

Eventi RAW, Toolbars ed Actions

Start:

Abbiamo già visto nei tutorials precedenti gli Alias, gli Eventi e i un po' i popup, adesso dobbiamo vedere altri 2 elementi fondamentali per chi vuole creare degli script altamente personalizzati, cioè gli eventi RAW e le Action.

Eventi RAW:

Per prima cosa bisogna capire cosa siano gli eventi RAW e poi imparare a gestirli.

Quello degli eventi RAW é sicuramente uno dei 'mondi' meno esplorati da chi ha iniziato da poco a scriptare.

Dovete sapere che ogni messaggio proveniente dal server (quindi ogni risposta del server alle nostre richieste, qualunque esse siano, whois, join nei canali, richiesta del topic etc etc) ha un formato di questo tipo:

<risposta>[argomento_1][argomento_2][argomento_n]

Ed ad ogni risposta del server viene sempre associato un numero univoco, detto numero RAW, che ci consente di identificarlo.

Quindi una richiesta di *whois*, ad esempio, avrà un suo numero RAW e con tale numero possiamo intercettarla, analizzarla e farne tutto quello che vogliamo come far eseguire al nostro script determinate azioni in base alle risposte RAW ricevute dal server.

Per prima cosa dovremmo avere una lista degli Eventi RAW e dei loro codici numerici, alla fine di questo tutorial troverete la lista degli eventi presa direttamente dal RFC 1459, così potrete averli a portata di mano (nella lista sono state omesse alcune spiegazioni degli eventi, ma i numeri e gli eventi sono stati riportati **tutti**).

Adesso andiamo un po' a vedere come sfruttarli.

Scegliamo un evento a caso, il whois ad esempio, e una 'vittima' =D (io ho scelto la mia amica 'SuSp|r|a' che sono sicuro non si arrabbierà).

Proviamo a fare un whois ad un nick.

Informazioni WHOIS per SuSp|r|a ricevute da 2night.azzurra.org

SuSp|r|a è SuSp|r|a!~aNiMa@83.225.Azzurra-3164FD86

Nome vero per SuSp|r|a: DoN't OpEn ThiS QuErY

SuSp|r|a è nei canali: @#xxx, #yyy, @#zzzzz, @#www, +#qqqq, @#kkk, @#KvirC.net

Il server di SuSp|r|a è: libero.azzurra.org - Libero Wind S.p.A.

Le informazioni di SuSp|r|a sono: has identified for this nick

Informazioni WHOIS per SuSp|r|a ricevute da 2night.azzurra.org

Bene questo è la risposta al whois fatto sulla mia amica, adesso se andiamo alla ricerca dei RAW nella lista ne troviamo alcuni molto interessanti e che probabilmente saranno gli interessati di questo risultato.

```
311 RPL_WHOSUSER
    "<nick> <user> <host> * :<real name>"
312 RPL_WHOSSEVER
    "<nick> <server> :<server info>"
313 RPL_WHOSOPERATOR
    "<nick> :is an IRC operator"
317 RPL_WHOSIDLE
    "<nick> <integer> :seconds idle"
318 RPL_ENDOFWHOIS
    "<nick> :End of /WHOIS list"
319 RPL_WHOSCHANNELS
    "<nick> :{[|+]<channel><space>}"
- Replies 311 - 313, 317 - 319 are all replies
generated in response to a WHOIS message. [OMISSIS]
```

Quindi proviamo a prenderne alcuni, per l'esattezza prendiamo il 311,312 il 319 che corrispondo a quei dati che abbiamo visto nel whois, *NomeVero*, *Server* e *Canal* e vediamo come fare a personalizzarci l'output del whois.

Apriamo l'editor degli eventi RAW, dopo facciamo click col tasto destro sulla colonna bianca a sinistra e poi facciamo "Aggiungi Eventi RAW", a questo punto ci chiederà il numero e inseriamo 311 e, una volta creato, modifichiamo il nome del gestore "predefinito" in *mywhois*.

A questo punto dobbiamo aver chiari quali siano i parametri che l'evento 311 ci invia, quindi inseriamo un piccolo script che ci permetta di analizzare i parametri:

```
# Code start:
```

```
echo $k(8)$u() $1
echo $k(4) Completo: $0-
echo Number: $1
if($2=="") return
echo Param1: $2
if($3=="") return
echo Param2: $3
if($4=="") return
echo Param3: $4
if($5=="") return
echo Param4: $5
if($6=="") return
echo Param5: $6
if($7=="") return
echo Param6: $7
if($8=="") return
echo Param8: $8
if($9=="") return
echo Param9: $9
if($10=="") return
echo Param10: $10
```

```
# Code end.
```

Una volta fatto ciò facciamo applica, e riproviamo a fare il *whois*:

311

```
Completo: 2night.azzurra.org 311 Gryf[out] SuSp|r|a ~aNiMa  
83.225.Azzurra-3164FD86 * DoN't OpEn ThiS QuErY
```

Number: 311

Param1: Gryf[out]

Param2: SuSp|r|a

Param3: ~aNiMa

Param4: 83.225.Azzurra-3164FD86

Param5: *

Param6: DoN't OpEn ThiS QuErY

e di seguito ci apparirà l'output del whois normale.

A questo punto sappiamo che il sesto parametro (\$7 dato che partono da 0) è il nome reale, quindi possiamo modificare l'output personalizzandolo mettendo, ad esempio (sempre nel gestore dell'evento raw 311), il codice:

```
# Inserisco l'output che voglio venga generato
```

```
echo $k(4) $u() $3
```

```
echo $k(4,8) In realtà si chiama \:$o $7
```

```
# ovviamente interrompo per evitare che il KVirc
```

```
# generi il suo di output oltre al mio.
```

```
halt
```

Adesso passiamo all'evento 312 e modifichiamo anche qui l'output, possiamo anche farlo utilizzando il comando **event** come abbiamo visto con gli eventi normali invece di aggiungere il codice tramite il raw editor semplicemente dando il comando:

```
/event(312,mywhois){echo $k(4,8) E' connesso al server \:$o $4 -$5;halt;}
```

e facciamo la stessa cosa per l'evento raw 319 (la risposta del server che ci dice su quali canali si trova la persona a cui abbiamo fatto il whois.

```
/event(319,mywhois){echo $k(4,8) Si trova sui canali \:$o $4;halt;}
```

Vogliamo pure che ci metta in bella mostra se la persona a cui stiamo facendo il whois è un IRCOP? Niente di più semplice!

```
/event(313,mywhois){echo $k(4) $b() $3 $b() IS AN IRCOP;}
```

A questo punto possiamo rifare il *whois* e goderci il nostro risultato personalizzato.

^ ^

Come potete notare è abbastanza semplice personalizzarsi i RAW sul KVirc (=) divertitevi un po' prima di passare al prossimo argomento ..le Action!

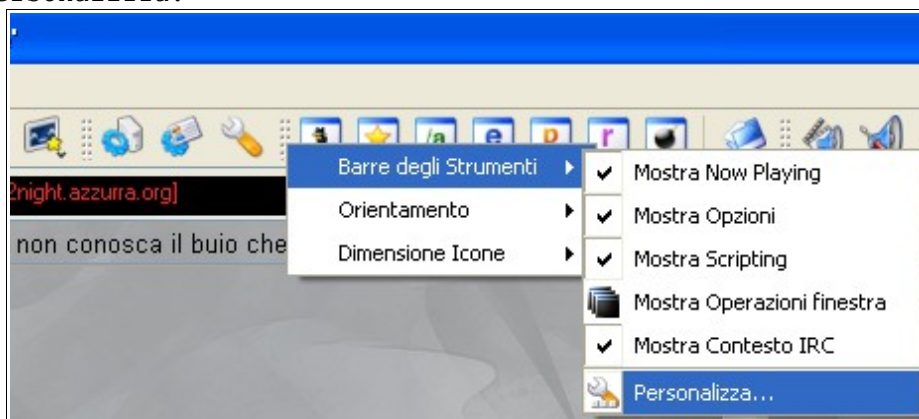
Action e ToolBars:

Una cosa molto comoda è avere a disposizione una barra degli strumenti con le azioni che facciamo più spesso suddivise in pulsanti.

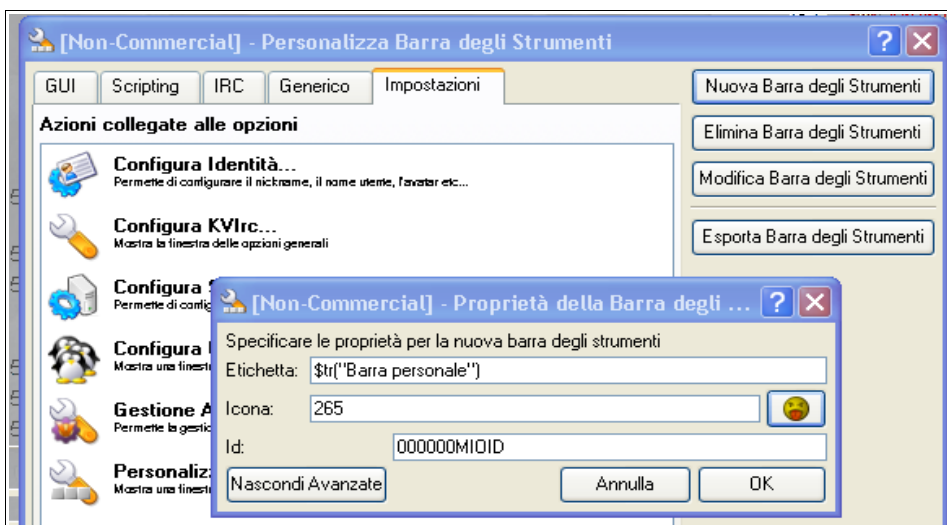
Per fare ciò abbiamo, anche qui, due strade: una grafica ed una tramite comandi, le vedremo tutte e due.

Modalità grafica

Tasto destro sulla barra degli strumenti poi Barre degli strumenti e poi personalizza:



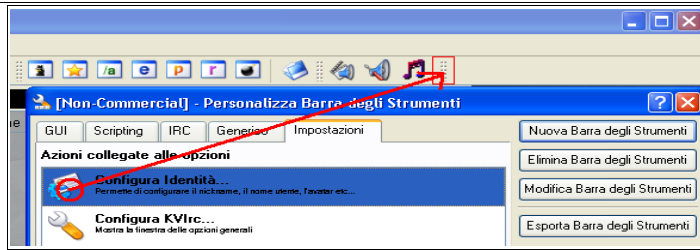
A questo punto apparirà un menù in cui noi potremo creare, modificare o eliminare le barre degli strumenti, creiamone una dandogli un nome, una icona ed un id (che non è necessario ma potrebbe servirci nel caso in cui poi volessimo poter risalire subito alla nostra barra dagli script che creiamo, magari per distruggerla).



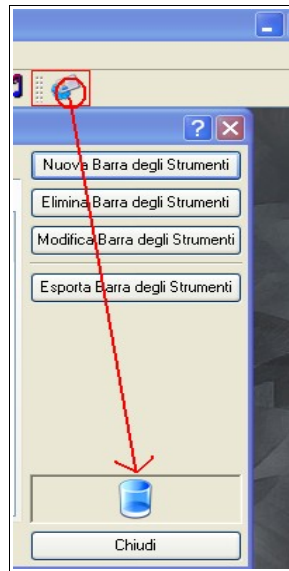
A questo punto vedremo la nostra bella barra degli strumenti apparire come per magia in alto tra le altre, evidenziata di rosso.



Ora non ci resta che trascinare l'oggetto che vogliamo aggiungere, come nella seguente immagine.



Come vedete si sarà aggiunto alla nostra barra.
Invece se volessimo eliminarlo basterebbe trascinarlo dalla barra nel cestino.



A questo punto dobbiamo capire come crearci noi questi 'pulsanti', e per questo ci serve l'Action editor (Icona a forma di stella nella barra dello scripting).

Apriamolo e ci troveremo di fronte ad un menù che potrebbe apparire abbastanza complesso, ma in realtà è molto semplice.

In basso a sinistra abbiamo i tasti per creare la nostra nuova Action, ed alla nostra destra abbiamo delle schede che ci permetteranno di selezionare tutte le caratteristiche che la nostra Action dovrà avere, l'icona, la categoria, dove e quando il pulsante della nostra Action dovrà o potrà essere visibile o abilitato, la categoria etc etc.

Facciamo subito una prova creandone una.

Clicchiamo su **"Nuova azione"**, appena apparirà nell'elenco delle azioni, diamole un *nome* ed un *etichetta* ad esempio *"JoinCanali"*

Poi inseriamo il codice che dovrà essere eseguito una volta che l'azione verrà attivata; ad esempio potremmo mettere:

```
# Code start:  
join #kvirc.net  
join #canale1  
join #canale2  
join #canale3  
join #canale4  
# Code end.
```

Così quando l'azione verrà eseguita (o cliccheremo sul nostro pulsante) il nostro KVIrc entrerà automaticamente nei nostri canali preferiti.

A questo punto andiamo sulla scheda **"Proprietà"** e assegniamogli un *icona* una breve *descrizione* e, se vogliamo, una sequenza di *tasti rapidi* e come *categoria* possiamo lasciare "Generico".

Se vogliamo possiamo anche farci un giretto nella sezione **"Opzioni"** e settare qualche altra cosa particolare, ma questo dipende dalle vostre esigenze =).

Applichiamo e usciamo.

A questo punto possiamo metterla nella nostra toolbar col metodo che abbiamo visto sopra, infatti, adesso, la nostra Azione comparirà tra le altre nella scheda **"Generale"** della personalizzazione delle barre degli strumenti e quindi potremo trascinarla, come abbiamo visto sopra, nella nostra toolbar.

Quindi quando andremo a premere il pulsante della nostra action, sulla nostra barra, verrà eseguito il codice che abbiamo scritto e, nel nostro caso, entreremo nei nostri canali preferiti.

Potremmo anche eseguire l'azione che abbiamo creato, direttamente, senza dover premere il pulsante, infatti basterà dare il comando

```
/action.trigger <nome azione>
```

quasi come fosse un alias.

In pratica (se mi permettete un paragone alla larga) le actions sono come degli alias molto più complessi e potenti che possono fare anche da pulsanti.

Cominciate a intuirne la potenza vero? =)

Per farvi un esempio io mi sono creato un alias **"at"** che non fa altro che eseguire il comando **"action.trigger \$0-"**, così se voglio eseguire ad esempio la mia Action **"joinall"** basta che do il comando **"/at joinall"** e la mia action viene eseguita =D comodo no? Pulsante ed alias tutto insieme!

Adesso dobbiamo fare un giretto nella creazione e manipolazione manuale (senza interfaccia) delle toolbars e delle Actions, questo perché i nostri script devono poter vivere di vita autonoma, riducendo al minimo la necessità di interagire con l'utente, i nostri script devono poter creare, mostrare e distruggere toolbars senza che l'utente debba fare "tasto destro sulla barra degli strumenti, mostra barra etc etc".

Vediamo quali comandi il buon KVIrc ci mette a disposizione per fare tutto ciò.

Se andiamo a vedere il nostro insostituibile file di aiuto del KVIrc nella sezione *comandi*, alla lettera "t" troveremo:

```
toolbar.create [-p] <id> <label> [;icon_id]
```

Come potete vedere, quindi per creare rapidamente una toolbar basta dare il comando

```
toolbar.create mytoolbarname Barra personale
```

dove **"mytoolbarname"** sarà l'identificativo della nostra toolbar.

Facciamo subito una prova dando in sequenza questi comandi:

```
/toolbar.create mytoolbarname Barra personale
```

La barra così verrà creata, ovviamente non la vedete.

```
/toolbar.show mytoolbarname
```

La barra così, se guardate bene, apparirà tra le altre.

```
/toolbar.hide mytoolbarname
```

La barra così verrà nascosta ma continuerà ad esistere.

```
/toolbar.destroy mytoolbarname
```

La barra così verrà distrutta.

Poi notiamo ancora che esistono i comandi

```
/toolbar.addItem <toolbarid> <action>
```

```
/toolbar.removeItem <toolbarid> <action>
/toolbar.clear <toolbarid> <action>
```

Che come intuirete servono ad aggiungere una Action, togliere una Action e rimuovere tutte le Action che abbiamo aggiunto alla nostra toolbar.

A questo punto ci andiamo a chiedere ..ma come possiamo creare una Action tramite comandi?

A rispondere è sempre il nostro file di help e questa volta, come sicuramente immaginerete la sintassi è abbastanza complessa =P

```
action.create [-i] [-c] [-l] [-w=<window_types>] [-s] [-t=<category>]
(<name>,<visible
label:string>,<description>,<big_icon_id>[,<small_icon_id>])
{
    <action body>
}
```

Per la spiegazione di tutti i flag vi rimando direttamente alla guida ufficiale del KVIrc, perché io non potrei fare altro che tradurlo alla lettera =P.

Qui vi elencherò, i comandi principali che troviamo realizzabili sulle Action.

action.destroy [-q] <name>

Per distruggere una action

action.disable [-q] [-f] <name>

Per disabilitarla

action.enable [-q] <name>

Per abilitarla

action.list

Per ottenere la lista di tutte le Actions.

A questo punto credo che anche con questo argomento abbiamo finito =D ora sta solo a voi prendere padronanza con questi comandi.

Digitiamo felicemente il nostro amato...

/ECHO STOP

```
-----
" Tu vedi cose e ne spieghi il perché, io invece immagino cose che
non sono mai esistite e mi chiedo perché no." (George Bernad Shaw)
-----
```

Grifisx

Allegato:

6. REPLIES

The following is a list of numeric replies which are generated in response to the commands given above. Each numeric is given with its number, name and reply string.

6.1 Error Replies.

```
401     ERR_NOSUCHNICK
        "<nickname> :No such nick/channel"
        - Used to indicate the nickname parameter supplied to a
          command is currently unused.
402     ERR_NOSUCHSERVER
        "<server name> :No such server"
        - Used to indicate the server name given currently
          doesn't exist.
```

403 ERR_NOSUCHCHANNEL
 "<channel name> :No such channel"
 - Used to indicate the given channel name is invalid.

404 ERR_CANNOTSENDOCHAN
 "<channel name> :Cannot send to channel"
 - Sent to a user who is either (a) not on a channel which is mode +n or (b) not a chanop (or mode +v) on a channel which has mode +m set and is trying to send a PRIVMSG message to that channel.

405 ERR_TOOMANYCHANNELS
 "<channel name> :You have joined too many \\
 channels"
 - Sent to a user when they have joined the maximum number of allowed channels and they try to join another channel.

406 ERR_WASNOSUCHNICK
 "<nickname> :There was no such nickname"
 - Returned by WHOWAS to indicate there is no history information for that nickname.

407 ERR_TOOMANYTARGETS
 "<target> :Duplicate recipients. No message \\
 delivered"
 - Returned to a client which is attempting to send a PRIVMSG/NOTICE using the user@host destination format and for a user@host which has several occurrences.

409 ERR_NOORIGIN
 ":No origin specified"
 - PING or PONG message missing the originator parameter which is required since these commands must work without valid prefixes.

411 ERR_NORECIPIENT
 ":No recipient given (<command>)"

412 ERR_NOTEXTTOSEND
 ":No text to send"

413 ERR_NOTOPLEVEL
 "<mask> :No toplevel domain specified"

414 ERR_WILDTOPLEVEL
 "<mask> :Wildcard in toplevel domain"
 - 412 - 414 are returned by PRIVMSG to indicate that the message wasn't delivered for some reason.[OMISSIS]

421 ERR_UNKNOWNCOMMAND
 "<command> :Unknown command"
 - Returned to a registered client to indicate that the command sent is unknown by the server.

422 ERR_NOMOTD
 ":MOTD File is missing"
 - Server's MOTD file could not be opened by the server.

423 ERR_NOADMININFO
 "<server> :No administrative info available"
 - Returned by a server in response to an ADMIN message when there is an error in finding the appropriate information.

424 ERR_FILEERROR
 ":File error doing <file op> on <file>"
 - Generic error message used to report a failed file operation during the processing of a message.

431 ERR_NONICKNAMEGIVEN
 ":No nickname given"
 - Returned when a nickname parameter expected for a command and isn't found.

432 ERR_ERRONEUSNICKNAME
 "<nick> :Erroneus nickname"
 - Returned after receiving a NICK message which contains characters which do not fall in the defined set. See section x.x.x for details on valid nicknames.

433 ERR_NICKNAMEINUSE
 "<nick> :Nickname is already in use"
 - Returned when a NICK message is processed that results in an attempt to change to a currently existing nickname.

436 ERR_NICKCOLLISION
 "<nick> :Nickname collision KILL"

- Returned by a server to a client when it detects a nickname collision (registered of a NICK that already exists by another server).

441 ERR_USERNOTINCHANNEL
 "<nick> <channel> :They aren't on that channel"

- Returned by the server to indicate that the target user of the command is not on the given channel.

442 ERR_NOTONCHANNEL
 "<channel> :You're not on that channel"

- Returned by the server whenever a client tries to perform a channel effecting command for which the client isn't a member.

443 ERR_USERONCHANNEL
 "<user> <channel> :is already on channel"

- Returned when a client tries to invite a user to a channel they are already on.

444 ERR_NOLOGIN
 "<user> :User not logged in"

- Returned by the summon after a SUMMON command for a user was unable to be performed since they were not logged in.

445 ERR_SUMMONDISABLED
 ":SUMMON has been disabled"

- Returned as a response to the SUMMON command. Must be returned by any server which does not implement it.

446 ERR_USERSDISABLED
 ":USERS has been disabled"

- Returned as a response to the USERS command. Must be returned by any server which does not implement it.

451 ERR_NOTREGISTERED
 ":You have not registered"

- Returned by the server to indicate that the client must be registered before the server will allow it to be parsed in detail.

461 ERR_NEEDMOREPARAMS
 "<command> :Not enough parameters"

- Returned by the server by numerous commands to indicate to the client that it didn't supply enough parameters.

462 ERR_ALREADYREGISTRED
 ":You may not reregister"

- Returned by the server to any link which tries to change part of the registered details (such as password or user details from second USER message).

463 ERR_NOPERMFORHOST
 ":Your host isn't among the privileged"

- Returned to a client which attempts to register with a server which does not been setup to allow connections from the host the attempted connection is tried.

464 ERR_PASSWDMISMATCH
 ":Password incorrect"

- Returned to indicate a failed attempt at registering a connection for which a password was required and was either not given or incorrect.

465 ERR_YOUREBANNEDCREEP
 ":You are banned from this server"

- Returned after an attempt to connect and register yourself with a server which has been setup to explicitly deny connections to you.

467 ERR_KEYSET
 "<channel> :Channel key already set"

471 ERR_CHANNELISFULL
 "<channel> :Cannot join channel (+l)"

472 ERR_UNKNOWNMODE
 "<char> :is unknown mode char to me"

473 ERR_INVITEONLYCHAN
 "<channel> :Cannot join channel (+i)"

474 ERR_BANNEDFROMCHAN
 "<channel> :Cannot join channel (+b)"

475 ERR_BADCHANNELKEY
 "<channel> :Cannot join channel (+k)"

481 ERR_NOPRIVILEGES

```

        ":Permission Denied- You're not an IRC operator"
- Any command requiring operator privileges to operate
  must return this error to indicate the attempt was
  unsuccessful.
482 ERR_CHANOPRIVSNEEDED
        "<channel> :You're not channel operator"
- Any command requiring 'chanop' privileges (such as
  MODE messages) must return this error if the client
  making the attempt is not a chanop on the specified
  channel.
483 ERR_CANTKILLSERVER
        ":You cant kill a server!"
- Any attempts to use the KILL command on a server
  are to be refused and this error returned directly
  to the client.
491 ERR_NOOPERHOST
        ":No O-lines for your host"
- If a client sends an OPER message and the server has
  not been configured to allow connections from the
  client's host as an operator, this error must be
  returned.
501 ERR_UMODEUNKNOWNFLAG
        ":Unknown MODE flag"
- Returned by the server to indicate that a MODE
  message was sent with a nickname parameter and that
  the a mode flag sent was not recognized.
502 ERR_USERSDONTMATCH
        ":Cant change mode for other users"
- Error sent to any user trying to view or change the
  user mode for a user other than themselves.

```

6.2 Command responses.

```

300 RPL_NONE
        Dummy reply number. Not used.
302 RPL_USERHOST
        ":[<reply>{<space><reply>}]"
- Reply format used by USERHOST to list replies to
  the query list. The reply string is composed as
  follows:
  <reply> ::= <nick>['*'] '=' <'+'|'-><hostname>
  The '*' indicates whether the client has registered
  as an Operator. The '-' or '+' characters represent
  whether the client has set an AWAY message or not
  respectively.
303 RPL_ISON
        ":[<nick> {<space><nick>}]"
- Reply format used by ISON to list replies to the
  query list.
301 RPL_AWAY
        "<nick> :<away message>"
305 RPL_UNAWAY
        ":You are no longer marked as being away"
306 RPL_NOWAWAY
        ":You have been marked as being away"
- These replies are used with the AWAY command (if
  allowed). [OMISSIS]
311 RPL_WHOSUSER
        "<nick> <user> <host> * :<real name>"
312 RPL_WHOSSEVER
        "<nick> <server> :<server info>"
313 RPL_WHOSOPERATOR
        "<nick> :is an IRC operator"
317 RPL_WHOSIDLE
        "<nick> <integer> :seconds idle"
318 RPL_ENDOFWHOSIS
        "<nick> :End of /WHOSIS list"
319 RPL_WHOSCHANNELS
        "<nick> :{[+]<channel><space>}"
- Replies 311 - 313, 317 - 319 are all replies
  generated in response to a WHOIS message. [OMISSIS]
314 RPL_WHOWASUSER
        "<nick> <user> <host> * :<real name>"
369 RPL_ENDOFWHOWAS
        "<nick> :End of WHOWAS"

```

```

- When replying to a WHOWAS message, a server must use
  the replies RPL_WHOWASUSER, RPL_WHOISSEVER or
  ERR_WASNOSUCHNICK for each nickname in the presented
  list. [OMISSIS]
321 RPL_LISTSTART
    "Channel :Users Name"
322 RPL_LIST
    "<channel> <# visible> :<topic>"
323 RPL_LISTEND
    ":End of /LIST"
- Replies RPL_LISTSTART, RPL_LIST, RPL_LISTEND mark[OMISSIS]
324 RPL_CHANNELMODEIS
    "<channel> <mode> <mode params>"
331 RPL_NOTOPIC
    "<channel> :No topic is set"
332 RPL_TOPIC
    "<channel> :<topic>"
- When sending a TOPIC message to determine the
  channel topic, one of two replies is sent. If
  the topic is set, RPL_TOPIC is sent back else
  RPL_NOTOPIC.
341 RPL_INVITING
    "<channel> <nick>"
- Returned by the server to indicate that the
  attempted INVITE message was successful and is
  being passed onto the end client.
342 RPL_SUMMONING
    "<user> :Summoning user to IRC"
- Returned by a server answering a SUMMON message to
  indicate that it is summoning that user.
351 RPL_VERSION
    "<version>.<debuglevel> <server> :<comments>"
- Reply by the server showing its version details.[OMISSIS]
352 RPL_WHOREPLY
    "<channel> <user> <host> <server> <nick> \
    <H|G|*|@|+> :<hopcount> <real name>"
315 RPL_ENDOFWHO
    "<name> :End of /WHO list"
- The RPL_WHOREPLY and RPL_ENDOFWHO pair are used
  to answer a WHO message. [OMISSIS]
353 RPL_NAMREPLY
    "<channel> :[[@|+]<nick> [[@|+]<nick> [...]]]"
366 RPL_ENDOFNAMES
    "<channel> :End of /NAMES list"
- To reply to a NAMES message, a reply pair consisting
  of RPL_NAMREPLY and RPL_ENDOFNAMES [OMISSIS]
364 RPL_LINKS
    "<mask> <server> :<hopcount> <server info>"
365 RPL_ENDOFLINKS
    "<mask> :End of /LINKS list"
- In replying to the LINKS message, a server must send
  replies back using the RPL_LINKS numeric and mark the
  end of the list using an RPL_ENDOFLINKS reply.
367 RPL_BANLIST
    "<channel> <banid>"
368 RPL_ENDOFBANLIST
    "<channel> :End of channel ban list"
- When listing the active 'bans' for a given channel,
  a server is required to send the list back using the
  RPL_BANLIST and RPL_ENDOFBANLIST messages. [OMISSIS]
371 RPL_INFO
    " :<string>"
374 RPL_ENDOFINFO
    " :End of /INFO list"
- A server responding to an INFO message is required to
  send all its 'info' in a series of RPL_INFO messages
  with a RPL_ENDOFINFO reply to indicate the end of the
  replies.
375 RPL_MOTDSTART
    " :- <server> Message of the day - "
372 RPL_MOTD
    " :- <text>"
376 RPL_ENDOFMOTD

```

```

        ":End of /MOTD command"
- When responding to the MOTD message and the MOTD file
  is found, the file is displayed line by line, with
  each line no longer than 80 characters, [OMISSIS]
381 RPL_YOUREOPER
        ":You are now an IRC operator"
- RPL_YOUREOPER is sent back to a client which has
  just successfully issued an OPER message and gained
  operator status.
382 RPL_REHASHING
        "<config file> :Rehashing"
- If the REHASH option is used and an operator sends
  a REHASH message, an RPL_REHASHING is sent back to
  the operator.
391 RPL_TIME
        "<server> :<string showing server's local time>"
- When replying to the TIME message, a server must send
  the reply using the RPL_TIME format above. [OMISSIS]

392 RPL_USERSSTART
        ":UserID Terminal Host"
393 RPL_USERS
        ":%-8s %-9s %-8s"
394 RPL_ENDOFUSERS
        ":End of users"
395 RPL_NOUSERS
        ":Nobody logged in"
- If the USERS message is handled by a server, [OMISSIS]
200 RPL_TRACELINK
        "Link <version & debug level> <destination> \
        <next server>"
201 RPL_TRACECONNECTING
        "Try. <class> <server>"
202 RPL_TRACEHANDSHAKE
        "H.S. <class> <server>"
203 RPL_TRACEUNKNOWN
        "???? <class> [<client IP address in dot form>]"
204 RPL_TRACEOPERATOR
        "Oper <class> <nick>"
205 RPL_TRACEUSER
        "User <class> <nick>"
206 RPL_TRACESERVER
        "Serv <class> <int>S <int>C <server> \
        <nick!user!*:*>@<host|server>"
208 RPL_TRACENEWTYPE
        "<newtype> 0 <client name>"
261 RPL_TRACELOG
        "File <logfile> <debug level>"

- The RPL_TRACE* are all returned by the server in
  response to the TRACE message. [OMISSIS]
211 RPL_STATSLINKINFO
        "<linkname> <sendq> <sent messages> \
        <sent bytes> <received messages> \
        <received bytes> <time open>"
212 RPL_STATSCOMMANDS
        "<command> <count>"
213 RPL_STATSCLINE
        "C <host> * <name> <port> <class>"
214 RPL_STATSNLINE
        "N <host> * <name> <port> <class>"
215 RPL_STATSILINE
        "I <host> * <host> <port> <class>"
216 RPL_STATSKLINE
        "K <host> * <username> <port> <class>"
218 RPL_STATSYLINE
        "Y <class> <ping frequency> <connect \
        frequency> <max sendq>"
219 RPL_ENDOFSTATS
        "<stats letter> :End of /STATS report"
241 RPL_STATSLINE
        "L <hostmask> * <servername> <maxdepth>"
242 RPL_STATSUPTIME

```

```

243     ":Server Up %d days %d:%02d:%02d"
RPL_STATSO LINE
244     "O <hostmask> * <name>"
RPL_STATSHLINE
221     "H <hostmask> * <servername>"
RPL_UMODEIS
     "<user mode string>"
     - To answer a query about a client's own mode,
       RPL_UMODEIS is sent back.
251     RPL_LUSERCLIENT
     ":There are <integer> users and <integer> \
       invisible on <integer> servers"
252     RPL_LUSEROP
     "<integer> :operator(s) online"
253     RPL_LUSERUNKNOWN
     "<integer> :unknown connection(s) "
254     RPL_LUSERCHANNELS
     "<integer> :channels formed"
255     RPL_LUSERME
     ":I have <integer> clients and <integer> \
       servers"
     - In processing an LUSERS message, the server
       sends a set of replies from RPL_LUSERCLIENT,
       RPL_LUSEROP, RPL_USERUNKNOWN,
       RPL_LUSERCHANNELS and RPL_LUSERME. [OMISSIS]
256     RPL_ADMINME
     "<server> :Administrative info"
257     RPL_ADMINLOC1
     " :<admin info>"
258     RPL_ADMINLOC2
     " :<admin info>"
259     RPL_ADMINEMAIL
     " :<admin info>"
     - When replying to an ADMIN message, a server
       is expected to use replies RPL_ADMINME
       [OMISSIS]

```

6.3 Reserved numerics.

These numerics are not described above since they fall into one of the following categories:

1. no longer in use;
 2. reserved for future planned use;
 3. in current use but are part of a non-generic 'feature' of the current IRC server.
- | | | | |
|-----|---------------------|-----|-------------------|
| 209 | RPL_TRACECLASS | 217 | RPL_STATSQLINE |
| 231 | RPL_SERVICEINFO | 232 | RPL_ENDOFSERVICES |
| 233 | RPL_SERVICE | 234 | RPL_SERVLIST |
| 235 | RPL_SERVLISTEND | | |
| 316 | RPL_WHOSCHANOP | 361 | RPL_KILLDONE |
| 362 | RPL_CLOSING | 363 | RPL_CLOSEEND |
| 373 | RPL_InfOSTART | 384 | RPL_MYPORTIS |
| 466 | ERR_YOUWILLBEBANNED | 476 | ERR_BADCHANMASK |
| 492 | ERR_NOSERVICEHOST | | |