

HC Installation and Setup Guide for A3EAI

Last updated: October 4, 2015

Introduction

A Headless Client can be used by missions to offload AI processing to a dedicated client, freeing up the dedicated server process from most AI processing. Thus more AI units can be spawned and the server process will be able to dedicate most of its processing towards communication with the clients. For a Headless Client/Dedicated Server to function together efficiently, they both need to be connected to each other via extremely low latency and high bandwidth. Effectively, they need to be on the same LAN at least, but preferably running on the same computer.

Source: https://community.bistudio.com/wiki/Arma_3_Dedicated_Server#Headless_Client

If the Arma 3 Headless Client and a Arma 3 Dedicated Server are running on the same Windows computer (VM or physical), it may be beneficial to set processor affinity to prevent execution intensive threads from being scheduled across the same (v)CPUs. The operating system will generally schedule threads efficiently, but manual CPU allocation is possible. This can be achieved by right-clicking on the process (e.g. Arma3Server.exe(*32)) in the processes tab of the Windows Task Manager and selecting Set Affinity.... Be aware that on a physical intel CPU, the odd numbered CPUs are Hyper-thread cores.

*Note: You can use automated solution for affinity assign via batch file with commandline CMD
/AFFINITY HEXvalue e.g. CMD /C START /AFFINITY 0xF3 arma3server.exe*

It has been observed that although the ARMA server and client processes will kick off multiple threads, the bulk of processing is used up by only one or two threads. For example, spawning 50 AI units does not generate 50 threads. There is one thread in the process that handles all of the AI units, irrespective of how many have been spawned. In this way, the ARMA server and client processes do not make maximal use of the processing capability found in modern processors and so AI counts do not scale easily. As such, faster CPU core speed is king and offloading the AI to multiple headless clients on the same computer will probably produce the best possible results for complex missions involving many AI units (albeit an expensive way to get the results).

Source:

https://community.bistudio.com/wiki/Arma_3_Dedicated_Server#Headless_Client_on_a_Dedicated_Server_Notes

Required/Recommended Tools:

1. Notepad ++ (Recommended): <https://notepad-plus-plus.org/download/>
2. Eliteness (Required): <https://dev.withsix.com/projects/mikero-pbodll/files>
3. PBO Manager (Required): <http://www.armaholic.com/page.php?id=16369>

Note: If you already have a basic headless client set-up, skip to Part B to set up an HC for A3EAI

Part A: Set up Headless Client

1. Edit your server's config.sqf using a text editor

- Recommended to use **Notepad++** (Notepad++ will be used in this guide).
- Typically, config.sqf is located in /SC/config.cfg

2. Add the following lines to the end of this file.

```
localClient[]={127.0.0.1};
```

```
headlessClients[] = {"127.0.0.1"};
```

```
battleyeLicense=1;
```

3. **If this is the first time the HC is being set up on your server:** Add the provided BattlEye filters contained in the text file named "BattlEye filters for HC".

- If you are running your headless client from a different physical machine on a LAN or VM, add its IP to localClient[] and headlessClients[].
- It is recommended to run your HC on the same physical machine to minimize latency and maximize bandwidth between dedicated server and HC.

- From most ideal to least ideal dedicated/headless client setup (red indicates setups that should be avoided)
 - HC on same physical machine and environment as dedicated server
 - Minimal added latency, maximum bandwidth. **Recommended.**
 - HC running on a different physical machine on same LAN as dedicated server
 - More added latency compared to above scenarios
 - HC running on a different physical machine, not connected to the dedicated server by a LAN
 - Much higher latency, high bandwidth requirement may cause issues with other players compared to above scenarios. **Not recommended.**

3. Using **Eliteness**, open your mission.sqm and decrypt mission.sqm using the below guide.

The screenshot shows the EliteNess x32Version 3.25 application window. The interface includes a menu bar (File, Edit, View, Options, Tools, Help), a toolbar with icons for file operations, and a main workspace divided into a file explorer on the left and a text editor on the right.

File Explorer (Left): Displays a directory tree with folders like Epoch_Server_0.3.0.1_b1, Epoch_Server_0.3.0.1_b17, Epoch_Server_0.3.0.2_b16, Epoch_Server_0.3.0.2_b19, Epoch_Server_0.3.0.2_b2, Epoch_Server_0.3.0.2_b3, Epoch_Server_0.3.0.3_b1, @epochhive, BEC, DB, keys, mpmissions, epoch.Altis (highlighted with a red box and labeled "1. Locate unpacked mission files"), and SC.

Text Editor (Right): Displays the contents of a mission.sqm file. The text includes a header section with version information and a class definition for Mission. The class definition includes various parameters like addOns, addOnsAuto, randomSeed, and class Intel with various settings.

Annotations:

- 3. Click DeRapify button:** Points to the DeRapify icon in the toolbar.
- 4. Save and replace file when prompted:** Points to the Save icon in the toolbar.
- 2. Select mission.sqm:** Points to the mission.sqm file in the file explorer.

5. Using **Notepad++**, edit the decrypted mission.sqm using the below guide.

```
class Groups    Located at around Line 36
{
    items = 100;    Increase this number by 1.
                   In this case here, change it to 101.
    class Item0
    {
        side = "CIV";
        class Vehicles
        {
            items = 1;
            class Item0
            {
                position[] = {23598.076,3.19,17997.086};
                azimuth = 136.241;
                special = "NONE";
                id = 0;
                side = "CIV";
                vehicle = "VirtualMan_EPOCH";
                player = "PLAY CDG";
                leader = 1;
                skill = 0.6;
            };
        };
    };
};
```

Then, further down the file at about **Line 2019**:

```
class Item99    Take note of this number (In this case: 99)
{
    side = "CIV";
    class Vehicles
    {
        items = 1;
        class Item0
        {
            position[] = {23605.094,3.19,17998.143};
            azimuth = 130.016;
            special = "NONE";    Take note of this highlighted section
                                enclosed in {}
            id = 99;
            side = "CIV";
            vehicle = "VirtualMan_EPOCH";
            player = "PLAY CDG";
            leader = 1;
            skill = 0.6;
        };
    };
};
```

Add the below block, refer to the image further below:

```
class Item100
{
    side="LOGIC";
    class Vehicles
    {
        items=1;
        class Item0
        {
            position[]={23605.094,3.19,17998.143};
            special="NONE";
            id=100;
            side="LOGIC";
            vehicle="HeadlessClient_F";
            player = "PLAY CDG";
            leader=1;
            skill=0.60000002;
            text="HC";
            forceHeadlessClient = 1;
        };
    };
};
```

It should look like this afterwards:

```
class Item99
{
    side = "CIV";
    class Vehicles
    {
        items = 1;
        class Item0
        {
            position[] = {23605.094,3.19,17998.143};
            azimuth = 130.016;
            special = "NONE";
            id = 99;
            side = "CIV";
            vehicle = "VirtualMan_EPOCH";
            player = "PLAY CDG";
            leader = 1;
            skill = 0.6;
        };
    };
};

class Item100
{
    side="LOGIC";
    class Vehicles
    {
        items=1;
        class Item0
        {
            position[]={23605.094,3.19,17998.143};
            special="NONE";
            id=100;
            side="LOGIC";
            vehicle="HeadlessClient_F";
            player = "PLAY CDG";
            leader=1;
            skill=0.60000002;
            text="HC";
            forceHeadlessClient = 1;
        };
    };
};
```

Ensure this number is one higher than the previous Item, highlighted in yellow above.

Name of HC, should be unique from other HCs

IMPORTANT: You must have one HC slot for each addon on your server that uses an HC. Failing to do this will cause HCs after the first connected one to fail to connect.

6. If your mission files are kept in a pbo, repack the mission files into pbo format. Otherwise, installation of the HC is done. Your next step is now to set up A3EAI on your HC (Part B).

Part B: Set up A3EAI HC

1. Copy the @A3EAI_HC folder from the downloaded install package, inside the "6. Headless Client" folder.
2. Paste the @A3EAI_HC folder into your Arma 3 directory.
3. Open @A3EAI_HC/Keys.
4. Copy A3EAI.bikey to your server's Keys folder.
5. Edit your mission.sqm and add a new HC slot (**This should have been done in Part A**)
6. In @A3EAI >> A3EAI_config.pbo >> config.cpp, set **enableHC = 1**; (You will need to unpack A3EAI_config.pbo to access config.cpp).
7. Repack A3EAI_config.pbo.
8. Start the HC by starting arma3server with these parameters: -client -mod=@Epoch;@A3EAI_HC;

Part C: Things to note

1. It should not matter whether you start your dedicated server or headless client first.
2. You may start, close, or restart your HC at any time while A3EAI is running on your dedicated server without issues.
3. A3EAI will transfer AI groups from the dedicated server to the headless client gradually (1 group per 5 seconds) until all have been transferred.
4. A3EAI's HC will only manage AI units that A3EAI (on the dedicated server) has spawned, **not** AI from other mods or addons.
5. Dedicated server will handle A3EAI's background tasks and AI spawning, while the HC will handle AI behavior directly.

Part D: Verifying HC is running normally

Inside your server's RPT log, you will see that the HC has successfully connected when you see this:

```
"[A3EAI] Headless client L Charlie 1-2:1 REMOTE (owner: 4) logged in successfully."
```

Inside your HC's RPT log, you will see this (it won't be this clean, but this is what you're looking for):

```
22:17:06 "[A3EAI] Initializing A3EAI HC build 0.1.8 using base path A3EAI."  
22:17:06 "[A3EAI] Waiting for HC player object setup to be completed."  
22:17:07 "[A3EAI] Attempting to connect to A3EAI server..."  
22:17:08 "[A3EAI] Loaded all A3EAI settings in 0.000999451 seconds."  
22:17:08 "[A3EAI] Headless client connection successful. HC authorization request granted."  
22:17:09 "[A3EAI] Compiling functions..."  
22:17:09 "[A3EAI] A3EAI HC functions loaded."  
22:17:09 "[A3EAI] Compiling A3EAI functions."  
22:17:09 "[A3EAI] A3EAI functions compiled."  
22:17:09 "[A3EAI] A3EAI HC PVEHs loaded."  
22:17:09 "[A3EAI] A3EAI HC started with Debug Level: 2."
```