

# IRC

## Der Internet Relay Chat

# IRC – Internet Relay Chat

1. Geschichte
2. Grundsätzlicher Netzwerkaufbau
3. Kommunikationsmethoden im Netzwerk
4. Das Protokoll selbst
  - 4.1. Grundsätzlicher Nachrichtenaufbau
  - 4.2. Einige Befehle von IRC genannt und erklärt
    - 4.2.1. Verbindungs- und Verwaltungs-Kommandos
    - 4.2.2. Channel-Kommandos
    - 4.2.3. Querybefehle
    - 4.2.4. zum Nachrichtenversand nötige Befehle
  - 4.3. Replies
    - 4.3.1. Fehlerantworten
    - 4.3.2. Kommandoantworten
5. Quellen
6. Fragen?

# Geschichte

- Geburtsort des IRC ist das BITNET
- Vater ein finnischer Student Namens Jarkko Oikarinen
- wurde im Sommer 1988 auf das Internet übertragen

# Geschichte

- Netzwerk wuchs zu unwartbarer Größe weshalb im Jahr 1993 weitere unabhängige Netze entstanden
- 1996 wurde das ursprüngliche Netzwerk aufgrund von Differenzen der Betreiber geteilt.

Heute findet man Teile davon in den Netzen IRCNet und EFNet

# Geschichte

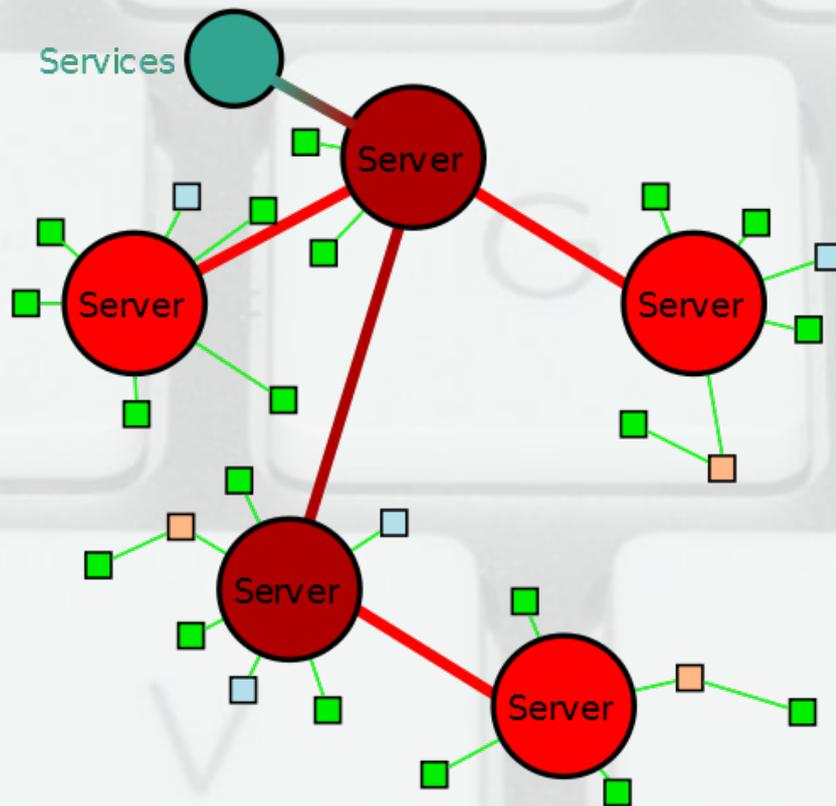
- Heute existieren tausende voneinander unabhängige IRC-Netzwerke

Unter anderem:

QuakeNet, EFNet, IRCNet, Undernet  
(sie gehören zu den größeren und)

Freenode, euIRC, FurNet  
(gehören zu den kleineren IRC-Netzwerken)

# Grundsätzlicher Netzwerkaufbau



Ein IRC Netzwerk besteht aus:

- Servern (rot)
- Services
- Clients (eckig)
  - echte Clients (grün)
  - Bouncer (orange)
  - Bots (bläulich)

# Grundsätzlicher Netzwerkaufbau

Tausende Benutzer schreiben gleichzeitig Nachrichten!

*Wer blickt da noch durch?*

- Möglichkeit der Organisation eines Netzwerks durch sogenannte Channel (früher Groups genannt).
- Channel sind virtuelle Räume innerhalb eines IRC-Netzwerkes, die von den Benutzern betreten werden können.
- Jeder kann einen Channel anlegen und dann über diesen bestimmen.
- Channelnamen beginnen mit # oder &

# Grundsätzlicher Netzwerkaufbau

- Anmeldung am IRC-Netz mit frei wählbarem Benutzernamen (nickname)
- Chefs im Netz sind die „Operatoren“
- Netzwerkoperatoren haben Kontrolle über das gesamte Netzwerk  
inkl. der einzelnen Server.
- Channeloperatoren haben Kontrolle über den eigenen Raum

# Grundsätzlicher Netzwerkaufbau

## Floodcontrol:

Was versteht man darunter?

Massennachrichten im Netzwerk werden durch das Protokoll verhindert.

WIE?

Ein Server merkt sich die Anzahl der Nachrichten innerhalb einer bestimmten Zeitspanne. Wird ein Maximalwert überschritten, trennt ein Server die Verbindung mit einem bestimmten Kommentar.

# Kommunikationsmethoden

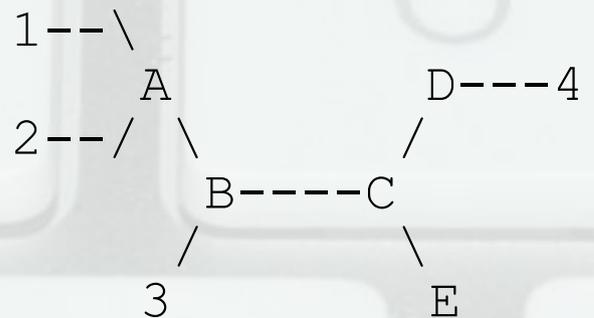
Grundsätzliche Unterscheidung:

1. One to One
2. One to Many
3. One to All

# Kommunikationsmethoden

## 1. One to One

- Aufgrund der asynchronen Netzwerkkommunikation wird eine Nachricht nicht an alle Clients geschickt.



*Server:* A,B,C,D,E  
*Clients:* 1,2,3,4

Beispiel:

Eine Nachricht geht von  
Client 1 an Client 3.

Wer bekommt sie alles zu  
sehen?

# Kommunikationsmethoden

## 2. One to Many

Aufteilung in:

### 2.1. One to a List

Benutzer gibt mehrere Empfänger an, die seine Nachricht erhalten.

### 2.2. One to a Group (Channel)

Benutzer schickt Nachricht an einen Channel. Sie wird von den Servern an alle Benutzer im Channel weitergeleitet.

### 2.3. One to a Host/Servermask

Benutzer kann damit alle User auf einem Server erreichen oder solche, die eine ähnliche Hostmask haben (über Wildcards z. B. \*.rdfnuernberg.de)

# Kommunikationsmethoden

## 3. One to All

- entspricht dem Broadcast in der Netzwerktechnik.
- eine Nachricht wird von einem Client/Server an alle anderen geschickt.
- aus Sicherheitsgründen nur für Operatoren und Server möglich.

# Das Protokoll

- Basiert auf dem TCP/IP Protokoll
- komplett Text-basiert
- früher nur ASCII-Code verwendet
- heute auch andere Zeichensätze möglich z. B. UTF-8
- im Normalfall übernimmt das Client-Programm die Übersetzung in den Protokollsyntax
- im IRC ist es (anders als in anderen Protokollen) üblich, dass der Benutzer direkt in die Kommunikation eingreifen kann

# Das Protokoll

## Aufbau einer Nachricht

Das Protokoll basiert auf Textnachrichten:

Aufbau einer solchen Nachricht:

```
[ ':' <prefix><SPACE> ] <command>  
<parameter><CRLF>
```

**Erklärung zum hier verwendeten Syntax:**

[] markiert optionale Parameter/Angaben

{ } markiert wiederholtes Auftreten des gleichen Parametertyps

<> markiert einen Parametertyp; an dieser Stelle wird anderer Text eingesetzt

| ist eine ODER-Entscheidung ( z. B. +|- == Plus oder Minus)

# Das Protokoll

## Aufbau einer Nachricht

Der Prefix-Aufbau:

```
' : ' <servername> | ' : ' <nick> [ ' ! ' <user> ] [ ' @ ' <host> ]
```

<servername> = der Name des sendenden Servers

Servername wird nur verwendet,  
wenn die Nachricht von einem anderen Server kommt.

<nick> = der IRC-Benutzername

<user> = der Rechner Benutzername

<host> = der Hostname des Client Rechners

' : ' = Zeigt an das ein Prefix vorhanden ist

Ein Prefix wird im Normalfall nur von Servern angehängt.  
Hängt ein Client ein Prefix an, ist nur sein Nickname erlaubt,  
da sonst die Nachricht ignoriert wird.

# Das Protokoll

## Aufbau einer Nachricht

Der Kommandoaufbau:

`<letter>{<letter>} | <number><number><number>`

`<letter>` = ein Buchstabe von a-z oder A-Z

`<number>` = eine Ziffer von 0-9

Beispiel:

JOIN  
KILL  
PART  
405  
413  
446

Ein Parameter hat den Aufbau:

`<letter>{<letter>}`

`<CRLF>` = die Carriage Return & Line Feed  
Kombination zum Abschluss einer  
Nachricht

# Das Protokoll

einige Befehle genannt und erklärt

**Verbindungs- und  
Verwaltungskommandos:**

**Kommando:** PASS

**Parameter:** <password>

Mit dem PASS Kommando wird das Verbindungspasswort gesetzt.

Muss bei Serververbindungen vor jedem Versuch, eine Verbindung zu etablieren, gesetzt werden.

**Beispiel:**

PASS ichbinsupergeheim

**Mögliche Antworten:**

ERR\_NEEDMOREPARAMS

ERR\_ALREADYREGISTERED

# Das Protokoll

einige Befehle genannt und erklärt

## Verbindungs- und Verwaltungskommandos:

Kommando: NICK

Parameter: <nickname> [<hopcount>]

Mit dem NICK-Befehl wird ein Nickname für die Verbindung festgelegt.

Der Parameter <hopcount> wird von Servern genutzt, um anzuzeigen, wie weit der Nick vom Heimserver entfernt ist.

Beispiel:

```
NICK WiZ ;Benutzername erstmalig gemeldet
:WiZ NICK Kilroy ;Kommando aus Sicht eines Servers
Hier ändert WiZ seinen Nick auf Kilroy
```

Mögliche Antworten:

```
ERR_NONICKNAMEGIVEN ERR_NICKCOLLISION
ERR_NICKNAMEINUSE ERR_ERRONEOUSNICKNAME
```

# Das Protokoll

einige Befehle genannt und erklärt

## Verbindungs- und Verwaltungskommandos:

Kommando: USER

Parameter: <username> <hostname> <servername> <realname>

Der Befehl USER wird dazu verwendet, diverse Informationen des Clients zu spezifizieren.

USER wird ebenfalls ausgeführt, um nach erfolgreicher Registrierung der Verbindung den Benutzer auf allen Servern im Netzwerk bekanntzumachen.

Der Parameter <realname> muss am Ende der Parameterliste stehen da er Leerzeichen enthalten kann. Aus diesem Grund wird er normalerweise mit einem Doppelpunkt geprefixt

Beispiel:

```
USER gast tolmoon tolsun :Ronnie Reagan ;client registriert sich  
beim Server
```

```
:testnick USER gast tolmoon tolsun :Ronnie Reagan  
;Nachricht zwischen den Servern  
um den obigen Benutzer im  
Netzwerk bekannt zu machen
```

Mögliche Antworten:

ERR\_NEEDMOREPARAMS

ERR\_ALREADYREGISTERED

# Das Protokoll

## einige Befehle genannt und erklärt

### Verbindungs- und Verwaltungskommandos:

Kommando: OPER

Parameter: <user> <password>

Das OPER Kommando dient dazu, einem normalen Benutzer Operatorenrechte (Serveroperator) zu gewähren.

Wenn das korrekte Passwort angegeben wurde, teilt der Heimserver des Benutzers dem übrigen Netzwerk die Rechteänderung mit einer MODE +o Nachricht mit.

Beispiel:

```
OPER maxi geheimespassword ;Benutzer maxi wird zum Operator gemacht
```

Mögliche Antworten:

```
ERR_NEEDMOREPARAMS  
ERR_PASSWDMISMATCH
```

```
ERR_NOOPERHOST  
RPL_YOUREOPER
```

# Das Protokoll

einige Befehle genannt und erklärt

## Verbindungs- und Verwaltungskommandos:

Kommando: `QUIT`

Parameter: [`<quit nachricht>`]

Mit einer `QUIT` Nachricht werden alle Verbindungen zum Server beendet.

Ein Server der eine `QUIT` Nachricht erhält muss die Ursprungsverbindung dieser Message umgehend trennen.

Optional kann eine Nachricht angegeben werden, die andere Benutzer sehen können.

Beispiel:

```
QUIT :Ich geh schlafen ;Der Doppelpunkt prefixt den  
Leerzeichenhaltige Parameter
```

Mögliche Antworten:

Keine da Verbindung sofort getrennt wird.

# Das Protokoll

einige Befehle genannt und erklärt

## Verbindungs- und Verwaltungskommandos:

Kommando: `CONNECT`

Parameter: `<target Server> [<port>[ <remote Server>]]`

Dieses Operator Kommando bringt einen `<remote Server>` dazu, sofort eine Verbindung zu dem mit `<target server>` angegebenen Server aufzubauen und dazu den mit `<port>` angegebenen Port zu nutzen.

Beispiel:

```
CONNECT tolsun.oulu.fi ;Befehl eine Verbindung zum server  
tolsun.oulu.fi aufzubauen
```

```
:WiZ CONNECT eff.org 6667 csd.bu.edu  
;Befehl von WiZ die server eff.org und  
csd.bu.edu auf port 6667 zu verbinden
```

Mögliche Antworten:

```
ERR_NOSUCHSERVER  
ERR_NEEDMOREPARAMS
```

```
ERR_NOPRIVILEGES
```

# Das Protokoll

einige Befehle genannt und erklärt

## Verbindungs- und Verwaltungskommandos:

Kommando: PING

Parameter: <server1>[ <server2>]

Genau wie in der Netzwerktechnik ist das PING Kommando da, um eine bestehende Verbindung zu prüfen.

PING wird bei Inaktivität regelmäßig gesendet.

Erfolgt keine Antwort wird die Verbindung getrennt

Beispiel:

```
PING tolsun.oulu.fi      ;ein Server sendet eine PING Nachricht an  
                          einen anderen um zu testen ob er noch  
                          lebt
```

```
PING WiZ                 ;PING Nachricht an Benutzer WiZ
```

Mögliche Antworten:

```
ERR_NOORIGIN
```

```
ERR_NOSUCHSERVER
```

# Das Protokoll

einige Befehle genannt und erklärt

## Verbindungs- und Verwaltungskommandos:

Kommando: PONG

Parameter: <daemon1> [<daemon2>]

Das PONG ist die Antwort auf ein PING.

Beispiel:

```
PONG csd.bu.edu tolsun.oulu.fi ;PONG Nachricht von csd.bu.edu an  
tolsun.oulu.fi
```

Mögliche Antworten:

ERR\_NOORIGIN

ERR\_NOSUCHSERVER

# Das Protokoll

## einige Befehle genannt und erklärt

### Channel-Kommandos:

**Kommando:** JOIN

**Parameter:** <channelname>{,<channelname>} [<passwort>{,<passwort>}]

Mit einem JOIN tritt ein Benutzer einem Channel bei.

Der Benutzername wird von den Servern in die Liste der Channelmitglieder eingetragen; somit erhält der Benutzer alle Nachrichten, die an <channelname> gehen; das beinhaltet KICK, PART, QUIT, PRIVMSG, NOTICE, JOIN.

#### Beispiel:

```
JOIN #foobar           ;benutzer tritt dem channel #foobar bei
JOIN &foobar barfoo    ;benutzer tritt dem channel &foobar bei mit
                        Passwort barfoo
JOIN #foo,&bar barfu    ;benutzer tritt #foo mit passwort barfu und
                        &bar ohne passwort bei
:Wiz JOIN #foobar      ;nachricht die alle Channel Benutzer erhalten
                        über den Beitritt von Benutzer Wiz
```

#### Mögliche Antworten:

```
ERR_NEEDMOREPARAMS    ERR_INVITEONLYCHAN    ERR_CHANNELISFULL
ERR_NOSUCHCHANNEL     ERR_BANNEDFROMCHAN    ERR_BADCHANNELKEY
ERR_BADCHANMASK       ERR_TOMANYCHANNELS    RPL_TOPIC
```

# Das Protokoll

## einige Befehle genannt und erklärt

### Channel-Kommandos:

Kommando: PART

Parameter: <channelname>{,<channelname>}

Mit PART verlässt ein Benutzer den mit <channelname> angegebenen Channel.  
Hierzu wird er von den Servern aus der Liste der aktuellen Channelmitglieder ausgetragen.

Beispiel:

```
PART #twilight_zone ;channel #twilight_zone verlassen
PART #oz-ops, &group5 ;die Channel #oz-ops und &group5
                        verlassen
```

Mögliche Antworten:

```
ERR_NEEDMOREPARAMS ERR_NOTONCHANNEL
ERR_NOSUCHCHANNEL
```

# Das Protokoll

## einige Befehle genannt und erklärt

### Channel-Kommandos:

**Kommando:** MODE

**Parameter:** <channel>{[+|-]|o|p|s|i|t|n|b|v} [<limit>][<user>][<ban mask>]  
<nickname> {{+|-}|i|w|s|o}

Das MODE Kommando kann auf zwei Arten aufgerufen werden.

1. Channelmodes setzen/löschen.
2. Usermodes setzen/löschen.

**Beispiel:**

**Channelmodes:**

```
MODE #fubar +im ;channel #fubar wird invite only und moderierter
status zugewiesen
```

```
MODE #barfoo +b *!*@*.t-ipconnect.de ;kein Telekom benutzer darf
#barfoo betreten
```

**Usermodes:**

```
MODE WiZ -o ;WiZ nimmt sich selbst Operator status
```

# Das Protokoll

einige Befehle genannt und erklärt

Channel-Kommandos:

Channel-Modes

- o - Operator Privilegien nehmen/geben (-/+)
- p - Privat Channel Flag (Channel taucht im WHOIS Befehl nicht auf)
- s - Secret Channel Flag (Channel taucht im WHOIS und im LIST nicht auf)
- i - invite-only (Channel kann nur nach Einladung eines Operators betreten werden)
- t - Topic nur vom Operator setzbar Flag
- n - keine Nachrichten von Nicht-Channelmitgliedern zeigen
- m - moderierter Channel
- l - Benutzerlimit für den Channel setzen
- b - Banmask setzen, um Benutzer(-gruppen) aus dem Channel zu verbannen
- v - Erlaubnis zum Sprechen erteilen (in moderierten Channels)
- k - Channel-Passwort setzen

# Das Protokoll

einige Befehle genannt und erklärt

## Channel-Kommandos:

### User-Modes

- i - Benutzer unsichtbar für andere
- s - Benutzer erhält Server Notices
- w - Benutzer erhält WALLOP Messages (Nachricht, die an alle Operatoren gehen)
- o - Benutzer ist Operator

### Mögliche Antworten:

ERR_NEEDMOREPARAMS	ERR_CHANOPRIVSNEEDED
ERR_NOSUCHNICK	ERR_NOTONCHANNEL
ERR_KEYSET	ERR_UNKNOWNMODE
ERR_NOSUCHCHANNEL	ERR_USERSDONTMATCH
ERR_UMODEUNKNOWNFLAG	RPL_BANLIST
RPL_ENDOFBANLIST	RPL_UMODEIS
RPL_CHANNELMODEIS	

# Das Protokoll

## einige Befehle genannt und erklärt

### Channel-Kommandos:

**Kommando:** TOPIC

**Parameter:** <channelname> [<topic>]

Es dient zum Setzen oder Auslesen des aktuellen Channel-Themas.  
Das Thema kann von einem User nur dann geändert werden wenn...

... das t Flag nicht gesetzt ist

**Beispiel:**

```
TOPIC #test ;gibt die aktuelle Topic von #test zurück
TOPIC #test :neue Topic ;setzt „neue Topic“ als neues Thema in
                        #test
```

**Mögliche Antworten:**

```
ERR_NEEDMOREPARAMS ERR_NOTONCHANNEL
RPL_NOTOPIC         ERR_CHANOPRIVSNEEDED
RPL_TOPIC
```

# Das Protokoll

## einige Befehle genannt und erklärt

### Channel-Kommandos:

**Kommando:** INVITE

**Parameter:** <nickname> <channelname>

INVITE lädt den mit <nickname> angegebenen Benutzer in <channelname> ein.

#### Voraussetzungen:

- <channelname> muss das +i Flag gesetzt haben
- Einladender muss Operator in <channelname> sein

#### Beispiel:

```
:Angel INVITE Wiz #foobar ;Benutzer Angel läd Wiz in Channel  
#foobar ein Kommando aus Sicht eines  
Servers
```

#### Mögliche Antworten:

```
ERR_NEEDMOREPARAMS ERR_NOSUCHNICK  
ERR_NOTONCHANNEL ERR_USERONCHANNEL  
ERR_CHANOPRIVSNEEDED RPL_INVITING  
RPL_AWAY
```

# Das Protokoll

## einige Befehle genannt und erklärt

### Channel-Kommandos:

**Kommando:** KICK

**Parameter:** <channel> <user> [<comment>]

Mit Hilfe des Befehls KICK kann ein Operator einen Benutzer aus einem Channel entfernen.

**Beispiel:**

```
KICK &foobar testuser ;Benutzer testuser aus channel
                        &foobar kicken
KICK #test foobar :wrong language ;Benutzer foobar wird aus
                                    channel #test gekickt mit
                                    der Begründungsnachricht
                                    „wrong language“
```

**Mögliche Antworten:**

```
ERR_NEEDMOREPARAMS
ERR_BADCHANMASK
ERR_NOTONCHANNEL
```

```
ERR_NOSUCHCHANNEL
ERR_CHANOPRIVSNEEDED
```

# Das Protokoll

## einige Befehle genannt und erklärt

### Querybefehle:

**Kommando:** WHOIS

**Parameter:** [<server>] <nickmask>[ {, <nickmask> } ]

Mit WHOIS können Informationen über einen Benutzer abgerufen werden.  
z. B. Realname, Hostmask und die Channel, in denen der Benutzer online ist.

### Beispiel:

WHOIS DonSilver

```
* [DonSilver] (n=DonSilve@g229076206.adsl.alicedsl.de): Christoph
* [DonSilver] #gentoo-anfaenger #ga-offtopic #linux-phc
* [DonSilver] kornbluth.freenode.net :Frankfurt, DE, EU
* [DonSilver] is identified to services
* [DonSilver] is signed on as account K_O-Gnom
* [DonSilver] untätig 00:35:34, Online seit: Sat Mar 14 13:03:19
* [DonSilver] Ende der WHOIS-Liste.
```

### Mögliche Antworten:

ERR_NOSUCHSERVER	ERR_NONICKNAMEGIVEN	RR_NOSUCHNICK
RPL_WHOISUSER	RPL_WHOISCHANNELS	RPL_WHOISSERVER
RPL_AWAY	RPL_WHOISOPERATOR	RPL_WHOISIDLE
RPL_ENDOFWHOIS		

# Das Protokoll

## einige Befehle genannt und erklärt

### Querybefehle:

**Kommando:** WHOWAS

**Parameter:** <nickname>[ <count>[ <server>]]

Mit WHOWAS können Informationen über einen Benutzer abgerufen werden, der nicht mehr länger online ist.

### Beispiel:

```
WHOWAS DonSilver
```

```
* [DonSilver] (n=DonSilve@g229076206.adsl.alicedsl.de): Christoph
```

```
* [DonSilver] irc.freenode.net :Sat Mar 14 13:52:17 2009
```

```
* DonSilver :End of WHOWAS
```

### Mögliche Antworten:

```
ERR_NONICKNAMEGIVEN
```

```
RPL_WHOWASUSER
```

```
RPL_ENDOFWHOWAS
```

```
ERR_WASNOSUCHNICK
```

```
RPL_WHOISSERVER
```

# Das Protokoll

## einige Befehle genannt und erklärt

### Querybefehle:

Kommando: LIST

Parameter: [<channelname>{,<channelname>} [<server>]]

Mit LIST können alle oder ein/mehrere bestimmte(r) Channel und deren Topics aufgelistet werden. Channel die als privat gekennzeichnet sind, werden nur mit Namen aufgelistet. Versteckte Channel tauchen nicht auf.

Beispiel:

```
LIST                ;Alle Channels des Netzwerks werden  
                   Ausgegeben
```

```
LIST #twilight_zone,#test,&foobar ;nur infos über die  
                                   Channel #twilight_zone,#test  
                                   und &foobar werden angezeigt
```

Mögliche Antworten:

```
ERR_NOSUCHSERVER  
RPL_LIST
```

```
RPL_LISTSTART  
RPL_LISTEND
```

# Das Protokoll

einige Befehle genannt und erklärt

## Kommandos zum Nachrichtenversand:

Kommando: PRIVMSG

Parameter: <receiver>[ {, <receiver>} ] :<text zum senden>

Das PRIVMSG Kommando schickt eine Nachricht an einen oder mehrere Empfänger.  
<receiver> kann ein Nickname, ein Channelname eine Hostmaske (#mask) oder eine servermask (\$mask) sein.

Beispiel:

```
:Angel PRIVMSG Wiz :Hello are you there? ;nachricht von Angel an Wiz
PRIVMSG jto@tolsun.oulu.fi :Hello! ;nachricht an den user jto
                                     auf dem Server tolsun.oulu.fi
PRIVMSG $*.fi :Server foobar.fi will reboot now
                                     ;nachricht an alle User auf
                                     einem Server der Domain *.fi
```

Mögliche Antworten:

ERR_NORECIPIENT	ERR_NOTEXTTOSEND	ERR_CANNOTSENDDTOCHAN
ERR_NOTOPLEVEL	ERR_WILDTOPLEVEL	ERR_TOOMANYTARGETS
ERR_NOSUCHNICK	RPL_AWAY	

# Das Protokoll

einige Befehle genannt und erklärt

## Kommandos zum Nachrichtenversand:

Kommando: NOTICE

Parameter: <nickname> :<text>

Das NOTICE Kommando ist sehr ähnlich dem PRIVMSG Kommando der einzige Unterschied ist, dass der Sender keine Antwort auf erfolgreiche Zustellung oder Fehler erhält.

Beispiel:

```
NOTICE Testuser :hallo bist du da?
```

Mögliche Antworten:

Keine.

# Das Protokoll

## Replies - Antworten

### **ERROR Replies:**

Nummer:	402	ERR_NOSUCHSERVER
Beispiel:	„<server name> :No such Server“	
Erklärung:	Zeigt an, dass der übergebene Servername nicht existiert	
Nummer:	405	ERR_TOOMANYCHANNELS
Beispiel:	„<channel name> :You have joined to many channels“	
Erklärung:	Meldet dem Benutzer, dass er das Limit der betretbaren Channels mit dem letzten JOIN überschritten hat und dieses nicht ausgeführt wurde.	
Nummer:	446	ERR_USERSDISABLED
Beispiel:	„USERS had been Disabled“	
Erklärung:	Meldet dem Client, dass das Kommando USERS von den Server Operatoren deaktiviert wurde.	

# Das Protokoll

## Replies - Antworten

### **Kommando Replies:**

- Nummer: 305 RPL\_UNAWAY  
Beispiel: „:You are no longer marked as being away“  
Erklärung: Bestätigung, dass der User gerade seinen Away-Status aufgehoben hat.
- Nummer: 318 RPL\_ENDOFWHOIS  
Beispiel: „<nick>:End of /WHOIS list“  
Erklärung: Zeigt das Ende der Informationen, die mit WHOIS ermittelt wurden, an.
- Nummer: 341 RPL\_INVITING  
Beispiel: „<channel> <nick>“  
Erklärung: Zeigt an, dass der mit INVITE angegebene Benutzer erfolgreich in den Channel eingeladen wurde.

# Quellen

[http://de.wikipedia.org/wiki/Internet\\_Relay\\_Chat](http://de.wikipedia.org/wiki/Internet_Relay_Chat)

<http://www.irchelp.org/irchelp/text/rfc1459.txt>

<http://webtoman.com/opera/panel/ircdmodes.html>

<http://de.wikipedia.org/wiki/BITNET>

**Vielen Dank für die  
Aufmerksamkeit!**

Fragen!?