cf/x

Dynamic Mission Library
for DCS

DML

QUICK REFERENCE

ME MODULE INTEGRATION

Table Of Contents

1 ME FLAG INTEGRATION

1.1 Bang! Method (output)

DML understands the following methods, each defined by a keyword:

- ‘on’
  Sets the flag’s value to one, no matter what it was before

- ‘off’
  Sets the flag’s value to 0 (zero), no matter what the value was before.

- ‘inc’
  Increases the flag’s value by 1 (one). If, for example, the flag’s value was previously 10, that is increased to 11

- ‘dec’
  Decreases the flag’s value by 1 (one). If, for example, the flag’s value was previously 10, this number is decreased to 9.

- ‘flip’
  This is the most effecting method if you intend to trigger on flag change. It flips the flag’s value between 0 (zero) and “not 0”: If the flag’s value was anything except zero, the new value is zero. If the flag’s value was zero, the new value is 1 (one). This way you can flip-flop flags, turning them on and off repeatedly.

-

1.2 DML Flag Naming Rules

	ME	**DML**	**DML Zone-Local**
Format	Positive Number	Alphanumeric, must not contain comma ‘,’, **must not start with asterisk ‘*’ or digit (‘0’…’9’)**	**Starts with asterisk** ‘*’, alphanumeric, must not contain comma ‘,’
Examples	123 99 2898	A12 With blank F***d up Yup “quotes” too	*1 *A12 *fireCloner *ok multi **
Scope / Visibility	Entire DCS	DML modules, Lua Scripts	Only DML modules in same zone
Invisible to		ME Triggers	Everyone outside Zone

2 Artillery Zones

2.1 Summary

Simulates an artillery barrage inside the zone.

2.2 Dependencies

dcsCommon, cfxZones

2.3 ME Integration

Name	Description
artilleryTarget	Marks this zone as an artillery zone. Value is ignored **MANDATORY**
coalition	Used with Artillery UI – the coalition that can give a fire command (the explosions are completely coalition agnostic – they kill anyone). When the artillery zone is marked on the map, only this side will see it. Defaults to 0. Supports “red” and “blue” as values
spotRange	Used with Artillery UI – the maximum range at which an FO can give a fire command. Measured from center of zone. Defaults to 3000 meters
shellStrength	Average power of **each** exploding shell. Defaults to 500. 3000 is enough to level big buildings, so be conservative.
shellNum	Number of shells (salvo) per fire cycle. Defaults to 17 shells per cycle
transitionTime	The time (in seconds) the shells take on average to reach the target zone. Note that not all shells arrive at once but are usually spread over a couple of seconds. Defaults to 20
addMark	Add the artillery target zone to the F10 map of coalition (see above). Defaults to **true**.
shellVariance	Difference in shell’s explosion power, in percent. Defaults to 0.2 (20%)
f? artillery?	The ME flag to watch to trigger a fire cycle. Whenever the value of that flag changes, a fire cycle is initiated. If a cooldown attribute was specified, the current cooldown status is ignored and also won’t be reset. Defaults to nil (no flag watched)
cooldown	Used with Artillery UI: Number of seconds before the next fire cycle can be initiated. Is ignored when initiating fire via ME flags. Defaults to 120 (= 2 Minutes)
baseAccuracy	The radius (in meters) around the center of the zone in which the projectiles will land. Defaults to the ME zone’s radius (meaning all projectiles will land inside the zone if this attribute is missing and fire cycle is invoked via trigger flag)
silent	Used with Artillery UI: if true, suppresses communication responses from artillery

Supports DML Flags

2.4 Demos

- Artillery with UI
- Artillery zones triggered

- Pulsing Fun

3 Artillery UI

3.1 Summary

Provides forward observation features for helicopters, along with a UI to trigger artillery zones.

3.2 Dependencies

Tcb

3.3 ME Integration

Tbc

3.4 Demos

- Artillery with UI

4 Cargo Receiver

4.1 Summary

A zone designed to generate an event / signal on a flag when cargo is landed in the zone. Cargo must be registered with CargoManager

4.2 Dependencies

dcsCommon, cfxZones, cfxPlayer, cfxCargoManager.

4.3 ME Integration

cargoReceiver	Marks this zone as a cargo receiver zone. Value is ignored **MANDATORY**
autoRemove	Delete any object immediately after it was successfully delivered. This is helpful for most ObjectSpawnZones set-ups to trigger their spawn cycle
silent	Set to true to turn off this zone's directions. Defaults to false (zone will talk to pilots)
f=1	Sets the flag specified in Value to 1 when cargo is delivered **(Deprecated – use method and f!)**
f=0	Sets the flag specified in Value to 0 when cargo is delivered **(Deprecated – use method and f!)**
f+1	Increases the value of the flag specified in Value by 1 when cargo is delivered **(Deprecated – use method and f!)**
f-1	Decreases the value of the flag specified in Value by 1 when cargo is delivered **(Deprecated – use method and f!)**
method	Describes how flag! is to be set during a pulse: • on – set the flag to 1 (one) • off – set the flag to 0 (zero) • inc – increases the flag's value by 1 (one) • dec – decreases the flag's value by 1 (one) • flip – set the flag's value to 1 if it's currently 0 (zero), or to 0 (zero) in every other case. Defaults to "inc"
f! cargoReceived!	The flag to bang! when the object is destroyed. Use only one synonym per zone.

4.4 Demos

- Helo Cargo

5 Clone Zone

5.1 Summary

Allows you to clone any groups, at any time, anywhere.

5.2 Dependencies

dcsCommon, cfxMX, cfxZones

5.3 ME Integration

Name	Description
cloner	Marks this ME Zone as a clone zone. **Value of this attribute is ignored**, use it to describe this cloner's function. **MANDATORY**
source	The source for the clone template, must be the name of clone zone. When a clone cycle is initiated, the template is fetched from the source zone, and the units are then spawned around If this zone's center. If this attribute is present, this zone is not scanned for units to create a template from. If you supply **more than one template zone names**, separated by comma (e.g., "SAM 9 small, SAM 9 big"), each time a clone cycle is initiated, the clone zone picks one template by random. Defaults to <not present, zone scanned for units to create a template from>
turn	Degrees in which the clones are turned relative to the template's original position, relative to the zone's center. Defaults to 0 (zero)
moveRoute	If this attribute's value is true, all waypoints are move the same amount as the cloned units upon spawn. Only relevant if the zone is cloning another zone's template. When not present or false, all spawned units use the template's waypoints. Defaults to false
onStart	When set to false (default), the cloner will not spawn during start. Note that if this spawn zone is used to create a template, this results in an empty zone, as all units used for the template are destroyed during template creation. Defaults to false (**no spawn on start** of mission). **To spawn units at mission start, set this attribute to true.** **Deprecated. Use a raiseFlag module instead**
masterOwner	If not present, all cloned units retain the exact ownership of the units that the template was created from. If present, all cloned units are owned by the faction (red/blue/grey) that owns masterOwner. Note that using masterOwner, the coalition for the units that this cloner spawns can change during the mission.

Name	**Description**
	Defaults to <none, retain template ownership>
spawn? f? in? clone?	Flag to watch for a change. If the value of this flag changes, a new clone cycle is initiated Defaults to <none> This flag has multiple synonyms. Use only one per zone.
preWipe	If this attribute is true, any remaining units from the previous cloning cycle are removed from the game when the next clone cycle starts. Use this to ‘refresh’ groups like SAMs or Tanks that can run out of ammo. Default: false
empty+1	The value of this flag is increased by one when all units that were spawned during the last clone cycle are destroyed. This includes static objects. Defaults to <none> **Deprecated. Use empty! instead**
empty!	The value of this flag is changed according to method when all units from the last spawn have been destroyed, including all static objects. **Note:** use this in favor of empty+1 **Note:** use either empty+1 or empty! but **not both**.
method	Describes how empty! is to be set during a pulse: • on – set the flag to 1 (one) • off – set the flag to 0 (zero) • inc – increases the flag’s value by 1 (one) • dec – decreases the flag’s value by 1 (one) • flip – set the flag’s value to 1 if it’s currently 0 (zero), or to 0 (zero) in every other case. Defaults to “inc”
deSpawn? deClone?	Flag to watch for a change. If the value of this flag changes, the remaining units / static objects from the previous spawn are removed. Note that if you trigger deSpawn?, empty+1 will **not** trigger subsequently. Defaults to <none>
trackWith:	Name of a groupTracker zone. All spawned groups are added to this groupTracker to track. If you have stacked the tracker on the same zone, you can use a single asterisk ‘*’ as zone name

Supports DML flags

5.4 Demos

- Attack of the CloneZ
- Once, twice, three times a maybe
- Clone Relations
- Frog Men Training
- Flag Fun
- Track This!

6 Count Down

6.1 Summary

A module that counts down flag signals (not time!) and can do various tricks with these signals.

6.2 Dependencies

dcsCommon, cfxZones

6.3 ME Integration

Name	Description
countDown	Marks this as a count down. The value of this attribute defines the number times until the count reaches zero This value supports ranges: if you specify a range (e.g., “3-5”) each time the count down is initialized (at start, and when looping), a random number in the range (including upper and lower limit) is chosen. Defaults to 1 (one) **MANDATORY**
loop	If this attribute is true, a count down restarts after reaching zero. If the count down is given as a range, an new random start value is taken from that range (including upper and lower limit)
method	Describes how flag! is to be set during a pulse: • on – set the flag to 1 (one) • off – set the flag to 0 (zero) • inc – increases the flag’s value by 1 (one) • dec – decreases the flag’s value by 1 (one) • flip – set the flag’s value to 1 if it’s currently 0 (zero), or to 0 (zero) in every other case. Defaults to “flip”
count? in?	Flag to watch. A change of value on this flag constitutes a signal to count down one Note: ‘count?’ and ‘in?’ are both names for the same flag. Use either, but not both. Defaults to <none>
zero! out!	Output flag to bang when countdown reaches zero. Note: ‘zero!’ and ‘out!’ are both names for the same flag. Use either, but not both. Defaults to <none>
tMinus!	Output flag to bang when the value is lowered, and has not yet reached zero. Defaults to <none>
counterOut!	A flag, when given, is set to the current value of the count. The flag is updated each time that the counter is triggered via count?

6.4 Demos

- Once, twice, three times a maybe
- The Zonal Countdown

7 CSAR Manager

7.1 Summary

CSAR Manager is a stand-alone module that adds CSAR (Combat Search And Rescue) operations to your missions. It consists of multiple ME parts, and in-game UI.

7.2 Dependencies

CSAR Manager requires dcsCommon, cfxZones, cfxPlayer, nameStats, cargoSuper

Optional: cfxCommander (when using CSAR Zones)

7.3 ME Integration

ME Integration consists of two parts: placing CSARBASES, which are zones in which a successful landing will complete the CSAR mission successfully (and return a lost pilot when using limited airframes), and placing CSAR Zones that essentially start CSAR Missions by placing downed pilots

Name	Description
CSARBASE	Must be present to identify this zone as CSAR Base where CSAR Missions can end. A helicopter must land inside this zone. Supports linked zones (for example if the BSAR Base is a ship). Each side that has CSAR Missions must have at least one such zone, or CSAR Missions can not be completed. There is no upper limit on the number of CSAR Bases a side can have. The value of this attribute can be used to name the CSAR Base, else the Zone's name is used. **MANDATORY**
coalition	The side that owns the CSAR Base. If neutral, both sides can use this as a base, else only the faction specified. Defaults to "neutral". Other possible values are "red" and "blue"
name	Optional name for CSARBASE.

Name	Description
CSAR	Identifies this as CSAR Zone that is converted into a CSAR mission upon mission start or when the startCSAR? flag is changed. **MANDATORY**
coalition	Faction (red/blue) for which this mission is generated
name csarName pilotName victimName	Name of this mission, recommended is to use a personal name, e.g. "Lt. Wesley Crasher"
freq	Frequency for the ELT (radio to home in on) in KHz. Random if not set
timeLimit	(currently not used)

weight	Weight of pilot (tbc)
deferred	If true, CSAR missions are only created when the startCSAR? flag changes. Default is false (a CSAR mission is automatically created when the main mission starts up)
in? startCSAR?	When the value of this flag changes, a new CSAR is created according to this Zone's attributes. Defaults to <none>

7.4 Demos

- CSAR of Georgia

8 Delay Flags (tbc)

8.1 Summary

Change a flag on the output side a while after a change was detected on the input side

8.2 Dependencies

dcsCommon, cfxZones

8.3 ME Integration

Name	Description
timeDelay	Marks this as a delayFlag module. The value of this attribute defines the number of seconds to wait after activation before the output flag is set. Value can be a range in which case delayFlag picks a random number inside the range (including bounds). Defaults to 1 second **MANDATORY**
out! delayDone!	The flag to bang! after the delay has passed. Use only one synonym per Zone
method	Describes how flag! is to be set during a pulse: • on – set the flag to 1 (one) • off – set the flag to 0 (zero) • inc – increases the flag's value by 1 (one) • dec – decreases the flag's value by 1 (one) • flip – set the flag's value to 1 if it's currently 0 (zero), or to 0 (zero) in every other case. Defaults to "flip"
f? in? startDelay?	The flag to watch for a change that starts the delay. Use any synonym, but only one per zone.
stopDelay?	Stops a running delay. Has no effect on a stopped delay

Supports DML flags

8.4 Demos

- Attack of the CloneZ
- Bottled Messages
- Clone Relations
- Flag Fun
- Track This!

9 FARP Zones

9.1 Summary

A zone linked to a FARP (and thus conquerable) that automatically provides service vehicles.

9.2 Dependencies

dcsCommon, cfxZones

9.3 ME Integration

Name	Description
FARP	Indicates that this zone is a FARP zone. Value is ignored. **MANDATORY**
rPhiHDef	Radius (in m), Phi (degrees) and Heading (degrees) of the center point around which the **defenders** deploy. Defaults to 0, 0, 0
rPhiHRes	Radius (in m), Phi (degrees) and Heading (degrees) of the center point around which the **resource** vehicles deploy as a line. Defaults to 0, 0, 0
redDefenders	typeStrings of defender vehicles. Example “ZSU-23-4 Shilka, ZSU-23-4 Shilka”. Defaults to “none” Special encoding: “none” – no vehicles
blueDefenders	typeStrings of defender vehicles. Example “Roland ADS, Roland Radar, Roland ADS”. Defaults to “none” Special encoding: “none” – no vehicles
formation	Formation of the defenders group. See dcsCommon for supported group formations. Defaults to ‘circle_out’.
rFormation	Radius of the circle that the defenders assemble in. Defaults to 100m
hidden	Set to “no” if FARP is visible on the F10 map (and colored according to owner). Defaults to “no”
hideRed hideBlue hideGrey	For any of these three attributes, the FARP is hidden if it belongs to that faction. For example, if hideRed is set to true, the FARP is shown on the map while it belongs to neutral or blue, but disappears when it is owned by red.

9.4 Demos

- FARP and away

10 Group Tracker

10.1 Summary

A module that counts the groups in a set, and changes flags when the number changes. Extremely versatile count based on the tracked groups’ survival.

10.2 Dependencies

dcsCommon, cfxZones

10.3 ME Integration

To add all groups that have at least one unit inside the zone to a groupTracker, add the following attribute to the zone:

Name	Description
addToTracker:	Add all groups that have at least one unit in this zone to the tracker whose zone name is given in the Value field. This happens only at mission start-up, and therefore only work for non-player-controlled planes (since player-controlled planes do not exist at mission start-up). If your player group contains AI planes, place one of those into the zone, and that group can be added to a tracker. If the same zone also hosts the groupTracker, you can use a single asterisk “*” as wildcard for the groupTracker’s zone name

To add a groupTracker to a zone

Name	Description
tracker	Marks this zone as a groupTracker. It can be referenced by the zone’s name passed in the trackWith: and addToTracker: attributes. When referenced locally, a single asterisk “*” can be used as wildcard name for easy copy/paste of the entire stack **MANDATORY**
addGroup!	Whenever a group is added to the tracker, the value of this flag is increased. If not changed by other modules, this flag also doubles as a running total of all groups added to the tracker Defaults to <none>
removeGroup!	Whenever a tracked group is destroyed, the value of this flag is increased. If not changed by other modules, this flag also doubles as a running total of all watched groups that have been destroyed while they were tracked Defaults to <none>
numGroups!	The value of this flag always represents the number of groups currently watched by this tracker. Its value is updated 1/ups times per second. Defaults to <none>

10.4 Demos

- Track This!

11 Guardian Angel

11.1 Summary

Provides out-of-the-box protection for aircraft against guided missiles.

11.2 Dependencies

Tbc

11.3 ME Integration

Tbc

11.4 Demos

- Missile Evasion
- Guardian Angel

12 Helo Troops

12.1 Summary

This module provides instant out-of-the box ability for troop transport helicopters to pick up and deploy infantry

12.2 Dependencies

Tbc

12.3 ME Integration

Tbc

12.4 Demos

- Helo Trooper

13 Map Markers

13.1 Summary

Allows you to place markers on the F10 map.

13.2 Dependencies

dcsCommon, cfxZones

13.3 ME Integration

Name	Description
mapMarker	Turns on the map marking feature. Simply must be present. Content of this property is displayed as text on the Map. Example “Destroy all vehicles in this area” **MANDATORY**
coalition	Side that sees this marker. Can be “red”, “blue”, “neutral”, or “all”. You can also substitute “1” for red, and “2” for blue. Defaults to “all”

13.4 Demos

14 Messenger

14.1 Summary

A module that generates a text and/or audio output.

14.2 Dependencies

dcsCommon, cfxZones

14.3 ME Integration

Name	Description
messenger	Attaches the messenger module to this zone. The value of this attribute is ignored. **MANDATORY**
f? in? messageOut?	Watches this flag for a change. **Each time the flag value changes,** the message is displayed, and sound is played You can use any synonym, but only one per zone
message	The text of the message to be displayed **TEXT WILDCARDS** *name is replaced by zone's name *value is replaced by the value currently held by the flag specified in messageValue?
messageValue?	A flag name that is read whenever a message is put out. The value that this flag contains can be accessed with the *value wildcard in the message attribute
clearScreen	If true, erase all existing messages. Defaults to false
spaceBefore	If true, an empty line is issued before the text message. Default false
spaceAfter	If true, an empty line is issued after the text message. Default is false
soundFile	Name of the sound file (including extension like '.wav') that is to be played. Defaults to '<none>. Note that the sound file's name must be specified relative to the mission's default location for sound files (l10n/DEFAULT/). If you use ME to import the sound files, you do not have to specify the location. Remember to import the sound file into the mission else no sound will play.
coalition msgCoalition	The coalition that the message/sound is played to. If no coalition is given, text and sound are played to all. Legal values are "red", "blue", "neutral", 0, 1, 2 Default is <none> (play to all).
messageOn?	When the value of this flag changes, the messenger is turned on. If it already was on, nothing happens All messengers start in 'On' state and require at least one signal on messageOff to disable it. Defaults to <none>

messageOff?	When the value of this flag changes, the messenger is turned off. Any further messages are suppressed. If the messenger already was turned off, nothing happens. Defaults to <none>

Supports DML Flags

14.4 Demos

- Bottled Messages
- Frog Men Training
- The Zonal Countdown
- Follow Me!
- CSAR of Georgia
- Track This!

15 NDB

15.1 Summary

Allows you to place an NDB in any zone. This includes linked zones so you can place NDBs that follow ships.

15.2 Dependencies

dcsCommon, cfxZones.

15.3 ME Integration

Name	**Description**
NDB	Creates an NDB at the zone's center. If the zone is linked to a unit, this NDB will automatically update to the unit's location. **The value of this attribute is the frequency (in MHz) at which the NDB transmits (e.g. 121.5 for 121.5 MHz, 0.42 for 420 kHz)** **MANDATORY**
fm	If true, the transmission is in FM, else in AM Defaults to false (AM)
ndbSound	Name of the sound file with extension that is to be transmitted. Defaults to '<none>. Note that the sound file's name must be specified relative to the missions default location for sound files (l10n/DEFAULT/). If you use ME to import the sound files, you do not have to specify the location. Remember to import the sound file into the mission else no sound will play.
watts	Transmission power (in watts) for the NDB. 100 Watts usually has a range of some 150 km. Defaults to 100 Watts
paused	If set to true, on mission start the NDB will not start up. Use the "on?" watch flag attribute or API to turn it on. Defaults to false
on?	Watches the flag <Value Number> for a change. **Each time the flag value changes, the NDB is started (will also cause the transmission sound to rewind).** The current paused value is ignored, and then set to false after the NDB has started. Defaults to no flag to watch
off?	Watches the flag <Value Number> for a change. **Each time the flag value changes, the NDB is stopped.** paused value is set to true after the NDB has stopped. Defaults to no flag to watch

15.4 Demos

- ADF and NDB Fun

16 Object Destruct Detector

16.1 Summary

Generates a signal (flag change) when the map object that is referenced by the zone is destroyed.

16.2 Dependencies

dcsCommon, cfxZones

16.3 ME Integration

Name	Description
OBJECT ID	THIS ATTRIBUTE IS FILLED BY ME AND MUST NOT BE CHANGED **MANDATORY**
NAME	THIS ATTRIBUTE IS FILLED BY ME AND MUST NOT BE CHANGED **MANDATORY**
f=1	Sets the flag specified in Value to 1 when object is destroyed **(deprecated – use method and f!)**
f=0	Sets the flag specified in Value to 0 when object is destroyed **(deprecated – use method and f!)**
f+1	Increases the value of the flag specified in Value by 1 when object is destroyed **(deprecated – use method and f!)**
f-1	Decreases the value of the flag specified in Value by 1 when object is destroyed **(deprecated – use method and f!)**
method	Describes how flag! is to be set during a pulse: • on – set the flag to 1 (one) • off – set the flag to 0 (zero) • inc – increases the flag's value by 1 (one) • dec – decreases the flag's value by 1 (one) • flip – set the flag's value to 1 if it's currently 0 (zero), or to 0 (zero) in every other case. Defaults to "flip"
f! destroyed!	The flag to bang! when the object is destroyed. Use only one synonym per zone.

16.4 Demos

- Object Destruct Detection

17 Object Spawn Zones

17.1 Summary

Allows static objects and cargo to spawn according to an attribute (type)

17.2 Dependencies

dcsCommon, cfxZones, (cfxCargoManager)

17.3 ME Integration

Name	Description
objectSpawner	Marks this ME Zone as a spawn zone. **Value of this attribute is ignored**, use it to describe what it spawns to make mission editor easier for you **MANDATORY**
f? spawn? spawnObjects?	An ME-compatible flag (e.g. 100) that this object spawner monitors for change. Whenever the value of the monitored flag changes, a new set of objects is spawned immediately, ignoring all maxSpawn and cooldown rules.
pause?	Flag to observe. Each time the flag's value changes, the spawner's 'paused' setting is forced to 'true. Used to 'pause a spawner
activate?	Flag to observe. Each time the flag's value changes, the spawner's 'paused' setting is forced to 'false'. Used to 'activate' a paused spawne
types	Type string array for the STATIC OBJECTS that are spawned. Example "White_Tyre, Red_Flag". These objects may look like units (if you use the type string for a ground unit or aircraft), but they are static. **WARNING**: Blanks are part of the type, and blanks directly before and after the last character are automatically stripped. All static objects given here are stacked on top of each other, and count as one instance (the example creates a tire with a red flag in the middle) **MANDATORY**
count	The number of times that the combined object in types is to be repeated. If count equals one (or is omitted), the objects defined in types are assembled in the center of the zone. Otherwise, the objects are distributed over the zone's circumference count times. Defaults to one
country	The country for which the static objects are spawned. Examples: 0 = Russia, 1 = Ukraine, 2 = USA etc. Defaults to 2 (USA)
baseName	Used to create the names that uniquely identify the objects that are spawned to DCS. **If provided, MUST BE UNIQUE** for each spawner. If you do not provide a baseName, a unique name is generated for you.
cooldown	Number of seconds after the last spawn was removed before new objects are spawned. Default is 60 seconds
autoRemove	Wait for the spawned objects to be removed or destroyed, immediately start cooldown, then re-spawn according to rules. Default is false

Name	Description
autoLink	Only used when the spawner is linked to a unit: should the spawned objects move with the unit that the zone is linked to (usually ships, but can also be other objects). Defaults to true. Set to false if the spawner should ‘drop’ the objects to the ground.
heading	Orientation of the objects when they are spawned. Default is 0 (North)
weight	Used with cargo objects: the weight of this object in kg. Defaults to zero.
isCargo	Are these objects to be picked up by helicopters? Defaults to false.
managed	Used only if the objects spawned are cargo. If true, cargo objects are automatically registered with cfxCargoManager when they are spawned and cfxCargoManager is loaded). Defaults to true
maxSpawns	Number of times that the spawner spawns the objects. Defaults to 1 (one)
paused	A paused spawner will not spawn automatically (but can be forced to spawn via API or query flag f?). Set to true to pause spawning. Defaults to false.
requestable	This spawner should only spawn on request (i.e. via API or from other zones). Forces paused to true. Default value is false

17.4 Demos

- ME Triggered Spawns
- Spawn Zones (training and lasing)
- Random Glory
- Helo Cargo

18 Owned Zone

18.1 Summary

Zones that can be conquered and generate a signal when conquered

18.2 Dependencies

dcsCommon, cfxZones, (cfxGroundTroops)

18.3 ME Integration

Name	**Description**
owner	Coalition that owns the zone at beginning of Mission. Can be 0, 1, 2 or “red”, “blue”, “neutral”. If nothing or some illegal value give, this defaults to neutral (0) **MANDATORY**
conq+1	Increase this flag’s value by 1 (one) each time it is conquered. Defaults to <none>
defendersRED	A string, coma separated, that specifies the types of troops to spawn when the zone is owned by RED. Example: "Soldier M4,Soldier M4” places two Infantry soldiers. **Warning:** these types need to *exactly* match DCS’s types. Be sure not to accidentally insert blanks. Special types: “none” – no troops Defaults to “none”
defendersBLUE	A string, coma separated, that specifies the types of troops to spawn when the zone is owned by RED. Example: "Soldier M4,Soldier M4” places two Infantry soldiers. **Warning:** these types need to *exactly* match DCS’s types. Be sure not to accidentally insert blanks. Special types: “none” – no troops Defaults to “none”
attackersRED	A string, coma separated, that specifies the types of troops to spawn when the zone is owned by RED. Example: "Soldier M4,Soldier M4” places two Infantry soldiers. **Warning:** these types need to *exactly* match DCS’s types. Be sure not to accidentally insert blanks. Special types: “none” – no troops Defaults to “none”
attackersBLUE	A string, coma separated, that specifies the types of troops to spawn when the zone is owned by RED. Example: "Soldier M4,Soldier M4” places two Infantry soldiers. **Warning:** these types need to *exactly* match DCS’s types. Be sure not to accidentally insert blanks. Special types: “none” – no troops Defaults to “none”
formation	Formation of the defenders group. See dcsCommon for supported group formations. Defaults to ‘circle_out’.
attackFormation	Formation of the attackers group. See dcsCommon for supported group formations. Defaults to ‘circle_out’.
spawnRadius	Radius of circle that the defenders are placed on. Defaults to slightly less than zone radius, so defenders are always inside the zone they are defending. Defaults to 0.

Name	Description
attackRadius	Radius of circle in which the attackers spawn after they are produced. Defaults to zone radius
attackDelta	Distance from center of zone in which attackers spawn circle is located. Defaults to 10.
attackPhi	Angle (direction) in degrees from zone center where attackers are spawning. Defaults to 0.
paused	Pauses zone. “true” or “yes” means that the zone is paused. A paused zone produces no attackers nor defenders, but will detect capture normally. Capturing a paused zone will currently not unpause the zone. Do that in the capture callback. Defaults to “no”
unbeatable	“true” or “yes” makes it unbeatable. Zone can’t be conquered by other side. Defaults to “no”
untargetable	“true” or “yes” makes it untargetable. Zone will not be targeted by troops with ‘attackOwnedZones’. Defaults to “no”
hidden	“true” or “yes” hides it. Zone is not shown on F10 Map. Defaults to “no”

18.4 Demos

- Owned Zones ME Integration

19 Player Score

19.1 Summary

A module to keep award and tabulate kill score for players

19.2 Dependencies

Tbc

19.3 ME Integration

Tbc

19.4 Demos

- Keeping The score
- More Score

20 Pulse Flags

20.1 Summary

Flag “Heartbeat” – (somewhat) regularly sets/changes a flag

20.2 Dependencies

dcsCommon, cfxZones

20.3 ME Integration

Name	Description
pulse **pulse!**	Marks this as a pulser. The value of this describes the flag (number) to change on each pulse. The flag is changed according to the method attribute **MANDATORY**
~~**flag!**~~	~~The flag (number) to change on each pulse. The flag is changed according to the method attribute~~ ~~**MANDATORY**~~
method	Describes how flag! is to be set during a pulse: • on – set the flag to 1 (one) • off – set the flag to 0 (zero) • inc – increases the flag’s value by 1 (one) • dec – decreases the flag’s value by 1 (one) • flip – set the flag’s value to 1 if it’s currently 0 (zero), or to 0 (zero) in every other case. Defaults to “flip”
~~done+1~~ done! pulsesDone!	This flag’s value is incremented by 1 (one) when the pulser completes a fully run of pulses. Can only happen when the *pulses* attribute supplies a positive number. Defaults to <none>
activate? startPulse?	When this flag’s value changes, a paused pulser is reset and then restarted. Defaults to <none>
pause? pausePulse?	When this flag’s value changes, a pulser is paused. Defaults to <none>
paused pulseStopped	When true, the pulser does not start at the beginning of a mission but waits for a change on activate?-flag. Defaults to false
pulses	The number of pulses to complete. If set to -1, the pulser runs until the mission ends or the pause?-flag changes
time	Seconds between pulses. You can supply a range (two numbers separated by a hyphen, e.g. “4-19”), the time between pulses is randomized after each pulse to a number in that range. Defaults to 1
zeroPulse	Usually, a pulser starts with an initial pulse (“pulse zero”). This initial pulse can be delayed by *time* by setting zeroPulse to false The effect is that the initial pulse happens after the first delay Default is true (initial pulse immediately)

20.4 Demos

- Pulsing Fun
- Frog Men Training
- The Zonal Countdown

21 Raise Flag

21.1 Summary

A simple, DML way to set flags to values. Supports randomization and delayed setting of flags.

21.2 Dependencies

dcsCommon, cfxZones

21.3 ME Integration

Name	Description
raiseFlag **raiseFlag!**	Marks this as a flag raiser. The value of this attribute is the flag that is to be raised. Use only one synonym per zone **MANDATORY**
value	Value (number) to set the flag to. Default is 1
afterTime	Amount of time (in seconds) after mission start to set the flag. Can be a range. If a range is given, the time is a random number from this range. **If this attribute is omitted, the flag is set 0.5 seconds after the mission starts.**
stopFlag?	Only useful in conjunction with afterTime. A flag to watch for a change. When the flag changes and raiseFlag is still waiting for afterTime, raiseFlag is ‘disarmed’ and will not set the flag later.

Supports DML Flags

21.4 Demos

- Flag Fun

22 Recon Mode

22.1 Summary

Provides out-of-the-box advanced recon abilities for aircraft.

22.2 Dependencies

Tbc

22.3 ME Integration

Tbc

22.4 Demos

- Recon Mode

23 rnd Flags

23.1 Summary

Can randomly set flags – from a pool of flags, in many different methods

23.2 Dependencies

dcsCommon, cfxZones

23.3 ME Integration

Name	Description
RND	Marks this as a randomizer. The value of this attribute is ignored **MANDATORY**
flags!	Set of flags, as a comma (‘,’) separated list of the flag names that can be chosen from. The flag names can appear in any sequence. Supports ranges like “2-7” (ME numbers only). Flag names can be included multiple times, including the same flag name multiple times simply increases the likelihood that this number is chosen. Examples: “2, 4, A, A, F, 6” “A9, 3-18, C33, endGame, 11-11” **MANDATORY**
method rndMethod	Describes how flags are to be set when chosen during a cycle. rndFlags currently supports the following methods: • on – set the flag to 1 (one) • off – set the flag to 0 (zero) • inc – increases the flag’s value by 1 (one) • dec – decreases the flag’s value by 1 (one) • flip – set the flag’s value to 1 if it’s currently 0 (zero), or to 0 (zero) in every other case. Defaults to “on”
pollSize	Number of items to choose from the set of flags during a cycle. Defaults to 1. Can be a range: two numbers separated by a hyphen, e.g. “2-5”. When a range is given, pollSize is randomized each cycle to a number between the lower and upper bounds, inclusive.
remove	When set to true, the flags that were chosen during a cycle are removed from the set of flags. Defaults to false.
reshuffle	When set, the original full set of flags is restored when all flags have been removed
f? in? rndPoll?	The flag to watch to trigger (fire) a randomize cycle. Each time that this flag changes in value, a new randomize cycle is initiated. Defaults to <none set> You can use any synonym, but only one per Zone
onStart	If true, a cycle is run for this randomizer 0.25 seconds after the mission starts. Defaults to false **NOTE**:

Name	Description
	if no f? (or synonym) is specified, and onStart is false, onStart is automatically set to true so the randomizer runs once
done+1	The flag number to increase when the randomizer has run out of flags to change, and reshuffle is false (randomizer did nothing) Is increased every time that the randomizer runs a cycle on an empty flag set

Supports DML Flags

23.4 Demos

- Random Glory
- Random Death
- Pulsing Fun
- Attack of the CloneZ

24 Smoke Zones

24.1 Summary

Places a colored permanently refreshing smoke at the center of the zone

24.2 Dependencies

dcsCommon, cfxZones

24.3 ME Integration

Name	Description
smoke	Adds a permanent smoke affect to the center of the zone. Possible values for the smoke effect are: • “green” or “0” • “red” or “1” • “white” or “2” • “orange” or “3” • “blue” or “4” • “random”, “?” or “rnd” (random color from above) **MANDATORY**
paused	When true, will not start smoke at mission start, but wait for a signal on the f? flag. Defaults to false (smoke starts at mission beginning). Note that if you set paused to true and omit the f? attribute, you can only start this smoke via API
f? startSmoke?	Flag to watch for a change. When the value of that flag changes, pause is set to false, and smoke starts. Defaults to <none>
altitude	Altitude (in meters) above ground that the smoke should be created. Defaults to 1m

Supports DML Flags

24.4 Demos

- Smoke’em DML Intro
- Random Glory
- Once, twice, three times a maybe

25 Spawn Zones

25.1 Summary

Allows spawning of ground units based on the types attribute (text)

25.2 Dependencies

dcsCommon, cfxZones, cfxCommander, cfxGroundTroops, (Helo Troops)

25.3 ME Integration

Name	Description
spawner	Marks this ME Zone as a spawn zone. **Value of this attribute is ignored**, use it to describe what it spawns to make mission editor easier for you **MANDATORY**
f? spawn? spawnUnits?	Flag (ME-compatible) to observe. Each time the value of that flag changes, a new spawn is forced, ignoring all other settings like maxSpawn, cooldown, paused, etc. Defaults to no flag to observe Use only one synonym per zone
pause?	Flag to observe. Each time the flag's value changes, the spawner's 'paused' setting is forced to 'true. Used to 'pause a spawner
activate?	Flag to observe. Each time the flag's value changes, the spawner's 'paused' setting is forced to 'false'. Used to 'activate' a paused spawne
types	Type string array for the ground units that are spawned. Example "Roland ADS, Roland Radar, Roland ADS" or "Soldier M4" – **WARNING**: Blanks are part of the type, and blanks before and after the last character are automatically stripped. For a full reference of objects and their types, see here https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB and use whatever is given as value for the "typeName" attribute, e.g. "Soldier M249" for the "INF Soldier M249.lua"
country	The country (a number) the units that spawn belong to, e.g. "22" for Switzerland (**Warning**: unlike many other zone extensions, we use a County, not a Coalition here. The coalition is determined by which Faction the country belongs to as is defined when you create the mission, or by using the faction editor. Common Countries are Russia = 0, Ukraine = 1, USA = 2, UN Peace Keepers = 82 You can find a reference of all country codes here: https://wiki.hoggitworld.com/view/DCS_enum_country).
masterOwner	A string that references another ME Zone by name. It must match that Zone's name exactly, and that zone must have an owner (e.g. defined as an cfxOwnedZone or FARPZone). A spawner only spawns automatically when the masterOwner's owning faction is the same as the spawner's country affiliation. On the map, the spawner does not have to be inside the masterOwner's zone, it can be hundreds of miles away. You can use this to start spawning reinforcements in a

Name	Description
	completely unrelated part of the map when units conquer the masterOwner zone. If no masterOwner is specified, the Spawner spawns as directed and disregards any surrounding zones that happen to be owned Optional, defaults to empty
baseName	A name (e.g. "Hill Marines") that is used to create units and groups from during unit spawning. **If provided, baseName MUST BE UNIQUE. If you do not assign a base name, a unique one will be generated for you**. If two spawners have the same baseName, one of them will not spawn, so if for some reason a spawner does not spawn, make it a habit to check this first.
cooldown	Time interval (in seconds) from when a new group can be produced (removed from the spawner) to the moment it is produced. Defaults to 60
autoRemove	Usually, a spawner retains ownership of a group that is produced, and will re-start the spawning cycle only after it was removed. If you add the autoRemove attribute with a "yes" or "true" value, the Spawner will automatically re-start the spawning cycle (cooldown, produce) as soon as the new group has spawned. You can use this to automatically give orders and have units move out after they have spawned (similar to how OwnedZones spawn attackers). Be advised that you can create a lot of vehicles on your map in a very short time, so be careful when using autoRemove. Defaults to 'false'
heading	The direction the spawned group it oriented to, from the center of the spawn zone. Defaults to 0
formation	Formation of the spawned group. See dcsCommon for supported group formations. Defaults to 'circle_out'.
paused	When paused, a spawner only spawns when other scripts tell it to (e.g. your own scripts, cfxHeloTroops, triggers). Defaults to "no"
orders	This is an optional interface to other troop-governing modules, e.g. cfxGroundTroops. Default is "guard", and spawners support in addition to those that cfxGroundTroops support (see → **Error! Reference source not found.**)
range	An attribute used to pass a range value to orders (e.g. JTAC laze range, detection/engage range)
target	An attribute used to pass a target zone when used in conjunction with the 'attackZone' orders
maxSpawns	The maximum number of times that this spawner spawns groups. Set it to a positive number (e.g. 3) to spawn that many time. Set it to a negative number for an unlimited number of spawns (default is -1). Set it to zero (0) and the spawner will never spawn.
requestable	Interfaces with other scripts, if you set this value to true, troops will only spawn on request via `cfxSpawnZones.spawnWithSpawner()`. See the API section on how to get a list of eligible spawners. Automatically interfaces with HeloTroops and other enhancements

25.4 Demos

- Random Death
- Moving Spawners
- Helo Troops

26 Unit Zone

26.1 Summary

Tests if a unit/group is inside/outside a zone and can change flags when the status changes.

26.2 Dependencies

dcsCommon, cfxZones

26.3 ME Integration

Name	Description
unitZone	Marks this ME Zone as an anchor for unitZone **MANDATORY**
lookFor	Name for the group or (player) unit to check zone status. If the last character in the name is an asterisk "*", exact matches are and all group/unit names that start with the string (minus asterisk) are accepted, e.g. if you supply "Hel*" all of the following would be accepted: • Hel • Hello World • Helo Rescue-1 • Hellfire You can use this feature to your advantage in conjunction with cloners or spawners, as these all produce groups with a known base name. If you only supplied "Hel",only (without the asterisk "*") only the group whose name exactly matches "Hel" is checked.
matching	What type of units to match. Currently supported are • group (default): look for group names • player – look only at player units and match their unit's (not group's) name against lookFor Default: group
coalition uzCoalition	The coalition (red/blue) units to check. Accepts 0, 1, 2, red, blue, neutral. Note that 0 (zero) and 'neutral' means 'both' in this case, neutral groups/units are never checked against the zone Defaults to 0 (both)
filterFor	Which categories to look for. If no attribute is given, **all** categories are checked against the zone (when their name pattern matches). When you supply a filterFor attribute, only that category is considered. Currently supported are • 0 (zero) or "aircraft" or "air" • 1 or "helo" or "heli2 or "helicopter" • 2 or "ground" • 3 or "ship"

Name	Description
	- 4 or “train” Defaults to no filtering
enterZone!	Change this flag when the first unit (player) or part of all groups that match the criteria enters the zone
exitZone!	Change this flag when the last unit (player) of all groups that match the criteria have exited the zone (being destroyed counts as leaving)
changeZone!	Changes this flag whenever enterZone! or exitZone! are triggered

26.4 Demos

- Follow Me!

27 Module Name

27.1 Summary

27.2 Dependencies

27.3 ME Integration

27.4 Demos