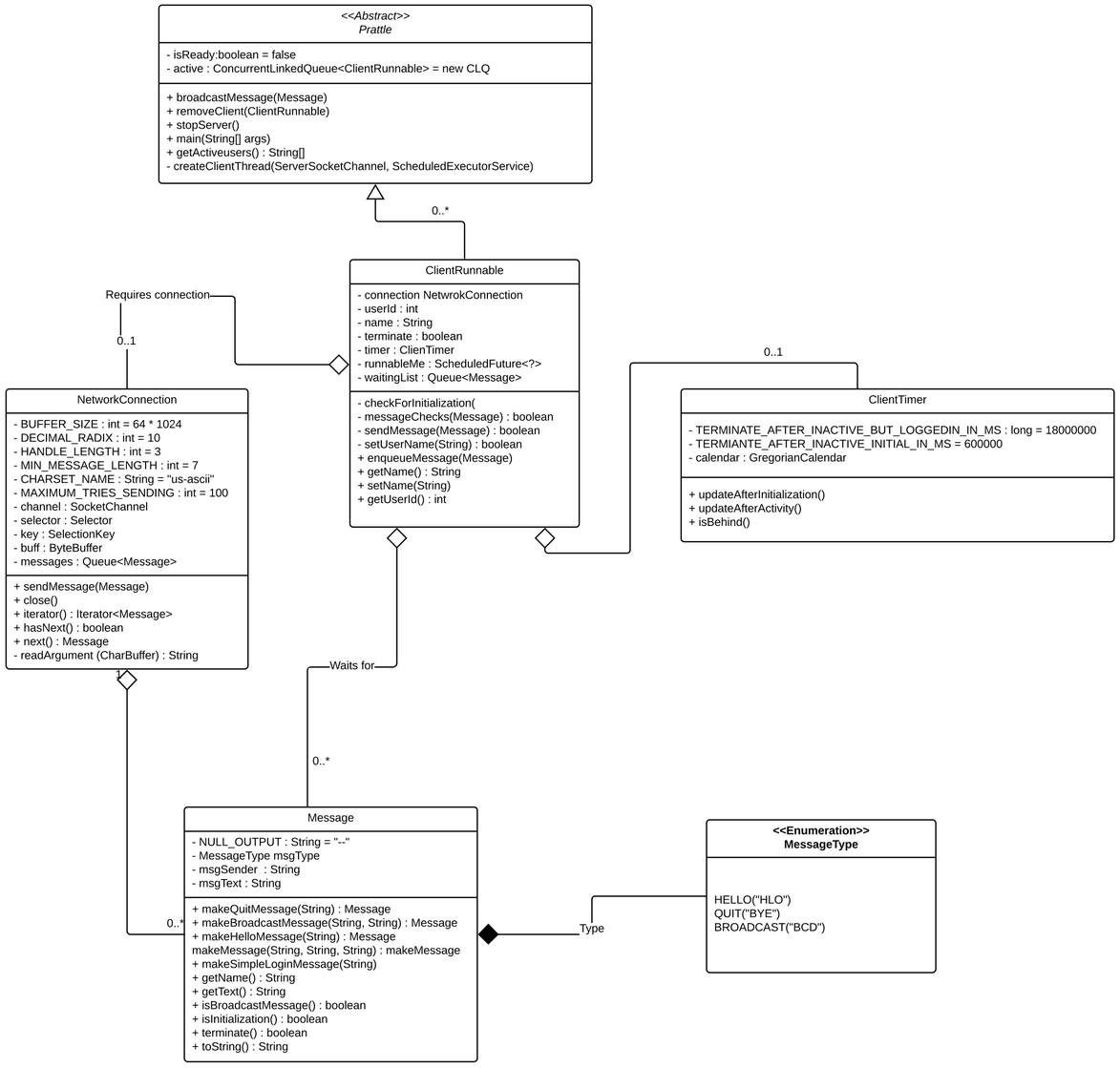
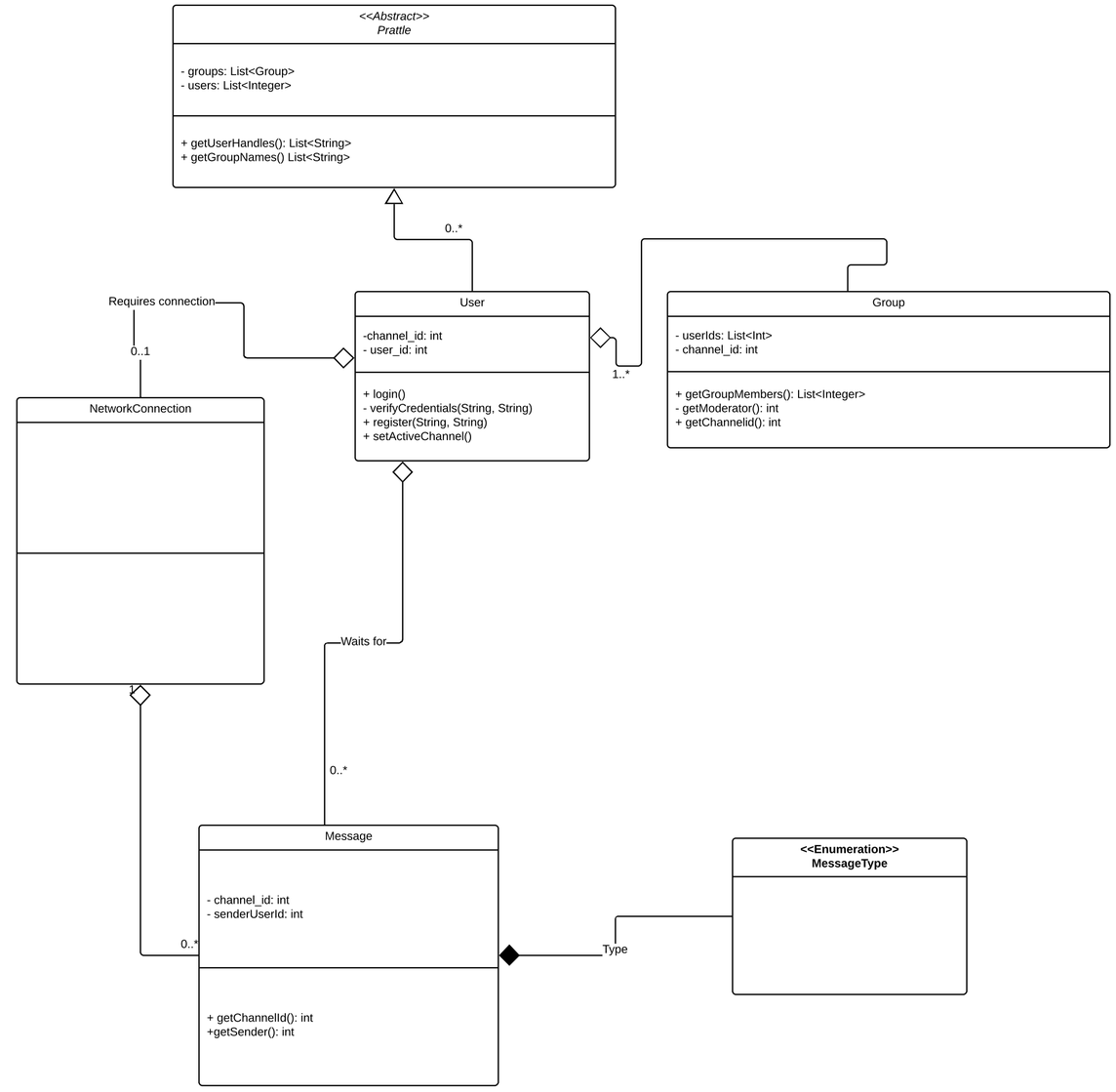
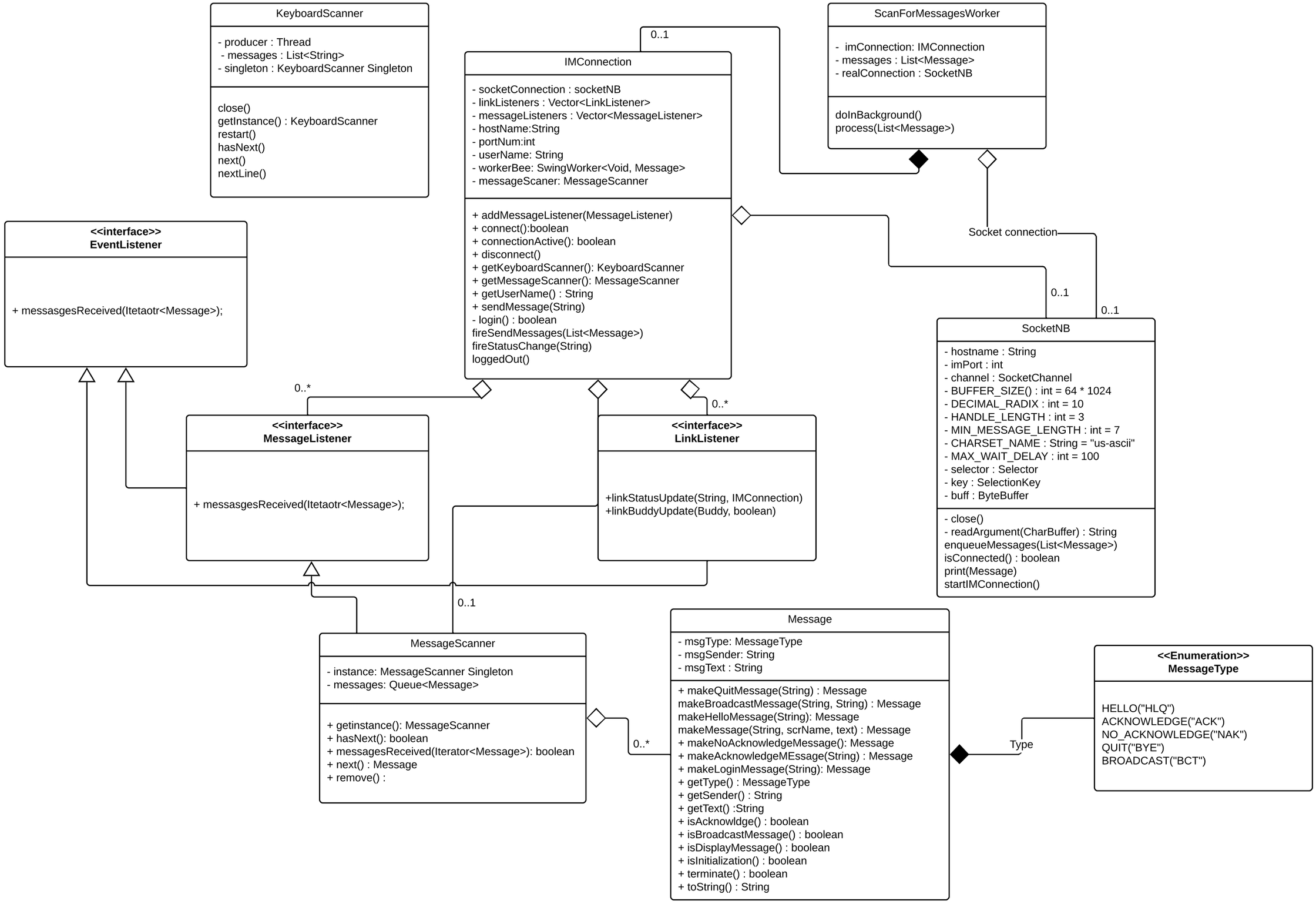

<<Abstract>>
Prattle
- isReady:boolean = false
- active : ConcurrentLinkedQueue<ClientRunnable> = new CLQ
+ broadcastMessage(Message)
+ removeClient(ClientRunnable)
+ stopServer()
+ main(String[] args)
+ getActiveusers() : String[]
- createClientThread(ServerSocketChannel, ScheduledExecutorService)
0..*
ClientRunnable
- connection NetwrokConnection
- userId : int
- name : String
- terminate : boolean
- timer : ClienTimer
- runnableMe : ScheduledFuture<?>
- waitingList : Queue<Message>
- checkForInitialization(
- messageChecks(Message) : boolean
- sendMessage(Message) : boolean
- setUserName(String) : boolean
+ enqueueMessage(Message)
+ getName() : String
+ setName(String)
+ getUserId() : int
Requires connection
0..1
NetworkConnection
- BUFFER_SIZE : int = 64 * 1024
- DECIMAL_RADIX : int = 10
- HANDLE_LENGTH : int = 3
- MIN_MESSAGE_LENGTH : int = 7
- CHARSET_NAME : String = "us-ascii"
- MAXIMUM_TRIES_SENDING : int = 100
- channel : SocketChannel
- selector : Selector
- key : SelectionKey
- buff : ByteBuffer
- messages : Queue<Message>
+ sendMessage(Message)
+ close()
+ iterator() : Iterator<Message>
+ hasNext() : boolean
+ next() : Message
- readArgument (CharBuffer) : String
0..1
ClientTimer
- TERMINATE_AFTER_INACTIVE_BUT_LOGGEDIN_IN_MS : long = 18000000
- TERMIANTE_AFTER_INACTIVE_INITIAL_IN_MS = 600000
- calendar : GregorianCalendar
+ updateAfterInitialization()
+ updateAfterActivity()
+ isBehind()
Waits for
1
0..*
Message
- NULL_OUTPUT : String = "--"
- MessageType msgType
- msgSender : String
- msgText : String
+ makeQuitMessage(String) : Message
+ makeBroadcastMessage(String, String) : Message
+ makeHelloMessage(String) : Message
makeMessage(String, String, String) : makeMessage
+ makeSimpleLoginMessage(String)
+ getName() : String
+ getText() : String
+ isBroadcastMessage() : boolean
+ isInitialization() : boolean
+ terminate() : boolean
+ toString() : String
0..*
Type
<<Enumeration>>
MessageType
HELLO("HLO")
QUIT("BYE")
BROADCAST("BCD")


<<Abstract>>
Prattle
- groups: List<Group>
- users: List<Integer>
+ getUserHandles(): List<String>
+ getGroupNames() List<String>
0..*
User
-channel_id: int
- user_id: int
+ login()
- verifyCredentials(String, String)
+ register(String, String)
+ setActiveChannel()
1..*
Group
- userIds: List<Int>
- channel_id: int
+ getGroupMembers(): List<Integer>
- getModerator(): int
+ getChannelid(): int
Requires connection
0..1
NetworkConnection
Waits for
1
0..*
Message
- channel_id: int
- senderUserId: int
+ getChannelId(): int
+getSender(): int
0..*
Type
<<Enumeration>>
MessageType


KeyboardScanner
- producer : Thread
- messages : List<String>
- singleton : KeyboardScanner Singleton
close()
getInstance() : KeyboardScanner
restart()
hasNext()
next()
nextLine()
IMConnection
- socketConnection : socketNB
- linkListeners : Vector<LinkListener>
- messageListeners : Vector<MessageListener>
- hostName:String
- portNum:int
- userName: String
- workerBee: SwingWorker<Void, Message>
- messageScaner: MessageScanner
+ addMessageListener(MessageListener)
+ connect():boolean
+ connectionActive(): boolean
+ disconnect()
+ getKeyboardScanner(): KeyboardScanner
+ getMessageScanner(): MessageScanner
+ getUserName() : String
+ sendMessage(String)
- login() : boolean
fireSendMessages(List<Message>)
fireStatusChange(String)
loggedOut()
0..1
ScanForMessagesWorker
- imConnection: IMConnection
- messages : List<Message>
- realConnection : SocketNB
doInBackground()
process(List<Message>)
Socket connection
0..1
0..1
<<interface>>
EventListener
+ messasgesReceived(Itetaotr<Message>);
SocketNB
- hostname : String
- imPort : int
- channel : SocketChannel
- BUFFER_SIZE() : int = 64 * 1024
- DECIMAL_RADIX : int = 10
- HANDLE_LENGTH : int = 3
- MIN_MESSAGE_LENGTH : int = 7
- CHARSET_NAME : String = "us-ascii"
- MAX_WAIT_DELAY : int = 100
- selector : Selector
- key : SelectionKey
- buff : ByteBuffer
- close()
- readArgument(CharBuffer) : String
enqueueMessages(List<Message>)
isConnected() : boolean
print(Message)
startIMConnection()
0..*
0..*
<<interface>>
MessageListener
+ messasgesReceived(Itetaotr<Message>);
<<interface>>
LinkListener
+linkStatusUpdate(String, IMConnection)
+linkBuddyUpdate(Buddy, boolean)
0..1
MessageScanner
- instance: MessageScanner Singleton
- messages: Queue<Message>
+ getinstance(): MessageScanner
+ hasNext(): boolean
+ messagesReceived(Iterator<Message>): boolean
+ next() : Message
+ remove() :
0..*
Message
- msgType: MessageType
- msgSender: String
- msgText : String
+ makeQuitMessage(String) : Message
makeBroadcastMessage(String, String) : Message
makeHelloMessage(String): Message
makeMessage(String, scrName, text) : Message
+ makeNoAcknowledgeMessage(): Message
+ makeAcknowledgeMEssage(String) : Message
+ makeLoginMessage(String): Message
+ getType() : MessageType
+ getSender() : String
+ getText() :String
+ isAcknowldge() : boolean
+ isBroadcastMessage() : boolean
+ isDisplayMessage() : boolean
+ isInitialization() : boolean
+ terminate() : boolean
+ toString() : String
Type
<<Enumeration>>
MessageType
HELLO("HLQ")
ACKNOWLEDGE("ACK")
NO_ACKNOWLEDGE("NAK")
QUIT("BYE")
BROADCAST("BCT")