

# EZ Multiplayer Reference Sheet

by Andrew Klassen

**net\_init**( *the dll location, true if you are using studio* );

**parameters:** (string, boolean)

**return type:** none

**description:** You need this function to use any of EZ Multiplayer's functions. The first parameter is the location of your dll which is "dll\_39dll.dll" if you have added it into your included files. I recommend using this in your object's create event.

## Server Functions

**net\_server\_start**( *port number, max clients* );

**parameters:** (integer, integer)

**return type:** none

**description:** This function starts the server. You can not use **server\_update**() without it.

**net\_server\_update**( );

**parameters:** (none)

**return type:** none

**description:** This function will take all incoming TCP packets, and distribute the same packets to all connected clients. This includes the client who sent the packet or packets.

## Built in Constants

**NET\_INT**

**description:** This is just an integer that is equal to 1.

**NET\_STRING**

**description:** This is just an integer that is equal to 2.

# Client Functions

**net\_client\_connect**( *ip address or domain name, port number* );

**parameters:** (string, integer)

**return type:** none

**description:** This function establishes a TCP connection with a server at the given address and port. If you do not connect, your client will not be able to send or receive any TCP packets. If you make a connection, you don't have to make another connection unless you become disconnected. I recommend using it in an obj\_client create event.

**net\_client\_update**( );

**parameters:** (none)

**return type:** none

**description:** This function updates the client with new TCP packets. Normally, you would need to use this at the top of obj\_client step event. You will not receive anything until this function is executed.

**net\_client\_send**( *packet id, amount of packets, variable type, var1, var2, . . .* );

**parameters:** (integer, integer, integer, integers or strings depending on parameters 2 and 3)

**return type:** none

**description:** This function sends 1 TCP packet to the server. 1 or **NET\_INT** should be used as the third parameter if you want to send integers. 2 or **NET\_STRING** should be used as the third parameter if you want to send strings. Gamemaker does not allow more than 16 parameters for any of its functions, which means that parameter 2 should never be more than 13.

**net\_client\_get**( *packet id, variable index starting at 1* );

**parameters:** (integer, integer)

**return type:** the variable according to the parameters, or 0 if there are no packets with the id

**description:** This function returns the most up to date packet with the given id. If you try to get the value of a packet index that doesn't exist and the packet id has been used, your program will crash.

**net\_client\_check**( *packet id* );

**parameters:** (integer)

**return type:** boolean

**description:** This function returns true if **net\_client\_update**() received a packet, with the given id, within the same irritation. It's best to make sure **net\_client\_update**() is used before this function. It will only return true once per packet.