MODE message

```
      Command: MODE
   Parameters: <target> [<modestring> [<mode arguments>...]]
```

The `MODE` command is used to set or remove options (or *modes*) from a given target.

User mode

If `<target>` is a nickname that does not exist on the network, the *`ERR_NOSUCHNICK`* `(401)` numeric is returned. If `<target>` is a different nick than the user who sent the command, the *`ERR_USERSDONTMATCH`* `(502)` numeric is returned.

If `<modestring>` is not given, the *`RPL_UMODEIS`* `(221)` numeric is sent back containing the current modes of the target user.

If `<modestring>` is given, the supplied modes will be applied, and a `MODE` message will be sent to the user containing the changed modes. If one or more modes sent are not implemented on the server, the server MUST apply the modes that are implemented, and then send the *`ERR_UMODEUNKNOWNFLAG`* `(501)` in reply along with the `MODE` message.

Channel mode

If `<target>` is a channel that does not exist on the network, the *`ERR_NOSUCHCHANNEL`* `(403)` numeric is returned.

If `<modestring>` is not given, the *`RPL_CHANNELMODEIS`* `(324)` numeric is returned. Servers MAY choose to hide sensitive information such as channel keys when sending the current modes. Servers MAY also return the *`RPL_CREATIONTIME`* `(329)` numeric following `RPL_CHANNELMODEIS`.

If `<modestring>` is given, the user sending the command MUST have appropriate channel privileges on the target channel to change the modes given. If a user does not have appropriate privileges to change modes on the target channel, the server MUST NOT process the message, and *`ERR_CHANOPRIVSNEEDED`* `(482)` numeric is returned. If the user has permission to change modes on the target, the supplied modes will be applied based on the type of the mode (see below). For type A, B, and C modes, arguments will be sequentially obtained from `<mode arguments>`. If a type B or C mode does not have a parameter when being set, the server MUST ignore that mode. If a type A mode has been sent without an argument, the contents of the list MUST be sent to the user, unless it contains sensitive information the user is not allowed to access. When the server is done processing the modes, a `MODE` command is sent to all members of the channel containing the mode changes. Servers MAY choose to hide sensitive information when sending the mode changes.

`<modestring>` starts with a plus `('+', 0x2B)` or minus `('-', 0x2D)` character, and is made up of the following characters:

- **`'+'`**: Adds the following mode(s).
- **`'-'`**: Removes the following mode(s).
- **`'a-zA-Z'`**: Mode letters, indicating which modes are to be added/removed.

The ABNF representation for <`modestring`> is:

```
  modestring  =  1*( modeset )
  modeset     =  plusminus *( modechar )
  plusminus   =  %x2B / %x2D
                   ; + or -
  modechar    =  ALPHA
```

There are four categories of channel modes, defined as follows:

- **Type A**: Modes that add or remove an address to or from a list. These modes MUST always have a parameter when sent from the server to a client. A client MAY issue this type of mode without an argument to obtain the current contents of the list. The numerics used to retrieve contents of Type A modes depends on the specific mode. Also see the *`EXTBAN`* parameter.

- **Type B**: Modes that change a setting on a channel. These modes MUST always have a parameter.

- **Type C**: Modes that change a setting on a channel. These modes MUST have a parameter when being set, and MUST NOT have a parameter when being unset.

- **Type D**: Modes that change a setting on a channel. These modes MUST NOT have a parameter.

Channel mode letters, along with their types, are defined in the *`CHANMODES`* parameter. User mode letters are always **Type D** modes.

The meaning of standard (and/or well-used) channel and user mode letters can be found in the *Channel Modes* and *User Modes* sections. The meaning of any mode letters not in this list are defined by the server software and configuration.

Type A modes are lists that can be viewed. The method of viewing these lists is not standardised across modes and different numerics are used for each. The specific numerics used for these are outlined here:

- ***Ban List* "+b"**: Ban lists are returned with zero or more *`RPL_BANLIST`* `(367)` numerics, followed by one *`RPL_ENDOFBANLIST`* `(368)` numeric.

- ***Exception List* "+e"**: Exception lists are returned with zero or more *`RPL_EXCEPTLIST`* `(348)` numerics, followed by one *`RPL_ENDOFEXCEPTLIST`* `(349)` numeric.

- ***Invite-Exception List* "+I"**: Invite-exception lists are returned with zero or more *`RPL_INVITELIST`* `(336)` numerics, followed by one *`RPL_ENDOFINVITELIST`* `(337)` numeric.

After the initial `MODE` command is sent to the server, the client receives the above numerics detailing the entries that appear on the given list. Servers MAY choose to restrict the above information to channel operators, or to only those clients who have permissions to change the given list.

Command Examples:

```
    MODE dan +i                     ; Setting the "invisible" user mode on dan.

    MODE #foobar +mb *@127.0.0.1    ; Setting the "moderated" channel mode and
                                    adding the "*@127.0.0.1" mask to the ban
                                    list of the #foobar channel.
```

Message Examples:

```
    :dan!~h@localhost MODE #foobar -bl+i *@192.168.0.1
                                    ; dan unbanned the "*@192.168.0.1" mask,
                                    removed the client limit from, and set the
                                    #foobar channel to invite-only.

    :irc.example.com MODE #foobar +o bunny
                                    ; The irc.example.com server gave channel
                                    operator privileges to bunny on #foobar.
```

Requesting modes for a channel:

```
    MODE #foobar
```

Getting modes for a channel (and channel creation time):

```
    :irc.example.com 324 dan #foobar +nrt
    :irc.example.com 329 dan #foobar 1620807422
```