

### Actionbuilder



#### Introduction

Actionbuilder is a powerful unit spawning tool to aid Arma 3 mission making.

The goal is to enable more varying community generated missions by increasing depth and randomization of the missions.

By using Actionbuilder you can create bigger and better missions without a need to spend countless hours in the mission editor, tinkering with the technicalities.

#### Requirements

Actionbuilder missions require Actionbuilder from both the server and the clients.

#### Backwards compatibility

Actionbuilder is highly backwards compatible and aims to stay that way. New features may emerge, old ones may improve, but the missions still keep working for as long as the major version is the same: 1.0.0 -> *major.minor.patch*.

### Bundled Modules

#### Actionpoint

Handles a singular Actionbuilder system. May contain conditions that must be fulfilled before the portals are triggered.



#### Portal

Spawns registered AI units. One Actionpoint can hold multiple portals. One portal can hold multiple waypoints.



#### Waypoint

Orders for the spawned units to follow. One Actionbuilder system may contain tens of waypoints, creating long chains of commands.



### Bundled Functions

#### Utility functions

Since the version 0.3.0 Actionbuilder has been bundled with countless of utility functions. These functions are scripts that are designed for different goals. The very same functions are used by the Actionbuilder.

You can combine these functions to your own scripts or launch them via in-game triggers.

- Functions are not necessary for creating an Actionbuilder mission.
- Functions do not require editor placed modules to work.

#### How to use?

You can easily browse Actionbuilder functions with Arma 3's built-in function browser. To access the browser, press the "fx" button in the mission editor.



Look for functions under *Utility* category.

Example call:

`[man, "KILL"] spawn Actionbuilder_fnc_punish;`

## Usage of Actionpoints and Portals

### Description

An actionpoint is synchronized with portals and the portals are synchronized with AI units, groups or objects.

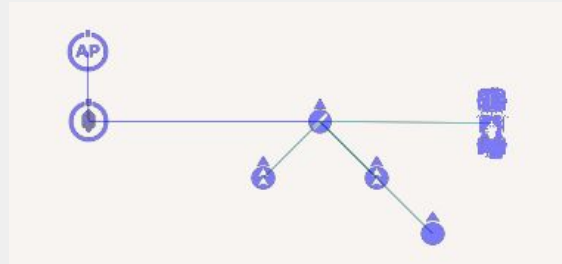
When the game starts, the original synchronized units are removed as they are registered to the portals. The portals spawn virtual copies of the original units.

### General notes

- ❑ All synchronizations must be done with the *synchronize* (F5) tool.
- ❑ Synchronizing a one unit of a group is enough for the entire group to be registered.
- ❑ Special attributes like *name* and *initialization* given to an unit are **not** registered!

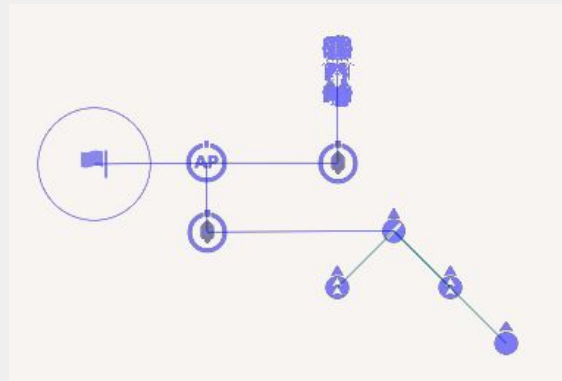
### Common issues

- ❑ If a portal doesn't seem to function, make sure the portal and actionpoint settings are correctly set. There may be some condition blocking the spawn action.
- ❑ If units are spawning to wrong positions, make sure there are no group connections directly to Actionbuilder modules and that portal settings are correctly set.



#### Example 1.1 - ONE PORTAL, ONE GROUP

The portal spawns a one group including four men and a manned vehicle. The group is spawned immediately as there are no triggers synchronized to the Actionpoint.



#### Example 1.2 - TWO PORTALS, TWO GROUPS

The trigger activates an Actionpoint that holds two distinct portals. The portals will spawn two groups: one including only infantry and another containing a manned vehicle.

Do note, a singular portal can hold multiple distinct groups.

## Usage of Waypoints

### Description

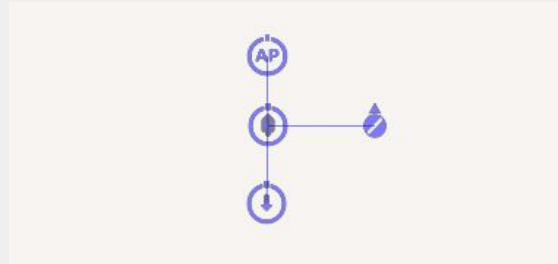
Waypoints are synchronized with portals and other waypoints. Portal spawned units will immediately start to execute available waypoints one by one.

### General notes

- ❑ In a case of multiple waypoint choices, the group will attempt to select the most suitable waypoint for them. Sharp u-turns are usually avoided.
- ❑ You can set some waypoints to be high priority. These waypoints are always selected over the others.
- ❑ Some of the waypoint types are instant. This means that the waypoints activate instantly without a need to reach the position of the waypoint.
- ❑ There can be max 144 Actionbuilder modules (Arma 3 limit).

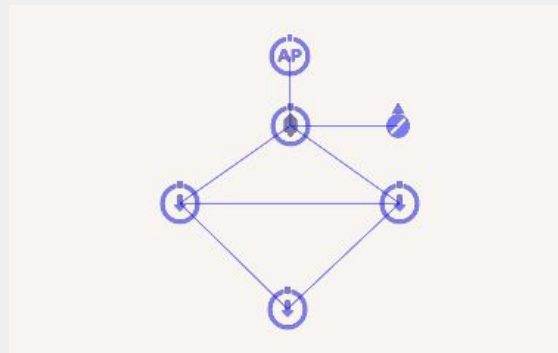
### Tips

- ❑ Same waypoints can be utilized by different groups. Use this feature to create highly randomized waypoint networks.
- ❑ Remember to set *placement of radius* to further increase unpredictability.
- ❑ Experiment with the waypoints. There are many “hidden” yet powerful features that didn’t fit into this guide.



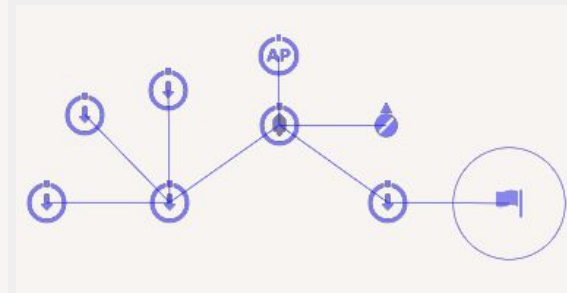
### Example 2.1 - MOVE

The spawned unit will *MOVE* to the position of the only available waypoint module.



### Example 2.2 - CYCLE (left)

The unit will select one of the two available waypoints. The selection is fully random as the waypoints are equal. After the first waypoint, the unit will have a second similar choice to be made. Notice how the waypoints produce a never-ending loop. The unit will never stop moving.



### Example 2.3 - CONDITIONAL WAYPOINT (right)

If the trigger condition is equal to *FALSE*, the unit will select the waypoint on the left and continue from there. If the trigger is active (*TRUE*), the unit will go either left or right. The waypoint on the right is called “conditional waypoint”.

## Solving problems

### Module errors

If modules are improperly synchronized, the Actionbuilder will inform the user about it on the bottom of the screen. Errors like these should **always** be fixed before publishing the mission.

Each error contains an unique **error code** that can help you to find the solution.

### Unexpected behaviour

At times improper module settings may cause unexpected behaviour. To solve issues like these, Actionbuilder has various debugging features. To enable some of these features, set variable `RHNET_AB_L_DEBUG` to `TRUE`. After this step, Actionbuilder will print out the reasoning behind some actions.

Example setup for debugging:

- 1) Create a new trigger.
- 2) Set condition to `TRUE`.
- 3) Set onAct to `RHNET_AB_L_DEBUG = true`;

**Please, remove debugging before publishing the mission!**

## Performance optimizations

### Areas of operation

It is always a good idea to have AI only there where they are relevant: close to players. Use triggers to create sectors that have AI units only after the player enters the area. Just make sure the players can't see the spawning: take advantage of buildings, cliffs and other visual barriers.

Make sure to be familiar with the advanced portal settings. For example you can set player free zones to hide AI spawning.

### Increased AI count

Missions tend to get bigger when spawning of the units is as easy as it is with Actionbuilder. Make sure the mission doesn't become too heavy for the server or the clients to handle. See portal settings to restrict the maximum AI count.

### Free performance gains

Actionbuilder provides various performance advantages against the traditional missions.

- Optimized AI gear and behaviour.
- Performance monitoring that prevents servers from dying.
- Optimized functions and modules.
- A full headless client support.

## Headless client

### What?

One of the key ways to increase server performance is to distribute the load. With headless clients the load can be handled to external workers, like other servers or even just cpu threads that Arma 3 can't fully utilize on its own.

If a mission has *playable* headless clients and the server has headless clients connected, Actionbuilder will automatically distribute the load.

The load is distributed between Actionpoints. It is a good idea to use as compact Actionbuilder systems as possible.

Headless clients are set in the editor.

[Game Logic](#) → [Virtual Entities](#) → [Headless client](#)



### Important to understand

Headless client feature may enable huge AI battles for your server but remember: also players must be able to play the mission. Make sure their game experience stays enjoyable!