ChatDrive IRC Protocol 1.0 *(Rough Draft)*

Table Of Contents

1. Introduction

This article describes ChatDrive IRC protocol 1.0 introduced by ChatDrive IRC Server, which is based on the IRCX protocol.

2. User Access Levels

- **Super Admin:** Super admin has full access to the server. A server can only have one super admin account which can be only accessed from the computer running the server.
- **Admins or Operators:** Admins or operators oversee and control the chat server. Their rights can be set on individual account basis by the super admin.
- **Channel Super Owner:** Channel super owner is assigned to the user who created the room by the server and a channel can have only one super owner. Super owner has full access to the room depending on the server configuration and can change the access rights of owners, hosts, and half-ops.
- **Channel Owner:** Same as IRCX.
  Channel Host: Same as IRCX.
- **Channel Half-Op:** Channel half-op is the help operator of the channel, and can kick, but s/he can't ban or change modes.
- **Channel Member:** Same as IRCX.
- **Chat User:** Same as IRCX.

3. Chat Room Properties

(3.a) Syntax

PROP (command)

Add, change or delete a channel data property.

To query a property:
Syntax 1: PROP <channel> <property>[,<property>]
Syntax 2: PROP <channel> list

To set or change a property:
Syntax 3: PROP <channel> <property> :<data>

To delete a property:
Syntax 4: PROP <channel> <property> :
Syntax 5: PROP <channel> <property> delete

(3.b) Properties

OID (R/O) [see ircx draft]
NAME (R/O) [see ircx draft]
CREATION (R/O) [see ircx draft]
LANGUAGE [see ircx draft]
OWNERKEY [see ircx draft]
HOSTKEY [see ircx draft]
MEMBERKEY [see ircx draft]
PICS [see ircx draft]
TOPIC [see ircx draft]
SUBJECT [see ircx draft]
ONJOIN [see ircx draft]
ONPART [see ircx draft]
LOCK: Locks a property (available to super owner only)
UNLOCK: Unlocks a property (available to super owner only)

(3.c) Results

IRCRPL_PROPLIST
IRCRPL_PROPEND

(3.d) Possible Errors

IRCERR_BADCOMMAND
IRCERR_BADPROPERTY
IRCERR_SECURITY
IRCERR_NOSUCHOBJECT
IRCERR_TOOMANYARGUMENTS
IRCERR_BADVALUE

4. Chat Room Modes

(4.a) Syntax (User)

MODE %#<room name> {+ | -}<room modes

For example, to set the modes: MODE %#MyChatRoom +mftsupiwWa
And to unset the modes: MODE %#MyChatRoom –mftsupiwWa

(4.b) Server Reply Format

:{<nick>!<ident>@<{domain | IP | mask}> | <server name>} MODE %#<room name> {+ | -}<room modes>

(4.c) List

PUBLIC (RFC1459 default)
Same as in IRCX and RFC1459.

PRIVATE (RFC1459 *[Changed]* +p)
The channel is private and is not listed, but the room can be queried only by the exact name.
This mode may be set by operators of the chat room.

PRIVATE (RFC1459 *[Changed]* +p)
The channel is private and is not listed, but the room can be queried only by the exact name.
This mode may be set by operators of the chat room.

SECRET (RFC1459 *[Changed]* +s)
The channel is private, not listed, and cannot be queried.
This mode may be set by operators of the chat room.

MODERATED (RFC1459 +m)
Same as in IRCX and RFC1459.

TOPICOP (RFC1459 +t)
Same as in IRCX and RFC1459.

KNOCK (IRCX +u)
Same as in IRCX.

NOFORMAT (IRCX +f)
Same as in IRCX.

NOWHISPER (IRCX +w)
Same as in IRCX.

NOWHISPER [Guests Only] (ChatDrive01 +W)
This mode is same as ‘+w’ (NOWHISPER) but only applied to Guests.

AUDITORIUM (IRCX +x)
Same as in IRCX.

REGISTERED (IRCX +r)
Same as in IRCX.

AUTHONLY (IRCX +a)
Same as in IRCX.

CLONEABLE (IRCX +d)
Same as in IRCX.

CLONE (IRCX +e)
Same as in IRCX.

(4.d) Chat room modes discarded
- HIDDEN (IRCX +h)
- NOEXTERN (RFC1459 +n): ChatDrive server does not allow users to send messages outside a channel. Only authorized server operators can send messages directly to users [subject to server configuration].

5. User Chat Room Modes

(5.a.i) Super Owner (ChatDrive01 +Q [Represented by: '])
The super owner mode indicates that the user is the creator of the chat room. Only the chat room creator can be the super owner; however, creator of the chat room can turn this mode on and off for him/herself.

Users see super owner mode nicknames with a "" prefix. Super owner, by default, has full control of the chat room [the level of control depends on server configuration]. Super owners can set, change, and lock all the properties of the chat room; and powers of owners, hosts and half-ops.

Syntax:
SUPEROP <chatroom> {ON | OFF}

Server Reply *(same as reply to other "moderator" modes)*:
:<server name> MODE <room name> {+ | -}Q <nick>

(5.a.ii) Owner (IRCX +q [.])
Works same as in IRCX.

(5.a.iii) Host (RFC1459 +o [@])
Works same as in IRCX.

(5.a.iv) Half Op [a.k.a. Help Op] (+h [%])
Help-op can kick and ban, but he/she cannot set the modes and properties of the room.

(5.a.v) Voiced (RFC1459 +v [+])
Works same as in RFC1459.

(5.a.vi) Gag (IRCX +z)
Works same as in IRCX.

(5.b) Results
Rpl_ChannelModeIs
Rpl_UniqOpIs

(5.c) Possible Errors

Err_NeedMoreParams	Err_Keyset
Err_NoChanModes	Err_ChanOprivsNeeded
Err_UserNotInChannel	Err_UnknownMode

6. User Personal Modes
Invisible (+i)
Masked (+x)

7. Commands
(7.a) Chat Commands List
(7.a.i) WHISPER
Works same as in IRCX.

Syntax:
WHISPER <room name> <nick-list>*[Note: max 5 users allowed]* :<message>
Server Reply:
: <nick>!<ident>@<{domain | IP | mask}> WHISPER <room name> <nick-list>*[Note: max 5 users allowed]* :<message>

(7.a.ii)PRIVMSG
Works same as in IRCX.

Syntax:
PRIVMSG <room name> :<message> *[Sends the message to the whole room]*
PRIVMSG <nick-list>*[Note: max 5 users allowed]* :<message> *[Sends the message to the target only in PRIVMSG format]*
PRIVMSG <room name> <nick-list>*[Note: max 5 users allowed]* :<message> *[Sends the message to the target only, in windowless WHISPER format]*

Server Reply:
: <nick>!<ident>@<{domain | IP | mask}> PRIVMSG <room name> :<message>
: <nick>!<ident>@<{domain | IP | mask}> PRIVMSG <nick-list> :<message>
: <nick>!<ident>@<{domain | IP | mask}> PRIVMSG <room name> <nick-list> : <message>

(7.a.iii)NOTICE
Works same as in IRCX.

Syntax:
NOTICE <target> :<message>

(7.a.iv) DATA
Works same as in IRCX.

Syntax:
DATA <target> <tag> :<message>

(7.b) Room Commands
(7.b.i) ACCESS
Same as in IRCX

Syntax:
ACCESS <room name> LIST *[Lists all access entries]*
ACCESS <room name> ADD|DELETE <DENY|GRANT|OWNER|HOST|VOICE> <mask> [<timeout> [:<reason>]]
ACCESS <room name> CLEAR [<level>]

Results:
IRCRPL_ACCESSADD
IRCRPL_ACCESSDELETE
IRCRPL_ACCESSSTART
IRCRPL_ACCESSLIST
IRCRPL_ACCESSEND

Possible Errors:
IRCERR_BADLEVEL
IRCERR_DUPACCESS
IRCERR_MISACCESS
IRCERR_TOOMANYACCESSES
IRCERR_TOOMANYARGUMENTS
IRCERR_BADCOMMAND
IRCERR_NOTSUPPORTED
IRCERR_NOACCESS

(7.b.ii) AWAY
Similar to RFC2812.

Syntax:
AWAY [<room name>] :[message] *[Sets the user away; if used with the room name than user is set away in this room only]*
AWAY [<room name>] *[Sets the user un-away]*

Replies:
RPL_UNAWAY
RPL_NOWAWAY

(7.b.iii) CREATE
Same as in IRCX.

Syntax:
CREATE <channel> [<modes> [<modeargs>]]

Results:
CREATE message
JOIN message
Rpl_Topic
Rpl_NamePly
Rpl_EndOfNames

Possible Errors:
IrcErr_ChannelExist
IrcErr_AlreadyOnChannel
Err_NeedMoreParams
Err_InviteOnlyChan
Err_ChannelIsFull
Err_BannedFromChan
Err_BadChannelKey
Err_TooManyChannels
Err_UnknownCommand

(7.b.iv) DATA
Same as 7.a.iv.

(7.b.v) INVITE
Works similar to as in RFC2812.

Syntax:
INVITE <nick> <room name>
Server Reply:
:{<nick>!<ident>@<{domain | IP | mask}> | <server name>} INVITE <nick> <room name>

Results:
Rpl_Inviting Rpl_Away

Possible Errors:
Err_NeedMoreParams Err_NoSuchNick
Err_NotOnChannel Err_UserOnChannel
Err_ChanOprivsNeeded

(7.b.vi) JOIN
Works similar to as in RFC2812.

Syntax:
JOIN <channel> [<key>]

Server Reply:
: <nick>!<ident>@<{domain | IP | mask}> JOIN :<channel>

Results:
RPL_TOPIC

Possible Errors:

ERR_NEEDMOREPARAMS	ERR_BANNEDFROMCHAN
ERR_INVITEONLYCHAN	ERR_BADCHANNELKEY
ERR_CHANNELISFULL	ERR_BADCHANMASK
ERR_NOSUCHCHANNEL	ERR_TOOMANYCHANNELS
ERR_TOOMANYTARGETS	ERR_UNAVAILRESOURCE

(7.b.vii) KICK
Works similar to as in RFC2812.

Syntax:
KICK <room name> <nick> [:<comment>]

Server Reply:
:{<nick>!<ident>@<{domain | IP | mask}> | <server name>} KICK <room name> <nick>

Possible Errors:

ERR_NEEDMOREPARAMS	ERR_NOSUCHCHANNEL
ERR_BADCHANMASK	ERR_CHANOPRIVSNEEDED
ERR_USERNOTINCHANNEL	ERR_NOTONCHANNEL

(7.b.viii) KNOCK
KNOCK message can be sent by a user to inform the channel members that he/she wants to come in. Knock message won't be sent to the channel if the user can join the channel without notifying the channel operators.

Syntax:
KNOCK <channel> [:<message>]

Server Reply:
:<nick>!<ident>@<{domain | IP | mask}> KNOCK <channel> [:<message>]

(7.b.ix) MODE
See chat room modes - section 4.

(7.b.x) NAMES
Works similar to as in RFC2812.

(7.b.xi) NOTICE
Works similar to as in RFC2812.

(7.b.xii)PROP
See chat room properties - section 3.

(7.b.xiii) STATS
<not documented in this draft>

(7.c) <u>User Commands List</u>
(7.c.i)NICK
Works similar to as in RFC2812.

(7.c.ii)USER
Works similar to as in RFC2812.

(7.c.iii)LOGIN
Authenticates a user.

Syntax:
LOGIN <username> <password>
LOGIN guest [lets user in as guest]

(7.c.iv)LOGINH
Same as login command but accepts md5 hash of the password.

Syntax:
LOGINH <username> <md5 hashed password>

(7.c.v)GETHASH
Returns hashed version of user's password.

Syntax:
GETHASH <username>

(7.c.vi)USERHOST
Works same as in RFC2812.

(7.d) Server & Operator Commands List

(7.d.i)KILL

Kills the connection of the user.

Syntax:

KILL <channel>|<nick> [:<message>]

(7.d.ii)KLINE

Bans a mask from the server.

Syntax:

KLINE +|-<mask> :<reason> [minus deletes the mask from ban list]

(7.d.iii)AKLINE

Grants a mask on the server.
Works same as kline.

(7.d.iv)ZLINE

Adds an IP to the ban list.
Works same as kline.

(7.d.v)AZLINE

Grants an IP on the server.
Works same as kline.

(7.d.vi)REHASH

Reloads configuration but the new settings doesn't affect channels created and users joined before the rehash.

(7.d.vii)RELOAD

Reloads the server.

(7.d.viii)OPER

Works same as login for users.

(7.d.ix)OPERH

Works same as loginh for users.