MSO Editor's Guide

Version 0.8 for MSO v4.5
17 Nov 2012
http://dev-heaven.net/projects/mso

What Is MSO?

Multi Session Operations is a modular mission framework. Essentially it is a collection of scripts that create a persistent, living scenario in which the player operates. Unlike regular missions that have pre-scripted objectives, MSO procedurally generates enemy forces that are unpredictable and (semi) random. Players have to think beyond the mere tactical level and build up the intelligence picture, carrying out recce patrols to identify enemy locations and conduct deliberate attacks to neutralise them. Everything is persistent, including equipment, ammo, vehicles and even player lives.

Downloading MSO

You can download the MSO zip file from - http://dev-heaven.net/projects/mso/files. Make sure you have the latest version.

Unzip the zip file into your ARMA2 folder. This will extract over 20 sample MSO missions - that are ready to play - into your MPMission folder.

Additionally, it will extract the necessary MSO files and folders into MPMIssions\MSO_BASE_MISSION_CODE_[VER] for any custom mission you wish to create .

MSO Mission Making

Creating a mission with MSO is fairly simple. The modules are procedural and self discovering. In other words, they will detect which map they are on, generate enemy locations automatically and do most of the hard work for you. Most of the general settings can be changed in the parameters at mission start, so there's very little need to dig into the scripts. This guide assumes you have a basic knowledge of mission editing.

Note: Map locations are defined in the map config. 3rd party maps can be added into the core/functions/common/fn_initLocations.sqf script. Some 3rd party addon maps do not have predefined locations, in which case MSO will automatically generate them via the ..\core\functions\server\fn_createLocations.sqf script. If the map bottom left position (0,0,0) is set incorrectly, this script will not work and locations will have to be defined manually.

HINT for ppl with knowledge of excel: If you want to place many locations yourself manually, put down markers named hill, hill_0, hill_1 and strongpoint, strongpoint_0 etc. in mission-editor (just copypaste the markers where you want em), save, and then copy the content of the mission.sqm to excel. You can easily change x,z,y coordinates to x,y,z of the regarding markers at once. It saved me hours.

In the Map Editor

Open the mission editor and pick a suitable location for a friendly operating base (and an OPFOR base if you're creating a TvT scenario), for example a military base or airfield. Each base location needs enough space to place the following:

- Some playable units
- A few respawnable vehicles (see vehicle respawn scripts below)
- Some defence stores and logistics equipment like sandbags and ammo crates (see resupply and logistics scripts below)
- An HQ tent (for recruitment and team status)
- UAV Terminal (for checking the weather) and possible playable UAV CAS
- A medic tent and a "Hospital' marker (for CASEVAC)
- A notice board (for posting messages for other players)
- Add names to any object or vehicle that you may wish to be saved to the database (when using the persistentDB module)

We need to register the factions with the game engine. The easiest way to do this is add a 'hidden' unit of each faction. Add an enemy rifleman, an independent and a civilian and set their probability of presence to zero for each of them.

Respawn markers - MSO MultiSpawn function allows multiple factions to respawn in different locations. So for example, USMC could respawn at a coastal FOB and US Army at a base in the centre of the map. For it to work properly, we need to add spawn points for each faction. Start by putting an empty marker at the bottom left of the map (by convention it's usually put at 0,0,0) and name it *respawn_west.* Then create a respawn marker in the base for each friendly faction and label them *respawn_<yourfaction>*. It's worth also adding a static spawn point in the main base - see the Support section for details.

Ammunition markers - In order to have MSO automatically support ACRE, place a marker named *ammo* in a suitable area, ideally near the base ammo crates. If you have a second base use *ammo_1*. If the server is running ACRE, an ACRE Radio Box will be automatically placed at the *ammo* and *ammo_1* markers.

ACE: The current version of MSO does not support automatic adding of ACE ammo crates. If you are making an ACE version you need to add ACE ammo crates manually.

Other markers - usually its a good idea to mark where the various MSO assets are on the map, such as vehicles, crew served weapons, building supplies etc. You can specify a custom CAS_Spawn marker to ensure aircraft spawn in a location where there are few mountains (useful for CLAfghan).

Finally, to prevent random enemy patrols appearing in the middle of your base, create a trigger to cover the whole area of your base, set to *BLUFOR Present* and activation *Repeatedly*. Name it *BIS_Zora_0* Now we need to add a few BIS modules and initialise MSO. You also have the option to place additional 10 Blacklist-markers on map, with name CQB_BL0 - CQB_BL9 for blacklists of 500 mtrs to stop CQB based enemy patrols from spawning.

Add the following modules in the editor with the init lines as indicated:

- Functions Module - Init="BIS_fnc_locations = compile preprocessFileLineNumbers "CA\modules\functions\systems\fn_locations.sqf"

- Military Symbols Module - Init="setGroupIconsVisible [true, false]; BIS_marta_mainscope setVariable [""delay"", 300];"
- Garbage Collector

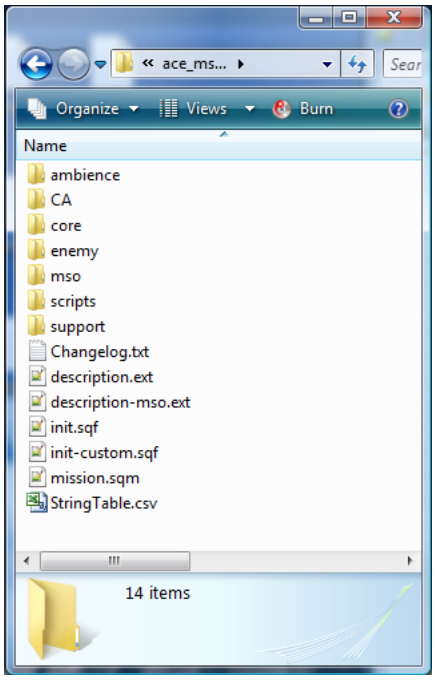
Optionally, you can add the Environmental Effects, Colours and Weather modules. These should all have “!isDedicated” in the Condition of Presence box. You should not however add ALICE, SILVIE or Ambient Animals as all the ambiance is handled separately by MSO.

MSO includes full integration with BIS Warfare. To enable this mode all you need to do is place the Warfare module in the editor and synch your playable units and starter vehicles. Various Warfare settings are available in the mission parameters.

ACE: If you are using ACE, you should add the ACE Respawn With Same Weapons module, the ACE medic modules of your choice and any other ACE modules you like to use, such as the Spare Wheels in Vehicles and Fast Roping modules.

Adding the MSO Modules

Save your mission, alt-tab out of the game, open the mission folder and find your mission.sqm. Now copy and paste all the MSO folders and files from the MSO_BASE_MISSION_CODE_VER folder into your mission folder. Be sure you do not overwrite your mission.sqm. It should look something like this:



- Description.ext must have the line*include <description-MS #O.ext>*, otherwise you can put any normal stuff here.
- If you need to initialise any other custom misadsion (non-MSO) scripts, add them to the init-custom.sqf as you would normally (this is called by init.sqf)

And that's it. You should be able to load the mission in game and enjoy all the fun of a Multi Session Operation.

Note: Almost every module has a _debug line in *main.sqf.* Set this to *true* if you want to get feedback on what MSO is doing, or check the server.rpt file. If you want to disable modules completely, the easiest method is to comment out the respective line in the module.hpp files that can be found in every modules folder.

Description of MSO Modules

MSO Core

This is the heart of MSO and provides all the persistence. Even if you don't include any other modules, you can add these persistent modules to other missions to use the functionality.

PersistentDB (PDB) - This is the system used to save player, mission, object, vehicle and CQB location data to an external database. You will require to set up a database as described here - https://dev-heaven.net/projects/mso/wiki. Do not use NOMAD if you are using PersistentDB. Objects and vehicles require names in the editor to be persisted. PDB also has both player and server auto-save functionality.

RMM NOMAD - The NOMAD system is responsible for saving the state of the player on disconnect, including position, health, weapons and ammo. It is tied to the player UID so you can't 'cheat' by joining in another slot. Parameters that can be set at game start include the option to restricts players to the same *class* type (e.g.if you start as a rifleman, you can only rejoin as a rifleman). You can also define how many times a player can respawn (number of 'lives') and how often these lives regenerate. The minimum requirements to run NOMAD in any mission are:

- In the editor: BIS Functions module and a Playable Unit
- In DESCRIPTION.EXT: #include <core\modules\nomad\params.hpp>
- In INIT.SQF: execNow "core\init.sqf";
- In the mission folder, copies of
  ..\core\init.sqf
  ..\core\modules\modules.hpp (#define RMM_NOMAD)
  ..\core\modules\nomad*.*

MP_Rights - this is another important function of MSO. It's where the mission maker can restrict access to certain functions or vehicles based on the player UIDs (so for example, if you want only NCOs to be able to create tasks or call Close Air Support). The file *mso_uids.txt* has a list of player UIDs with their related accesses (pilots, vehicle crew or admins who are the "leaders" and can do everything). MP_Rights can be disabled altogether in the mission parameters.

Note: The server admin will need to create am MSO folder in the root ArmA2 folder (e.g. ..\arma 2 operation arrowhead\mso) and copy the file *mso_uids.txt* to it.

Caching - This reduces server load by compressing groups of enemy units beyond visual range of the player into single units, although this can break any scripts applied to those groups. Caching is enabled by default. You can disable it in the mission parameters or by commenting out the respective module in ..\core\modules\modules.hpp. We recommend CEP Caching over NOUJAY.

Debug - includes an in game debug console for error checking and executing scripts on the fly. Access can be restricted to admins in *mso_uids.txt.* You can also switch on Debug to see a visual representation on the map of MSO features.

Settings - adds an in game menu for dynamic client side view distance and terrain detail settings, accessed via the interaction key.

Weather - synchronises time and weather across all clients and provides weather forecasts. Add *0 = this execVM "core\modules\weather\addAction.sqf";* to the init line of an appropriate object (UAVTerminal).

Help - as it says on the tin. Basic in game help for new players, displayed at mission start. Intro help can be disabled in init.sqf at line 15 by commenting out or deleting *execNow "core\scripts\intro.sqf"*

MSO Enemy

Zora - based on the BIS module, this spawns random enemy patrols in the vicinity of players and is self balancing depending on how many players are active in the area.

rmm_enemypop - this module automatically generates enemy fixed bases and defences within a random radius of suitable locations around the map (2000m by default). The intensity can be set in mission parameters at game start. Be aware that very high intensity may cause performance issues on low end gaming servers. Since 4.4 there is a spawning/deleting caching method implemented. If all units of a location have been killed it is cleared, otherwise it will reeinforce if player moves out of spawnrange and comes back in, this can be switched off in mission settings but will improve performance a lot. If you experience odd spawn locations, the random radius can be lowered in ..\enemy\modules\rmm_enemypop at line 33: *ep_dist = 2000;*

CQB_EnPop - Uses houses and fortification as spawnpoints, provides Close Quater Combat and brings a new dimension to MSO. If players gets in range AI is spawned in houses, if out of range they are moving back to their houses and get deleted (incl. groups). Uses a Serverside Garbage Collector, cleared positions are stored to PDB. Settings /// CQB AI Limit: Max. 10 AI spawned local on each client (20 players x 10 AI = 200 AI in crowded areas), “Off” means the spawnpoint is suspended after 1 group has spawned
(recommended) /// CQB Group limit: Groupcount must be lower than this limit for CQB to spawn. Recommended settings are 10% / off / off (not change that until you know what you do). Use markers ammo and ammo_1 for a safezone of at least 1000 mtrs around base (it uses safe-zone setting from rmm_enpop or falls back to 1000 default). You have the option to place additional 10 Blacklist-markers on map, with name CQB_BL0 - CQB_BL9 for blacklists of 500 mtrs.

CRB_Terrorists - terrorist cells recruit at intervals set in mission parameters. Each cell is based around a weapons crate or arms cache. Intel reports of suspected terrorist activity are marked on the map with a distance indicating the rough area. If left unchecked the terrorist can overrun the place a bit and impact

server performance. Note that they will employ VB-IEDs (car bombs). *Note: Currently Terrorist cells modules suffers from some performance issues when hosting more than 10 players.*

ROY_PatrolOps - PO2 module is a task generator that generates special operations based on map-locations. There is a variety of different tasks from destroying a radar tower to capturing a corrupt officer. Easily add your custom tasks by creating a scripted mission within the PO2 framework, see roy_patrolops\tasks\types for examples and add your custom tasks in enemy\modules\roy_patrolops\patrol_ops.sqf afterwards.

TUP_IED - Places random IEDs in towns based on enemy proximity or on a random basis. IEDs are a mixture of IED objects and other clutter (such as rubble, tyres, crates, sacks etc). TUP_IED will also place EOD mod IEDs if you are running that mod. TUP_IED places IEDs in areas such as roads, entrances to buildings, low walls, mosques and around houses.

TUP_IEDs can be detected and disarmed if you are an Engineer or using an ACE_Minedetector_US or EOD mod SR5_THOR3 device. Be careful though, getting too close to IEDs can be lethal and sometimes you may come across IEDs that are complex to disarm. EOD mod IEDs cannot be disarmed (yet) but can sometimes be detected by the THOR3 if they are radio controlled. TUP_IEDs and EOD_IEDs can be destroyed by gunfire or grenades. TUP_IEDs will detonate on a proximity basis. EOD mod IEDs are either radio controlled by a spotter or are pressure based (vehicles and/or people).

TUP_IED also provides suicide bombers who will hunt down BLUFOR in towns and VB-IEDs where certain civ cars maybe wired with an IED.

WICT_enemypop - based on the World In Conflict Tool by ArmAIIholic, this provides a number of base locations around the map that spawn groups of enemy who will dynamically create a front line and attempt to recapture lost bases. It works particularly well with smaller groups of players who stick together and work as a team. It is not recommended for long term play (esp. having up the server for several days).

BIS_Warfare - this is the regular BIS Warfare game mode integrated into MSO. Please see the BI Wiki for more details.

CRB_Convoys - enemy logistics convoys regularly move across the AO. Friendly intel will indicate the start, midpoint and end of these convoys (Green, Yellow and Red markers respectively) and generate a Task to interdict and destroy or capture them. At the stage of version 4.2 Patrol Ops 2 Module takes over CRB_convoys to a PO2 task (obsolete).

MSO Support

The support modules include three main functions: situational awareness for the players, combat support, and combat service support and logistics.

J**IP Markers, JIP Tasks, After Actions Reports** and the **Logbook** are all covered in the player guide. Restrict who can use them by adding UIDs to admin group in *mso_uid.txt.* Logbooks can be added to any object init line with *0 = this execVM "support\modules\rmm_logbook\addAction.sqf"*

Close Air Support aircraft can be changed or added to the array in ..\support\modules\rmm_cas\main.sqf at line 3: *RMM_cas_types*. You can also tweak the time on station (default is 10 mins). Note that CAS is

controlled by the group leader of the group that calls it. Place a marker named CAS_spawn to your map for optional start position (f.e. over an airfield)

Flippable works on ATV and Motorcyles by default. Other vehicles can be added in the array ..\support\modules\crb_flippable\main.sqf at line 3: *_flipvehs*

CASEVAC defaults to a marker placed on the map and named *hospital*. The evac chopper will fly to the location of the player that made the call, wait for all wounded of the player faction who cannot stand and are within 100m of the LZ to be loaded, then fly back to the hospital marker. After all wounded are loaded or if no wounded are found nearby, the chopper will wait 30 secs before moving off. You will need a suitable radio to call for CASEVAC. If using ACE/ACRE this will require a manpack or long range radio.

Recruitment & Team status. Add *0 = this execVM "support\modules\rmm_recruitment\addAction.sqf"*; to the init line of a suitable object (e.g. HQ tent). The recruitment module detects the player faction and loads the appropriate preset. There are presets included for AAW, BAF, BIS US and Cz Forces. Copy & edit one of the dialog.hpp files to include other addons. You must be Lieutenant or above to recruit AI.

RMM_Cnstrct. An alternative (or addition) to the defence stores cluttering the base for building FOBs, the construction module is based loosely on the Warfare 3D placement module. Add the following to the init line of a suitable object or vehicle:

0 = this execVM "support\modules\rmm_cnstrct\addAction.sqf"; this setvariable ["cnstrct_supplies",100000];

Where *cnstrct_supplies* is the amount of supplies you have available. Items and their relative 'cost' can be added to the *cnstrct_items* array in ..\support\modules\rmm_cnstrct\fn_open.sqf at line 19

WHB_Multispawn provides multiple spawn locations at a deployable MHQ. Players have to 'sign in' to move their spawn point. The default vehicle is LAV25_HQ but others can be added to the array in ..\support\modules\WHB_FOBspawn\server\init_server_MHQs.sqf at line 7: P*V_hqArray = [];* You can also make any object into a 'Set Spawn Point' by putting in the following in the Init line: *this addAction ["Set Spawn", "support\modules\WHB_Multispawn\common\fn_addAction_SignInFOB.sqf", position player, 0, false, true, "", "true"];*

R3F_logistics provides a complete logistics module for lifting, carrying, towing and loading object in cargo. All objects are listed in the relevant arrays in ..\support\modules\R3F_logistics\R3F_LOG\config.sqf, which includes instructions. To exclude individual objects in the editor, add *this setVariable ["R3F_LOG_disabled", true]* to the init line.

ACE: If you are using ACE, you may wish to disable RMM_Revive, RMM_Tyres and R3F_Logistics in the support modules.hpp as these are all covered by alternative ACE versions. Also note there are several ACE settings set in init-custom.sqf, change as necessary.

Static Resupply can be set for any object in game (e.g. ammo crates) by adding *0 = [this, 14400] execVM "support\scripts\resupply.sqf"* to the init line, where the number is time in seconds before resupply takes place. Note that items will not resupply until they have been physically moved from their initial location. Suggested resupply times in seconds are:

- Backpacks - 60
- Ammo Crates - 14400 (4 hrs)
- Defence Stores - 28800 (8 hrs)

Vehicle Resupply (respawn) is handled by adding *0 = [this, 172800, 9999999] execVM "support\scripts\vehicle.sqf";* where the first number indicate time in seconds for a destroyed vehicle to respawn and the second number is time for an abandoned vehicle to respawn. There are more options available. See ..*support\scripts\vehicle.sqf* for details. Suggest resupply times in seconds are:

- Runabouts (ATVs) - 18000 (5 mins) if abandoned or destroyed
- Transport Vehicles - 43200 (12 hrs) if destroyed, none if abandoned (99999)
- Assault Vehciles - 86400 (24 Hrs) if destroyed, none if abandoned (99999)

TUP_Logistics can be used by players in game to get vehicles, crates, aircraft, support weapons and defence stores delivered to the battlefield. Only team leaders with a MAN PACK radio can call in a Logistics Demand. Any item ordered with TUP_Logistics will be stored to the database. There are options to deliver supplies via Paradrop, Airlift or Road Convoy. Mission editors can customize what is available in game by editing the *logistics.hpp* file in the support/modules/tup_logistics folder. The file contains the following areas that can be edited (that are functional):

```
////////////////////////////////////////////////////////////////////////
// Add items here if you wish to define the logistics player can order
// Uncomment the lines below and add items to the array
//tup_logistics_defence = ["Hedgehog","Hhedgehog_concrete"];
//tup_logistics_air = ["C130J"];
//tup_logistics_land = ["HMMWV_M2"];
//tup_logistics_crate = ["USBasicWeaponsBox"];
//tup_logistics_static = ["M2StaticMG"];

//////////////////////////////////////////
// Set maximum cost allowed per session (default in mission set to 50,000,000 - aircraft are around 10-20m each)
// tup_logistics_orderlimit = 50000000;

//////////////////////////////////////////////
// Replen Dem - a standard template order, that can be ordered at a single click
// Add items to the standard Replen Dem order
tup_logistics_replendem =
[[1,"USBasicWeaponsBox"],[1,"USLaunchersBox"],[1,"USBasicAmmunitionBox"]];
```

There are several useful scripts included in the MSO package. Full instructions are included in the sqf files themselves:

- CRB_HousePos.sqf - Move a patrol or unit to random house positions nearby.
- CRB_staticRearm.sqf - Rearm fixed enemy emplacements
- Guardpost.sqf - Simple guard post script
- BIN_taskDefend.sqf - Group will man all nearby static defenses and vehicle turrets and guard/ patrol the position and surrounding area

ACRE: If you are using ACRE, the init-custom.sqf script can be used to set various ACRE settings and also to automatically place ACRE Radio Boxes at ammunition markers if a player is using the Add-on. Also note that XEH_preinit.sqf will initialise ACRE variables. Finally, an ACRE Sync script accessible via the interation key allows players to synchronize ACRE if players JIP a considerable distance from others.

MSO Ambience

These are a collection of server safe ambient modules to add life to the map. Some may cause server fps to drop on lower end machines but they can be disabled in mission parameters.

Wildlife - Crows, flies and wild dogs (be warned, they bite). Shepherds carry weapons (don't mistake them for OPFOR). Guard dogs can be added to friendly groups by putting *0 = [this] spawn dogs_fnc_blitzy;* in the init line of the group leader.

Environment - Sandstorms, fly swarms, war torn and burnt out urban areas and the Islamic Call To Prayer can add a lot to the ambience on appropriate maps.

AEG - the ArmA Electrical Grid by Loyalguard can be enabled for maps that support it.

CRB_Civilians - Civi life continues, with busy markets, working bus routes, commuters going about their daily business and emergency services responding to casualties (CRB_Emergency).

Sea & Air Traffic - these modules look for ports and airports on the map, detect the dominant faction present (BLUFOR, OPFOR, Independent or Civilian) and spawn appropriate boats and aircraft. You can influence what spawns by placing empty vehicles of that faction at the airports. Note that anything blocking runways or taxiways will confuse the AI pilots, although they will avoid busy heli pads. You can set the intensity of these modules in the mission parameters. You can also define whether the traffic should be civilian only or include military units and the combat mode.

Mission Parameters

All in game mission parameter defaults can be edited via the params.hpp file in each module. We highly recommend that you use the recommended defaults. You may want to change the factions that are enabled (msofactionsdefault.hpp) or possibly some of the ambient settings.

PERFORMANCE: Due to the dynamic nature of MSO and its reliance on scripting to generate ambience and dynamic enemy, it can be very resource intensive. The default settings are the recommended ones for most maps (although it is recommended to use All Factions for ambient air because the civilian AN2 gets annoying fast). If you have a large map with lots of locations (e.g. Chernarus) or a large number of players you may want to consider disabling or reducing ambience or enemy. CLAfghan on the other hand is an example of a large map with few locations and therefore actually benefits from higher intensity settings for enemy.

Server Settings

Although it's perfectly possible to play on a locally hosted LAN game, MSO is designed and works best on a persistent dedicated server. Ensure you have *persistent = 1;* in your server.cfg

Credits

All the scripts included have appropriate credits included inside. No permission is needed to use MSO modules or scripts in your own missions.

Thanks to:
- MSO development team: Wolffy.au, Tupolov, highhead, friznit, WobblyHeadedBob, zorrobyte, HateDread, kieran, Ryan, Kolmain, and Rommel
- PersistentDB dev "team" - JMan, Scott W, Firefly
- Ryan, Antipop, Rommel, Militant, Scarecrow and Australian Armed Forces (AAF)
- Friznit, Tupolov, DaveP and Volunteer Commando Battalion (VCB)
- Krause, Beta, Sandiford, JollyResq - United Operations (UO)
- Kremator, SCAJolly, Katipo66, vengeance1, VRCRaptor, Dingbat, CebAlmighty, Jayc-astv, 16AA, VCB and AEF clans for testing
- Alef and Sickboy for CBA conversion for A2Free
- Xeno and Mikero for their Mission Builder Script
- Spyder for Object Network Update (ONU), LoyalGuard for ARMA Electrical Grids (AEG), ArmaIIholic and Highhead for World In Conflict (WICT), LurchiDerLurch for AC130 CAS
- Robalo, USMCWall, Muzzleflash, Wobbleyheadedbob, Hatedread, Zonekiller, VoW Widow, Swedge, Lonestar, SBSMac, Prymsuspec, zGuba, Fireball, Nou, Jaynus, Norrin for bits and pieces we got from you
- CWR² team, IceBreakr, Shezan, Commander, Old Bear for making your mods and islands MSO friendly
- Roy86 for Patrol Ops 2 Module
- Enigma for Weather Module
- PixelArt Studios for PXS SATCOMs