, CHANMODES, KICKLEN, MAXLIST, STD, TOPICLEN.

Appendix B. Acknowledgements

The author gratefully acknowledges the contributions of Bill Fenner ("fenestro"), Perry Lorier ("Isomer"), Kurt Roeckx ("Q") and John Midgley ("CrazyEddy") in the preparation of this document.

This document is heavily based on a previous document entitled "The 005 numeric".

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2004). 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 assignees.

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 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.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.