Network Working Group
Request for Comments: 2811
Updates: 1459
Category: Informational

C. Kalt

April 2000

Internet Relay Chat: Channel Management

Status of this Memo

Copyright Notice

Abstract

One of the most notable characteristics of the IRC (Internet Relay Chat) protocol is to allow for users to be grouped in forums, called channels, providing a mean for multiple users to communicate together.

There was originally a unique type of channels, but with the years, new types appeared either as a response to a need, or for experimental purposes.

This document specifies how channels, their characteristics and properties are managed by IRC servers.

Table of Contents

1. Introduction

This document defines in detail on how channels are managed by the IRC servers and will be mostly useful to people working on implementing an IRC server.

While the concepts defined here are an important part of IRC, they remain non essential for implementing clients. While the trend seems to be towards more and more complex and "intelligent" clients which are able to take advantage of knowing the internal workings of channels to provide the users with a more friendly interface, simple clients can be implemented without reading this document.

Many of the concepts defined here were designed with the IRC architecture [IRC-ARCH] in mind and mostly make sense in this context. However, many others could be applied to other architectures in order to provide forums for a conferencing system.

Finally, it is to be noted that IRC users may find some of the following sections of interest, in particular sections 2 (Channel Characteristics) and 4 (Channel Modes).

2. Channel Characteristics

A channel is a named group of one or more users which will all receive messages addressed to that channel. A channel is characterized by its name, properties and current members.

2.1 Namespace

Channels names are strings (beginning with a '&', '#', '+' or '!' character) of length up to fifty (50) characters. Channel names are case insensitive.

Apart from the the requirement that the first character being either '&', '#', '+' or '!' (hereafter called "channel prefix"). The only restriction on a channel name is that it SHALL NOT contain any spaces (' '), a control G (^G or ASCII 7), a comma (',' which is used as a list item separator by the protocol). Also, a colon (':') is used as a delimiter for the channel mask. The exact syntax of a channel name is defined in "IRC Server Protocol" [IRC-SERVER].

The use of different prefixes effectively creates four (4) distinct namespaces for channel names. This is important because of the protocol limitations regarding namespaces (in general). See section 6.1 (Labels) for more details on these limitations.

2.2 Channel Scope

A channel entity is known by one or more servers on the IRC network. A user can only become member of a channel known by the server to which the user is directly connected. The list of servers which know

of the existence of a particular channel MUST be a contiguous part of the IRC network, in order for the messages addressed to the channel to be sent to all the channel members.

Channels with '&' as prefix are local to the server where they are created.

Other channels are known to one (1) or more servers that are connected to the network, depending on the channel mask:

If there is no channel mask, then the channel is known to all the servers.

If there is a channel mask, then the channel MUST only be known to servers which has a local user on the channel, and to its neighbours if the mask matches both the local and neighbouring server names. Since other servers have absolutely no knowledge of the existence of such a channel, the area formed by the servers having a name matching the mask has to be contiguous for the channel to be known by all these servers. Channel masks are best used in conjunction with server hostmasking [IRC-SERVER].

2.3 Channel Properties

Each channel has its own properties, which are defined by channel modes. Channel modes can be manipulated by the channel members. The modes affect the way servers manage the channels.

Channels with '+' as prefix do not support channel modes. This means that all the modes are unset, with the exception of the 't' channel flag which is set.

2.4 Privileged Channel Members

In order for the channel members to keep some control over a channel, and some kind of sanity, some channel members are privileged. Only these members are allowed to perform the following actions on the channel:

```
INVITE  - Invite a client to an invite-only channel (mode +i)
KICK    - Eject a client from the channel
MODE    - Change the channel's mode, as well as
          members' privileges
PRIVMSG - Sending messages to the channel (mode +n, +m, +v)
TOPIC   - Change the channel topic in a mode +t channel
```

2.4.1 Channel Operators

The channel operators (also referred to as a "chop" or "chanop") on a given channel are considered to 'own' that channel. Ownership of a channel is shared among channel operators.

Channel operators are identified by the '@' symbol next to their nickname whenever it is associated with a channel (i.e., replies to the NAMES, WHO and WHOIS commands).

Since channels starting with the character '+' as prefix do not support channel modes, no member can therefore have the status of channel operator.

2.4.2 Channel Creator

A user who creates a channel with the character '!' as prefix is identified as the "channel creator". Upon creation of the channel, this user is also given channel operator status.

In recognition of this status, the channel creators are endowed with the ability to toggle certain modes of the channel which channel operators may not manipulate.

A "channel creator" can be distinguished from a channel operator by issuing the proper MODE command. See the "IRC Client Protocol" [IRC-CLIENT] for more information on this topic.

3. Channel lifetime

In regard to the lifetime of a channel, there are typically two groups of channels: standard channels which prefix is either '&', '#' or '+', and "safe channels" which prefix is '!'.

3.1 Standard channels

These channels are created implicitly when the first user joins it, and cease to exist when the last user leaves it. While the channel exists, any client can reference the channel using the name of the channel.

The user creating a channel automatically becomes channel operator with the notable exception of channels which name is prefixed by the character '+', see section 4 (Channel modes). See section 2.4.1 (Channel Operators) for more details on this title.

In order to avoid the creation of duplicate channels (typically when the IRC network becomes disjoint because of a split between two servers), channel names SHOULD NOT be allowed to be reused by a user if a channel operator (See Section 2.4.1 (Channel Operators)) has recently left the channel because of a network split. If this happens, the channel name is temporarily unavailable. The duration while a channel remains unavailable should be tuned on a per IRC network basis. It is important to note that this prevents local users from creating a channel using the same name, but does not prevent the channel to be recreated by a remote user. The latter typically happens when the IRC network rejoins. Obviously, this mechanism only makes sense for channels which name begins with the character '#', but MAY be used for channels which name begins with the character '+'. This mechanism is commonly known as "Channel Delay".

3.2 Safe Channels

Unlike other channels, "safe channels" are not implicitly created. A user wishing to create such a channel MUST request the creation by sending a special JOIN command to the server in which the channel identifier (then unknown) is replaced by the character '!'. The creation process for this type of channel is strictly controlled. The user only chooses part of the channel name (known as the channel "short name"), the server automatically prepends the user provided name with a channel identifier consisting of five (5) characters. The channel name resulting from the combination of these two elements is unique, making the channel safe from abuses based on network splits.

The user who creates such a channel automatically becomes "channel creator". See section 2.4.2 (Channel Creator) for more details on this title.

A server MUST NOT allow the creation of a new channel if another channel with the same short name exists; or if another channel with the same short name existed recently AND any of its member(s) left because of a network split. Such channel ceases to exist after last user leaves AND no other member recently left the channel because of a network split.

Unlike the mechanism described in section 5.2.2 (Channel Delay), in this case, channel names do not become unavailable: these channels may continue to exist after the last user left. Only the user creating the channel becomes "channel creator", users joining an existing empty channel do not automatically become "channel creator" nor "channel operator".

To ensure the uniqueness of the channel names, the channel identifier created by the server MUST follow specific rules. For more details on this, see section 5.2.1 (Channel Identifier).

4. Channel Modes

The various modes available for channels are as follows:

O - give "channel creator" status;
o - give/take channel operator privilege;
v - give/take the voice privilege;

a - toggle the anonymous channel flag;
i - toggle the invite-only channel flag;
m - toggle the moderated channel;
n - toggle the no messages to channel from clients on the outside;
q - toggle the quiet channel flag;
p - toggle the private channel flag;
s - toggle the secret channel flag;
r - toggle the server reop channel flag;
t - toggle the topic settable by channel operator only flag;

k - set/remove the channel key (password);
l - set/remove the user limit to channel;

b - set/remove ban mask to keep users out;
e - set/remove an exception mask to override a ban mask;
I - set/remove an invitation mask to automatically override the invite-only flag;

Unless mentioned otherwise below, all these modes can be manipulated by "channel operators" by using the MODE command defined in "IRC Client Protocol" [IRC-CLIENT].

4.1 Member Status

The modes in this category take a channel member nickname as argument and affect the privileges given to this user.

4.1.1 "Channel Creator" Status

The mode 'O' is only used in conjunction with "safe channels" and SHALL NOT be manipulated by users. Servers use it to give the user creating the channel the status of "channel creator".

4.1.2 Channel Operator Status

The mode 'o' is used to toggle the operator status of a channel member.

4.1.3 Voice Privilege

The mode 'v' is used to give and take voice privilege to/from a channel member. Users with this privilege can talk on moderated channels. (See section 4.2.3 (Moderated Channel Flag).

4.2 Channel Flags

The modes in this category are used to define properties which affects how channels operate.

4.2.1 Anonymous Flag

The channel flag 'a' defines an anonymous channel. This means that when a message sent to the channel is sent by the server to users, and the origin is a user, then it MUST be masked. To mask the message, the origin is changed to "anonymous!anonymous@anonymous." (e.g., a user with the nickname "anonymous", the username "anonymous" and from a host called "anonymous."). Because of this, servers MUST forbid users from using the nickname "anonymous". Servers MUST also NOT send QUIT messages for users leaving such channels to the other channel members but generate a PART message instead.

On channels with the character '&' as prefix, this flag MAY be toggled by channel operators, but on channels with the character '!' as prefix, this flag can be set (but SHALL NOT be unset) by the "channel creator" only. This flag MUST NOT be made available on other types of channels.

Replies to the WHOIS, WHO and NAMES commands MUST NOT reveal the presence of other users on channels for which the anonymous flag is set.

4.2.2 Invite Only Flag

When the channel flag 'i' is set, new members are only accepted if their mask matches Invite-list (See section 4.3.2) or they have been invited by a channel operator. This flag also restricts the usage of the INVITE command (See "IRC Client Protocol" [IRC-CLIENT]) to channel operators.

4.2.3 Moderated Channel Flag

The channel flag 'm' is used to control who may speak on a channel. When it is set, only channel operators, and members who have been given the voice privilege may send messages to the channel.

This flag only affects users.

4.2.4 No Messages To Channel From Clients On The Outside

When the channel flag 'n' is set, only channel members MAY send messages to the channel.

This flag only affects users.

4.2.5 Quiet Channel

The channel flag 'q' is for use by servers only. When set, it restricts the type of data sent to users about the channel operations: other user joins, parts and nick changes are not sent. From a user's point of view, the channel contains only one user.

This is typically used to create special local channels on which the server sends notices related to its operations. This was used as a more efficient and flexible way to replace the user mode 's' defined in RFC 1459 [IRC].

4.2.6 Private and Secret Channels

The channel flag 'p' is used to mark a channel "private" and the channel flag 's' to mark a channel "secret". Both properties are similar and conceal the existence of the channel from other users.

This means that there is no way of getting this channel's name from the server without being a member. In other words, these channels MUST be omitted from replies to queries like the WHOIS command.

When a channel is "secret", in addition to the restriction above, the server will act as if the channel does not exist for queries like the TOPIC, LIST, NAMES commands. Note that there is one exception to this rule: servers will correctly reply to the MODE command. Finally, secret channels are not accounted for in the reply to the LUSERS command (See "Internet Relay Chat: Client Protocol" [IRC-CLIENT]) when the <mask> parameter is specified.

The channel flags ’p’ and ’s’ MUST NOT both be set at the same time. If a MODE message originating from a server sets the flag ’p’ and the flag ’s’ is already set for the channel, the change is silently ignored. This should only happen during a split healing phase (mentioned in the "IRC Server Protocol" document [IRC-SERVER]).

4.2.7 Server Reop Flag

The channel flag ’r’ is only available on channels which name begins with the character ’!’ and MAY only be toggled by the "channel creator".

This flag is used to prevent a channel from having no channel operator for an extended period of time. When this flag is set, any channel that has lost all its channel operators for longer than the "reop delay" period triggers a mechanism in servers to reop some or all of the channel inhabitants. This mechanism is described more in detail in section 5.2.4 (Channel Reop Mechanism).

4.2.8 Topic

The channel flag ’t’ is used to restrict the usage of the TOPIC command to channel operators.

4.2.9 User Limit

A user limit may be set on channels by using the channel flag ’l’. When the limit is reached, servers MUST forbid their local users to join the channel.

The value of the limit MUST only be made available to the channel members in the reply sent by the server to a MODE query.

4.2.10 Channel Key

When a channel key is set (by using the mode ’k’), servers MUST reject their local users request to join the channel unless this key is given.

The channel key MUST only be made visible to the channel members in the reply sent by the server to a MODE query.

4.3 Channel Access Control

The last category of modes is used to control access to the channel, they take a mask as argument.

In order to reduce the size of the global database for control access modes set for channels, servers MAY put a maximum limit on the number of such modes set for a particular channel. If such restriction is imposed, it MUST only affect user requests. The limit SHOULD be homogeneous on a per IRC network basis.

4.3.1 Channel Ban and Exception

When a user requests to join a channel, his local server checks if the user's address matches any of the ban masks set for the channel. If a match is found, the user request is denied unless the address also matches an exception mask set for the channel.

Servers MUST NOT allow a channel member who is banned from the channel to speak on the channel, unless this member is a channel operator or has voice privilege. (See Section 4.1.3 (Voice Privilege)).

A user who is banned from a channel and who carries an invitation sent by a channel operator is allowed to join the channel.

4.3.2 Channel Invitation

For channels which have the invite-only flag set (See Section 4.2.2 (Invite Only Flag)), users whose address matches an invitation mask set for the channel are allowed to join the channel without any invitation.

5. Current Implementations

The only current implementation of these rules as part of the IRC protocol is the IRC server, version 2.10.

The rest of this section deals with issues that are mostly of importance to those who wish to implement a server but some parts may also be of interest for client writers.

5.1 Tracking Recently Used Channels

This mechanism is commonly known as "Channel Delay" and generally only applies to channels which names is prefixed with the character '#' (See Section 3.1 "Standard channels").

When a network split occurs, servers SHOULD keep track of which channels lost a "channel operator" as the result of the break. These channels are then in a special state which lasts for a certain period of time. In this particular state, the channels cannot cease to

exist. If all the channel members leave the channel, the channel becomes unavailable: the server local clients cannot join the channel as long as it is empty.

Once a channel is unavailable, it will become available again either because a remote user has joined the channel (most likely because the network is healing), or because the delay period has expired (in which case the channel ceases to exist and may be re-created).

The duration for which a channel death is delayed SHOULD be set considering many factors among which are the size (user wise) of the IRC network, and the usual duration of network splits. It SHOULD be uniform on all servers for a given IRC network.

5.2 Safe Channels

This document introduces the notion of "safe channels". These channels have a name prefixed with the character '!' and great effort is made to avoid collisions in this name space. Collisions are not impossible, however they are very unlikely.

5.2.1 Channel Identifier

The channel identifier is a function of the time. The current time (as defined under UNIX by the number of seconds elapsed since 00:00:00 GMT, January 1, 1970) is converted in a string of five (5) characters using the following base: "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890" (each character has a decimal value starting from 0 for 'A' to 35 for '0').

The channel identifier therefore has a periodicity of 36^5 seconds (about 700 days).

5.2.2 Channel Delay

These channels MUST be subject to the "channel delay" mechanism described in section 5.1 (Channel Delay). However, the mechanism is slightly adapted to fit better.

Servers MUST keep track of all such channels which lose members as the result of a network split, no matter whether the user is a "channel operator" or not.

However, these channels do NOT ever become unavailable, it is always possible to join them even when they are empty.

5.2.3 Abuse Window

Because the periodicity is so long, attacks on a particular channel (name) may only occur once in a very long while. However, with luck and patience, it is still possible for a user to cause a channel collision. In order to avoid this, servers MUST "look in the future" and keep a list of channel names which identifier is about to be used (in the coming few days for example). Such list should remain small, not be a burden for servers to maintain and be used to avoid channel collisions by preventing the re-creation of such channel for a longer period of time than channel delay does.

Eventually a server MAY choose to extend this procedure to forbid creation of channels with the same shortname only (then ignoring the channel identifier).

5.2.4 Preserving Sanity In The Name Space

The combination of the mechanisms described in sections 5.2.2 and 5.2.3 makes it quite difficult for a user to create a channel collision. However, another type of abuse consists of creating many channels having the same shortname, but different identifiers. To prevent this from happening, servers MUST forbid the creation of a new channel which has the same shortname of a channel currently existing.

5.2.5 Server Reop Mechanism

When a channel has been opless for longer than the "reop delay" period and has the channel flag ’r’ set (See Section 4.2.7 (Server Reop Flag)), IRC servers are responsible for giving the channel operator status randomly to some of the members.

The exact logic used for this mechanism by the current implementation is described below. Servers MAY use a different logic, but that it is strongly RECOMMENDED that all servers use the same logic on a particular IRC network to maintain coherence as well as fairness. For the same reason, the "reop delay" SHOULD be uniform on all servers for a given IRC network. As for the "channel delay", the value of the "reop delay" SHOULD be set considering many factors among which are the size (user wise) of the IRC network, and the usual duration of network splits.

a) the reop mechanism is triggered after a random time following the expiration of the "reop delay". This should limit the eventuality of the mechanism being triggered at the same time (for the same channel) on two separate servers.

b) If the channel is small (five (5) users or less), and the "channel delay" for this channel has expired,
Then reop all channel members if at least one member is local to the server.

c) If the channel is small (five (5) users or less), and the "channel delay" for this channel has expired, and the "reop delay" has expired for longer than its value,
Then reop all channel members.

d) For other cases, reop at most one member on the channel, based on some method build into the server. If you don't reop a member, the method should be such that another server will probably op someone. The method SHOULD be the same over the whole network. A good heuristic could be just random reop.
(The current implementation actually tries to choose a member local to the server who has not been idle for too long, eventually postponing action, therefore letting other servers have a chance to find a "not too idle" member. This is over complicated due to the fact that servers only know the "idle" time of their local users)

6. Current problems

There are a number of recognized problems with the way IRC channels are managed. Some of these can be directly attributed to the rules defined in this document, while others are the result of the underlying "IRC Server Protocol" [IRC-SERVER]. Although derived from RFC 1459 [IRC], this document introduces several novelties in an attempt to solve some of the known problems.

6.1 Labels

This document defines one of the many labels used by the IRC protocol. Although there are several distinct namespaces (based on the channel name prefix), duplicates inside each of these are not allowed. Currently, it is possible for users on different servers to pick the label which may result in collisions (with the exception of channels known to only one server where they can be averted).

6.1.1 Channel Delay

The channel delay mechanism described in section 5.1 (Tracking Recently Used Channels) and used for channels prefixed with the character '#' is a simple attempt at preventing collisions from happening. Experience has shown that, under normal circumstances, it

is very efficient; however, it obviously has severe limitations keeping it from being an adequate solution to the problem discussed here.

6.1.2 Safe Channels

"Safe channels" described in section 3.2 (Safe Channels) are a better way to prevent collisions from happening as it prevents users from having total control over the label they choose. The obvious drawback for such labels is that they are not user friendly. However, it is fairly trivial for a client program to improve on this.

6.2 Mode Propagation Delays

Because of network delays induced by the network, and because each server on the path is REQUIRED to check the validity of mode changes (e.g., user exists and has the right privileges), it is not unusual for a MODE message to only affect part of the network, often creating a discrepancy between servers on the current state of a channel.

While this may seem easy to fix (by having only the original server check the validity of mode changes), it was decided not to do so for various reasons. One concern is that servers cannot trust each other, and that a misbehaving servers can easily be detected. This way of doing so also stops wave effects on channels which are out of synch when mode changes are issued from different directions.

6.3 Collisions And Channel Modes

The "Internet Relay Chat: Server Protocol" document [IRC-SERVER] describes how channel data is exchanged when two servers connect to each other. Channel collisions (either legitimate or not) are treated as inclusive events, meaning that the resulting channel has for members all the users who are members of the channel on either server prior to the connection.

Similarly, each server sends the channel modes to the other one. Therefore, each server also receives these channel modes. There are three types of modes for a given channel: flags, masks, and data. The first two types are easy to deal with as they are either set or unset. If such a mode is set on one server, it MUST be set on the other server as a result of the connection.

As topics are not sent as part of this exchange, they are not a problem. However, channel modes 'l' and 'k' are exchanged, and if they are set on both servers prior to the connection, there is no mechanism to decide which of the two values takes precedence. It is left up to the users to fix the resulting discrepancy.

6.4 Resource Exhaustion

The mode based on masks defined in section 4.3 make the IRC servers (and network) vulnerable to a simple abuse of the system: a single channel operator can set as many different masks as possible on a particular channel. This can easily cause the server to waste memory, as well as network bandwidth (since the info is propagated to other servers). For this reason it is RECOMMENDED that a limit be put on the number of such masks per channels as mentioned in section 4.3.

Moreover, more complex mechanisms MAY be used to avoid having redundant masks set for the same channel.

7. Security Considerations

7.1 Access Control

One of the main ways to control access to a channel is to use masks which are based on the username and hostname of the user connections. This mechanism can only be efficient and safe if the IRC servers have an accurate way of authenticating user connections, and if users cannot easily get around it. While it is in theory possible to implement such a strict authentication mechanism, most IRC networks (especially public networks) do not have anything like this in place and provide little guaranty about the accuracy of the username and hostname for a particular client connection.

Another way to control access is to use a channel key, but since this key is sent in plaintext, it is vulnerable to traditional man in the middle attacks.

7.2 Channel Privacy

Because channel collisions are treated as inclusive events (See Section 6.3), it is possible for users to join a channel overriding its access control settings. This method has long been used by individuals to "take over" channels by "illegitimately" gaining channel operator status on the channel. The same method can be used to find out the exact list of members of a channel, as well as to eventually receive some of the messages sent to the channel.

7.3 Anonymity

The anonymous channel flag (See Section 4.2.1) can be used to render all users on such channel "anonymous" by presenting all messages to the channel as originating from a pseudo user which nickname is "anonymous". This is done at the client-server level, and no anonymity is provided at the server-server level.

It should be obvious to readers, that the level of anonymity offered is quite poor and insecure, and that clients SHOULD display strong warnings for users joining such channels.

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://coombs.anu.edu.au/pub/irc

Newsgroup: alt.irc

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.

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

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

[IRC-CLIENT] Kalt, C., "Internet Relay Chat: Client Protocol", RFC 2812, 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

Acknowledgement

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