

Channel Types

IRC has various types of channels that act in different ways. What differentiates these channels is the character the channel name starts with. For instance, channels starting with # are regular channels, and channels starting with & are local channels.

Upon joining, clients are shown which types of channels the server supports with the *CHANTYPES* parameter.

Here, we go through the different types of channels that exist and are widely-used these days.

Regular Channels (#)

The prefix character for this type of channel is ('#' , 0x23).

This channel is what's referred to as a normal channel. Clients can join this channel, and the first client who joins a normal channel is made a *channel operator*, along with the appropriate channel membership prefix. On most servers, newly-created channels have then *protected topic "+t"* and *no external messages "+n"* modes enabled, but exactly what modes new channels are given is up to the server.

Regular channels are persisted across the network. If two clients on different servers join the same regular channel, they'll be able to see that each other are joined, and will see messages sent to the channel by the other client.

On servers that support the concept of 'channel ownership' (a client being able to own a channel and retain control of it with their account), clients may not receive channel operator privileges on joining an otherwise empty channel.

Local Channels (&)

The prefix character for this type of channel is ('&' , 0x26).

This channel is what's referred to as a local channel. Clients can join this channel as normal, and the first client who joins a normal channel is made a *channel operator*, but the channel is not persisted across the network. In other words, each server has its own set of local channels that the other servers on the network don't see.

If a client on server A and a client on server B join the channel &info, they will not be able to see each other or the messages each posts to their server's local channel &info. However, if a client on server A and another client on server A join the channel &info, they will be able to see each other and the messages the other posts to that local channel.

Generally, the concept of channel ownership is not supported for local channels. Local channels also aren't as widely available as regular channels. As well, some networks disable or disallow local channels as ops across the network can't see nor administrate them.

Modes

Modes affect the behaviour and reflect details about targets – clients and channels. The modes listed here are the ones that have been adopted and are used by the IRC community at large. If we say a mode is ‘standard’, that means it is defined in the official IRC specification documents.

The status and letter used for each mode is defined in the description of that mode.

We only cover modes that are widely-used by IRC software today and whose meanings should stay consistent between different server software. For more extensive lists (including conflicting and obsolete modes), see the external `irc-defs` [client](#) and [channel](#) mode lists.

User Modes

Invisible User Mode

This mode is standard, and the mode letter used for it is "+i".

If a user is set to ‘invisible’, they will not show up in commands such as [WHO](#) or [NAMES](#) unless they share a channel with the user that submitted the command. In addition, some servers hide all channels from the [WHOIS](#) reply of an invisible user they do not share with the user that submitted the command.

Oper User Mode

This mode is standard, and the mode letter used for it is "+o".

If a user has this mode, this indicates that they are a network [operator](#).

Local Oper User Mode

This mode is standard, and the mode letter used for it is "+0".

If a user has this mode, this indicates that they are a server [operator](#). A local operator has [operator](#) privileges for their server, and not for the rest of the network.

Registered User Mode

This mode is widely-used, and the mode letter used for it is typically "+r". The character used for this mode, and whether it exists at all, may vary depending on server software and configuration.

If a user has this mode, this indicates that they have logged into a user account.

IRCV3 extensions such as [account-notify](#), [account-tag](#), and [extended-join](#) provide the account name of logged-in users, and are more accurate than trying to detect this user mode due to the capability name remaining consistent.

WALLOPS User Mode

This mode is standard, and the mode letter used for it is "+w".

If a user has this mode, this indicates that they will receive [WALLOPS](#) messages from the server.

Channel Modes

Ban Channel Mode

This mode is standard, and the mode letter used for it is "+b".

This channel mode controls a list of client masks that are 'banned' from joining or speaking in the channel. If this mode has values, each of these values should be a client mask.

If this mode is set on a channel, and a client sends a JOIN request for this channel, their nickmask (the combination of `nick!user@host`) is compared with each banned client mask set with this mode. If they match one of these banned masks, they will receive an [ERR_BANNEDFROMCHAN](#) (474) reply and the JOIN command will fail. See the [ban exception](#) mode for more details.

Exception Channel Mode

This mode is used in almost all IRC software today. The standard mode letter used for it is "+e", but it SHOULD be defined in the [EXCEPTS](#) RPL_ISUPPORT parameter on connection.

This channel mode controls a list of client masks that are exempt from the '[ban](#)' channel mode. If this mode has values, each of these values should be a client mask.

If this mode is set on a channel, and a client sends a JOIN request for this channel, their nickmask is compared with each 'exempted' client mask. If their nickmask matches any one of the masks set by this mode, and their nickmask also matches any one of the masks set by the [ban](#) channel mode, they will not be blocked from joining due to the [ban](#) mode.

Client Limit Channel Mode

This mode is standard, and the mode letter used for it is "+l".

This channel mode controls whether new users may join based on the number of users who already exist in the channel. If this mode is set, its value is an integer and defines the limit of how many clients may be joined to the channel.

If this mode is set on a channel, and the number of users joined to that channel matches or exceeds the value of this mode, new users cannot join that channel. If a client sends a JOIN request for this channel, they will receive an [ERR_CHANNELISFULL](#) (471) reply and the command will fail.

Invite-Only Channel Mode

This mode is standard, and the mode letter used for it is "+i".

This channel mode controls whether new users need to be invited to the channel before being able to join.

If this mode is set on a channel, a user must have received an [INVITE](#) for this channel before being allowed to join it. If they have not received an invite, they will receive an [ERR_INVITEONLYCHAN](#) (473) reply and the command will fail.

Invite-Exception Channel Mode

This mode is used in almost all IRC software today. The standard mode letter used for it is "+I", but it SHOULD be defined in the [INVEX](#) RPL_ISUPPORT parameter on connection.

This channel mode controls a list of channel masks that are exempt from the *invite-only* channel mode. If this mode has values, each of these values should be a client mask.

If this mode is set on a channel, and a client sends a JOIN request for that channel, their nickmask is compared with each 'exempted' client mask. If their nickmask matches any one of the masks set by this mode, and the channel is in *invite-only* mode, they do not need to require an INVITE in order to join the channel.

Key Channel Mode

This mode is standard, and the mode letter used for it is "+k".

This mode letter sets a 'key' that must be supplied in order to join this channel. If this mode is set, its value is the key that is required. Servers may validate the value (eg. to forbid spaces, as they make it harder to use the key in JOIN messages). If the value is invalid, they SHOULD return *ERR_INVALIDMODEPARAM*. However, clients MUST be able to handle any of the following:

- *ERR_INVALIDMODEPARAM*
- *ERR_INVALIDKEY*
- MODE echoed with a different key (eg. truncated or stripped of invalid characters)
- the key changed ignored, and no MODE echoed if no other mode change was valid.

If this mode is set on a channel, and a client sends a JOIN request for that channel, they must supply <key> in order for the command to succeed. If they do not supply a <key>, or the key they supply does not match the value of this mode, they will receive an *ERR_BADCHANNELKEY* (475) reply and the command will fail.

Moderated Channel Mode

This mode is standard, and the mode letter used for it is "+m".

This channel mode controls whether users may freely talk on the channel, and does not have any value.

If this mode is set on a channel, only users who have channel privileges may send messages to that channel. The *voice* channel mode is designed to let a user talk in a moderated channel without giving them other channel moderation abilities, and users of higher privileges (such as *halfops* or *chanops*) may also speak in moderated channels.

Secret Channel Mode

This mode is standard, and the mode letter used for it is "+s".

This channel mode controls whether the channel is 'secret', and does not have any value.

A channel that is set to secret will not show up in responses to the *LIST* or *NAMES* command unless the client sending the command is joined to the channel. Likewise, secret channels will not show up in the *RPL_WHOSCHANNELS* (319) numeric unless the user the numeric is being sent to is joined to that channel.

Protected Topic Mode

This mode is standard, and the mode letter used for it is "+t".

This channel mode controls whether channel privileges are required to set the topic, and does not have any value.

If this mode is enabled, users must have channel privileges such as *halfop* or *operator* status in order to change the topic of a channel. In a channel that does not have this mode enabled, anyone may set the topic of the channel using the *TOPIC* command.

No External Messages Mode

This mode is standard, and the mode letter used for it is "+n".

This channel mode controls whether users who are not joined to the channel can send messages to it, and does not have any value.

If this mode is enabled, users MUST be joined to the channel in order to send *private messages* and *notices* to the channel. If this mode is enabled and they try to send one of these to a channel they are not joined to, they will receive an *ERR_CANNOTSENDOCHAN* (404) numeric and the message will not be sent to that channel.

Channel Membership Prefixes

Users joined to a channel may get certain privileges or status in that channel based on channel modes given to them. These users are given prefixes before their nickname whenever it is associated with a channel (ie, in *NAMES*, *WHO* and *WHOIS* messages). The standard and common prefixes are listed here, and MUST be advertised by the server in the *PREFIX* RPL_ISUPPORT parameter on connection.

Founder Prefix

This mode is used in a large number of networks. The prefix and mode letter typically used for it, respectively, are "~" and "+q".

This prefix shows that the given user is the 'founder' of the current channel and has full moderation control over it – ie, they are considered to 'own' that channel by the network. This prefix is typically only used on networks that have the concept of client accounts, and ownership of channels by those accounts.

Protected Prefix

This mode is used in a large number of networks. The prefix and mode letter typically used for it, respectively, are "&" and "+a".

Users with this mode cannot be kicked and cannot have this mode removed by other protected users. In some software, they may perform actions that operators can, but at a higher privilege level than operators. This prefix is typically only used on networks that have the concept of client accounts, and ownership of channels by those accounts.

Operator Prefix

This mode is standard. The prefix and mode letter used for it, respectively, are "@" and "+o".

Users with this mode may perform channel moderation tasks such as kicking users, applying channel modes, and set other users to operator (or lower) status.

Halfop Prefix

This mode is widely used in networks today. The prefix and mode letter used for it, respectively, are "%" and "+h".

Users with this mode may perform channel moderation tasks, but at a lower privilege level than operators. Which channel moderation tasks they can and cannot perform varies with server software and configuration.

Voice Prefix

This mode is standard. The prefix and mode letter used for it, respectively, are "+" and "+v".

Users with this mode may send messages to a channel that is *moderated*.