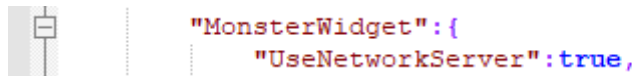


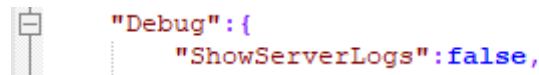
Monsters Data Distribution Explanation

This new experimental feature allows all team party that use SmartHunter to see Monsters parts and statuses even if they are not the current host player. Please note that this will work only if the current host of the lobby is using SmartHunter with this feature enabled. The current server is hosted by Altvista (aka free, aka very slow).

This feature is **DISABLED** by default and in order to enable it just set to 'true' the key named 'UseNetworkServer' under the root dictionary with key 'MonsterWidget' inside the file 'Config.json' as shows in the picture below. **Please note that if this is disabled everything that comes after in this document will be ignored and the application will NOT communicate with the server.**



I've also added a new option to display Debug log for this feature, so if you are interested you can enable it by setting to 'true' the key named 'ShowServerLogs' under the root dictionary 'Debug' (in the picture below I have it disabled)



If you are worried about what is being uploaded and downloaded please note that SmartHunter will identify you with a unique KEY calculated with the SHA1's algorithm of your SessionID + LobbyID and send to the server this KEY among with Monsters Data (if you are the host) or download that data, that the host (if there's any) is uploading to the server.

key	host_connected	player_count	data	last_modify
ZXE3Wgh4A6ymO0glko60CY4iWXk=	1	1	NULL	2020-03-01 17:20:05

After the end of every mission SmartHunter will log the number of operations that were done and the total of bytes that were uploaded and downloaded to the server.

```
[2020-03-01 16:20:05] Joined lobby with id 'ZXE3Wgh4A6ymO0glko60CY4iWXk='  
[2020-03-01 16:20:37] Left lobby with id 'ZXE3Wgh4A6ymO0glko60CY4iWXk='  
[2020-03-01 16:20:37] Total network operations 9, failed 0, sent 860 byte and received 329 byte
```

Now I would like to talk about the operations that are taken to make this new feature work. The picture below is a state machine diagram (is it really?) that sums the whole procedure and actions. The COMMANDS that are sent to the server are the squared rectangles in red:

1. **ALIVE**: this command check if the server is online and is made only once as soon the application is started. In case of failure you **MUST** restart the application to utilize the server.
2. **HELLO**: this command allows you to subscribe to the current lobby. On server side this creates a new row with the current KEY, if there's none, or updates it to increment the current number of players connected and if you are the host of the lobby set up a flag to indicate that the host is uploading data.
3. **CHECK**: this command is made after the **HELLO** command if it succeeds, and it checks if the current lobby have at least two players connected, if you are the host, or if the host is connected if you are a normal player in the lobby. This is done to avoid the host uploading unnecessary data if no players are in the lobby utilizing this feature (or SmartHunter) or downloading unnecessary data if the host is not

connected to the lobby for normal players. This command is done 1 time every 5 seconds till it's succeeds (host connected or at least 2 players are connected) or the lobby terminates.

4. PUSH: this command is done by the host of the lobby if CHECK's command succeeds, and it uploads current Monsters data to the server. This operation is done 1 per second to avoid overhead.
5. PULL: this command is done by normal players in the lobby if CHECK's command succeeds, and it downloads current Monsters data from the server and displays it on the overlay. This operation is done 1 per second to avoid overhead.
6. DONE: this command is done when the lobby terminates, either the mission ends or you exit the lobby, and it allows the server to decrement the current number of players connected to the server for that specific lobby. Note that when this counter reaches 0 or there are no updates for 60 minutes (the host is not uploading), the row corresponding to that lobby is deleted from the server.

