

- sendPacket: DatagramPacket
- receivePacket: DatagramPacket
- serverSocket: DatagramSocket
- clientSocket: DatagramSocket
- unknownSocket: DatagramSocket
- sendPort: int
- sendToServer: boolean
- packetNumber: int
- eOpFlag: boolean
- eFnFlag: boolean
- eMdFlag: boolean
- eBlockNumber: int
- eDfFlag: boolean
- errorCode: int
- filename: String
- delay: int
- unknownTID: boolean
- packetCount: int
- typeCount: int
- clientAddress: InetAddress
- MAX_DATA: int

- ToClient(DatagramPacket,PacketType,PacketDo,boolean,int,boolean,boolean,boolean,int,boolean,int,String,DatagramSocket,DatagramSocket,int,InetAddress,int,boolean)
- ToClient(DatagramPacket,DatagramSocket,DatagramSocket,int,InetAddress)
- run():void
- threadName():String
- action(byte[],int):byte[]
- createPacket():byte[]
- matchType(int):boolean
- receive(DatagramSocket):DatagramPacket
- processDatagram(DatagramPacket):byte[]
- send(byte[],InetAddress,int,DatagramSocket):void
- twoBytesToInt(byte,byte):int
- parseRequest(byte[]):String
- getFilename(byte[]):String
- getMode(byte[]):String
- parseAck(byte[]):String
- parseData(byte[]):String
- parseError(byte[]):String

<<Java Class>>



(default package)

sendPacket: DatagramPacket
receivePacket: DatagramPacket
serverSocket: DatagramSocket
clientSocket: DatagramSocket
unknownSocket: DatagramSocket

sendPort: int

saddr: InetAddress

sendToServer: boolean

packetNumber: int

eOpFlag: boolean

eFnFlag: boolean

eMdFlag: boolean

eBlockNumber: int

eDfFlag: boolean

errorCode: int

filename: String

delay: int

unknownTID: boolean

packetCount: int

typeCount: int

clientAddress: InetAddress

MAX_DATA: int

ToServer(DatagramPacket,PacketType,PacketDo,boolean,int,boolean,boolean,boolean,int,boolean,int,String,DatagramSocket,DatagramSocket,int,int,boolean)

ToServer(DatagramPacket,DatagramSocket,DatagramSocket,int)

run():void

threadName():String

processDatagram(DatagramPacket):byte[]

action(byte[],int):byte[]

createPacket():byte[]

matchType(int):boolean

receive(DatagramSocket):DatagramPacket

send(byte[],InetAddress,int,DatagramSocket):void

twoBytesToInt(byte,byte):int

parseRequest(byte[]):String

getFilename(byte[]):String

getMode(byte[]):String

parseAck(byte[]):String

parseData(byte[]):String

parseError(byte[]):String

