GX Addons
v2.4

(by Geloxo)
www.ust101.com

Introduction

This addon features scripts mainly intended to create air formations and allow real operations with AI. This release is an enhancement of the old Handy Tools already created by me in the past but improved to easy the configuration and usage. The scripts also include a set of common functions usable with general purpose on your missions. The addon is fully MP compatible and does not create conflicts with other addons. The addon is also compatible with land and sea units too.

This addon is configured for **Arma 3** by default but can be also used with other addons (override of default variables is needed then in the mission init.sqf, addon gx_fn_init.sqf or via any external script).

GX Zeus Advanced Script is also included in the addon pack. Please refer to the separate documentation to find information about this script.

Credits

- Functions from CBA and BIS
- Trap shout sounds included taken from DSIA mod (Credits to Wolfrug)
- Arty sounds from BON_ARTY renamed and repacked (not modified)
- Sound files from BIS Arma series renamed and repacked (not modified)
- Fastrope units animation by da12thMonkey (already included in old Handy Tools mod)

BIS Content licenses: http://www.bistudio.com/licenses

Requirements

This addon requires CBA and BIS functions to work.
UPSMON script also required only if mission uses the generation of units in the ambient modules.

Main features in current version

- Improved and smoother flight model calculations
- Anticollision feature during turnings
- Terrain avoidance system
- Dynamic formation changes based on predefined templates
- Custom formations based on mission editor objects placing
- Missed waypoint avoidance
- Navigation based on bearing via intermediate sub-waypoints
- Air Support radio and interface to request arty, CAS or airstrikes
- Misc handy functions for mission edition
- MP compatible
- Fully customizable internal variables for mission makers
- Custom arty and air support sound effects
- Resources and transports configurable for each player side
- Basic radio chatter added to support modules
- Predefined configuration for Unsung and Iron Front mods
- GX Zeus advanced script included in the addon pack
- Intel & Interaction module added
- Population and traffic modules added

Important note: due to changes in the **Support Cfg** module since v2.21, existing missions created using the old version of the module need to be updated for the support dialog to work properly. In order to do so simply reopen the mission in editor, delete the module, place it again, adjust its configuration parameters (if needed) and save mission again.

Changelog

v2.4:

- GX Zeus advanced script included in the mod pack
- Existing functions now available as editor modules
- Intel & Interaction module added
- Ambient and traffic modules added
- Helicopter attack added to support dialog

v2.31:

- Native support to Unsung and Iron Front mods
- Basic radio chatter and ambience sounds added

v2.21:

- Support cfg module variables changed
- Support dialog improvement
- Flight model calculations improved

v2.0:

- initial release

Support Modules

The following modules are available in the editor to configure several types of support:

Support: request arty, helicopter CAS and bombing via custom dialog on mouse wheel menu. Dependant on player´s side. You don´t need to place any module in editor, but just to enable the player´s mouse wheel action menu (see below).

To enable support on a unit or object without any condition use in its init:

```
this call gx_asp_action
```

To enable support on a unit requiring it to own GX Support Radio item in its inventory (item classname is "gx_supportradio") use in its init:

```
this call gx_asp_radio
```

To add the required GX Radio to an ammo box add to its init:

```
this call gx_asp_addradio
```

To clear the box completely and add the radio use the following command:

```
this call gx_asp_addradioclear
```

NOTE: to define your own support configuration use the support cfg module or refer to the variables override section of this manual (advanced users only).

A basic radio chatter soundset is enabled by default in the module. To disable simply set the following variable to false and **publish the new variable to all clients during MP**:

```
gx_asp_radiochatter_enable = false;
publicvariable “gx_asp_radiochatter_enable”;
```

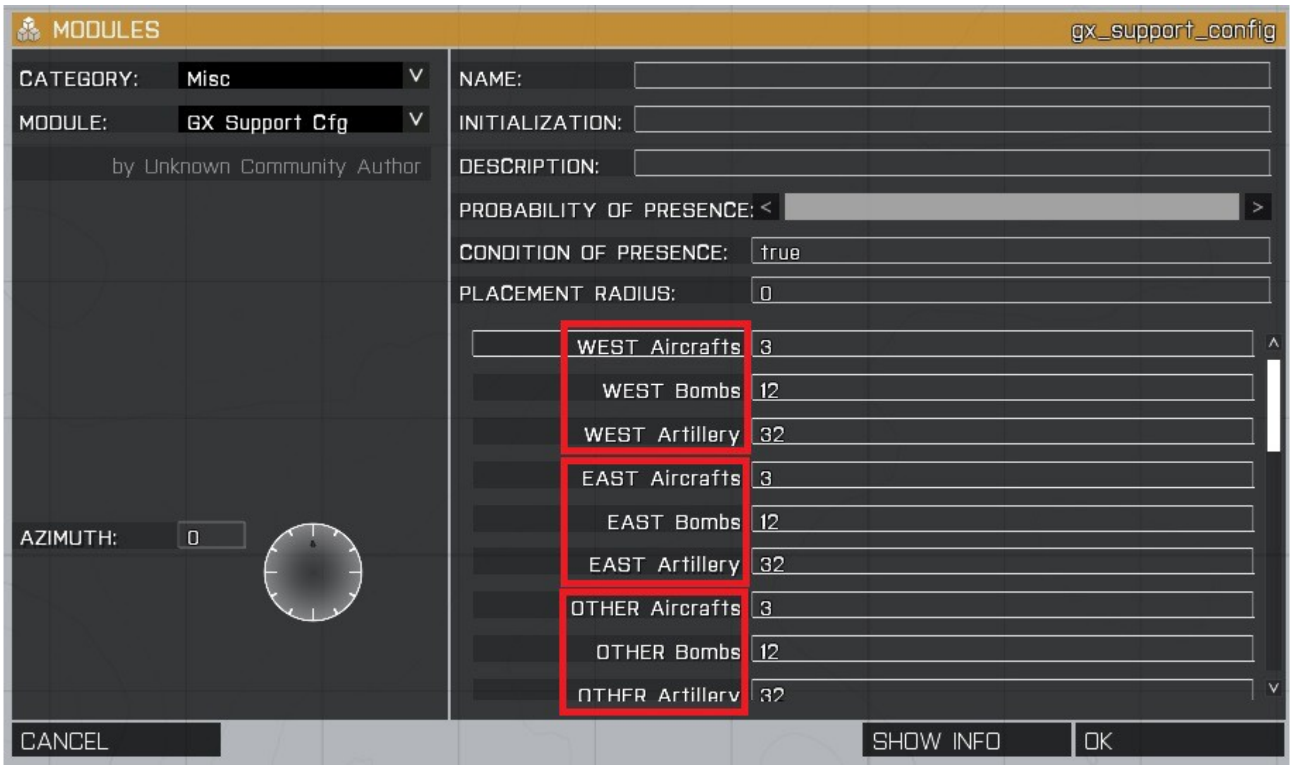
To disable the side chat messages used when asking for support use the following code (notice that true must be a string, that means, between commas):

```
gx_asp_zeus_cond = “true”;
publicvariable “gx_asp_zeus_cond”;
```

Vehicle chat messages will still be displayed for their crews and cargo. **Zeus units** have the chat disabled by default to hide their actions to other players.

Support Cfg: support configurator module

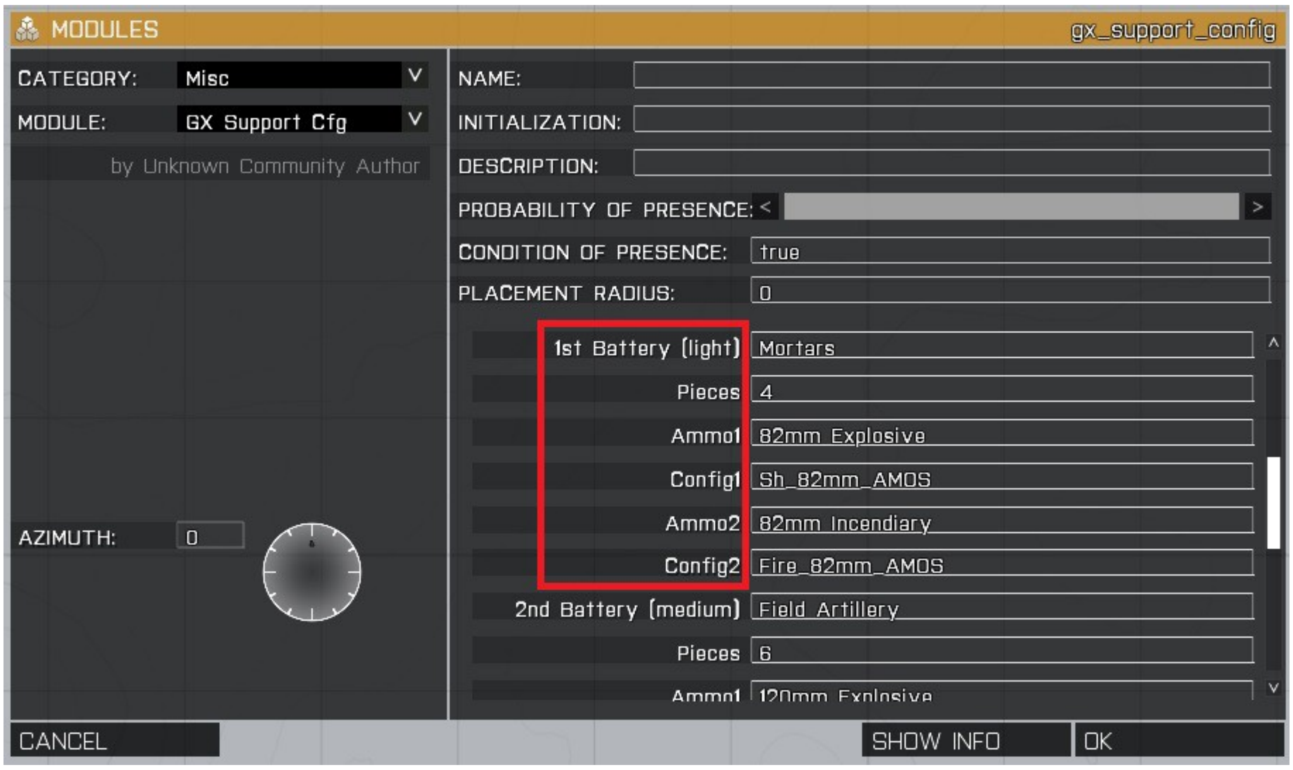
Place only one module in editor and set the desired values for supports in the module fields. This will override default values, allowing you to customize that mission. Variables for each side can be set here too.


MODULES gx_support_config

CATEGORY: Misc
MODULE: GX Support Cfg
by Unknown Community Author
AZIMUTH: 0

NAME:
INITIALIZATION:
DESCRIPTION:
PROBABILITY OF PRESENCE:
CONDITION OF PRESENCE: true
PLACEMENT RADIUS: 0

WEST Aircrafts 3
WEST Bombs 12
WEST Artillery 32
EAST Aircrafts 3
EAST Bombs 12
EAST Artillery 32
OTHER Aircrafts 3
OTHER Bombs 12
OTHER Artillery 32


MODULES gx_support_config

CATEGORY: Misc
MODULE: GX Support Cfg
by Unknown Community Author
AZIMUTH: 0

NAME:
INITIALIZATION:
DESCRIPTION:
PROBABILITY OF PRESENCE:
CONDITION OF PRESENCE: true
PLACEMENT RADIUS: 0

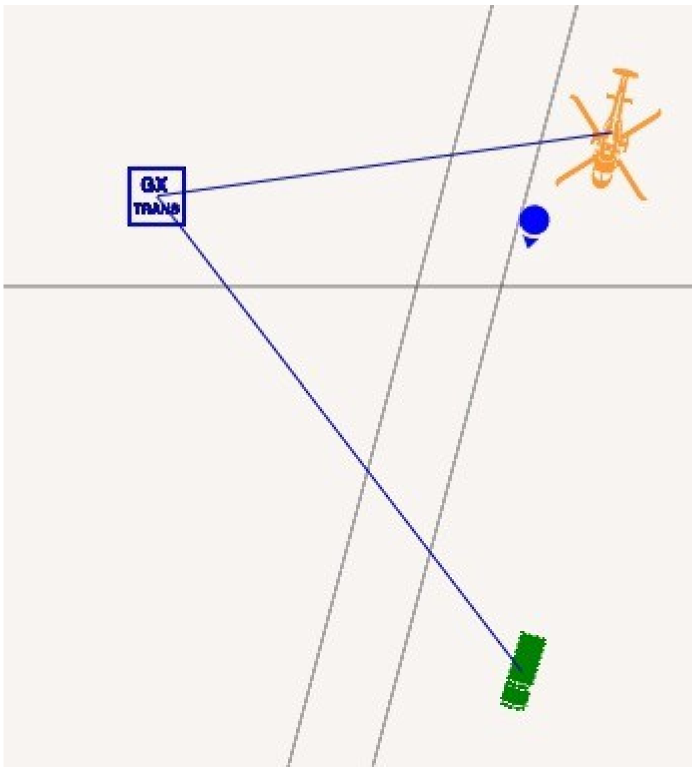
1st Battery (light) Mortars
Pieces 4
Ammo1 82mm Explosive
Config1 Sh_82mm_AMOS
Ammo2 82mm Incendiary
Config2 Fire_82mm_AMOS
2nd Battery (medium) Field Artillery
Pieces 6
Ammo1 120mm Explosive

NOTE: to disable any of the batteries so that it´s not available during mission set its “Pieces” entry to 0.

Transport: extraction request via radio. Dependant on player´s side.

All vehicles that are being used with this module must have a name in the mission editor.

Place only one module in editor and synchronize it to all units you want to use as transports (AI controlled or empty ones). When module is sync on vehicles from different sides the extraction will only be available for the vehicles on the player side only. You can use empty vehicles and assign a driver from any side by using moveindriver command.



Vehicle will be activating actions on the mouse wheel menu. Destination is set on map single click.

To enable transport request on a unit or object without any condition use in its init:

```
this call gx_trh_action
```

To enable transport request on a unit requiring it to own GX Support Radio item in its inventory (item classname is "gx_supportradio") use in its init:

```
this call gx_trh_radio
```

To immediatelly trigger the extraction during mission via a particular vehicle, name the vehicle in editor and use (for instance in a radio trigger Juliet):

```
Example: chinook1 call gx_trh_extract
```

To set a custom name for the extraction action menu add to the vehicle init the custom name you want:

```
Example:  [this, "Evasan UH-1H"] call gx_trh_setname
```

For planes to hold engines off while waiting at base add to their init:

```
this action ["engineOff", this]
```

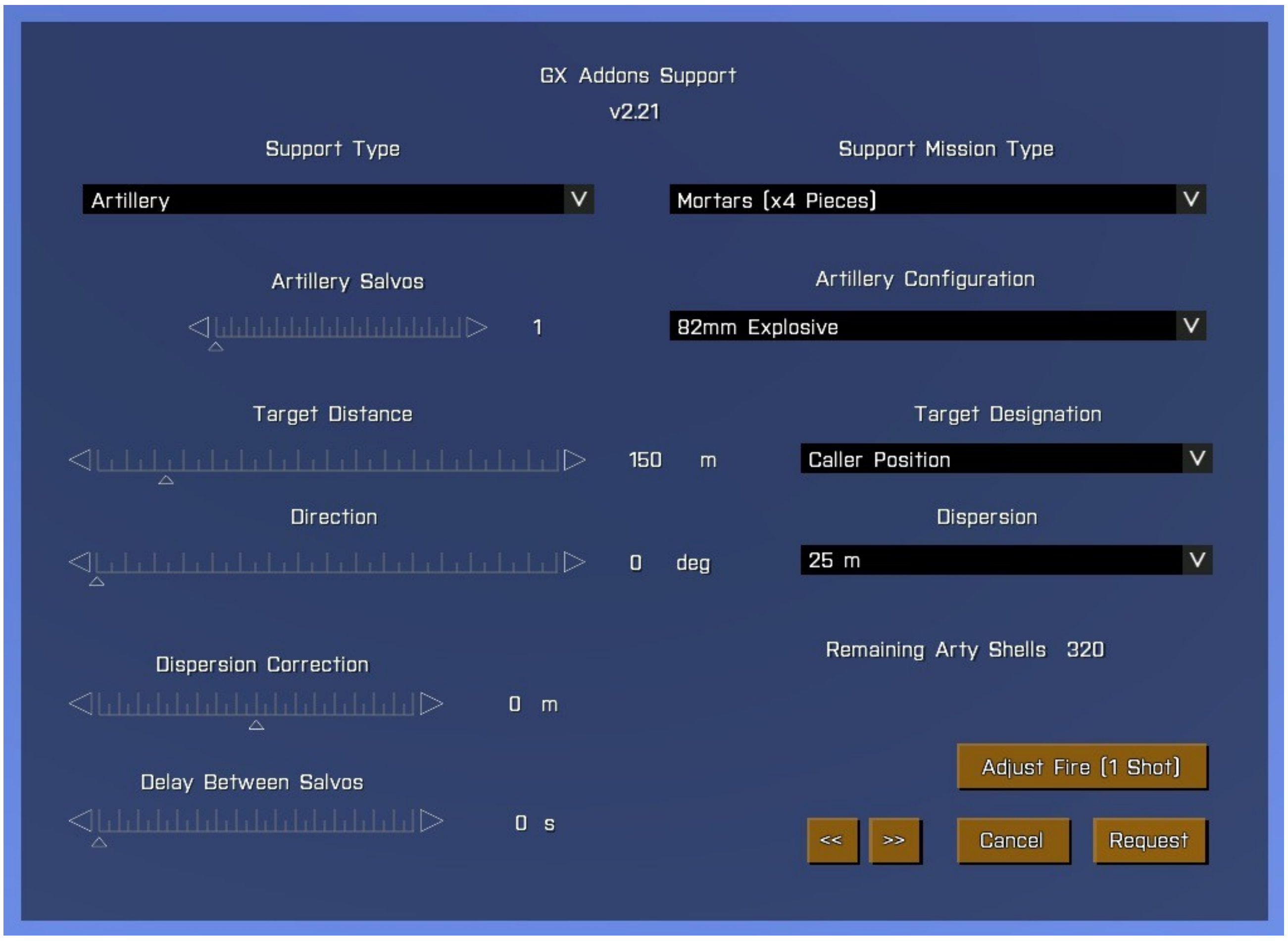
Support Modules GUI

Custom dialogs are available for the modules. Select the desired support by choosing the support type in the combo. This will display different submenus depending on the choosen support.

The << and >> orange buttons are used to swith between the support and transport dialogs, provided that player´s unit is enabled to use those support types in the mission editor.

Last used values are stored in case the dialog is closed and after the mission request. Remember to check the remaining aircrafts, bombs and shells prior to ask for the mission.

Artillery request


GX Addons Support
v2.21

Support Type: Artillery

Support Mission Type: Mortars (x4 Pieces)

Artillery Salvos: 1

Artillery Configuration: 82mm Explosive

Target Distance: 150 m

Target Designation: Caller Position

Direction: 0 deg

Dispersion: 25 m

Remaining Arty Shells 320

Dispersion Correction: 0 m

Delay Between Salvos: 0 s

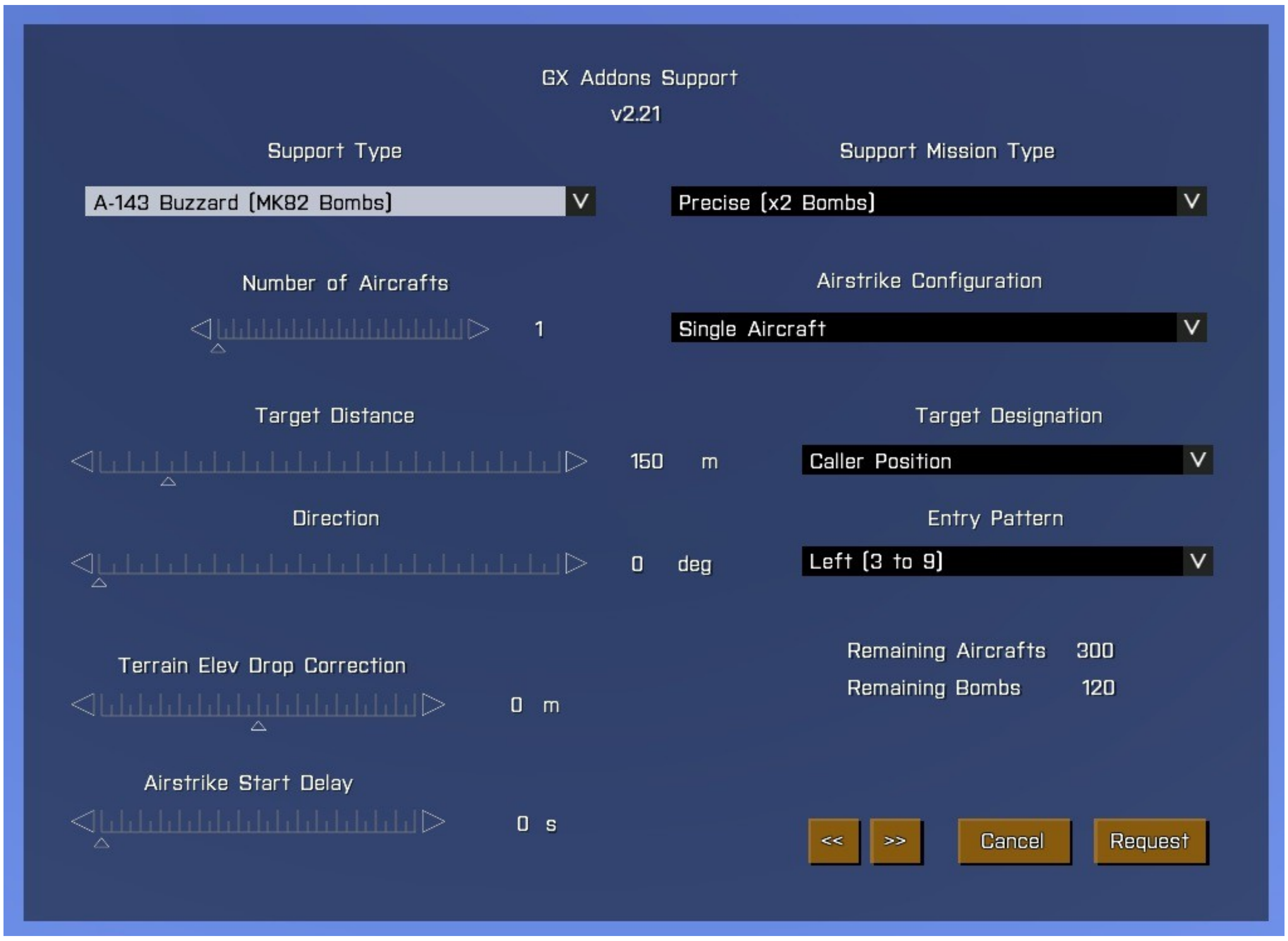
Use the support mission type combo to configure the battery to fire. This will define the number of pieces the fire mission will use. You can also define how many salvos the battery will fire and the delay between each salvo using the artillery salvos and delay between salvos sliders (e.g: a 4 pieces battery firing 2 salvos will shoot 8 shells in total).

Some batteries can fire several types of ammunition, such as explosive, smoke, illumination, etc. To define this use the artillery configuration combo.

Once the battery is set you have to provide the vectors to target from your current position. In order to do so define the target distance and direction using the sliders. The dispersion of shells in target can be defined by the dispersion combo presets and/or fine tuned with the dispersion correction slider.

If your mission is ready you can call for an adjustment shot and readjust the parameters or directly request fire for effect on the target using the orange buttons.

Bombing and helicopter attack request


GX Addons Support
v2.21

Support Type: A-143 Buzzard (MK82 Bombs)
Support Mission Type: Precise (x2 Bombs)
Number of Aircrafts: 1
Airstrike Configuration: Single Aircraft
Target Distance: 150 m
Target Designation: Caller Position
Direction: 0 deg
Entry Pattern: Left (3 to 9)
Remaining Aircrafts 300
Remaining Bombs 120
Terrain Elev Drop Correction: 0 m
Airstrike Start Delay: 0 s

Use the support mission type combo to configure the bomber type or the helicopter attack to use and the number of aircrafts with the slider. Now you can select the diferent payload and mission type in the support mission type combo. Here you can request precise bombings or spread the bombs on target.

The airstrike configuration combo defines the sequence of aircrafts on target. This can be a formation bombing at once or a sequence of aircrafts one after the other.

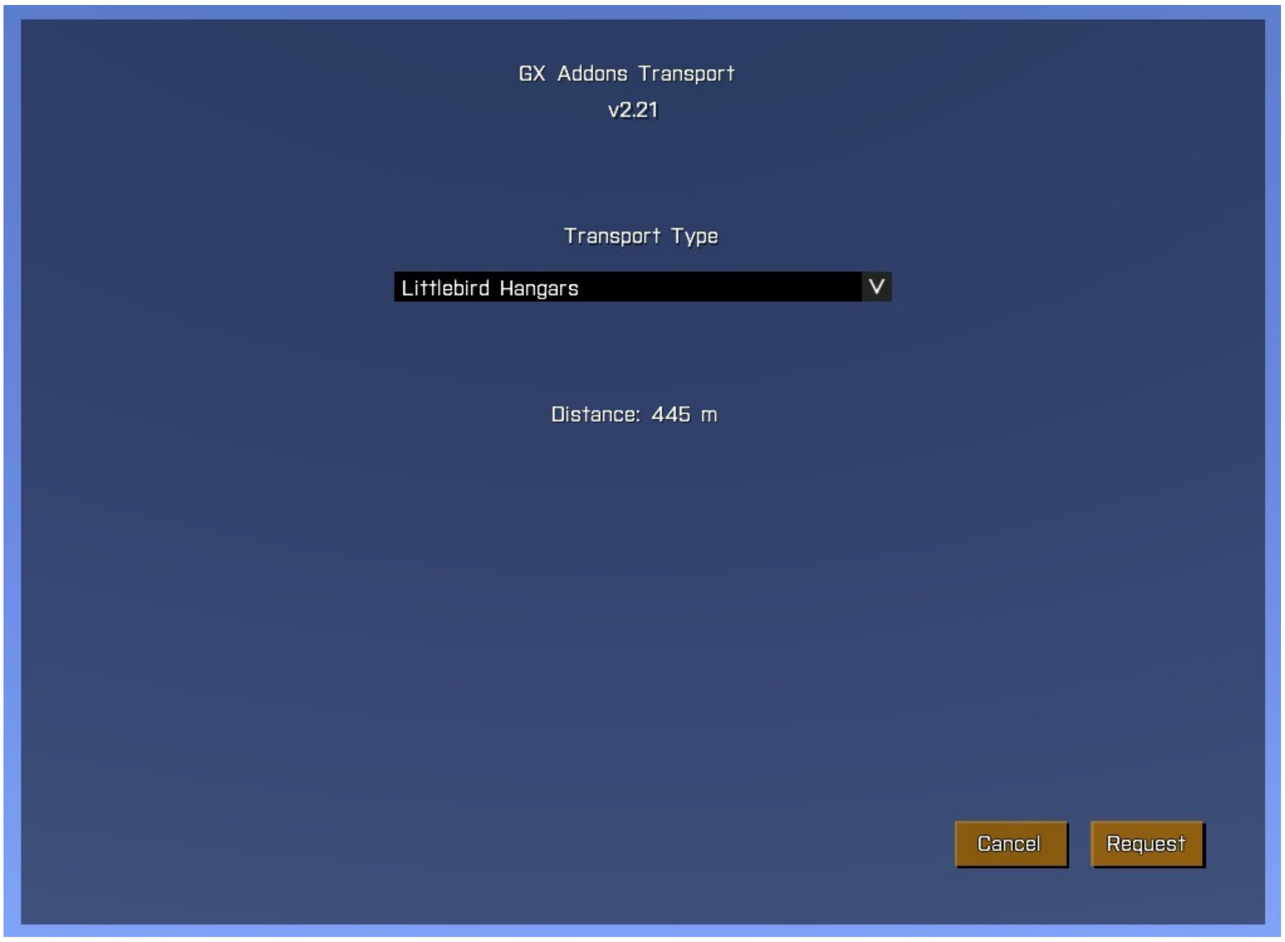
The airstrike can be programmed to start after some time once the request was performed. To define it use the start delay slider.

Last step is the definition of vectors to target and target designation mode. As in the case of artillery you need to define distance and direction from your possion but now you can choose to designate target (and therefore distance and direction) using also smoke markers. To do so use the target designation combo. Pilot will check for smoke grenades prior to attack and may abort mission if no smoke marker was visible in a target designation by smoke.

Last settings will be the entry pattern to objetive and the terrain elevation correction. Those are very important to succeed in the request when you want to drop bombs on valleys or hills. As a **general rule** try always to tell the bombers to enter in a direction to target where terrain is as flat as possible, because pilots will find easier to lock the target and precision will be higher. If your target is on a hill adjust the terrain elevation correction to a value closer to the hill´s height using a positive value and if your target is on a valley or downhill use a negative value. If the entry pattern crosses some elevations in terrain close to the bombing point you will also need to add a positive value to the correction. Just test it yourself.

In addition to this the **CAS mission** type is also available. Bomber will search for enemies and engage at will.

Transport request


GX Addons Transport
v2.21

Transport Type

Littlebird Hangars

Distance: 445 m

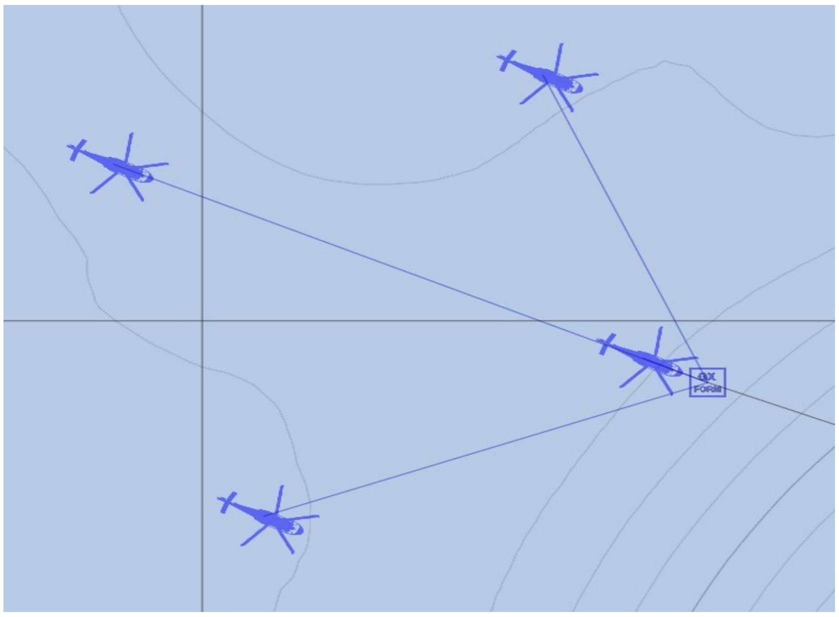
The transport dialog is really simple. Just select the transport from the combo (has to be configured in the mission editor) and it will come to your position. In case of emergency (e.g: designated LZ is under fire) or if transport is delayed in the landing just request it again to ask for an emergency extraction. That will force it to stop or land.

Air Formation Module

Creates custom air formations (compatible with sea and land units too).

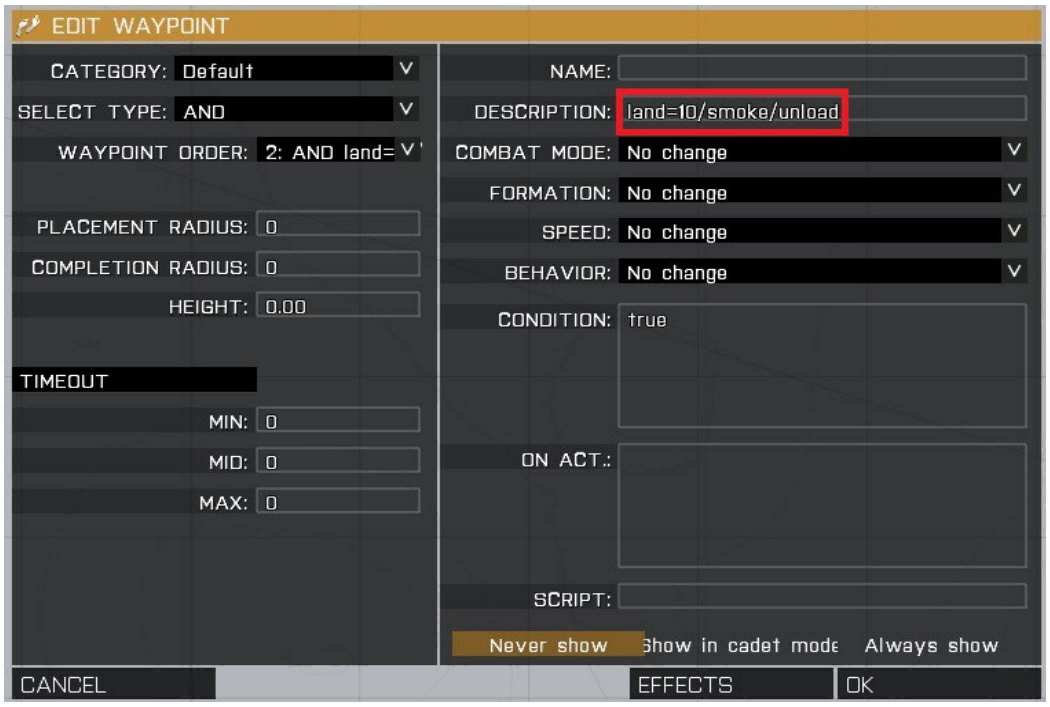
Place the module in editor and synchronize it to each of the air units in the desired order. First synchronized unit will be the formation leader but it will be reassigned in case it´s downed. The initial position of the units in the map will define their custom formation structure but predefined formations can be used as well dynamically.

Add waypoints to the module to define the path for the formation and the actions to execute on them. Intermediate sub-waypoints will be defined automatically to improve the navigation, which is based on bearing to the destinations.



You can place as many modules as formations and paths you want to create.

The landing procedure involves several options such as loading, unloading and secuential LZ approaching. Further wp configuration can be set on the module wp description field, setting variables with "=" and separating different actions with "/".



EDIT WAYPOINT

CATEGORY:	Default
SELECT TYPE:	AND
WAYPOINT ORDER:	2: AND land=
PLACEMENT RADIUS:	0
COMPLETION RADIUS:	0
HEIGHT:	0.00

TIMEOUT

MIN:	0
MID:	0
MAX:	0

NAME:	
DESCRIPTION:	land=10/smoke/unload
COMBAT MODE:	No change
FORMATION:	No change
SPEED:	No change
BEHAVIOR:	No change
CONDITION:	true
ON ACT.:	
SCRIPT:	

Some examples below:

alt=20/vel=75 --> fly at 20m altitude with 75 km/h speed
baro=100 --> fly at 100m barometric altitude (ASL)
altoff=10 --> apply an altitude offset to each unit in the formation
unload --> unload troops (used to deploy troops at hover or for ground vehicles)
land --> land (and terminate script)
land=10 --> land (and wait 10s on LZ)

fire --> fire main weapon
fire=1 --> fire weapon 1
cycle or repeat --> repeat the whole path
delete --> delete the unit
wait=10 --> pause script (and wait 10s to resume)
freeze=10 --> freeze vehicle (and wait 10s). Used to precisely sync events in a mission
freeze=name --> freeze (untill variable name comes true)
name --> pause script untill variable name comes true

Additional parameters for landing action

When landing action is set the following commands can be added as well:

helipad --> precise landing via helipad (for complex areas)
unload --> unload troops at LZ
load --> load nearest troops at LZ
lz --> creates a sequence landing on a single point
lz=2 --> sequence landing (2 choppers landing at the same time)
smoke --> mark LZ with smoke
arty --> creates a hot LZ under arty fire
arty=200 --> hot LZ (arty within 200m)

Formation presets

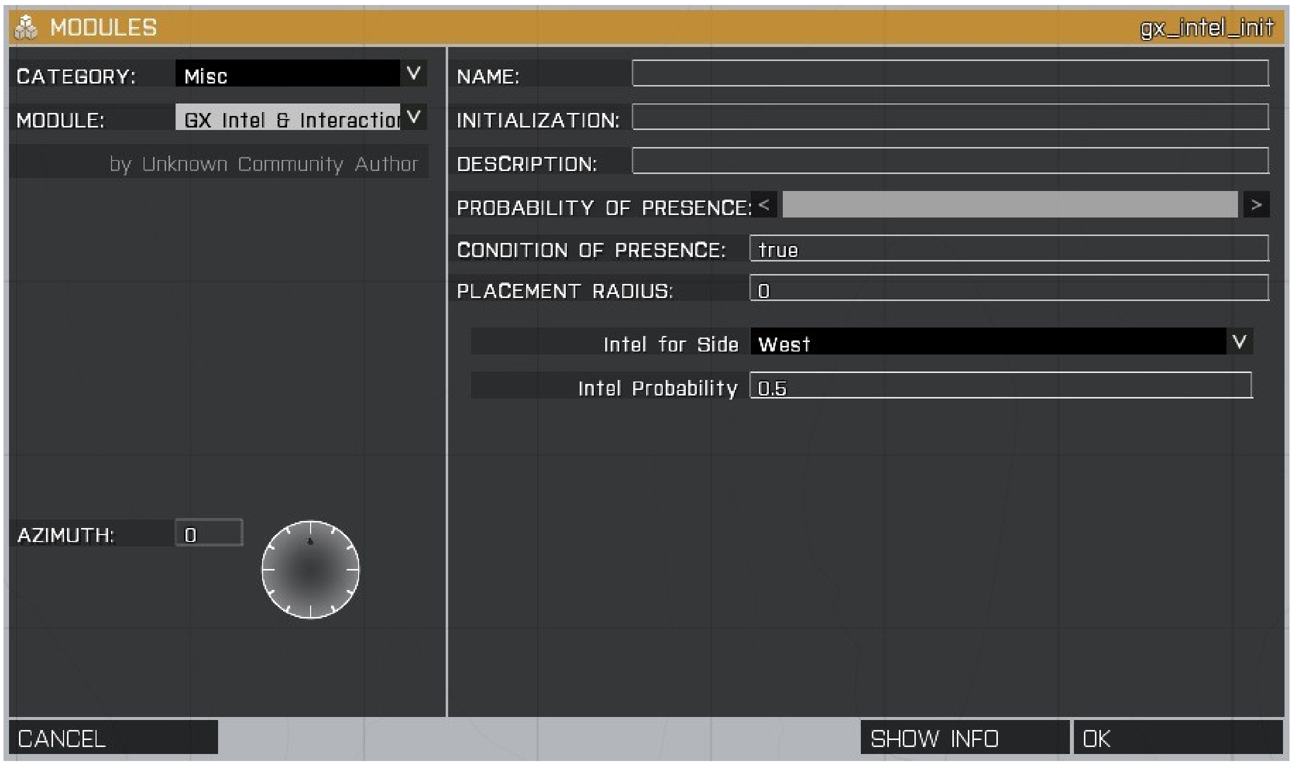
A set of predefined formations is available. Custom separation can be defined as well.

formcolumn --> column
formcolumn=70 --> column with 70m separation
formleft --> echelon left
formright --> echelon right
formline --> line (up to 10 units only)
formwedge --> wedge (up to 10 units only)
formdiamond --> diamond (up to 4 units only)
formreset --> reset to the default positions (defined in mission editor)

Intel & Interaction

This module enables interaction with units to allow talking, searching for intel information, checking and arresting civilian and enemy units. This module is based on the base idea of great Civilian and Interaction module by Nielsen, but addapted to the needs of the intel system.

Simply place the module in editor and configure which side will be capable of searching for info and the probability to find such info.



CATEGORY: Misc
MODULE: GX Intel & Interaction
by Unknown Community Author
AZIMUTH: 0
NAME:
INITIALIZATION:
DESCRIPTION:
PROBABILITY OF PRESENCE:
CONDITION OF PRESENCE: true
PLACEMENT RADIUS: 0
Intel for Side West
Intel Probability 0.5

All players will have the basic actions on mouse wheel menu. The actions will be available while any of the interaction keys is pressed (LShift or Rshift keys by default). The keys can be configured by the variables: gx_intel_key1 and gx_intel_key2.

To activate the actions: point to the desired object or unit, hold down any of the action keys, scroll down to select the action and click it. The tipical actions include:

- Stop land vehicles, ships and units
- Order crews to disembark from vehicles and ships
- Interrogate units
- Register units (search for items and intel)
- Immobilize units on the ground
- Arrest and release prisoners

Anytime the conditions are met the searching for intel will drop an intel item to the ground in front of the searched unit and information will be auto added to players intel menu (map menu) for further reviewing, with any of the following results:

- CardID: basic identifycation card
- Photo: rebel propaganda or any other picture
- Document: intel document with information about enemy targets locations (needs to be open via intel menu available on map to collect the info about target locations)

Documents can be optionally linked to mission targets locations and will include any of the predefined locations set by mission maker. Those locations are units or markers placed on mission editor previously. They are defined by the following array variable: gx_intel_targets. For example, to include two markers positions just add their names to the array: gx_intel_targets = [“marker1”, “marker2”].

You can hide those markers to players via the following code:

```
{_x setmarkeralpha 0} foreach gx_intel_targets;
```

The intel items pictures (1024x1024 .paa format) are defined in the following files (add your files names to the arrays): gx_intel_pic_doc, gx_intel_pic_card and gx_intel_pic_photo

Examples:

```
gx_intel_pic_doc = ["doc1","doc2"];
gx_intel_pic_card = ["idcard1"];
gx_intel_pic_photo = ["prop1"];
```

By default Arma3, Unsung and Iron Front ones are natively supported by the module. To add your own pictures to the mission define your pictures path in gx_intel_modpath variable as follows:

```
gx_intel_modpath = "<your mission intel folder>\Default\";      for default Arma3 missions
gx_intel_modpath = "<your mission intel folder>\NAM\";          for Unsung based missions
gx_intel_modpath = "<your mission intel folder>\WWII\";         for Iron Front based missions
```

You can also define which sentences the units will speak to the side defined as intel capable. Use the following arrays to include yours: gx_intel_talk_for to include sentences supporting your side and providing useful info and gx_intel_talk_against when unit is against your side (eg: when interrogating rebels).

Examples:

```
gx_intel_talk_for = ["Register people to search for info", "Enemy is close. Take care"];
gx_intel_talk_against = ["Fuck you killers!"];
```

You can override those variables in your mission init.sqf for instance.

If you want to add specific intel information to a unit use the following:

```
[<unit>, <intel type>, <target marker>, <content text>] call gx_intel_add
```

Example: place the following into an object init field to generate a document linked to marker1 position and including a customized text.

```
[this, "doc", "marker1”, "Possible meeting point position"] call gx_intel_add
```

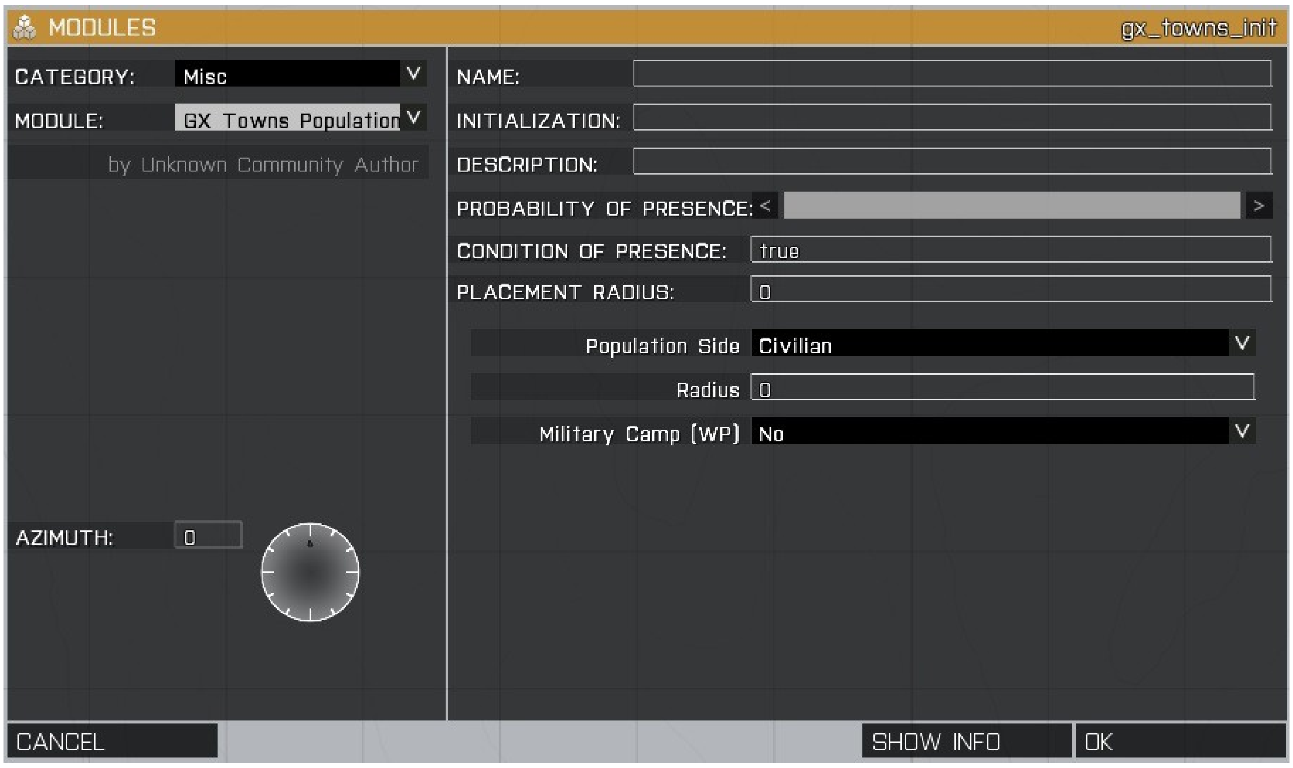
Population and Traffic Modules

There are some other modules to generate traffic, units and misc ambience to the mission. For them some common variables are used to define the classnames of units to spawn per side, whenever the module allows such configuration. Support for NAM and WWII is included by default. They are explained in each particular module section.

Towns Population: this module allows to generate civilian population and road traffic in particular areas of the map. It can also be used to spawn soldiers in town areas or bases. Two methods are available for extra flexibility: module position or waypoints position. You can place as many modules as you want to populate diferent map areas depending on the mission needs.

NOTE: If the module is used in a mission it requires **UPSMON script** to be included in mission folder as well to be able to spawn units using UPSMON.

In the method one you just place the module and configure the side for spawned population and radius to search for villages, towns and cities locations in the map.

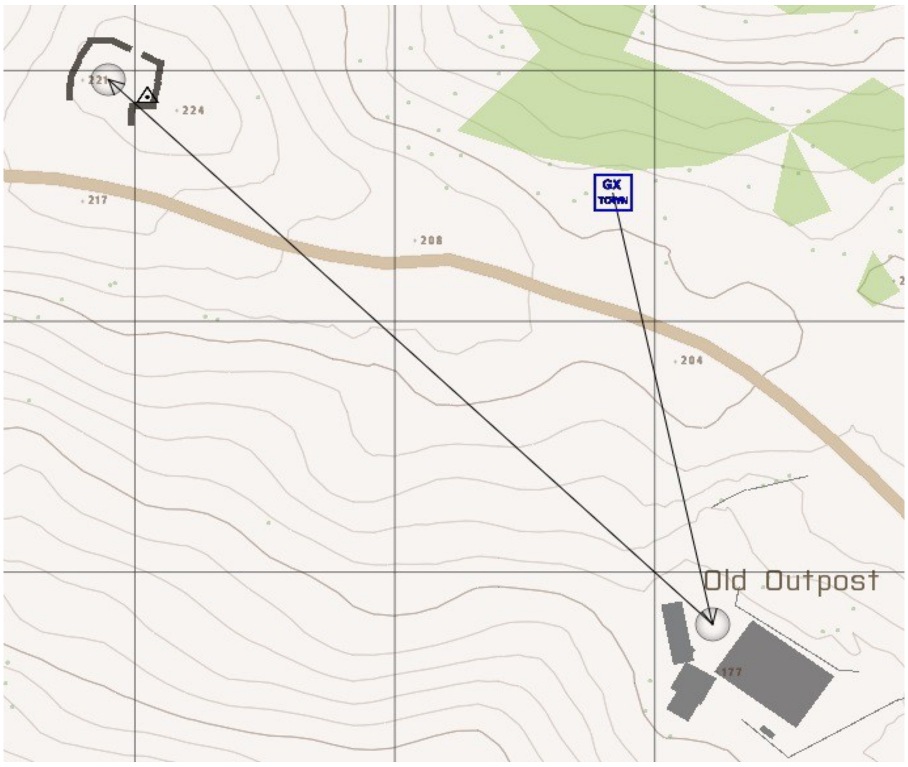


CATEGORY:	Misc
MODULE:	GX Towns Population
AZIMUTH:	0
NAME:	
INITIALIZATION:	
DESCRIPTION:	
PROBABILITY OF PRESENCE:	
CONDITION OF PRESENCE:	true
PLACEMENT RADIUS:	0
Population Side	Civilian
Radius	0
Military Camp (WP)	No

In the second method you place the module and add waypoints to it. In this case the module will only use the waypoints positions to create population (radius is ignored). You can also set the waypoints positions as military camps. Use this method to place population in areas not configured as villages in the map or to spawn units in custom areas such as bases.

The military camp option is mainly intended to spawn tunnels, bunkers and sniper trees in Unsung mod but it could be expanded in future releases.

In both cases, when civilian or independent sides are used, in addition to population also road traffic in nearby roads will be spawned using civilian vehicles.



The following variables are used to define the class names of units and road vehicles to be spawned by the module:

gx_var_unittype_civ, gx_var_unittype_east, gx_var_unittype_west, gx_var_unittype_other and gx_var_cartype_civ

Example:

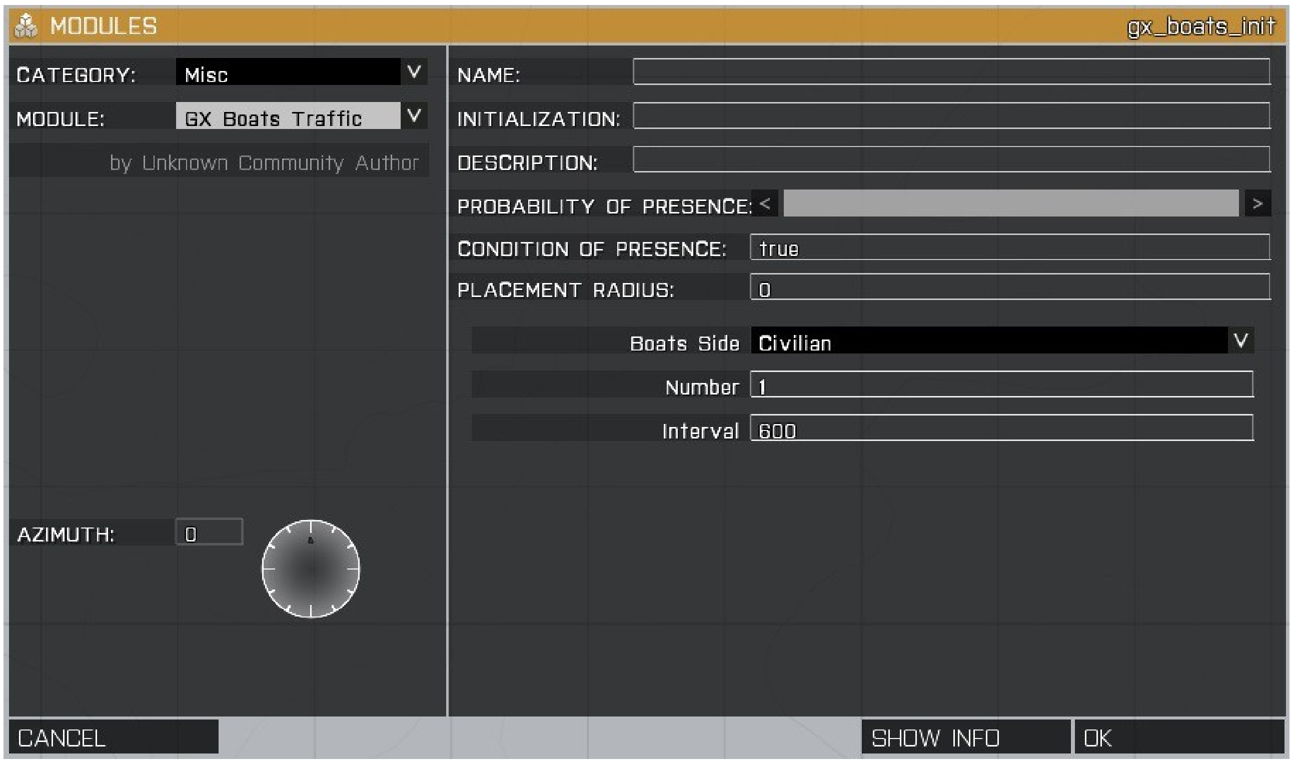
```
gx_var_unittype_civ = ["C_man_1","C_man_polo_2_F","C_man_polo_6_F"];
gx_var_cartype_civ = ["C_Offroad_01_F","C_Van_01_transport_F","C_Van_01_box_F"];
```

In case of civilian population mission maker can define the probability of generation of hostile civilians (rebels). They will be enemy to the side which is capable of searching for intel and may also include some additional intel information about targets. The probability of presence of such rebels is set in the following variable: gx_village_rebelprob.

You can override those variables in your mission init.sqf for instance.

Boats Traffic: enables the creation of boats traffic patterns in open sea or rivers. Place a module in editor and add waypoints to it to define the path for spawned boats. You can also define the side, number of spawned boats and delay interval for each spawned boat. You can place as many modules as traffic patterns you desire, for frienly, enemy or civilian routes.

If the last waypoint is placed closer than 200m from the first one then the pattern will cycle automatically to the first one again.



CATEGORY:	Misc
MODULE:	GX Boats Traffic
AZIMUTH:	0
NAME:	
INITIALIZATION:	
DESCRIPTION:	
PROBABILITY OF PRESENCE:	
CONDITION OF PRESENCE:	true
PLACEMENT RADIUS:	0
Boats Side	Civilian
Number	1
Interval	600

To configure the class names of spawned boats use the following variables:

gx_var_boattype_civ, gx_var_boattype_east, gx_var_boattype_west and gx_var_boattype_other

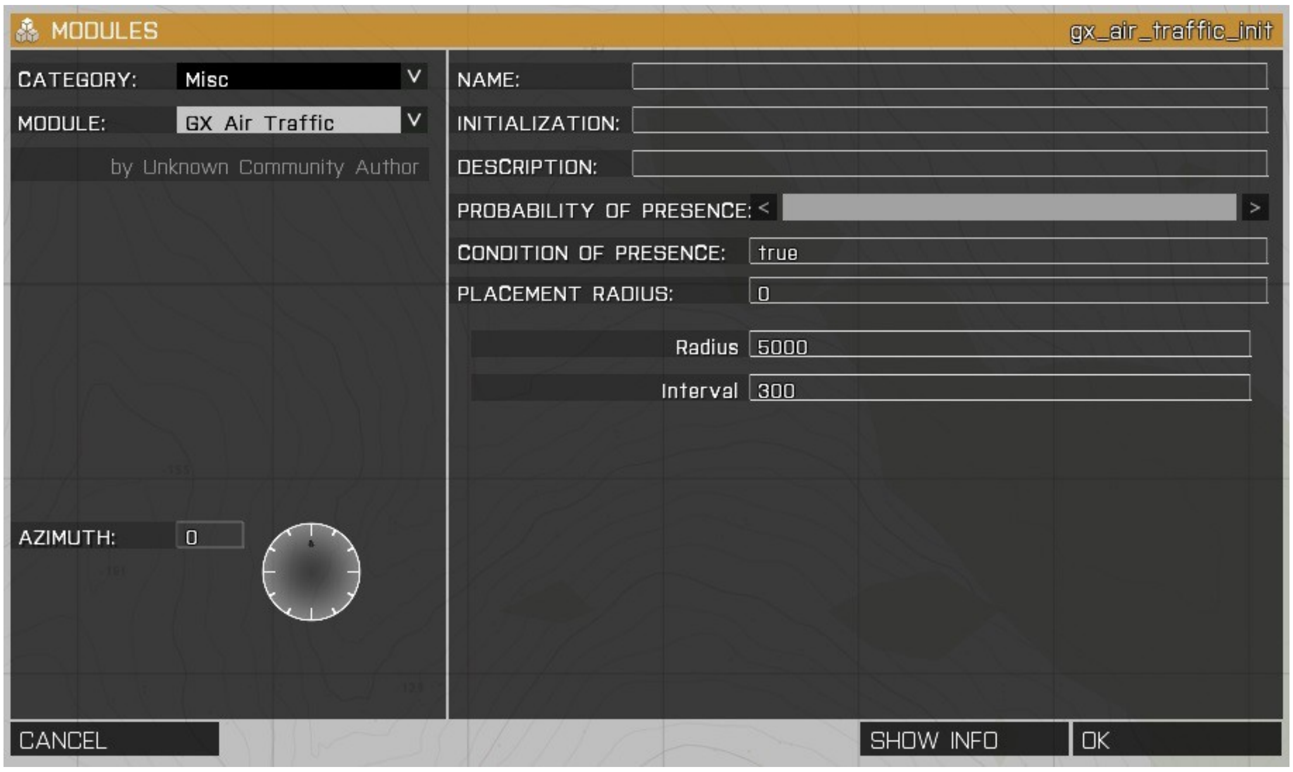
Examples:

```
gx_var_boattype_civ = ["B_G_Boat_Transport_01_F"];
```

The crew class names will be taken from the population module class names.

You can override those variables in your mission init.sqf for instance.

Air Traffic: enables the creation of random air formations traffic patterns in the map near players. Place only one module in editor and configure the radius and delay interval. Best is to place it in the middle of the map and define half the map size as radius.



MODULES gx_air_traffic_init

CATEGORY: Misc
MODULE: GX Air Traffic
by Unknown Community Author
NAME:
INITIALIZATION:
DESCRIPTION:
PROBABILITY OF PRESENCE:
CONDITION OF PRESENCE: true
PLACEMENT RADIUS: 0
Radius 5000
Interval 300
AZIMUTH: 0

The following variables are used to define the class names of aircrafts (light transports, heavy transports or jets) to be spawned by the module.

Air traffic:
gx_var_airtraffic_light, gx_var_airtraffic_heavy and gx_var_airtraffic_jets

Example:

```
gx_var_airtraffic_light = ["B_Heli_Transport_01_F","I_Heli_light_03_F"];
```

This classification is just used to group the aircrafts, as the module will cycle between light, medium and jets lists every time.

You can override those variables in your mission init.sqf for instance.

Other Modules

Wild Animals: enables the simulation of wild animals which will attack the soldiers and produce damage on them. Place only one module in editor and the supported animal types will attack when close enough to them.

Currently the wildboar is supported by default. For Unsung case also the water buffalo and snakes are included.

To add animals to this module add their class names to the variable array: gx_wild_animals_type

Example:

```
gx_wild_animals_type = ["CSJ_Snake2","CSJ_Snake3","uns_waterbuffalo01"];
```

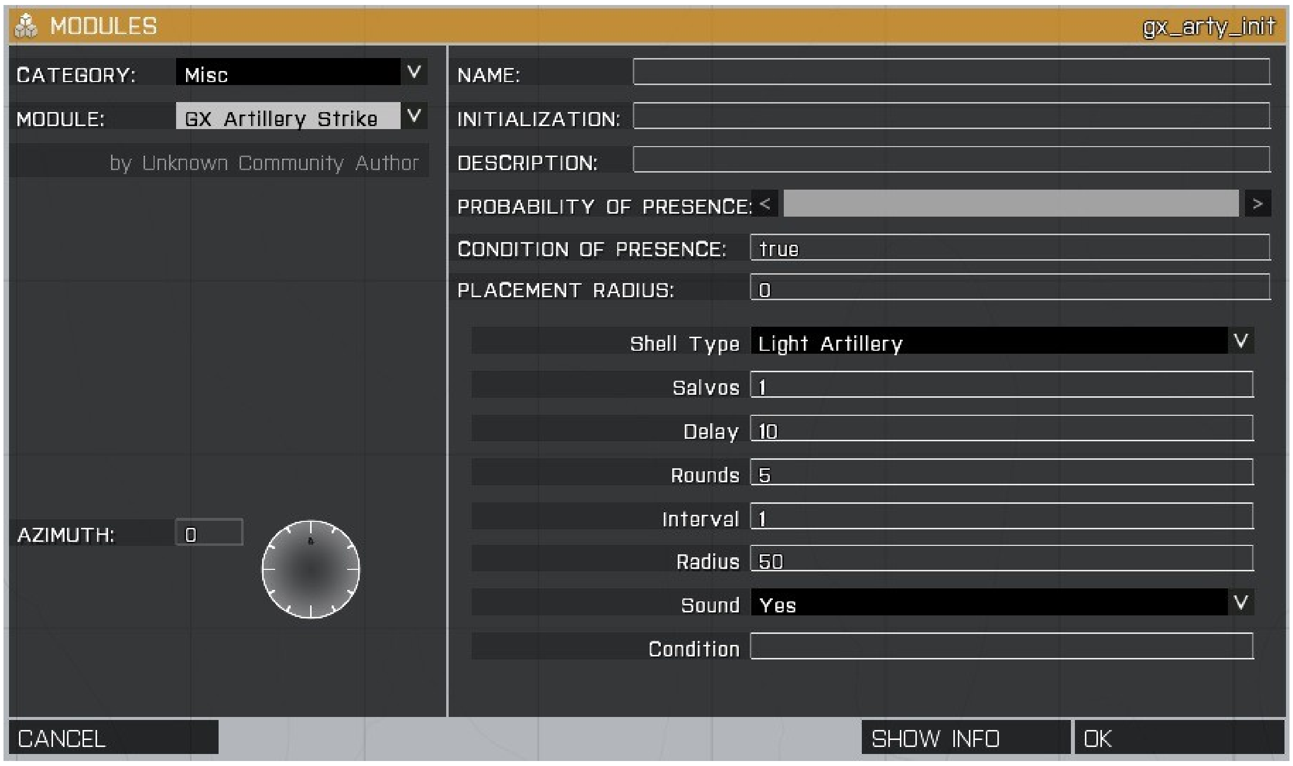
You can override those variables in your mission init.sqf for instance.

UNS Traps: enables the activation of gx_fn_trap function to any Unsung explosive trap placed in mission. This module is just intented to avoid having to activate the function on each trap init field individually.

Standalone artillery and bomber strike modules

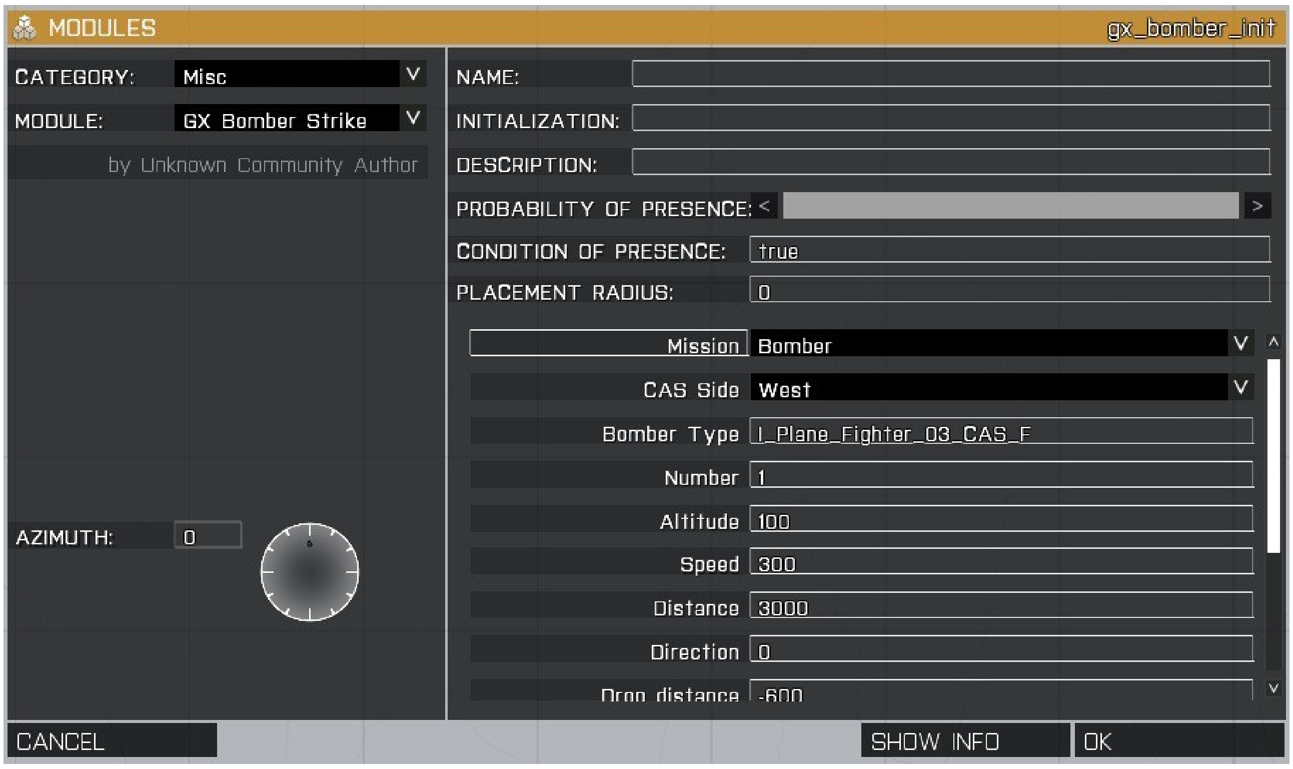
Independently from the main support module you can create artillery or bomber strikes in mission, for instance to generate ambientation, using the following two modules. Place them in the mission and define their parameters and optionally a start condition, which will be the one to trigger the module (e.g: time > 30).

Both modules strike positions can be defined at a single position (the module position) or using waypoints on the modules, creating then simultaneous strikes at several defined positions.


MODULES — gx_arty_init

CATEGORY:	Misc
MODULE:	GX Artillery Strike
	by Unknown Community Author
AZIMUTH:	0
NAME:	
INITIALIZATION:	
DESCRIPTION:	
PROBABILITY OF PRESENCE:	
CONDITION OF PRESENCE:	true
PLACEMENT RADIUS:	0
Shell Type	Light Artillery
Salvos	1
Delay	10
Rounds	5
Interval	1
Radius	50
Sound	Yes
Condition	

Bomber strike module can be set to spawn a bomber, CAS support or simply a flyby air traffic.


MODULES — gx_bomber_init

CATEGORY:	Misc
MODULE:	GX Bomber Strike
	by Unknown Community Author
AZIMUTH:	0
NAME:	
INITIALIZATION:	
DESCRIPTION:	
PROBABILITY OF PRESENCE:	
CONDITION OF PRESENCE:	true
PLACEMENT RADIUS:	0
Mission	Bomber
CAS Side	West
Bomber Type	I_Plane_Fighter_03_CAS_F
Number	1
Altitude	100
Speed	300
Distance	3000
Direction	0
Drop distance	-600

Standalone functions

The following functions are available at any time since the addon is initialized (no extra configuration is needed):

Artillery (creates arty or smoke rounds at a defined position, playing a round sound or not)

Usage: [array of variables] call gx_fn_arty

Array of variables:
[unit or position or marker,number,interval,radius,["GrenadeHand","Sh_105_HE"], sound]

Example:
[player, 10, 1, 100, ["Sh_105_HE"], true] call gx_fn_arty

This will create an strike at player position, with 10 SH105 rounds, with 1s delay within 100m radius and play a sound for each incomming round

Aircraft (spawns an aircraft in mid air to create individual air traffic)

Usage: [array of variables] call gx_fn_aircraft

Array of variables:
[unit type or unit,unit or position or marker,altitude,speed,distance,dir]

Example:
[“F35B”,”marker1”,100,300,1000,90] call gx_fn_aircraft

This will create a F35 with reference to marker1. Jet will be spawned at 1000m from marker position, fly in heading 90, at 100m altitude and 300 km/h

Bomber (spawns a vehicle which fires its weapons at a defined position)

Usage: [array of variables] call gx_fn_bomber

Array of variables:
[unit type or unit,unit or position or marker,altitude,speed,distance,dir, bombing distance offset,weapon type,number,interval]

Example:
[“F35B”,”marker1”,100,300,1000,90,-150,"Mk82Bomb",2,0.25] call gx_fn_bomber

This will create a F35 with reference to marker1. Jet will be spawned at 1000m from marker position, fly in heading 90, at 100m altitude and 300 km/h and will fire 2 times Mk82 bombs, 150m before the marker with an interval of 0.25s between each bomb drop

NOTE: The selected weapon will be added to the bomber even if it didn´t have it before

CAS (spawns a vehicle which engages ANY units at a defined position. Friendly side parameter is optional)

Usage: [array of variables] call gx_fn_cas

Array of variables:
[unit type or unit,unit or position or marker,altitude,speed,distance,dir,time, <optional side>]

Example:
[“F35B”,”marker1”,100,300,1000,90,240,west] call gx_fn_cas

This will create a F35 with reference to marker1. Jet will be spawned at 1000m from marker position, fly in heading 90, at 100m altitude and 300 km/h and will search for units on other sides than the west one during 240s

Cargo (moves instantly a squad into a vehicle assigning them as cargo)

Usage: [squad leader, vehicle] call gx_fn_cargo

Example:
[soldier1, jeep1] call gx_fn_cargo

This will move squad1 into jeep1

Getin (orders a squad to enter a vehicle assigning them as cargo)

Usage: [squad leader, vehicle] call gx_fn_getin

Example:
[soldier1, jeep1] call gx_fn_getin

This will order squad1 to get enter jeep1

Getout (orders a squad to exit from a vehicle)

Usage: [squad leader, vehicle] call gx_fn_getout

Same as getin, but commands the exit from vehicle

Jump (orders a squad to paradrop from a vehicle)

Usage: [squad leader, vehicle] call gx_fn_jump

Same as getout but with paradrop

Fire weapon (orders a unit to fire a selected weapon)

Usage: [unit,weapon type,number,interval] call gx_fn_fireweap

Examples:
[tank1,0,100,0.5] call gx_fn_fireweap

This will make tank1 to fire 100 times its weapon number 0 with an interval of 0.5s

[man1,"M16",100,0.5] call gx_fn_fireweap

This will make man1 to fire 100 times its M16 weapon with an interval of 0.5s

NOTE: to display the weapons for an unit and its type use:
[man1] call gx_fn_fireweap

Trap (creates an explosive trap on a selected object)

Usage:
[target object or unit, side which activates trap, probability to explode] call gx_fn_trap

Example:
[box,west,0.5] call gx_fn_trap

This will make a placed ammo box (named box) to become a trap with 50% probability and detonate if west units approach it

Intel Item (creates an interactive intel item on a selected object)

Usage:
[target object or unit, doc type, marker, text] call gx_intel_add

Example:
[man1,”pic”] call gx_intel_add

This will add a picture intel type to a soldier.

Zeus Advanced Script

Read the separate documentation for the usage and configuration of this module.

Support module variables override (advanced users only)

NOTE: change those variables only if you know what you are doing as wrong configuration values will drop errors on the scripts. Use the **support cfg module instead**. Remember to **publish the new variable to all clients during MP** by using publicvariable if you altered the variables after mission started.

In the mission editor or dynamically in time, the following variables values for the support module can be set:

gx_var_pilot --> defines the classname for spawned pilots (generic civilian)
gx_var_pilot_west, gx_var_pilot_east, gx_var_pilot_other --> defines the side spawned pilots
gx_var_markdot --> defines the marker classname for map icons

The support parameters are defined by the following parameters:

gx_asp_delayarty --> defines the delay since arty was called to strike start
gx_asp_delaybomber --> defines the delay since bomber was called to strike start
gx_asp_casduration --> defines the duration of CAS missions

The following variables are used to configure the artillery support individually per side (west, east or other):

gx_asp_callsign_west, gx_asp_callsign_east, gx_asp_callsign_other
gx_asp_aircrafts_west, gx_asp_aircrafts_east, gx_asp_aircrafts_other
gx_asp_bombs_west, gx_asp_bombs_east, gx_asp_bombs_other
gx_asp_arty_west, gx_asp_arty_east, gx_asp_arty_other

gx_asp_callsign --> defines callsign for support center
gx_asp_aircrafts --> defines max number of bombers available
gx_asp_bombs --> defines max number of bombs available for the bombers
gx_asp_arty --> defines max number of arty rounds available

The batteries configuration (number of pieces and ammunition) is global to all sides.

Arty rounds and bomber classnames can be set as well (refer to external file with pre-configurations examples). Artillery batteries and ammunition are global to all sides. The only configurable parameters which are side dependant for the artillery are the number of rounds available.

The artillery ammunition can be set by the following variables:

gx_asp_lshell --> defines the light battery configuration
gx_asp_mshell --> defines the medium battery configuration
gx_asp_hshell --> defines the heavy battery configuration
gx_asp_illum --> defines the ammo type for illumination
gx_asp_smokemarker --> defines the ammo type for smoke

The bombers configuration can be set global or individually per side (west, east or other) by the following variables:

gx_asp_cfgjets

gx_asp_cfgjets_west
gx_asp_cfgjets_east
gx_asp_cfgjets_other

To define your bombers configuration you have to provide the arrays for each side in the following format:

```
gx_asp_cfgjets = [<bomber1 config>, <bomber2 config>, <bomber3 config>, <bomber4 config>]
```

Each bomber config set uses the following variables:

<bomber1 config> = [Name, Aircraft type, altitude, [drop distances], weapon type, [number of bombs], drop interval]

- The name entry defines the display name for bomber in the GUI
- Aircraft type defines the classname of the bomber
- Altitude is the flight altitude for the bomber
- The drop distances and number of bombs arrays will define the missions types in the GUI
- Weapon type entry is the classname of the weapon to fire
- Drop interval is the delay between each bomb drop

In orther to tune your values just put yourself in a flat area such as the runway and call the bombing to see where bombs hit the target to adjust the values. Remember you can always use the CAS support available in GUI instead and forget about custom configurations.

You can define up to 4 bomber configuration per side and even swap them between sides during mission with scripts or triggers. The easiest way is to create a template file in the mission init.sqf with:

```
bombertemplate1 = [<config1>, <config2>, <config3>, <config4>]
bombertemplate2 = [<config5>, <config6>, <config7>, <config8>]
```

Then during mission you can reassign the side variables by doing (remember to publish them to clients):

```
gx_asp_cfgjets_east = bombertemplate1; publicvariable “gx_asp_cfgjets_east”
gx_asp_cfgjets_west = bombertemplate2; publicvariable “gx_asp_cfgjets_west”
```

The helicopter CAS configuration can be set global or individually per side (west, east or other) by the following variables:

```
gx_asp_cfgcas

gx_asp_cfgcas_west
gx_asp_cfgcas_east
gx_asp_cfgcas_other
```

Those variables are much simpler than the bombers ones as they just include the GUI name description and class name for the helicopters. Also up to 4 variants can be used here:

<helicopter1 config> = [Name, helicopter type]

NOTE: since version v2.31 native support to Unsung and Iron Front mods is included by default. User does not need to define bombers configuration while in mission as mods are autodetected and predefined bombers configuration is set automatically. You can still override it in the mission if you need so.