

Documentation

Controlled Spawn & Waypoints Randomizr (CSWR)

Last update: v6.5.2.1

[What is this](#)

[If you need an SQF editor](#)

[Run the script for a first look](#)

[Install the script in my own mission ★★★](#)

[Define the sides to spawn](#)

[Define who the enemy is](#)

[Define the weaponry and the side size](#)

[Spawn-points for side](#)

[Spawn-points > Working with sectors](#)

[Spawn-points > Spawn Delay methods](#)

[Which group-type is allowed for each spawn-point-type](#)

[Destinations \(waypoints\)](#)

[Destinations > Hold-marker](#)

[Destinations > Watch-marker](#)

[Destinations > Occupy-marker](#)

[Destinations > Working with sectors](#)

[Behavior for each group](#)

[Customizing the loadouts](#)

[Electronic Warfare Resources](#)

[Fixing: vehicles blow up when they spawn](#)

[Fixing: trying to edit the unit's loadout](#)

[Fixing: making CSWR lighter for servers](#)

[Fixing: helicopters over ship floors](#)

[Fixing: investigate further what is happening \(Debugging\)](#)

[Fixing: units are not in a crouch when inside a ruin](#)

[Contribute to the CSWR script](#)

[Author](#)

What is this

CSWR is an *Arma 3* script that allows the Mission Editor (you) to spawn AI units and vehicles (by ground or air paradrop) and makes those groups move randomly to waypoints forever in life, where spawn-points and waypoints are easily pre-defined by the Mission Editor through *Eden* marker's positions. CSWR accepts side loadout customization, including additional customizations for sniper groups and paratroopers. CSWR almost doesn't change any original Arma AI behavior, saving server performance and *Arma 3* integrity.

Creation concept: bring life to the mission through non-stop units' movements with some level of unpredictability without losing control of server performance and what AI units can do.

For multiplayer and single-player

CSWR works for all game purposes like multiplayer and single-player.

For Hosted and Dedicated servers

CSWR was built for both server types.

Populate missions with civilians only if needed

Yes and easily. You can spawn just a side, or two, or all of them. As you wish.

Compatible with ACE, RHS, CUP, or any others

Yes! You can call whatever classnames of units and vehicles in the *fn_CSWR_population.sqf* file.

CSWR cannot spawn planes and nautical vehicles yet

Since CSWR v5 has been allowed to spawn helicopters. Despite you might spawn soldiers and civilians in water when needed, nautical vehicles or planes aren't available yet.

Loadout customization

Yes, including uniforms, vests, helmets, night-visions, and backpacks, respecting the items inside the original ones and transferring them to the new gear. Check [how to customize the side loadouts](#).

Debug monitor and feedback available

CSWR was built to make the Mission Editor life easier, so if you want to get to know the [debug monitor now, check this out](#).

If you need an SQF editor

Sure, I'm using *Visual Studio Code* with this customs specific for *Arma 3*:

<https://forums.bohemia.net/forums/topic/239960-vs-code-tutorial-how-to-config-vs-code-for-arma-3-2023/>

If you need something simpler:

<https://notepad-plus-plus.org/>, install it and, when you open some script file, go to Notepad++ main menu, “Language” and select “C” as file language. That's it.

Run the script for a first look

1. Go to <https://steamcommunity.com/sharedfiles/filedetails/?id=2740912514>
2. Subscribe and wait for Steam to download it;
3. Open *Arma 3*, go to Multiplayer > Server browser > Host server > click Host Server button;
4. Select “Stratis” map and, after that, select “CSWR (...);
5. Play.

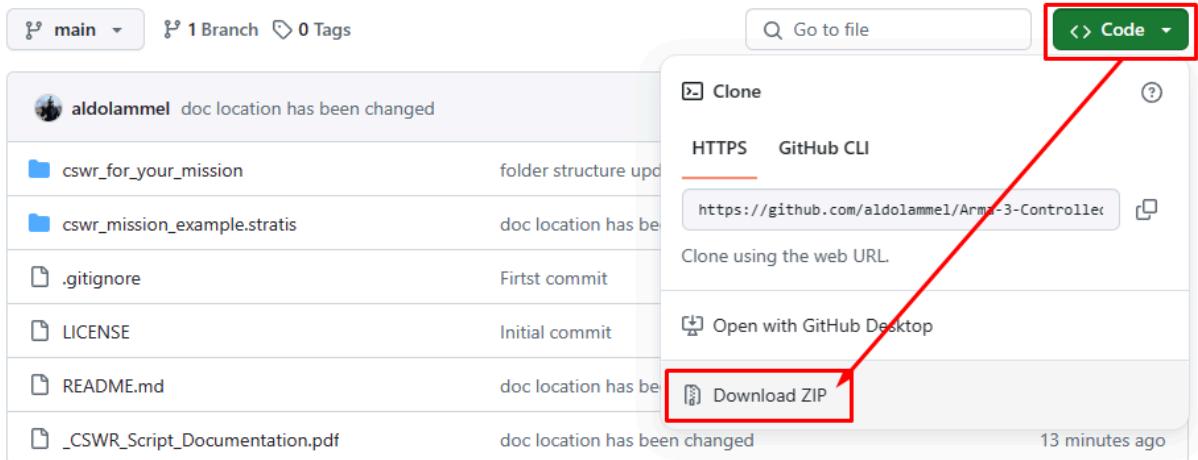
Important: if you want to apply this script in your own missions, [check this out](#).

Install the script in my own mission ★★★

1. Go to:

<https://github.com/aldolammel/Arma-3-Controlled-Spawn-And-Waypoints-Randomizr-Script>

2. Download the zip and open it;



3. In zip, get in the "cswr_for_your_mission" folder;
4. Copy all "cswr_for_your_mission" content to your mission folder root:
DRIVE:\Users\yourName\Documents\Arma 3\yourProfile\missions\yourMission
5. **WARNING:** in the mission folder root, if you already have a "description.ext" file, don't replace the current file with the new one. Just add the code below in the current file:

```
class cfgFunctions
{
    // CSWR: CONTROLLED SPAWN AND WAYPOINTS RANDOMIZR
    #include "CSWRandomizr\THY_CSWR_functions.hpp"
};
```

6. **WARNING:** still in the mission folder, if you already have an "init.sqf" file, don't replace the current file with the new one. Just add the code below in the current file:

```
// CSWR > HIDE THE SCRIPT MARKERS:
if !CSWR_isOnDebugGlobal then {{private _mk = toUpper _x; private _mkChecking = _mk splitString " "; if (_mkChecking find "CSWR" != -1) then {_x setMarkerAlpha 0}} forEach allMapMarkers};
```

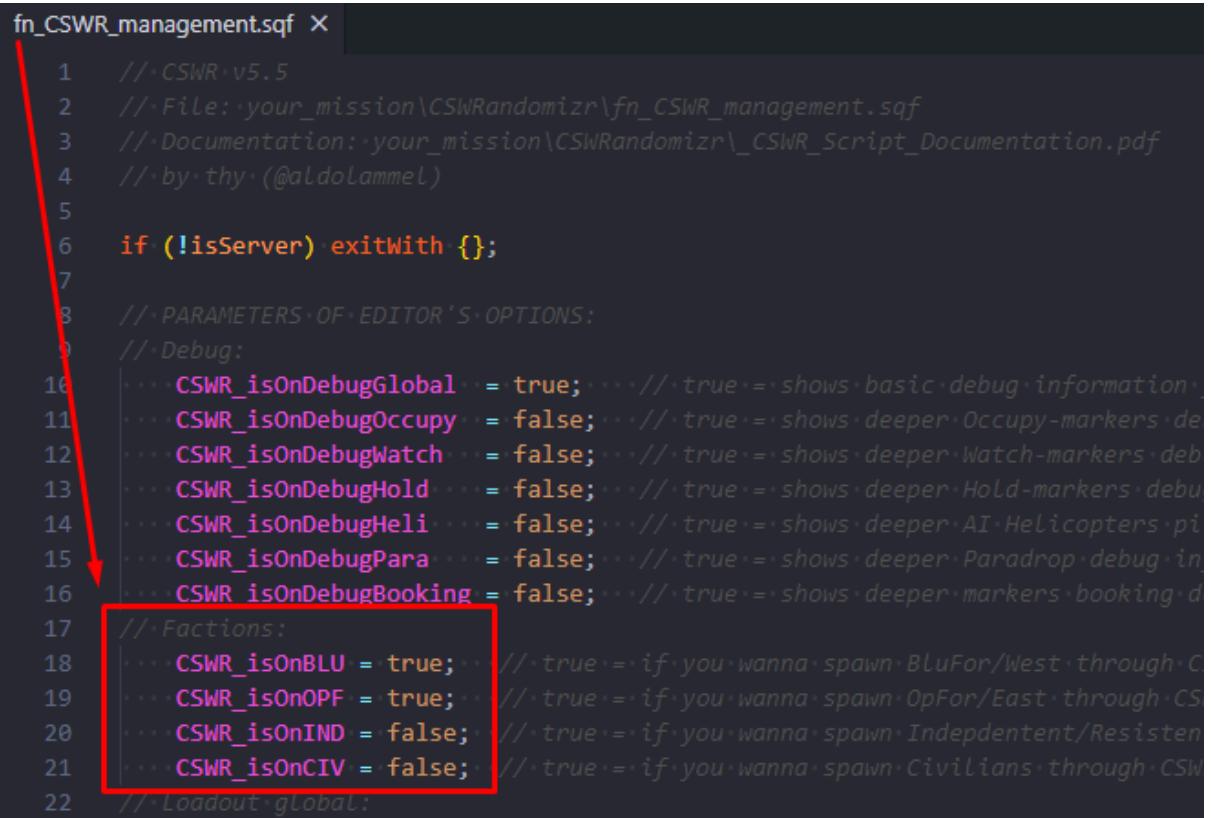
7. Now, [let's create your custom side\(s\)!](#)

Define the sides to spawn

In *Arma 3* there are 4 sides available: Blufor, Opfor, Independent, and Civilian. All of them are built by groups that are composed of units (soldiers/people). With CSWR you will spawn dynamically one or more sides. It's up to you. Let's do it, by opening the file:

\CSWRandomizr\fn_CSWR_management.sqf

Set as *true* the side you want to spawn through CSWR:

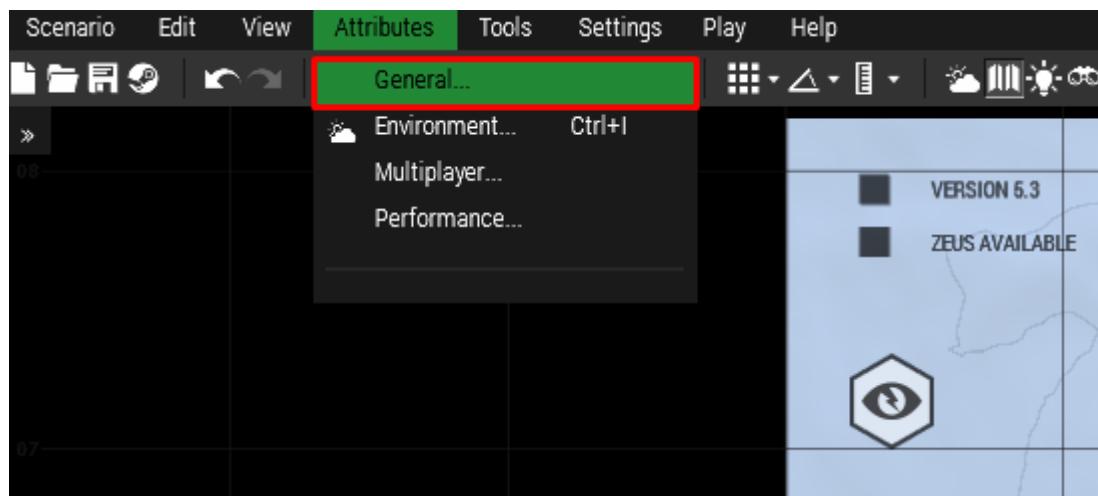


```
fn_CSWR_management.sqf X
1 // CSWR v5.5
2 // File: your_mission\CSWRandomizr\fn_CSWR_management.sqf
3 // Documentation: your_mission\CSWRandomizr\_CSWR_Script_Documentation.pdf
4 // by thy (@aldolammel)
5
6 if (!isServer) exitWith {};
7
8 // PARAMETERS OF EDITOR'S OPTIONS:
9 // Debug:
10    CSWR_isOnDebugGlobal = true; // true == shows basic debug information
11    CSWR_isOnDebugOccupy = false; // true == shows deeper Occupy-markers debug
12    CSWR_isOnDebugWatch = false; // true == shows deeper Watch-markers debug
13    CSWR_isOnDebugHold = false; // true == shows deeper Hold-markers debug
14    CSWR_isOnDebugHeli = false; // true == shows deeper AI Helicopters debug
15    CSWR_isOnDebugPara = false; // true == shows deeper Paradrop debug
16    CSWR_isOnDebugBooking = false; // true == shows deeper markers booking debug
17 // Factions:
18    CSWR_isOnBLU = true; // true == if you wanna spawn Blufor/West through CSWR
19    CSWR_isOnOPF = true; // true == if you wanna spawn Opfor/East through CSWR
20    CSWR_isOnIND = false; // true == if you wanna spawn Independent/Resistance
21    CSWR_isOnCIV = false; // true == if you wanna spawn Civilians through CSWR
22 // Loadout global:
```

Define who the enemy is

CSWR works with what you have defined through *Eden Editor*:

1. On *Eden's* main menu, go to Attributes > General;



2. In the "General" window, look for "Misc":



3. As BluFor and OpFor are always enemies of each other, you only need to define who is the Independent enemy, by clicking over the side flag.

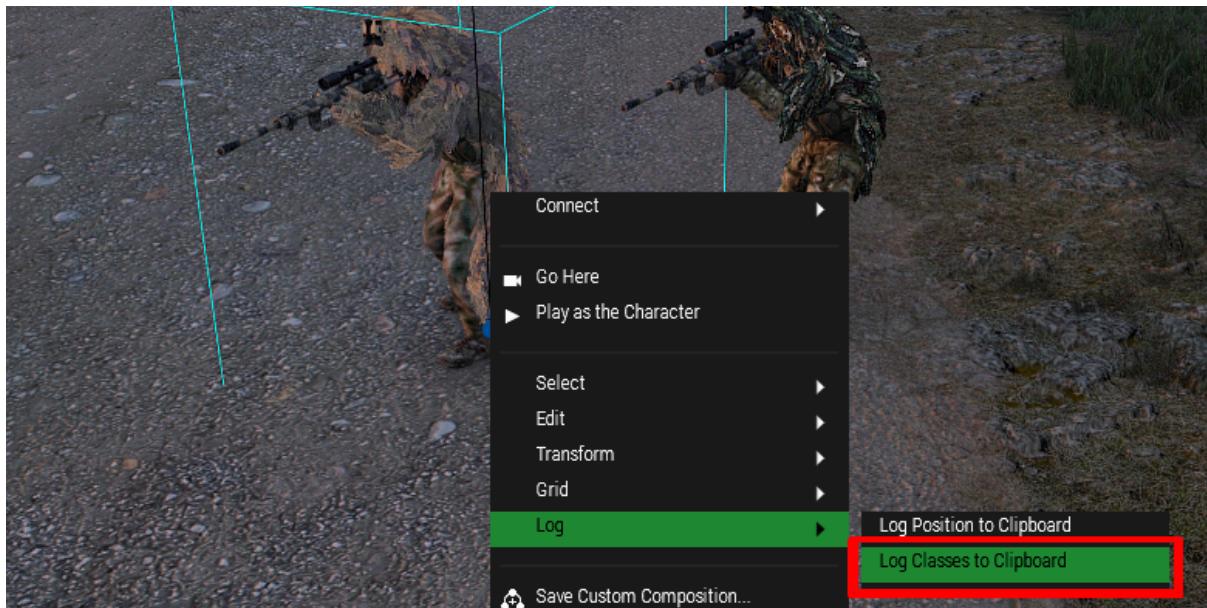
Important: about Arma 3 logic, Civilians don't fear Independent and their allies.

Define the weaponry and the side size

Now, let's request CSWR to spawn the [side allowed to do that](#). In the `fn_CSWR_population.sqf` file, search the section "SPawning GROUPS: BLUFOR", for example:

Step 1/3: select the weaponry

Crucial: before everything, make sure what kind of infantry weaponry you want to give to your groups because it's not safe to change weaponry in the customization section. You can custom whatever you want but weaponry is not a safety task against errors or inconsistencies. So to avoid headaches, start your army by selecting the soldiers you want in your *Eden Editors* and collecting their classnames.



Collecting the classname of a soldier through *Eden Editor*. Next, use **CTRL+V** [on your script editor](#).

Tip for further: you will see [how to customize your entire army loadouts](#).

Step 2/3: increase or not the size of each group-type

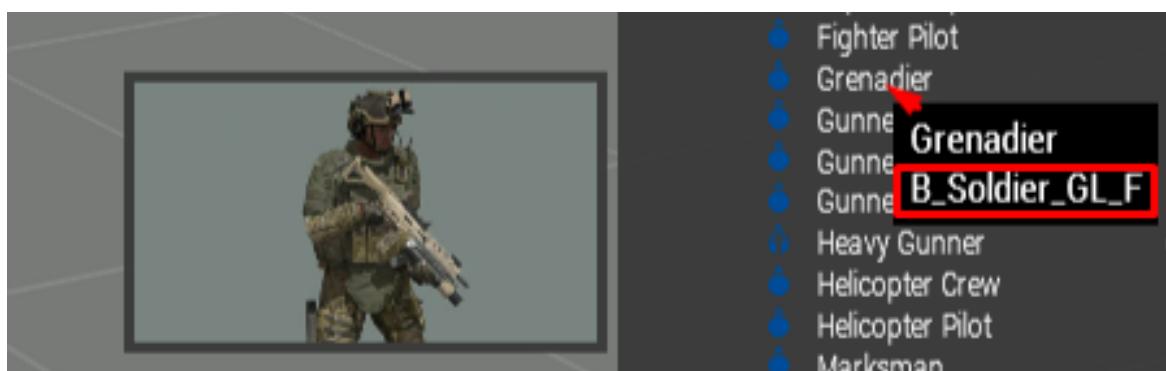
Now let's build your group types via `fn_CSWR_population.sqf` file:

```
fn_CSWR_population.sqf X
22 //DEFINING FACTION: BLUFOR
23 //Define the number of soldiers and vehicles of BluFor side.
24
25 //Vehicles
26 CSWR_vehicle_BLU_light = "B_G_Offroad_01_armed_F";
27 CSWR_vehicle_BLU_medium = "B_MRAP_01_hmg_F";
28 CSWR_vehicle_BLU_heavy = "B_MBT_01_TUSK_F";
29 CSWR_vehicle_BLU_custom_1 = "";
30 CSWR_vehicle_BLU_custom_2 = "";
31 CSWR_vehicle_BLU_custom_3 = "";
32
33 //Soldiers groups
34 CSWR_team_BLU_light = ["B_Soldier_TL_F", "B_Soldier_F"];
35 CSWR_team_BLU_medium = ["B_Soldier_TL_F", "B_Soldier_F", "B_Soldier_F"];
36 CSWR_team_BLU_heavy = ["B_Soldier_TL_F", "B_Soldier_F", "B_Soldier_F"];
```

Group/team of soldiers supports many types of soldiers in a single group. Feel free to customize the type of soldier you want for each group “template”. Those green names are “classnames” and they are unique from *Arma 3*, *Arma 3 DLC*, and *Arma 3* mods like *RHS* and *CUP*. Each type of soldier on each side has its own classname.

```
THIS IS A CLASSNAME
soldiers
adLightBLU = ["B_Soldier_TL_F", "B_Soldier_F"];
adRegularBLU = ["B_Soldier_TL_F", "B_Soldier_F", "B_Soldier_F"];
adHeavyBLU = ["B_Soldier_TL_F", "B_Soldier_F", "B_Soldier_F"];
adCustomOneBLU = [];
adCustomTwoBLU = [];
```

On *Eden Editor*, drag and drop the unit or vehicle you want to use on CSWR, copying the classname and pasting that into the group you’re building like the image above. Here is another example of figuring out the classname of something through the `Assets` menu:



Important 1: feel free to use any unit/vehicle classname from mods you want.

Important 2: never use vehicle classnames in groups of soldiers, as well as the opposite. Example of what not to do:

```
→ // Soldier groups  
→ CSWR_team_BLU_light = ["B_Soldier_TL_F", "B_Soldier_F", "B_G_Offroad_01_armed_F"];
```

Arma 3 doesn't work well when a single group is composed of infantry and vehicles together.

Step 3/3: define how many groups will spawn

Each line represents a group you want to spawn. You will see the vehicle groups separated from the soldier's groups, but the logic is the same.

```
fn_CSWR_population.sqf 1, M ×  
67 // Soldier groups  
68 // [Side; [Spawns; CSWR_spwnsBLU, CSWR_spwnsVehBLU, CSWR_spwnsParadropBLU]; Group type; Group formation; _form_BLU_1, _form_BLU_2]  
69 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_sniper, _form_BLU_1, _be_COMBAT, [_move_WATCH], []] call THY_fnc_CSWR_add_group;  
70 [BLUFOR, [CSWR_spwnsBLU, "A"], CSWR_group_BLU_light, _form_BLU_2, _be_SAFE, [_move_RESTRICTED, "A"], []] call THY_fnc_CSWR_add_group;  
71 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_2, _be_AWARE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
72 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_COMBAT, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
73 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_COMBAT, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
74 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
75 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
76 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
77 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
78 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;
```

Change (or not) the group type in each line:

```
fn_CSWR_population.sqf 1, M ×  
64 // Soldier groups  
65 // [Side; [Spawns; CSWR_spwnsBLU, CSWR_spwnsVehBLU, CSWR_spwnsParadropBLU]; Group type; Group formation; _form_BLU_1, _form_BLU_2]  
66 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_sniper, _form_BLU_1, _be_COMBAT, [_move_WATCH], []] call THY_fnc_CSWR_add_group;  
67 [BLUFOR, [CSWR_spwnsBLU, "A"], CSWR_group_BLU_light, _form_BLU_2, _be_SAFE, [_move_RESTRICTED, "A"], []] call THY_fnc_CSWR_add_group;  
68 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_2, _be_AWARE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
69 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_COMBAT, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
70 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_COMBAT, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
71 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
72 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
73 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
74 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
75 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
76 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
77 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;  
78 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc_CSWR_add_group;
```

Some of the options when groups of BLU side:

- CSWR_group_BLU_light
- CSWR_group_BLU_medium
- CSWR_group_BLU_heavy

Some of the options when vehicles of BLU side:

- CSWR_vehicle_BLU_light
- CSWR_vehicle_BLU_medium
- CSWR_vehicle_BLU_heavy

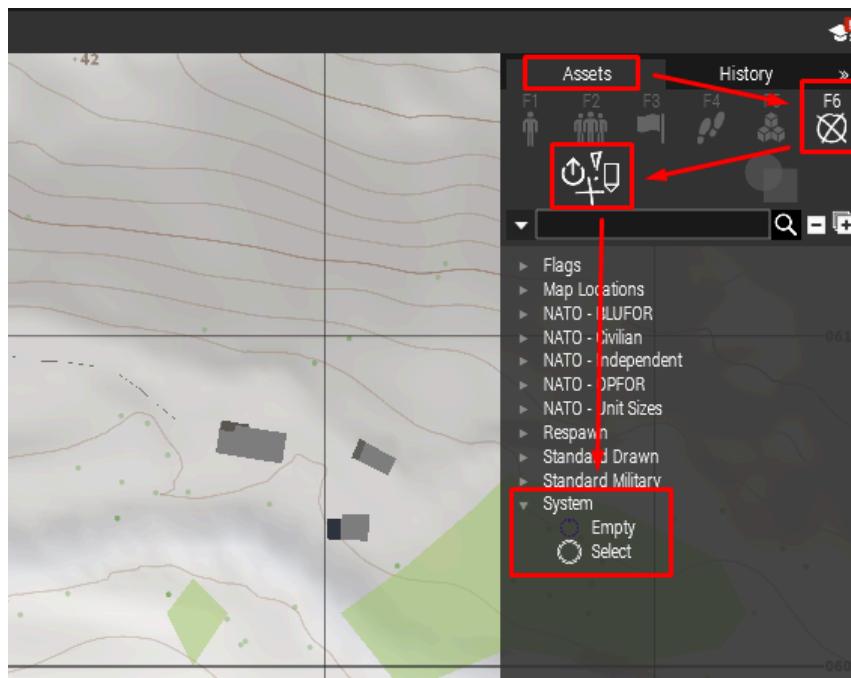
Important: be careful to not abuse the number of the lines. [It impacts server performance directly.](#)

Spawn-points for side

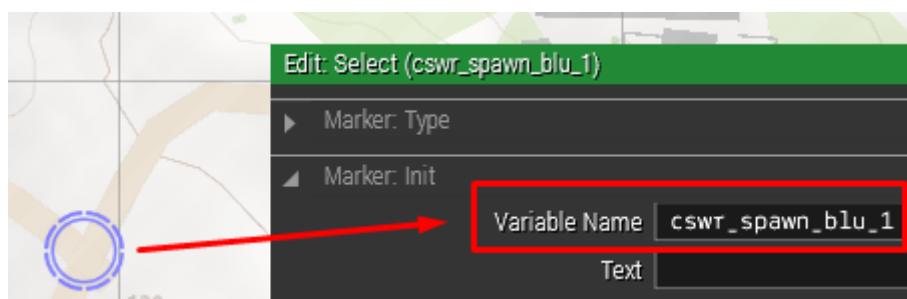
In CSWR, the AI will get also some pre-defined markers as spawn-points which they will use only one type of spawn in two ways: 1) right before the mission gets started; or 2) using the [Spawn Delay methods](#). The regular spawn can take a while like 1 or 2 minutes to be fully finished, depending on how many units and vehicles CSWR will create for you. Each side should have its own spawn-points, never sharing them with another side. If you apply more than one spawn-point of the same type, each group/vehicle assigned to spawn there will select one of those options randomly.

Define the spawn-points on *Eden Editor*, dropping a “Select” marker on the map and name it based on the example down below, respecting the tag of what side(s) you got enabled in [*fn_CSWR_management.sqf*](#) previously.

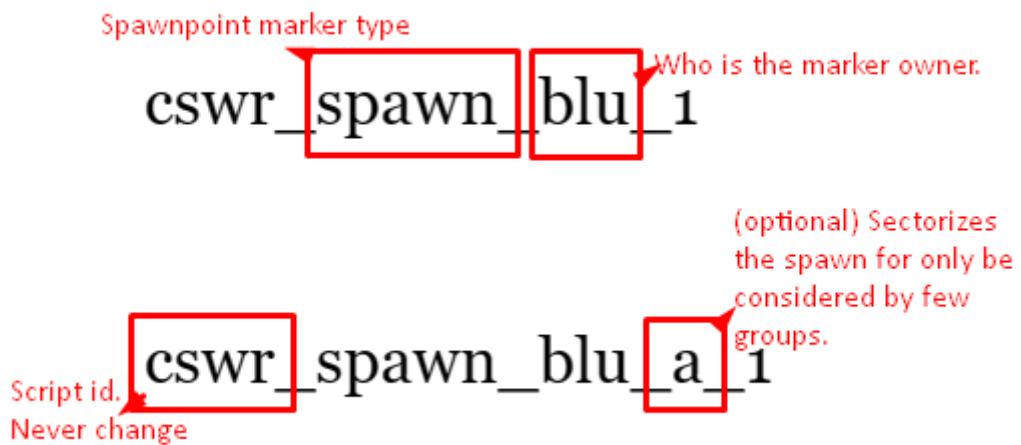
To find the “Select” marker on *Eden*, in the **Assets** menu, click on “**Markers**” (F6) and, after that, “**System**” on the category list. There, you will find out “**Select**” marker.



Above, the Asset-Browser on *Eden Editor*.



Above, a spawn-point for the BLU side. This is the marker attribute on *Eden Editor*.



Once you set the spawn-points in your mission, you're ready to rename each of them with one of these options ("Spawn-type names" column):

Can spawn:	Description:	On Eden Editor Spawn-type names: XXX = Side tag (blu, opf, ind, civ) / A = any sector letter
Groups can spawn Ground vehicles can spawn Helicopters cannot spawn	Only those are not sectorized.	<ul style="list-style-type: none"> • cswr_spawn_XXX_1 • cswr_spawn_XXX_2 ...
	Only those are sectorized.	<ul style="list-style-type: none"> • cswr_spawn_XXX_A_1 • cswr_spawn_XXX_A_2 ...
Ground vehicles can spawn Helicopters cannot spawn Groups cannot spawn	Only those are not sectorized.	<ul style="list-style-type: none"> • cswr_spawnveh_XXX_1 • cswr_spawnveh_XXX_2 ...
	Only those are sectorized.	<ul style="list-style-type: none"> • cswr_spawnveh_XXX_A_1 • cswr_spawnveh_XXX_A_2 ...
Helicopters can spawn Ground vehicles cannot spawn Groups cannot spawn	Only those are not sectorized.	<ul style="list-style-type: none"> • cswr_spawnheli_XXX_1 • cswr_spawnheli_XXX_2 ...
	Only those are sectorized.	<ul style="list-style-type: none"> • cswr_spawnheli_XXX_A_1 • cswr_spawnheli_XXX_A_2 ...
Groups can spawn Ground vehicles can spawn Helicopters cannot spawn CIV ground veh. cannot spawn	Only those are not sectorized.	<ul style="list-style-type: none"> • cswr_spawnparadrop_XXX_1 • cswr_spawnparadrop_XXX_2 ...

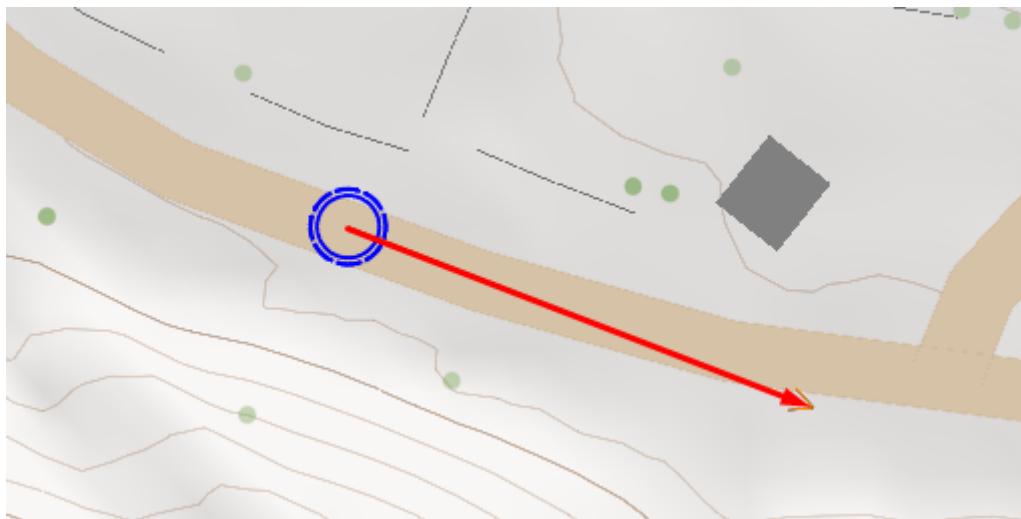
	Only those are sectorized.	<ul style="list-style-type: none"> • cswr_spawnparadrop_XXX_A_1 • cswr_spawnparadrop_XXX_A_2 ...
--	----------------------------	---

Of course, you might add or remove spawn-points as much as you want since you don't forget to [make sure the side is true](#) on the `fn_CSWR_management.sqf` file.

Set the spawn direction

Get used to setting the spawn direction:

1. Toggle the map to 2D view, pressing "M";
2. [Pressing Left-Shift](#), hold down the main mouse button over the marker;
3. Adjust the direction the vehicles should be facing right after the spawn.



Important: despite groups of soldiers/people not caring about the spawn marker direction, vehicles always will consider that.

Spawn-points > Working with sectors

To sectorize spawns of groups and vehicles is super easy. If you want to make, for example, only 2 groups of all your army spawn in a specific region, you need to use sectors in some of your spawn-point markers.

csWR_spawn_blu_1
csWR_spawn_blu_a_1

Examples of a regular and sectorized spawn marker name on *Eden Editor*.

```
[BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_RESTRICTED], []]
```

Above, the group will spawn in some of the side non-sectorized spawn-points.

```
[BLUFOR, [CSWR_spwnsBLU, "A"], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_RESTRICTED], []]
```

In this case above, the group will spawn only through those sectorized spawn-points with "A" letter.

You can use the English alphabet letters to set a sector.

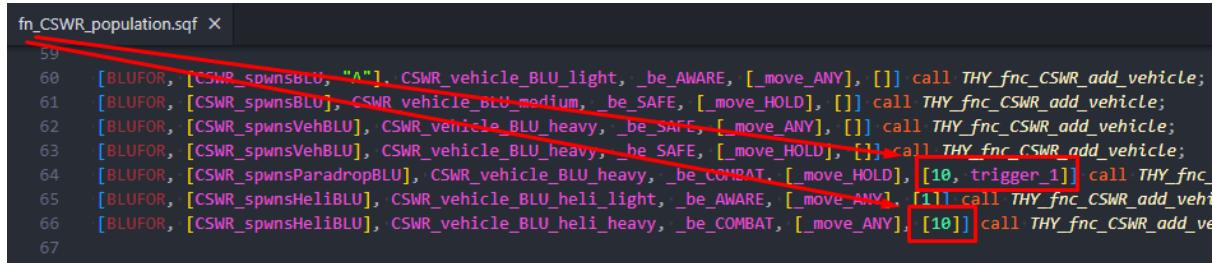
Important 1: groups and vehicles with no sector defined will be randomized only through spawns that are NOT using sectors.

Important 2: it's not possible to set more than one sector to a group/vehicle.

Spawn-points > Spawn Delay methods

If you want to make the groups and/or vehicles of one or more sides spawn lately in the mission, you can use 3 spawn delay methods to trigger those late spawns.

In the `fn_CSWR_population.sqf` file, through the *Spawning groups' section*, the last array “[]” of each group or vehicle is responsible for setting which method (or methods) the group will execute in the game.



```
fn_CSWR_population.sqf X
59
60  [BLUFOR, [CSWR_spwnsBLU, "A"], CSWR_vehicle_BLU_light, _be_AWARE, [_move_ANY], [] call THY_fnc_CSWR_add_vehicle;
61  [BLUFOR, [CSWR_spwnsBLU], CSWR_vehicle_BLU_medium, _be_SAFE, [_move_HOLD], [] call THY_fnc_CSWR_add_vehicle;
62  [BLUFOR, [CSWR_spwnsVehBLU], CSWR_vehicle_BLU_heavy, _be_SAFE, [_move_ANY], [] call THY_fnc_CSWR_add_vehicle;
63  [BLUFOR, [CSWR_spwnsVehBLU], CSWR_vehicle_BLU_heavy, _be_SAFE, [_move_HOLD], {} call THY_fnc_CSWR_add_vehicle;
64  [BLUFOR, [CSWR_spwnsParadropBLU], CSWR_vehicle_BLU_heavy, _be_COMBAT, [_move_HOLD], [10, trigger_1] call THY_fnc_CSWR_add_vehicle;
65  [BLUFOR, [CSWR_spwnsHeliBLU], CSWR_vehicle_BLU_heli_light, _be_AWARE, [_move_ANY], [11] call THY_fnc_CSWR_add_vehicle;
66  [BLUFOR, [CSWR_spwnsHeliBLU], CSWR_vehicle_BLU_heli_heavy, _be_COMBAT, [_move_ANY], [18] call THY_fnc_CSWR_add_vehicle;
```

Editing the Spawn Delay of BluFor side in `fn_CSWR_population.sqf` file.

So in that file, use these options (“Spawn Delay Config” column below) if you want to set some delay for some groups/vehicles:

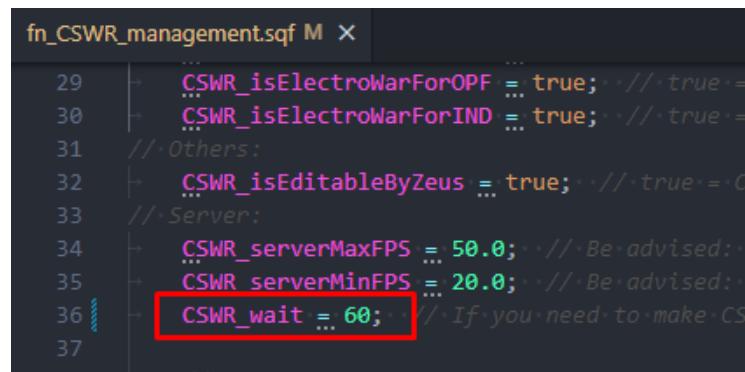
Method:	Description:	In Population file Spawn Delay Config: possibilities in each group/vehicle line
Timer	Define a unique integer or float number (in minutes) that represents the time for the group to execute its spawn later in the mission. This time begins to count not when the mission gets started but when the group is requested to be created by CSWR instead.	[10] Or [selectRandom[10,20,30,...]]
Trigger	Define one or more triggers on <i>Eden Editor</i> that, when activated, allow the group to execute its spawn later in the mission. Trigger names can be anything as long as they match the <i>Population</i> file and the trigger names configured by you on <i>Eden</i> .	[my_trigger_1] Or [my_trigger_1, my_trigger_2]
Target	Define one or more objects (soldiers/people/building) on <i>Eden Editor</i> that, when killed or destroyed, allowed the group to execute its spawn later in the mission.	[target_1] Or [target_1, target_2, target_3]

	Object names can be anything as long as they match the <i>Population</i> file and the object names configured by you on Eden.	
Timer + Trigger	Mix of methods. The first one reached allows the spawn immediately.	[10, my_trigger_1] Or [selectRandom[30,90], my_trigger_1] Or [10, my_trigger_1, my_trigger_2]
Timer + Target	Mix of methods. The first one reached allows the spawn immediately.	[5, target_1] or [5, target_1, target_2]
Timer + Trigger + Target	Mix of methods. The first one reached allows the spawn immediately.	[90, my_trigger_1, target_1] or [90, my_trigger_1, target_1, target_2]
Trigger + Target	Mix of methods. The first one reached allows the spawn immediately.	[trigger_1, my_trigger_1]
No Spawn delay	Set the group/vehicle to spawn right after the mission starts. Default!	[]

Important: for groups and vehicles with more than one Spawn Delay method, it's important to know that the first method reached to allow the spawn will ignore the other methods configured for that group, allowing the group to spawn immediately.

But I want to delay all units

In this case, you can use the *CSWR_wait* to delay when the script will be triggered by the server in-game. After this time the CSWR will initiate its functions normally.



```

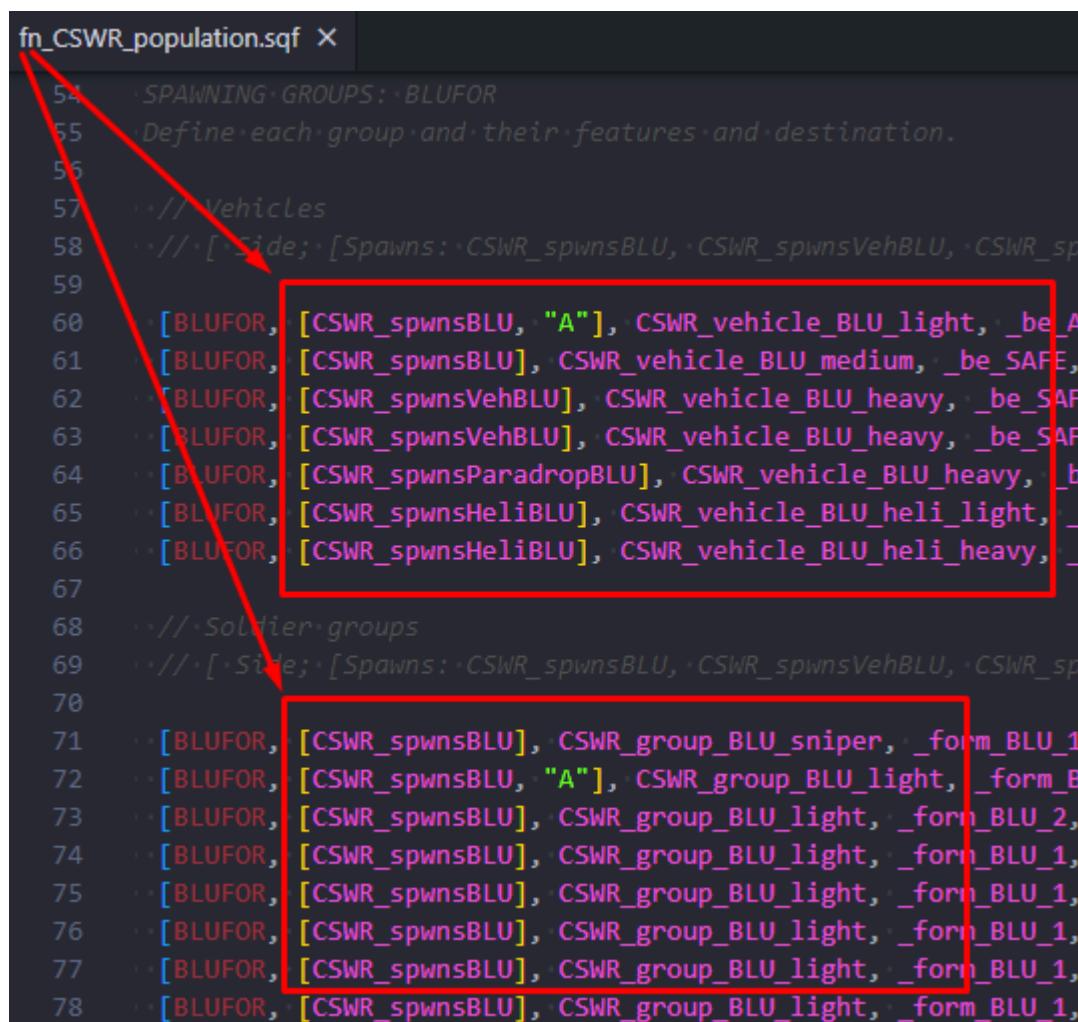
fn_CSWR_management.sqf M ×
29 →   CSWR_isElectroWarForOPF = true; //true
30 →   CSWR_isElectroWarForIND = true; //true
31 //Others:
32 →   CSWR_isEditableByZeus = true; //true
33 //Server:
34 →   CSWR_serverMaxFPS = 50.0; //Be advised:
35 →   CSWR_serverMinFPS = 20.0; //Be advised:
36 →   CSWR_wait = 60; //If you need to make CS
37

```

CSWR will start its functions after 60 **seconds** (not minutes here) later once the mission gets started.

Which group-type is allowed for each spawn-point-type

Initially, the first versions of CSWR soldiers and vehicles spawned at the same spawn-points. Over time, CSWR has incorporated specific spawn-points for vehicles to expand the possibilities for mission editors if the editor desires a starting point from which only ground vehicles will depart. As of v5, CSWR also has exclusive spawn-points for helicopters and paratroopers. With so many options, it was necessary to include some rules about which groups-types (units or vehicles) can take advantage of which types of spawn-points.



```
fn_CSWR_population.sqf X

54  SPAWNING_GROUPS: BLUFOR
55  Define each group and their features and destination.
56
57  // Vehicles
58  // Side; [Spawns: CSWR_spwnsBLU, CSWR_spwnsVehBLU, CSWR_sp
59
60  [BLUFOR, [CSWR_spwnsBLU, "A"], CSWR_vehicle_BLU_light, _be_A
61  [BLUFOR, [CSWR_spwnsBLU], CSWR_vehicle_BLU_medium, _be_SAFE,
62  [BLUFOR, [CSWR_spwnsVehBLU], CSWR_vehicle_BLU_heavy, _be_SAF
63  [BLUFOR, [CSWR_spwnsVehBLU], CSWR_vehicle_BLU_heavy, _be_SAF
64  [BLUFOR, [CSWR_spwnsParadropBLU], CSWR_vehicle_BLU_heavy, _b
65  [BLUFOR, [CSWR_spwnsHeliBLU], CSWR_vehicle_BLU_heli_light, _b
66  [BLUFOR, [CSWR_spwnsHeliBLU], CSWR_vehicle_BLU_heli_heavy, _b
67
68  // Soldier groups
69  // Side; [Spawns: CSWR_spwnsBLU, CSWR_spwnsVehBLU, CSWR_sp
70
71  [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_sniper, _form_BLU_1
72  [BLUFOR, [CSWR_spwnsBLU, "A"], CSWR_group_BLU_light, _form_B
73  [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_2,
74  [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1,
75  [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1,
76  [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1,
77  [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1,
78  [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1,
```

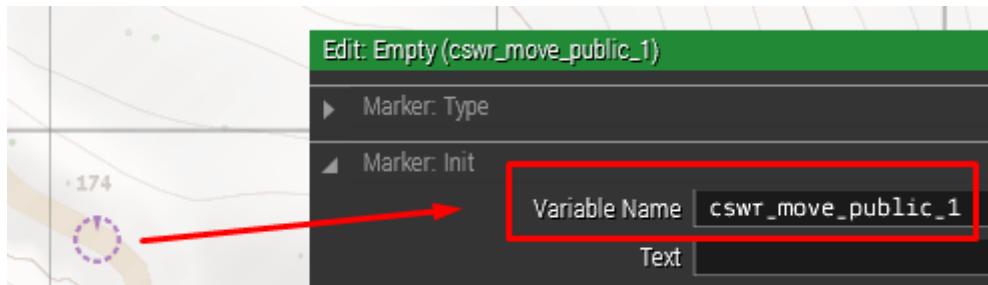
Editing the Spawn type of BLU side in *fn_CSWR_population.sqf* file.

Down below, you see all the possibilities to spawn each type of group for each Arma 3 side:

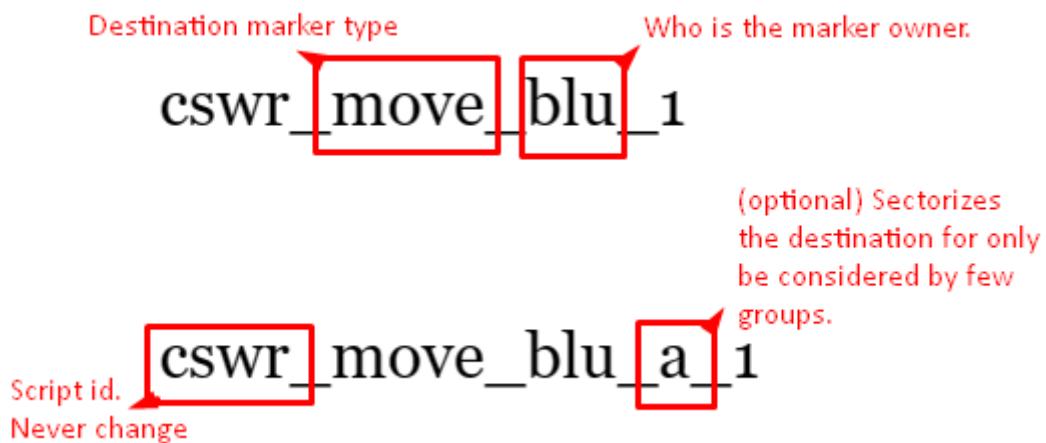
In Population file Spawn-point types allowed: XXX = Side tag	In Population file Group-types: XXX = Side tag (blu, opf, ind, civ)
<ul style="list-style-type: none"> • CSWR_spwnsXXX • CSWR_spwnsParadropXXX <p style="margin-top: 10px;">Groups of people are allowed to spawn on the ground and drop by air.</p>	CSWR_group_XXX_light CSWR_group_XXX_medium CSWR_group_XXX_heavy CSWR_group_XXX_custom_1 CSWR_group_XXX_custom_2 CSWR_group_XXX_custom_3 CSWR_group_XXX_sniper
<ul style="list-style-type: none"> • CSWR_spwnsXXX • CSWR_spwnsVehXXX • CSWR_spwnsParadropXXX <p style="margin-top: 10px;">Vehicles (with their crew) are allowed to spawn on the ground with people, in spawns exclusive for vehicles, and dropped by air*. (*) CIV vehicles CANNOT be spawned by air.</p>	CSWR_vehicle_XXX_light CSWR_vehicle_XXX_medium CSWR_vehicle_XXX_heavy CSWR_vehicle_XXX_custom_1 CSWR_vehicle_XXX_custom_2 CSWR_vehicle_XXX_custom_3
<ul style="list-style-type: none"> • CSWR_spwnsHeliXXX <p style="margin-top: 10px;">Helicopters (with their crew) are allowed to spawn only in exclusive helicopter spawns.</p>	CSWR_vehicle_XXX_heli_light CSWR_vehicle_XXX_heli_heavy

Destinations (waypoints)

In CSWR, there's almost no end-point, which means after the group spawns, it will move through the map directly to the pre-defined destinations (waypoints), randomly, and can take a break just a while before going to the next same destination type.



An example of destination marker and how *CSWR* recognizes that on *Eden Editor*.



Each side's group configured to spawn can have (or not) a different destination. All types of groups/vehicles (except sniper groups) will move forever from one marker to the other until the mission is over or the group has been eliminated.

Below, there are all kinds of destinations available for you name it on *Eden Editor* markers:

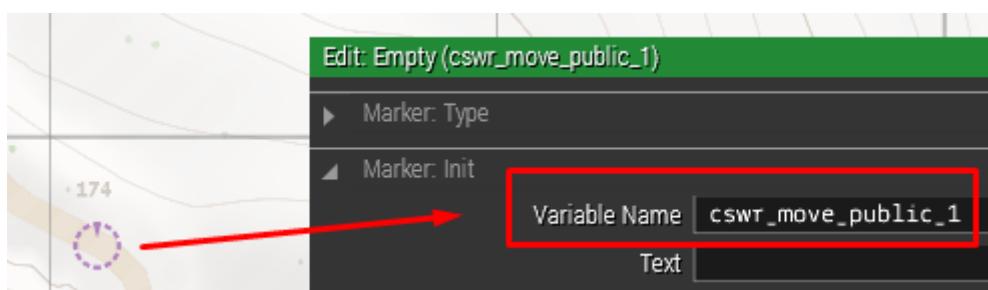
Destination types:	Description:	On Eden Editor Destination marker names: XXX = Side tag (blu, opf, ind, civ) 1 = Any number / A = any sector letter
Move markers	Move destinations are the simplest markers used in CSWR. A group of soldiers or a lone vehicle using it, after spawning, will go to one of them randomly, spend a while, and move on to the next marker of the	<ul style="list-style-type: none">• <code>cswr_move_public_1</code>• <code>cswr_move_public_A_1</code> <p>Public move destination is used by everyone assigned to use that, including civilians.</p> <ul style="list-style-type: none">• <code>cswr_move_XXX_1</code>

	<p>same type. If the move has a sector-letter in the name, the group will move only to move-destinations of the same sector. There are 2 types of move markers: public and restricted.</p>	<ul style="list-style-type: none"> • cswr_move_XXX_A_1 <p>Restricted-move-destination (it's when a move marker has a side tag declared) is used only by the side assigned in the marker name.</p> <p>Civilians cannot use this.</p>
Hold markers	<p>Hold destinations are markers with the goal is AI holding the position for a long time. The hold marker was designed mainly for tracked vehicles (tank, IFV) that, after the spawn, will go to the hold marker and execute the hold-maneuver <u>almost exactly over the marker position and heading for the hold marker direction</u> (configured on Eden). Soldiers using hold destinations always will avoid being over the roads/streets and never in the hold-marker center position (reserved for tracked vehicles). After spawning, the hold-groups will go to one of them randomly, spend a long time there, and then move on to the next same type marker. If the hold has a sector-letter in the marker name, the group will hold only to hold-destinations of the same sector.</p> <p>It's advised to not use hold markers over the roads/streets to avoid vehicle crashes, nor on very rough terrain.</p>	<ul style="list-style-type: none"> • cswr_hold_XXX_1 • cswr_hold_XXX_A_1 <p>Hold destination is used only by the side assigned in the marker name.</p> <p>Helicopters cannot use this.</p>
Watch markers	<p>Watch destinations are markers designed exclusively for soldiers (not vehicles) of <u>sniper groups with one or two units at maximum</u>. Once the spawning runs, the sniper team assigned with a watch destination will go around the marker (1000m by default), looking for high spots with some vision of the marker position. After the group's arrival, the sniper group will stay there (never changing the location) until</p>	<ul style="list-style-type: none"> • cswr_watch_XXX_1 • cswr_watch_XXX_A_1 <p>Watch destination is used only by the side assigned in the marker name.</p> <p>Vehicles cannot use this. Civilians cannot use this. Non-sniper-group cannot use this.</p>

	<p>the mission ends. If the watch has a sector-letter in the name, the group will watch only to watch-destinations of the same sector.</p>	
<u>Occupy markers</u>	<p>Occupy destinations are markers where a group of soldiers (<u>at a max of six units</u>) will get inside one of all available buildings inside a range (of 200m by default) from the marker center. Once inside the building, the group will stay there <u>maybe</u> for a long time, and if there's an enemy around, the group will fight from inside and not leave their covered position. If the occupy has a sector-letter in the name, the group will occupy only occupy-destinations of the same sector.</p>	<ul style="list-style-type: none"> • cswr_occupy_XXX_1 • cswr_occupy_XXX_A_1 <p>Occupy destination used by Blufor groups assigned to use that.</p> <p>Vehicles cannot use this.</p>

Step 1/2: including the destinations on *Eden Editor*

On *Eden Editor*, drop an “Empty” marker on the map, rename it with one of those examples from the table above, and with *Copy & Paste*, add how many markers (destinations of that type) your mission asks because the numeral of the markers name will increase automatically:

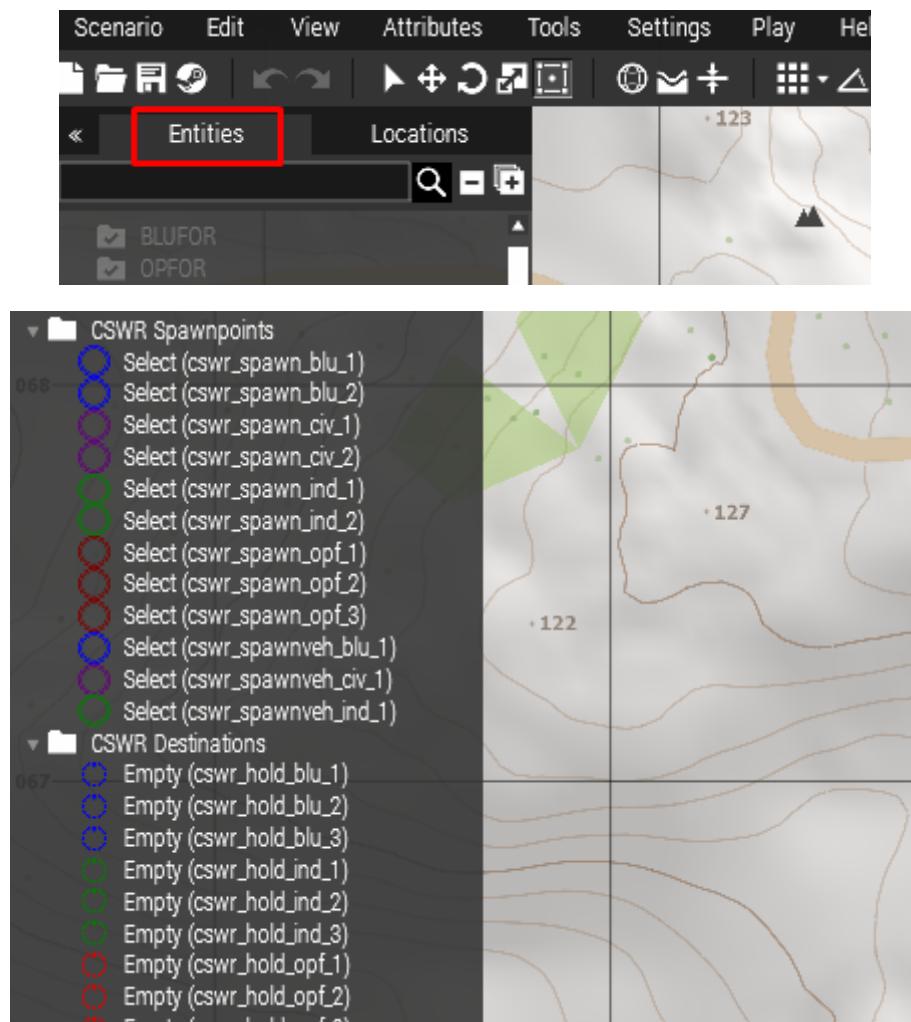


An example of a destination marker and its attributes on *Eden Editor*.

To find in *Eden Editor* “Empty” marker, in the Assets menu, click on “Markers” (F6) and, after that, “System” on the category list. There, you will find out “Empty” marker.

Drag and drop the marker where you want troops and vehicles going through. Soon you will understand each section of the marker's name structure.

A tip for you: use the Entities menu to organize what markers are spawn and what of them are destinations:



Step 2/2: define where each spawned group will move

Open the file: `fn_CSWR_population.sqf`

```
fn_CSWR_population.sqf ×
67
68 // Soldier groups
69 // [ Side; [Spawns; [CSWR_spwnsBLU, CSWR_spwnsVehBLU, CSWR_spwnsParadropBLU]; Group type; Group format
70
71 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_sniper, _form_BLU_1, _be_COMBAT, [_move_WATCH], []] call THY_fn
72 [BLUFOR, [CSWR_spwnsBLU, "A"], CSWR_group_BLU_light, _form_BLU_2, _be_SAFE, [_move_RESTRICTED, "A"], []
73 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_2, _be_AWARE, [_move_ANY], []] call THY_fn
74 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_COMBAT, [_move_ANY], []] call THY_fn
75 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_COMBAT, [_move_ANY], []] call THY_fn
76 [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fn
```

Above, each line is a Blufor group of soldiers. In the column before the last one, you define with type of destinations that the group must be assigned. Below, are the options (Destination types column):

Debug Dest. classes:	In Population file Destination types:	Description:
Common destinations	_move_ANY	<p>Group assigned for this goes to any common destination marker, including restricted move markers from other sides, no matter whether the destination-markers are sectorized or not.</p> <p>It's recommended if you want to provoke clashes as fast as possible between two or more sides spawned by CSWR.</p> <p>Sectorizing is not allowed.</p> <p>Civilians cannot use this.</p>
	_move_PUBLIC	<p>Group assigned for this goes to where anyone can go.</p> <p>Highly recommended to be used mainly for civilian side.</p> <p>Sectorizing is allowed.</p>
	_move_RESTRICTED	<p>Group assigned for this goes only to restricted destination markers of their own side.</p> <p>Sectorizing is allowed.</p> <p>Civilians cannot use this.</p>
Specialized destinations	<u>move_OCCUPY</u>	<p>Group assigned for this goes only to occupy destination markers of their own side.</p> <p>Sectorizing is allowed.</p> <p>Vehicles cannot use this.</p>
	<u>move_WATCH</u>	<p>Group assigned for this goes only to one watch destination marker of their own side.</p> <p>Sectorizing is allowed.</p> <p>Vehicles cannot use this.</p> <p>Civilians cannot use this.</p> <p>Non-sniper-groups cannot use this.</p>
	<u>move_HOLD</u>	<p>Group assigned for this goes only to hold destination markers of their own side.</p> <p>Highly recommended to be used mainly for tracked-vehicles as MTBs and IFVs.</p> <p>Sectorizing is allowed.</p>

Example of a destination type (occupy) sectorized ("A"):

```
[BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_2, _be_COMBAT, [_move_OCCUPY, "A"]]
```

Can the groups move free, without a pre-defined destination?

No. In CSWR the focus is the Editor has control of the areas where the troops should move through with a good level of unpredictability.

Destinations > Hold-marker

Helicopters cannot use this.

Hold-markers make units, mainly tracked vehicles, hold in that position and stay there for long periods, meanwhile, tracked vehicles always will take place in the center of the marker and face a specific direction configured by the Mission Editor. From all destinations available, only hold-markers are able to consider the *Eden* marker's direction:

1. Toggle the map to 2D view, pressing “M”;
2. Pressing Left-Shift, hold down the main mouse button over the marker;
3. Adjust the direction the tracked-vehicles must face right after their arrival.



Above, an example of a hold-marker facing to the northeast on *Eden Editor*.



Above, (when Debug is `true` through Zeus view) the same example in-game a tank holds that position.



Above, in-game the same example.

Important: groups of soldiers/people and non-tracked-vehicles don't care about the destination marker direction.

Destinations > Watch-marker

Vehicles cannot use this.

Civilians cannot use this.

Non-sniper-groups cannot use this.

The watch-marker was designed specifically for Sniper groups (max of 2 members). When the group spawns, and is assigned to go to a watch destination, first the group will check on the map if there are highlands around the target zone (position of *Eden* watch-marker itself) and if there is a clear visual of the target zone. After that, the group will choose the best position possible to maintain that zone under watch for the whole game, never changing their position again.



Above, an example of a watch-marker (target zone) in *Eden Editor*.



Above, same example but in-game when Debug is `true`. That blue dot is the highland found around.



Above, sniper group behavior when with two members watching the zone.

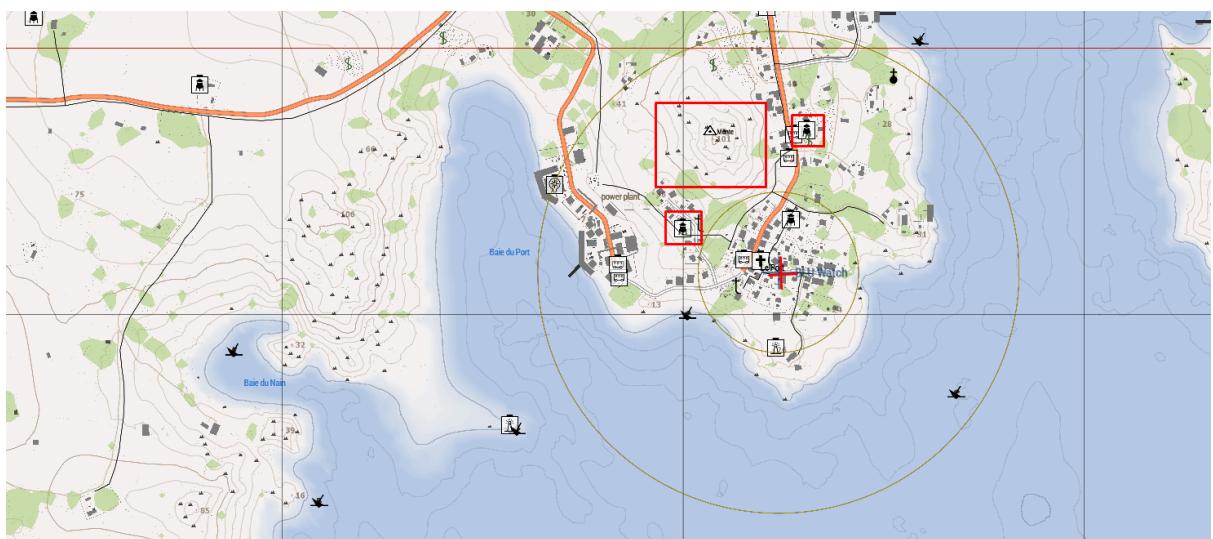
Important 1: if no good wild spot is found, the watcher-group will try to find an urban tower with a satisfactory view of the area to be watched.



Above, the watcher group in a tower, watching the zone right after opening the windows.

All acceptable towers that the watcher group will consider are configured as classnames' list called `CSWR_acceptableTowersForWatch` in `fn_CSWR_management.sqf` file.

Important 2: originally, the watcher group will look for spots between the range of 200 to 600m, ignoring all wild and urban spots out of that. To force the watcher-group to stay in urban areas only, you could decrease the maximum range value by looking for the `CSWR_watchMkrRange` variable in the `fn_CSWR_management.sqf` file.



Above, red squares are the potential watcher group positions, never in the small circle, never out of the big one.

Important 3: even if you set acceptable towers inside the correct range from the target zone if the tower is not high enough compared to the target zone, the watcher group will ignore that building.

Destinations > Occupy-marker

Vehicles cannot use this.

The occupy-marker is exclusive for groups of soldiers/people (with a max of 6 members). As smaller the group is, easier to find a good building to cover all group members inside. When the group spawns, and is assigned to go to an occupied destination, first the group will check on the map the options around the occupy-marker and will go there, staying there for a long time before changing to another building. If the enemy is spotted, the group doesn't leave the building until the tread has been neutralized.



Above, an example of an occupy-marker in *Eden Editor*.

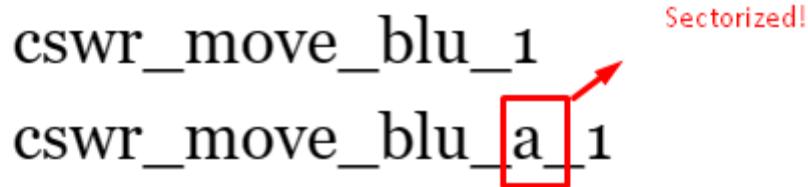


Above, through Zeus' view, the group members spread out inside the chosen building.

Important: if no good buildings are found or all buildings around the occupy-markers are destroyed, the group will stand still in its current position forever. To avoid this behavior at least right after the mission gets started, make sure you are setting the occupy-marker in urban areas or places with one or more buildings around. By default, CSWR will search those places 200m in all directions from the occupy-marker selected by the group. To change this value, look for the `CSWR_occupyMkrRange` variable in `fn_CSWR_management.sqf`.

Destinations > Working with sectors

To sectorize destinations of groups and vehicles is super easy. If you want to make, for example, only 2 groups of your army go straight to a specific region, you need to use sectors in some of your destination markers.



Examples of a regular and sectorized move-destination marker on *Eden Editor*.

```
[BLUFOR, -[CSWR_spwnsBLU], CSWR_group_BLU_light, -_form_BLU_1, -_be_SAFE, -[_move_RESTRICTED], -[]]
```

Above, the group will go to one of multiple exclusive side-destinations through the map.

```
[BLUFOR, -[CSWR_spwnsBLU], CSWR_group_BLU_light, -_form_BLU_1, -_be_SAFE, -[_move_RESTRICTED, "A"], []]
```

In this example above, the group will go only to those exclusive side-destinations sectorized with “A” letter. You can use the English alphabet letters to set a sector.

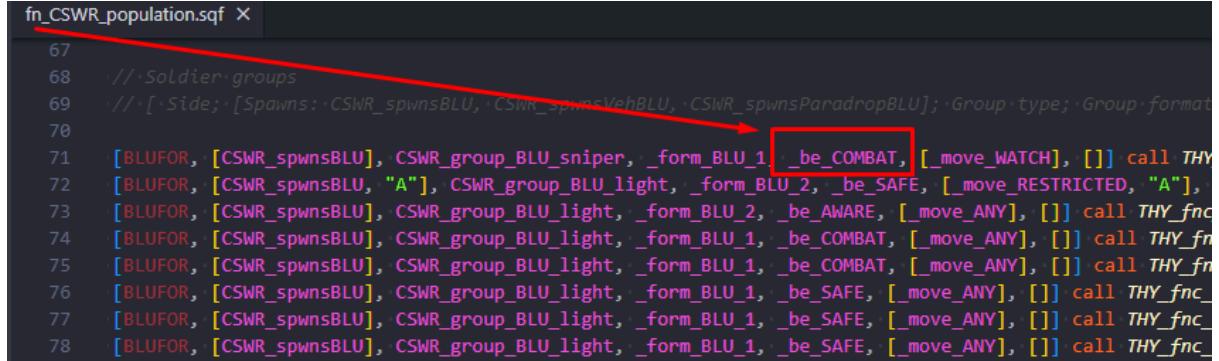
Important 1: groups and vehicles with no sector defined will be randomized only through destinations that are NOT using sectors.

Important 2: it's not possible to set more than one sector to a group/vehicle.

Behavior for each group

CSWR allows the Mission Editor to define the behavior of each group/vehicle spawned through the `fn_CSWR_population.sqf` file.

For example, go to the “SPAWNING GROUPS: BLUFOR” section and check it out:



```
fn_CSWR_population.sqf X
67
68    // Soldier groups
69    // [Side; [Spawns; CSWR_spwnsBLU, CSWR_spwnsVehBLU, CSWR_spwnsParadropBLU]; Group type; Group format
70
71    [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_sniper, _form_BLU_1, _be_COMBAT, [_move_WATCH], []] call THY_fnc
72    [BLUFOR, [CSWR_spwnsBLU, "A"], CSWR_group_BLU_light, _form_BLU_2, _be_SAFE, [_move_RESTRICTED, "A"], []
73    [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_2, _be_AWARE, [_move_ANY], []] call THY_fnc
74    [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_COMBAT, [_move_ANY], []] call THY_fnc
75    [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_COMBAT, [_move_ANY], []] call THY_fnc
76    [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc
77    [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc
78    [BLUFOR, [CSWR_spwnsBLU], CSWR_group_BLU_light, _form_BLU_1, _be_SAFE, [_move_ANY], []] call THY_fnc
```

Above, each line represents a Blufor group that will spawn, and each group has its personality configured for all game long. Below are the options (Behavior types column) you might use for one or more groups, up to you:

In Population file Behavior types:	Description:
_be_SAFE	<ul style="list-style-type: none">Group walking with the guns low, looking around, and feeling safe.Vehicle slow and on roads. If a tank or another vehicle with turret, the crew out into the hatch.
_be_AWARE	<ul style="list-style-type: none">Group walking but with guns ready, looking around and concerned.Vehicle slow and on roads, but concerned with the crew inside when a tank or another vehicle with a turret.
_be_COMBAT	<ul style="list-style-type: none">Group in combat mode, running, ready to fight and keep the formation.Vehicle in combat mode, running, ready to fight, leaving the roads to take shortcuts.
_be_STEALTH	<ul style="list-style-type: none">Group in combat mode, but much more in a cautious manner, moving always along the cover.Vehicle in combat mode, turning the lights off. Even if it's a heavy vehicle, it'll avoid knocking down trees.
_be_CHAOS	<ul style="list-style-type: none">Group full speed, ready to fight and lose the formation.Vehicle full speed, rides everywhere, firing at will.

Customizing the loadouts

You can easily customize each side loadout, setting some specific stuff for infantry, crewmen, snipers, and paratroopers. Yes, you can use stuff from mods easily too since the mod is loaded correctly in your server.

The easiest part of the customization is present in `fn_CSWR_management.sqf` file:

```
fn_CSWR_management.sqf X
22 // Loadout global:
23     CSWR_isBackpackForAllByFoot = false; // true == all units by foot (including
24     CSWR_isVestForAll          = false; // true == all units (including CIV) will
25 // Loadout by faction:
26     // Blu
27         CSWR_canNvgInfantryBLU = false; // true == BLU infantry/armoured will
28         CSWR_canNvgParatroopsBLU = true; // true == BLU paratroops will receive N
29         CSWR_canNvgSnipersBLU   = true; // true == BLU snipers will receive N
30         CSWR_nvgDeviceBLU       = "NVGoggles"; // Set the NightVision classname
31         CSWR_canFlashlightBLU  = true; // true == BLU units with no NVG will
32         CSWR_flashlightDeviceBLU = "acc_flashlight"; // Set the flashlight clas
33     // Opf
34         CSWR_canNvgInfantryOPF = false; // true == OPF infantry/armoured will
35         CSWR_canNvgParatroopsOPF = true; // true == OPF paratroops will receive N
36         CSWR_canNvgSnipersOPF   = true; // true == OPF snipers will receive N
37         CSWR_nvgDeviceOPF       = "NVGoggles_OPFOR"; // Set the NightVision cl
38         CSWR_canFlashlightOPF  = true; // true == OPF units with no NVG will
39         CSWR_flashlightDeviceOPF = "acc_flashlight"; // Set the flashlight clas
40     // Ind
41         CSWR_canNvgInfantryIND = false; // true == IND infantry/armoured will
42         CSWR_canNvgParatroopsIND = true; // true == IND paratroops will receive N
43         CSWR_canNvgSnipersIND   = true; // true == IND snipers will receive N
44         CSWR_nvgDeviceIND       = "NVGoggles_INDEP"; // Set the NightVision cl
45         CSWR_canFlashlightIND  = true; // true == IND units with no NVG will
46         CSWR_flashlightDeviceIND = "acc_flashlight"; // Set the flashlight clas
47     // civ
48         CSWR_canNvgCIV        = false; // true == CIV people will receive N
49         CSWR_nvgDeviceCIV      = "NVGoggles"; // Set the NightVision classname for
50 // end of customization
```

Loadout global:	If true:
<code>CSWR_isBackpackForAllByFoot</code>	<ul style="list-style-type: none">• Affects all sides, except Civilian.• It doesn't affect crewmen groups.
<code>CSWR_isVestForAll</code>	<ul style="list-style-type: none">• Affects all sides, except Civilian.

Before you jump into the advanced loadout customization, get to know some important points pre-defined automatically by CSWR:

Group classes: XXX = Side tag (blu, opf, ind, civ)	Loadout pre-defined details: Mandatory = you cannot remove that
Infantry <p>They're all groups you configured as XXX_light, XXX_medium, XXX_heavy, XXX_custom1, XXX_custom2 and XXX_custom3 in <i>fn_CSWR_population.sqf</i>. If it's a light ground vehicle, its crew will be infantry class too.</p>	<ul style="list-style-type: none"> ● Helmet customizable. ● Uniform customizable. ● Vest customizable. ● Backpack customizable. ● Goggles customizable. ● NVG semi-customizable. ● Flashlight semi-customizable. ● Items customizable. <p>All other classes inherit all or part of the infantry loadout.</p>
Infantry > Heavy Crew <p>They're created automatically when you configure a ground-heavy vehicle to spawn in <i>fn_CSWR_population.sqf</i>.</p>	<ul style="list-style-type: none"> ● Helmet customizable. ● Goggles customizable. ● Vest customizable. ● Uniform inherited from Infantry. ● Items inherited from Infantry. ● Backpack not available. ● NVG not available. ● Flashlight not available.
Infantry > Sniper groups <p>They're the group configured as CSWR_group_XXX_sniper in <i>fn_CSWR_population.sqf</i></p>	<ul style="list-style-type: none"> ● Rifle customizable. ● Rifle setup customizable. ● Uniform customizable. ● Helmet customizable. ● Goggles customizable. ● Backpack customizable. ● NVG semi-customizable. ● Vest customizable (mandatory). ● Binoculars customizable (mandatory). ● Items inherited from Infantry. ● Pistol no-customizable (mandatory); ● Flashlight not available.
Paratroopers <p>They're automatically all groups and light vehicles configured to spawn at CSWR_spwnsParadropXXX in <i>fn_CSWR_population.sqf</i></p>	<ul style="list-style-type: none"> ● Helmet customizable. ● Uniform customizable. ● NVG semi-customizable. ● Vest customizable (mandatory). ● Goggles customizable (mandatory). ● Flashlight inherited from Infantry. ● Items inherited from Infantry. ● Backpack not available yet (WIP). ● Helmet inherited from Inf. H. Crew.

<p>Paratroopers > Heavy Crew</p> <p>They're automatically all heavy vehicles configured to spawn at <code>CSWR_spwnsParadropXXX</code> in <code>fn_CSWR_population.sqf</code></p>	<ul style="list-style-type: none"> • Goggles inherited from Inf. H. Crew. • Uniform inherited from Paratroopers. • Items inherited from Infantry. • Backpack not available. • Flashlight not available.
<p>Paratroopers > Sniper groups</p> <p>They're automatically all heavy vehicles configured to spawn at <code>CSWR_spwnsParadropXXX</code> in <code>fn_CSWR_population.sqf</code></p>	<ul style="list-style-type: none"> • Items inherited from Infantry. • Rifle inherited from Inf. Sniper. • Rifle setup inherited from Inf. Sniper. • NVG* inherited from Inf. Sniper. • Pistol inherited from Inf. Sniper. • Vest inherited from Paratroopers. • Uniform inherited from Paratroopers. • Helmet inherited from Paratroopers. • Goggles inherited from Paratroopers. • Backpack not available yet (WIP). • Flashlight not available. <p>(*) Even when you remove the NVG from paratroopers, if they're Sniper groups too, and you allowed snipers to get NVG, sniper paratrooper groups will get NVG.</p>
<p>Air > Crew</p> <p>They're created automatically when you configure an air vehicle to spawn in <code>fn_CSWR_population.sqf</code></p>	<ul style="list-style-type: none"> • Not customizable yet!

Now you're prepared for the advanced customization. Open the file: `fn_CSWR_loadout.sqf`

```

//>BLU-INFANTRY-UNITS:
//>Loadout-replacement//empty("")>results.no-change.
[ 
    "",           //>infantry-uniform
    "H_Booniehat_tan", //>infantry-helmet
    "",           //>infantry-goggles
    "V_Chestrig_khk", //>infantry-vest-(one)
    "B_Carryall_cbr", //>infantry-backpack

    _unit, _grpType, _grpSpec, _tag] call THY_fnc_CSWR_U

//>BLU-PARATROOP-UNITS:
//>Loadout-replacement//empty("")>results.no-change.
[ 
    "U_B_CombatUniform_mcam_tshirt", //>Paratrooper
    "H_HelmetB_light_desert", //>Paratrooper
    "G_Lowprofile", //>Paratrooper
    "V_HarnessO_brown", //>Paratrooper

    _unit, _grpType, _grpSpec, _tag] call THY_fnc_CSWR_U

//>BLU-INF.-HEAVY-CREW-UNITS:
//>Loadout-replacement//empty("")>results.no-change.
[ 
    "H_HelmetCrew_I", //>heavy-crew-helmet
    "G_Balaclava_lowprofile", //>heavy-crew-goggles
    "V_Chestrig_khk", //>heavy-crew-vest

    _unit, _grpType, _grpSpec, _tag] call THY_fnc_CSWR_U

//>BLU-INF.-SNIPER-GROUP-UNITS:
//>Loadout-replacement//empty("")>results.no-change.
[ 
    "U_B_FullGhillie_sard", //>sniper-uniform
    "", //>sniper-helmet
    "", //>sniper-goggles
    "", //>sniper-vest-(cannot)
    "", //>sniper-backpack

    "srifle_EBR_F", //>sniper-rifle-(cannot)
    "20Rnd_762x51_Mag", //>sniper-rifle-magazine
    "optic_SOS", //>sniper-rifle-sight-(cannot)
    "", //>sniper-rifle-rail
    "muzzle_snds_B", //>sniper-rifle-muzzle
    "", //>sniper-rifle-bipod
    "Rangefinder", //>sniper-binoculars-(cannot)

    _unit, _grpType, _grpSpec, _tag] call THY_fnc_CSWR_U

```

Keep in mind you are able to easily customize almost* everything, including uniforms, vests, backpacks, and their contents. (*) Weaponry is strongly advised to use only the weapons that the original unit used. [That already was mentioned.](#)

Nowadays, only Sniper groups have total support from CSWR to change the main rifle easily.

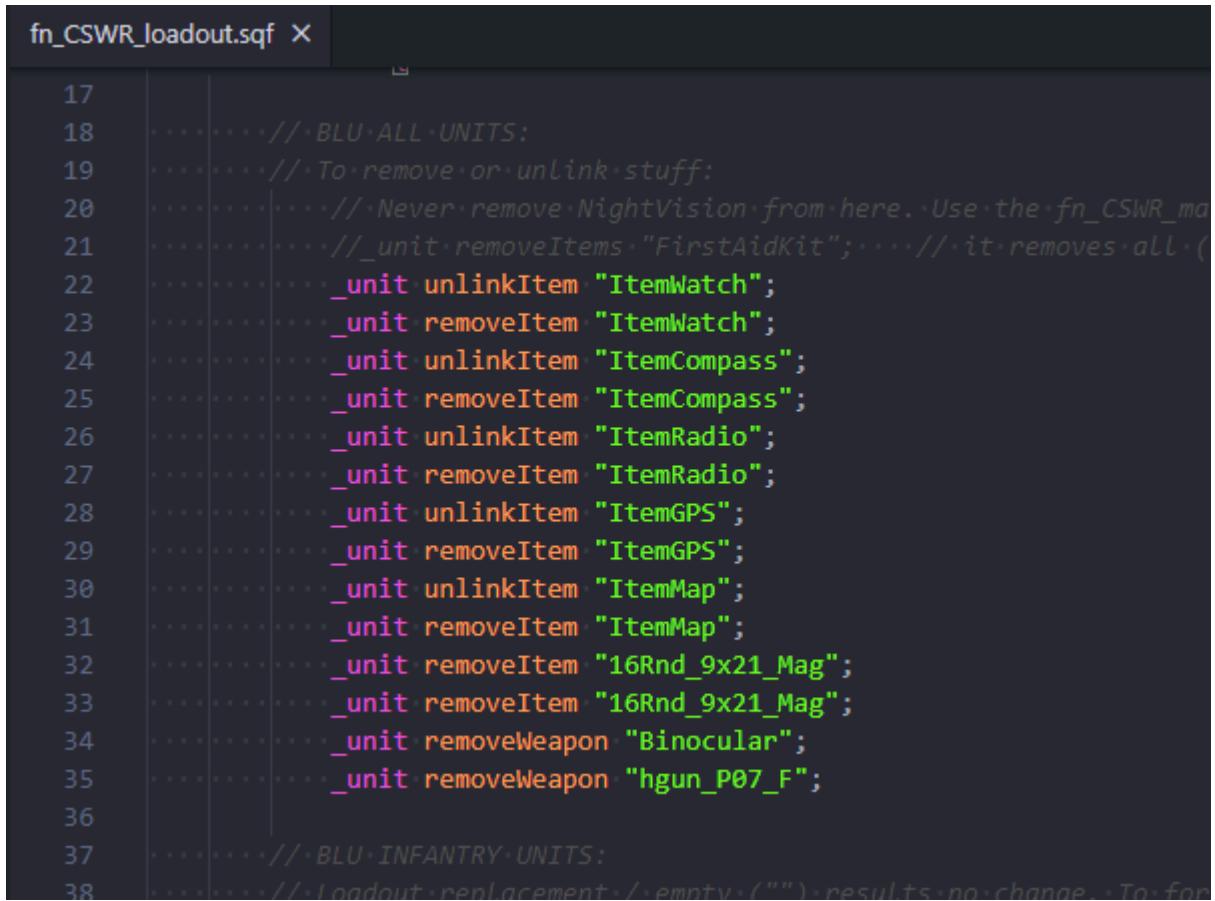
For example, you can remove all GPS devices (or any other items easily) from each CSWR spawned Blufor unit only, if you want. Or just exchange the standard vest from all Opfor units, without losing their items inside the original vest. With CSWR you're allowed to customize all unit loadouts spawned dynamically through the CSWR script.

Critical: keep in mind if you change the primary weapon, for example, you'll remove specialized weapons from specific soldier classes such as machine-gunners, marksmen, and crewmen, adding the same primary gun for everyone on the side if you force their addition through CSWR.

Use loadout editing wisely, remembering that anything you add or remove as items will impact ALL units on that side. Then open the `fn_CSWR_loadout.sqf` file and let's customize your army:

Step 1/6: removing and unlinking items

Here you will remove (and sometimes unlink and, after that, remove) items unnecessary for all units by a side:



The screenshot shows a code editor window with the file name "fn_CSWR_loadout.sqf" at the top. The code is written in Arma 3's mission configuration language (MCL). It contains two main sections: one for "BLU_ALL_UNITS" and one for "BLU_INFANTRY_UNITS". Both sections contain a series of commands to remove or unlink specific items from unit inventories. The items listed include "ItemWatch", "ItemCompass", "ItemRadio", "ItemGPS", "ItemMap", "16Rnd_9x21_Mag", "Binocular", and "hgun_P07_F". The code uses the syntax `_unit removeItem "ItemName";` and `_unit unlinkItem "ItemName";` to perform these operations.

```
fn_CSWR_loadout.sqf X
17      .... // BLU_ALL_UNITS:
18      .... // To remove or unlink stuff:
19      .... // Never remove NightVision from here. Use the fn_CSWR_ma
20      .... // _unit removeItems "FirstAidKit"; // it removes all (
21      .... _unit unlinkItem "ItemWatch";
22      .... _unit removeItem "ItemWatch";
23      .... _unit unlinkItem "ItemCompass";
24      .... _unit removeItem "ItemCompass";
25      .... _unit unlinkItem "ItemRadio";
26      .... _unit removeItem "ItemRadio";
27      .... _unit unlinkItem "ItemGPS";
28      .... _unit removeItem "ItemGPS";
29      .... _unit unlinkItem "ItemMap";
30      .... _unit removeItem "ItemMap";
31      .... _unit removeItem "16Rnd_9x21_Mag";
32      .... _unit removeItem "16Rnd_9x21_Mag";
33      .... _unit removeWeapon "Binocular";
34      .... _unit removeWeapon "hgun_P07_F";
35
36      .... // BLU_INFANTRY_UNITS:
37      .... // Loadout replacement. // empty ("") results no change. To force
38
```

For *Arma 3*, “Binocular” is a weapon and not an item, so be careful with the commands to remove things. Always test your customs, mainly when you’re removing and adding things in unit inventories.

Step 2/6: customizing the infantry

Infantry groups are all those groups by foot with no specialties created by CSWR. The infantry loadout is the base for the other group classes that you’ll see soon. Each side has its infantry loadout customization. Looking just at the green words on the image down below.

When you leave an empty field (””) it means that you don’t want to change the [original gear of the unit you’ve set previously through this file](#). When you want to force the gear removal, you can use “**REMOVED**” in uppercase. And of course, if you desire to change, for example, the uniform of the infantry, type into the uniform quotes the uniform classname desired.

```
fn_CSWR_loadout.sqf X

37     ... // BLU INFANTRY UNITS:
38     ... // Loadout replacement. / empty ("") results no change. To force re
39     ...
40         ... // infantry uniform
41         "H_Booniehat_tan", // infantry helmet
42         "", // infantry goggles
43         "V_Chestrig_khk", // infantry vest (only units wi
44         "B_Carryall_cbr", // infantry backpack (only unit
45
46     _unit, _grpType, _grpSpec, _tag] call THY_fnc_CSWR_Loadout_inf
47
```

Step 3/6: understanding the uniform, the vest, and the backpack

In *Arma 3*, the uniform, vest, and backpack are items' containers for each soldier. If you set an item for a unit, if you check their inventory in-game, there will be the item, maybe inside the uniform, backpack, or vest. If the unit has no uniform, vest, or backpack, the unit won't be able to store the item. When you set a new vest for a side, CSWR will scan all items inside the current vest, save them, change the vest to the new one, and, finally, transfer all original items to the new vest. The same happens when the uniform and backpack are replaced, for example, preserving the original unit weapons magazines.

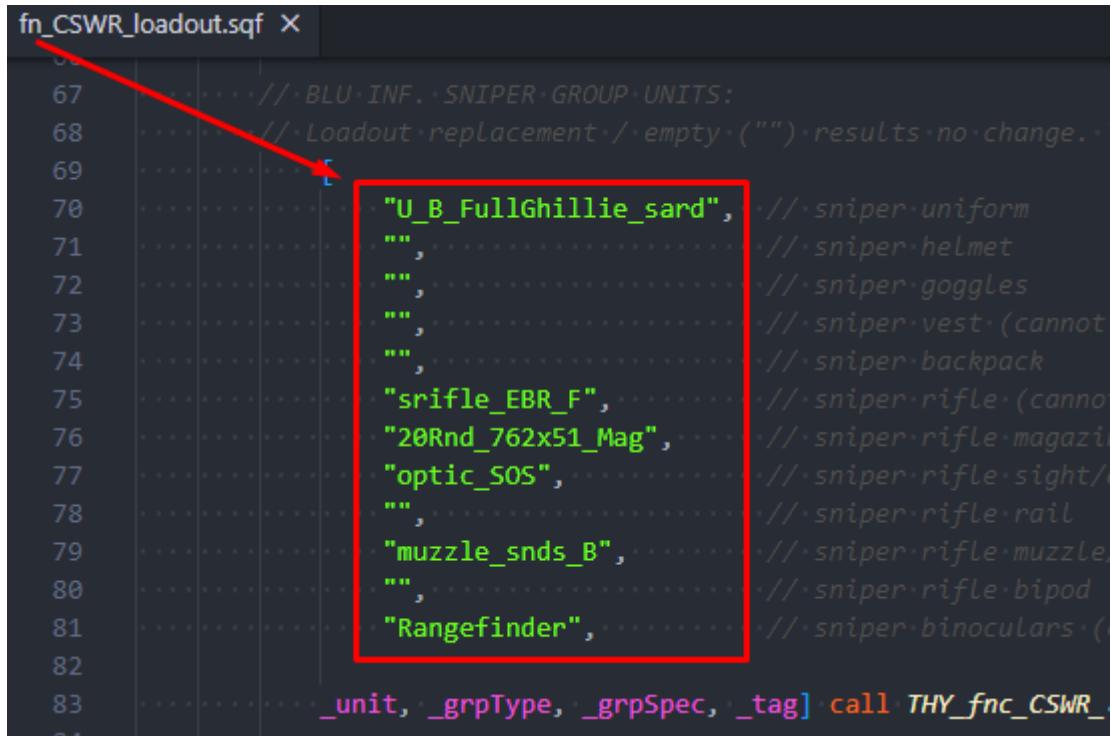
Important: Notice that not every soldier has a backpack or vest. CSWR will respect that logic but if you want to force every unit to get a backpack and/or a vest, you can do it through the file *fn_CSWR_management.sqf*, setting as true the options:

```
fn_CSWR_management.sqf X
17 // Factions:
18 CSWR_isOnBLU = true; // true == if you wanna spawn BluFor/West
19 CSWR_isOnOPF = true; // true == if you wanna spawn OpFor/East
20 CSWR_isOnIND = false; // true == if you wanna spawn Independents
21 CSWR_isOnCIV = false; // true == if you wanna spawn Civilians
22 // Loadout global:
23 CSWR_isBackpackForAllByFoot = false; // true == all units by foot
24 CSWR_isVestForAll = false; // true == all units (incl
25 // Loadout by faction
```

The option `CSWR_isBackpackForAllByFoot` when `true` sets the pre-defined new backpacks for all soldiers of all army sides, except paratroopers and heavy crewmen.

Step 4/6: customizing snipers' loadout, including the rifle

Sniper groups can be customized separately, exactly like Heavy Crewmen, and Paratroopers. As in the image below, you can change the sniper group's uniform, helmet, goggles, vest, backpack, rifle, magazine type, optics, rail, suppressor, bipod, and binoculars.



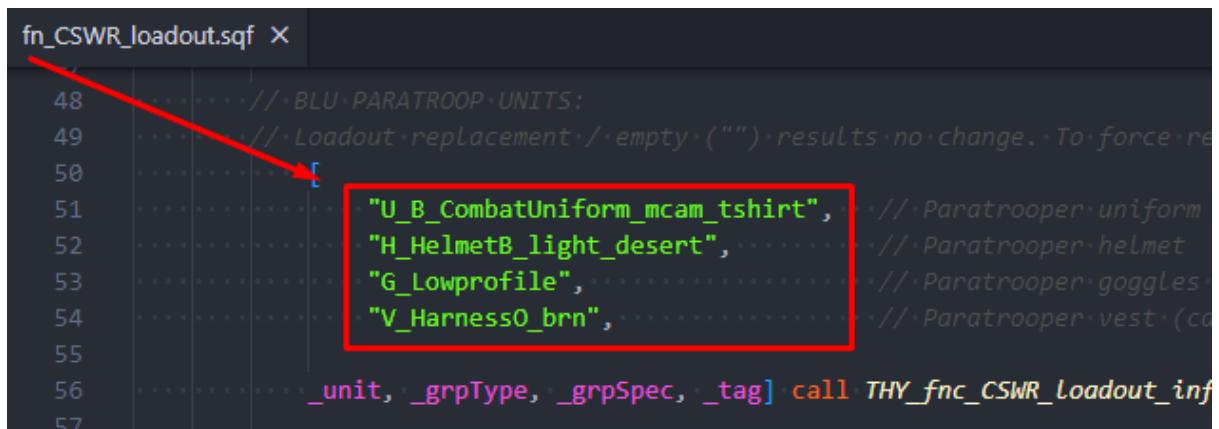
```
fn_CSWR_loadout.sqf X
5
67      // BLU-INF-SNIPER-GROUP-UNITS:
68      // Loadout replacement-/empty("")-results-no-change-
69
70      "U_B_FullGhillie_sard",           // sniper-uniform
71      "",                                // sniper-helmet
72      "",                                // sniper-goggles
73      "",                                // sniper-vest-(cannot)
74      "",                                // sniper-backpack
75      "srifle_EBR_F",                   // sniper-rifle-(cannot)
76      "20Rnd_762x51_Mag",               // sniper-rifle-magazi
77      "optic_SOS",                      // sniper-rifle-sight/
78      "",                                // sniper-rifle-rail
79      "muzzle_snds_B",                  // sniper-rifle-muzzle
80      "",                                // sniper-rifle-bipod
81      "Rangefinder",                    // sniper-binoculars-(

82
83      _unit, _grpType, _grpSpec, _tag] call THY_fnc_CSWR_
84
```

If you don't want to change the original stuff of the sniper group, just let those green classnames as "" (empty).

Step 5/6: customizing paratroopers' loadout

What defines if a group is a paratrooper group or a regular one is [which spawn-point type they will use](#). Said that, like other loadout customization, you must use the `fn_CSWR_loadout.sqf` file to edit the snippet below. Paratroopers can get their own uniform and vest, including, of course, the parachute bag:



```
fn_CSWR_loadout.sqf X
4
48      // BLU-PARATROOP-UNITS:
49      // Loadout-replacement-/empty("")-results-no-change-To-force-re
50
51      "U_B_CombatUniform_mcam_tshirt",           // Paratrooper-uniform
52      "H_HelmetB_light_desert",                  // Paratrooper-helmet
53      "G_Lowprofile",                          // Paratrooper-goggles
54      "V_HarnessO_brown",                     // Paratrooper-vest-(co
55
56      _unit, _grpType, _grpSpec, _tag] call THY_fnc_CSWR_Loadout_inf
57
```

Important: if the Sniper group is a paratrooper group too (it's possible), they will inherit some paratroop loadout like the uniform and helmet. Actually, all group classes that are using the paratroop specialty will inherit at least the paratroopers uniform if there's one declared by you. Otherwise, will inherit the uniform of their infantry.

Step 6/6: adding and linking new items

To finish, the last section of loadout customization is the area to add things. No doubt, this is the section that I use the least in my customizations. Generally, I just remove stuff from units, but when customizing the civilian side for example, I love to add the cellphone item to each unit just for fun. Be aware for some items like GPS that is important to “addItem” first and, right after “linkItem” too. Otherwise, if you just add the GPS, the item will be in the unit’s inventory but not in use.

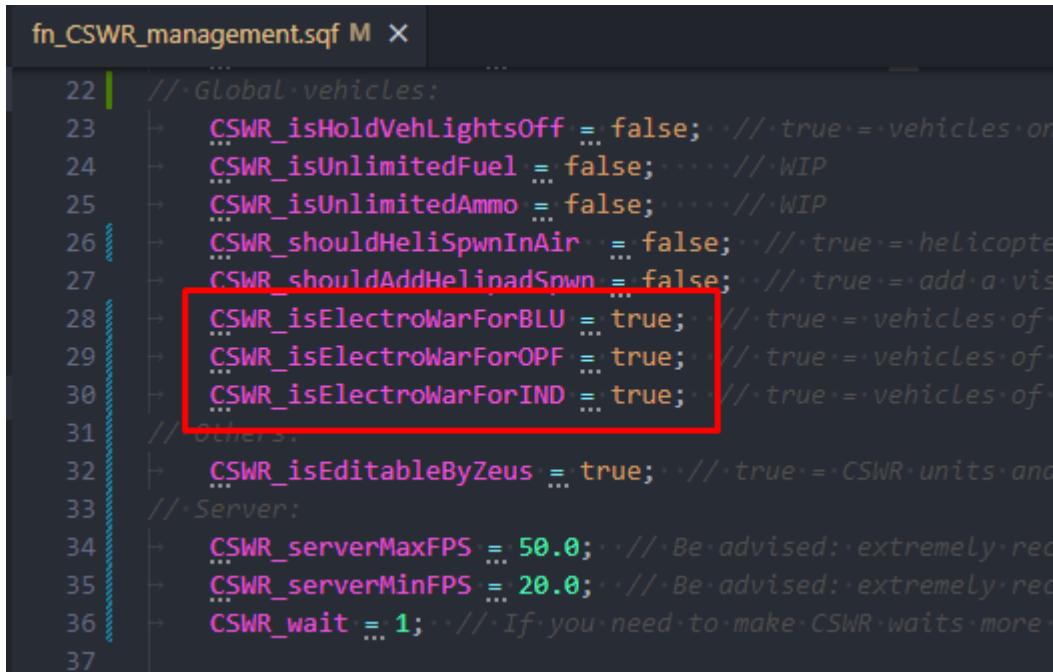
```
.... //• BLU•ALL•UNITS:  
.... //• To•add•or•Link•stuff:  
.... //• Never•add•NightVision•from•here. •Use•the•fn_CSWR_management.sqf•to•set•this.  
.... _unit addItem "FirstAidKit"; //•create•just•one•bandage•in•unit•inventory.  
.... _unit addItem "FirstAidKit"; //•create•another•bandage•in•unit•inventory.  
....};
```

Critical: about adding lots of things, always take care not to exceed the load limit of the chosen uniform, vest, and backpack. If no more space in their inventory, items start to disappear.

Important: if [something is wrong in your loadout changing tests or you want to learn how to see the original unit loadout code, check this out.](#)

Electronic Warfare Resources

You can turn on or off the electronic warfare resources for vehicles by each side. For example, if you want to make BLU a Brazilian Army against a poor OPF Argentine militia, you can turn *true* the electronic war systems for BLU, and turn it *false* for OPF in *fn_CSWR_management.sqf* file:



```
fn_CSWR_management.sqf M X
22 // Global vehicles:
23     CSWR_isHoldVehLightsOff = false; // vehicles on
24     CSWR_isUnlimitedFuel = false; // WIP
25     CSWR_isUnlimitedAmmo = false; // WIP
26     CSWR_shouldHeliSpwnInAir = false; // true = helicopter
27     CSWR_shouldAddHelipadSpwn = false; // true = add a vis
28     CSWR_isElectroWarForBLU = true; // true = vehicles of
29     CSWR_isElectroWarForOPF = true; // true = vehicles of
30     CSWR_isElectroWarForIND = true; // true = vehicles of
31 // Others:
32     CSWR_isEditableByZeus = true; // true = CSWR units and
33 // Server:
34     CSWR_serverMaxFPS = 50.0; // Be advised: extremely rec
35     CSWR_serverMinFPS = 20.0; // Be advised: extremely rec
36     CSWR_wait = 1; // If you need to make CSWR waits more
37
```

What happens when *true*?

- Side vehicles are capable of reporting their own positions;
- Side vehicles are capable of receiving targets;
- Side vehicles are capable of reporting targets;

Important: to improve the game experience, all helicopters always will receive targets from their allies, even the Mission Editor turned off electronic warfare for a side.

Fixing: vehicles blow up when they spawn

Different from people/soldiers, vehicles need a big empty space on the map to spawn on the ground safely, with no collisions. So, if it happens yet, make sure the side spawn-points are using them on clear and flat grounds, for example, over a road or in an empty flat field. Unfortunately, urban streets can be a challenge to spawn vehicles, resulting in explosions. Since CSWR v5.5, there's a booking system, especially for spawn-points where if a vehicle is spawning and too close to the spawn-point center, other vehicles scheduled to spawn in the same point will wait for the zone to be clear or deleting possible wrecks or other assets that suddenly show up in that spot.

Important 1: if more well-located spawn-points are available, fewer accidents such as infantry being run over and vehicles colliding will happen.

Important 2: **NEVER** set a spawn-point out of the map limits. *Arma 3* doesn't recognize positions beyond the map edges. CSWR will prevent you from doing that, deleting the marker automatically.

Fixing: trying to edit the unit's loadout

If you are facing a hard time editing your side's loadouts, maybe you are trying to remove a thing that doesn't exist or calling weird commands. Let's investigate:

- 1) On *Eden Editor*, drag and drop on map one unit of the original side you are trying to make changes;
- 2) After that, go to unit Attributes and click "Edit Loadout" > "BI Virtual Arsenal";
- 3) Now, just click on "Export" button;



- 4) Open your Notepad and CTRL+V;
- 5) Now, check carefully if you are editing the same items the original unit brings.

Fixing: making CSWR lighter for servers

First of all: the number of AI you ask to CSWR spawn in the mission is the CSWR feature with the highest impact on server performance.

Which number of AI is good enough

Remember that in `fn_CSWR_population.sqf` pretty often each new line is a group of units. If one side you have 20 lines, maybe you are asking CSWR to spawn 60 units or even much more, all depends on [how many units each one of the group-types](#) you've built. In 2023 my game computer was able to manage around 300 CSWR AI's without huge performance drops. Arma 3 servers with less than 20 FPS will impact the performance of all its players. Make some tests and, if you really know what you're doing, you can edit some numbers in `fn_CSWR_management.sqf` to help CSWR recalc stuff:

```
fn_CSWR_management.sqf M X
33 // Server:
34 CSWR_serverMaxFPS = 50.0; // Be advised: extremely recommended do
35 CSWR_serverMinFPS = 20.0; // Be advised: extremely recommended do
36 CSWR_wait = 1; // If you need to make CSWR waits more for other s
37
```

Sides CSWR will spawn

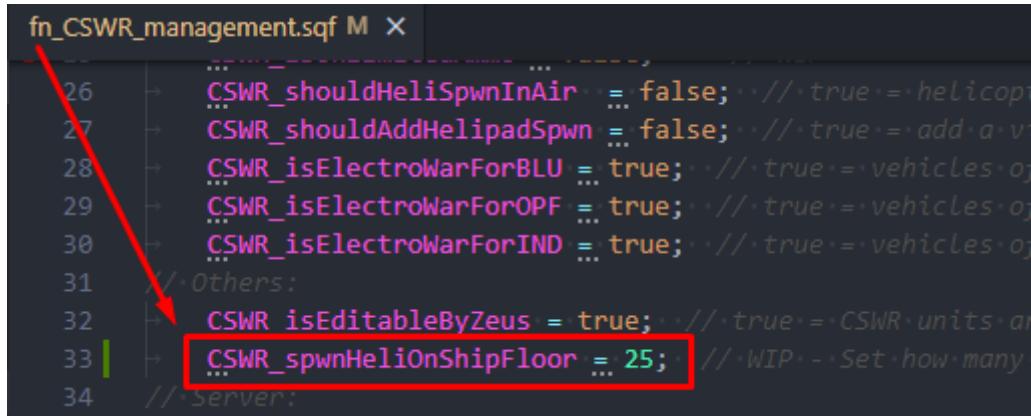
It doesn't matter for performance if the script will spawn one or all sides. The most important will be how many AI units you are dropping in the mission.

Using the Dynamic Simulation

All AI created by CSWR doesn't use Dynamic Simulation. Because of the CSWR creation concept, the script doesn't want to work with [DynamicSimulation](#) enabled. Of course, you can add manually how many units and objects using that feature through the *Eden Editor* or another script, CSWR won't care about it.

Fixing: helicopters over ship floors

I'm working to make this easier but for now, if you want to set spawns for helicopters over an asset floating in water (like a ship) you will need to set the height between the water surface and the asset's solid floor:



```
fn_CSWR_management.sqf M X
26     CSWR_shouldHeliSpwnInAir = false; // true == helicop
27     CSWR_shouldAddHelipadSpwn = false; // true == add a v
28     CSWR_isElectroWarForBLU = true; // true == vehicles.o
29     CSWR_isElectroWarForOPF = true; // true == vehicles.o
30     CSWR_isElectroWarForIND = true; // true == vehicles.o
31     /> Others:
32     CSWR_isEditableByZeus = true; // true == CSWR_units.a
33     CSWR_spwnHeliOnShipFloor = 25; // WIP -- Set how many
34 // Server:
```



The solid floor of the *USS Freedom* is 25 meters above sea level.



The solid floor of the *USS Liberty* is 9 meters above sea level.

Critical: extremely recommended you have more spawns dedicated to helicopters by the number of helicopters you want to spawn. This temporary solution is fragile and can easily provoke explosions if more than one helicopter chooses the same spawn-point at the same time.

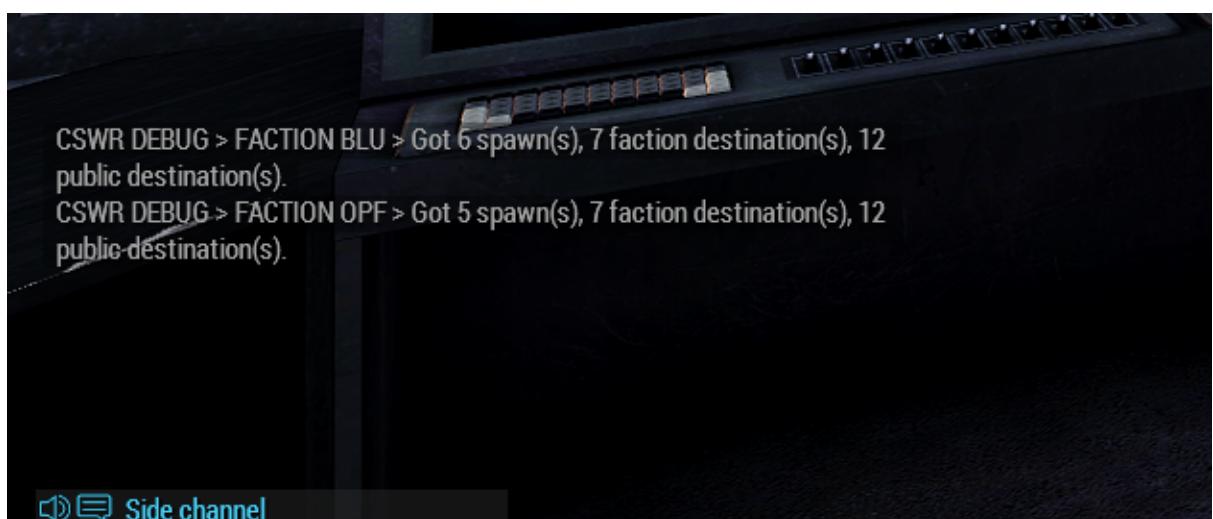
Fixing: investigate further what is happening (Debugging)

If you're facing something weird or unexpected or even just to check the CSWR AI behaviors with the settings you have set, it's recommended to turn `Debug Mode true`. This resource will describe a lot of actions the script is doing and, if you turn *true* on the Debug extensions (for Hold, Watch, Occupy, and Helicopters) you have powerful information on the screen. To see deeper information about Occupy movements, for example, it's needed to turn to *true* `CSWR_isOnDebugOccupy` as well as maintain `CSWR_isOnDebugGlobal` activated.



```
fn_CSWR_management.sqf X
1  //•CSWR•v5.5
2  //•File:•your_mission\CSWRandomizr\fn_CSWR_management.sqf
3  //•Documentation:•your_mission\CSWRandomizr\_CSWR_Script_Documentation.
4  //•by•thy•(@aldolammel)
5
6  if (!isServer) exitWith {};
7
8  //•PARAMETERS•OF•EDITOR'S•OPTIONS:
9
10 //Debug:
11   CSWR_isOnDebugGlobal = true;           //•true•=•shows•basic•debug•inform
12   CSWR_isOnDebugOccupy = false;          //•true•=•shows•deeper•Occupy•mark
13   CSWR_isOnDebugWatch = false;           //•true•=•shows•deeper•Watch•marke
14   CSWR_isOnDebugHold = false;            //•true•=•shows•deeper•Hold•marker
15   CSWR_isOnDebugHeli = false;             //•true•=•shows•deeper•AI•Helicopt
16   CSWR_isOnDebugPara = false;            //•true•=•shows•deeper•Paradrop•de
17   CSWR_isOnDebugBooking = false;          //•true•=•shows•deeper•markers•bo
```

The header of `fn_CSWR_management.sqf` file.



An example of CSWR debug messages right after the mission starts with debugging as `true`.

```
CSWR DEBUG > SPAWN DELAY > 1 OPF vehicle and its crew will spawn LATER.  
CSWR DEBUG > WATCH > 2 location(s) found, OPF 'O Alpha 1-5' moving: Location  
Hill at 3751, 4888.  
CSWR DEBUG > OCCUPY > OPF 'O Alpha 3-1' going to 1 of 87 building(s) found.
```

```
CSWR DEBUG > SPAWN DELAY > A OPF vehicle was granted TO SPAWN by  
TIMER (it was 120 secs).  
CSWR DEBUG > BLU 'B Alpha 2-1' helicopter returning to base!
```

```
CSWR DEBUG > OPF 'O Bravo 1-1' helicopter is TAKING OFF!  
CSWR DEBUG > HOLD > BLU 'B Alpha 1-6' tracked-vehicle hold [Desired:  
152.665° | Executed: 152.963°].
```

Important: don't forget to turn the (at least) `CSWR_isOnDebugGlobal` `false` to save the server performance, ignoring lots of debug code lines that, when your mission is ready, would be unnecessary still to be running.

Fixing: units are not in a crouch when inside a ruin

It is because of one of these two factors: 1) the ruin is not listed as a building to be ignored by CSWR, and 2) it's not listed as an acceptable ruin to be used by CSWR units doing occupy movement.



Above, it's how the group in occupy stays in an unknown ruin.

Below, in an acceptable ruin.



If you want to allow the units to use a ruin as a place to be (in a crouch in these cases), add the ruin classname in the `CSWR_occupyAcceptableRuins` list. Otherwise, add the classname in `CSWR_occupyIgnoredBuildings`.

Both lists are in the `fn_CSWR_management.sqf` file.

Contribute to the CSWR script

Discussion on Bohemia Forums

<https://forums.bohemia.net/forums/topic/237504-release-controlled-spawn-and-waypoints-randomizr/>

Changelog on GitHub

<https://github.com/aldolammel/Arma-3-Controlled-Spawn-And-Waypoints-Randomizr-Script#changelog>

Author

Based in Porto Alegre, Brazil



thy [@aldolammel](#)

If you care

Just give a like for CSWR on Workshop to spread the word :)

<https://steamcommunity.com/sharedfiles/filedetails/?l=swedish&id=2740912514>