

cf/x  
Dynamic Mission Library  
for DCS

# DML

## QUICK REFERENCE

### ME MODULE INTEGRATION

Copyright © 2022 by cf/x AG and Christian Franz

# Table Of Contents

1	ME FLAG INTEGRATION .....	3
2	Artillery Zones .....	7
3	Artillery UI .....	9
4	Cargo Receiver .....	10
5	Changer .....	11
6	CivAir .....	13
7	Clone Zone .....	15
8	Count Down .....	18
9	CSAR Manager .....	20
10	Delay Flags .....	22
11	FARP Zones .....	23
12	Group Tracker .....	24
13	Guardian Angel .....	26
14	Helo Troops .....	27
15	Map Markers .....	28
16	Messenger .....	29
17	NDB .....	31
18	Object Destruct Detector .....	33
19	Object Spawn Zones .....	34
20	Owned Zone .....	36
21	Player Score .....	38
22	Pulse Flags .....	39
23	Radio Trigger .....	41
24	Raise Flag .....	42
25	Recon Mode .....	44
26	rnd Flags .....	46
27	Smoke Zones .....	48
28	Spawn Zones .....	49
29	Unit Zone .....	52
30	Wiper .....	54
31	xFlags (Flag Testing) .....	56
32	Module Name .....	58

# 1 ME FLAG INTEGRATION

## 1.1 Understanding DML Flags

A DML module has two independent methods to communicate:

- The way it *sends* signals (the **output method**, usually designated by the exclamation point "!" at the end of the attribute's name). It describes what is sent when the module wants to convey information. This can be 'inc' (the most common, increases the flag's value), or 'On', 'Off', 'flip', a fixed value etc. **A module sets an output flag whenever it sees fit**, usually as a response to something happening in the mission (a group enters a zone, an object is destroyed etc)
- The way a module *receives* signals: how it looks at their inputs (flag attributes with question mark "?" in their name) and tries to decide if it should trigger. Most common is the 'change' method: the module triggers when the flag's current value changes.

### Note:

**Modules look at their input flags regularly**, usually once a second (usually set globally for the module with the ups attribute in the config zone). This means that they do not immediately detect a signal/change, *only the next time they look at their input flags*. This means that they can even miss a signal if the input flag changes too quickly.

## 1.2 Bang! Method (output, sending signals)

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

- 'on'  
Sets the flag's value to one, no matter what it was before. Same as using the number 1 (one)
- 'off'  
Sets the flag's value to 0 (zero), no matter what the value was before. Same as using the number 0 (zero)
- '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.

- (number)  
Sets the flag to the fixed value (number), no parentheses. Example: 33 – sets the flag's value to (the number) 33

### 1.3 Multiple Output Flags

DML has the ability to bang! multiple flags at the same time. Unless otherwise specified, all output bang! flags (those that end on an exclamation point "!") support this ability. To bang! multiple flags, they need to be listed as the attribute's value and are separated by a comma; leading/trailing blanks are ignored.

counterOut!	*cVal
tMinus!	*counted
zero!	isZero, startStageTwo

Please note the following:

- All flags are banged with the same method
- All flags are set simultaneously, meaning that there is no guaranteed order in which they are changed.

### 1.4 DML? Watchflag Method (input, receiving signals)

DML understands the following conditions to look for in a flag, each defined by a keyword. Watchflags are inspected regularly and can only trigger after a flag's value changes; they trigger when the following conditions are met:

- 'change' or '#'  
triggers whenever the flag's value changes
- 'off' or '0' or 'no' or 'false'  
triggers when the flag's value changes to zero
- 'on' or '1' or 'yes' or 'true'  
triggers when the flag's value changes from zero to non-zero  
(**Warning:** DML will not detect a transition between two non-zero numbers e.g., 3→4, it only triggers on a change from ZERO to a non-zero value)
- 'inc' or '+1'  
triggers when the flag's value changes to exactly the flag's previous value plus one
- 'dec' or '-1'  
triggers when the flag's value changes to exactly the flag's previous value minus one
- '>(number)' or '>(name)'  
triggers when the flag's value changes, and the value is larger than the number given or flag identified by name

#### Examples:

- >4 triggers when inFlag is larger than the number 4

- `>*landings` triggers when inFlag is larger than the value of local flag 'landings'
- ``=(number) '`` or ``=(name) '``  
triggers when the flag's value changes, and the value is equal to the number given or flag identified by name

**Examples:**

- `=4` triggers when inFlag is equal to the number 4
- `=*landings` triggers when inFlag is equal to the value of local flag 'landings'
- ``<(number) '`` or ``<(name) '``  
triggers when the flag's value changes, and the value is less than the number given or flag identified by name

**Examples:**

- `<4` triggers when inFlag is less than the number 4
- `<*landings` triggers when inFlag is less than the value of local flag 'landings'
- ``#(number) '`` or ``#(name) '``  
triggers when the flag's value changes, and the value is not equal to the number given or flag identified by name

**Examples:**

- `#4` triggers when inFlag is not equal to the number 4
- `#*landings` triggers when inFlag is not equal to the value of local flag 'landings'

### Quoting Numbered Flags

Early versions of DCS used only numbers for flag names. For example, "22" was (and still is) a legal flag name. This can create confusion when using Watchflags as it can't tell the difference between a number and a flag whose name happens to be a number.

In order to allow DML to distinguish between a number and flag whose name happens to be a number, such a flag's name must be put into double quotes "" and "" to be interpreted as a flag number. Hence, if you want to trigger only if the input flag was equal to flag named 22, the condition would be

`= "22"`

DML then (and only then) recognizes "22" as meaning the flag named 22 rather than the number 22.

## 1.5 DML Flag Naming Rules

	ME	DML	DML Zone-Local
<b>Format</b>	Positive Number	Alphanumeric, must not contain comma ',', <b>must not start with asterisk '*'</b> , <b>double quote ""</b> or <b>digit ('0'...'9')</b>	<b>Starts with asterisk '*'</b> , alphanumeric, must not contain comma ','
<b>Examples</b>	123 99 2898	A12 With blank F***d up Yup "quotes" too	*1 *A12 *fireCloner *ok multi **
<b>Scope / Visibility</b>	Entire DCS	DML modules, Lua Scripts	Only DML modules in same zone
<b>Invisible to</b>		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 <b>MANDATORY</b>
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 <b>each</b> 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 <b>true</b> .
shellVariance	Difference in shell’s explosion power, in percent. Defaults to 0.2 (20%)
f? in? artillery?	<b>DML Watchflag</b> . When triggered, the artillery bombardment starts. Defaults to <none> You can use any synonym, but only one per zone
triggerMethod artyTriggerMethod	Defines the trigger condition for the DML Watchflag. Defaults to “change”
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**

**Supports Watchflags**

## **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 <b>MANDATORY</b>
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 <b>(Deprecated – use <i>method</i> and <i>f!</i>)</b>
f=0	Sets the flag specified in Value to 0 when cargo is delivered <b>(Deprecated – use <i>method</i> and <i>f!</i>)</b>
f+1	Increases the value of the flag specified in Value by 1 when cargo is delivered <b>(Deprecated – use <i>method</i> and <i>f!</i>)</b>
f-1	Decreases the value of the flag specified in Value by 1 when cargo is delivered <b>(Deprecated – use <i>method</i> and <i>f!</i>)</b>
method cargoMethod	Standard DML flag method for output. Defaults to “inc”
f! cargoReceived!	The flag to bang! when the object is destroyed. Use only one synonym per zone.

### Supports DML Flags

### 4.4 Demos

- Helo Cargo

## 5 Changer

### 5.1 Summary

This module provides a convenient way to transform flags on-the-fly, and to provide a flexible gated switch

### 5.2 Dependencies

dcsCommon, cfxZones

### 5.3 ME Integration

Name	Description
change?	Marks this ME Zone as a clone zone. The value of this attribute designated the input flag that is to be changed to output values. <b>MANDATORY</b>
triggerMethod triggerChangeMethod	Watchflag method to apply to input flag when enabled with inEval. Defaults to 'change'
inEval	Tells changer to apply triggerMethod to evaluate the change? flag. If set to false, the value of the change? flag is read directly.
changeTo	Value transformation that is to be applied to the input flag's value after it has been read. Currently defined values are <ul style="list-style-type: none"><li>• bool Input is to be interpreted as a boolean value. If the value of the input flag (after applying triggerMethod when enabled) is zero, the output value is zero (0). In all other cases the output value is one (1)</li><li>• not Input is to be interpreted as an inverted Boolean value. If the value of the input flag (after applying triggerMethod when enabled) is zero, the output value is one (1). In all other cases the output value is zero (0)</li><li>• val send the result of all transformations of input to the output flag. Most commonly used when changer is used as a gated switch.</li></ul> Defaults to 'val'
out! changeOut!	The flags that are to be set to output values after transformation from the value read from change?
paused changePaused	Defines the initial state of changer when the mission starts up. True means that the changer is paused and will not change the output flags. Defaults to false

Name	Description
on? changeOn?	Turns a paused changer to active, enabling it to put signals on the out! flags. When already active has no effect. Always reacts to change of flag value Defaults to <none>
off? changeOff?	Pauses an active changer, making it no longer change the out! flags. Has no effect if the changer is already paused Defaults to <none>

**Supports DML Flags**  
**Supports Watchflags**  
**Supports zone-local verbosity**

## 5.4 Demos

## 6 CivAir

### 6.1 Summary

Drop-in to generate civilian air traffic. Runs out of the box, can be easily customized.

### 6.2 Dependencies

dcCommon, cfxZones

### 6.3 ME Integration

CivAir is customized via a config zone “civAirConfig” with the following attributes:

Name	Description
verbose	A value of “true” turns on debugging messages. Default is “false”
aircraftTypes	<p>A comma-separated list of Types (as defined in <a href="https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Aircraft">https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Aircraft</a> ) that define the aircrafts used for civilian flights. These must be fixed wing aircraft (i.e not helicopters).</p> <p>All airframe types are picked from this list, and each entry has the same chance to be picked. This means that if you list the same type twice, you increase the chance of that type to be picked.</p> <p>civAir defaults to the following list:</p> <ul style="list-style-type: none"><li>• Yak-40, Yak-40, C-130, C-17A, IL-76MD, An-30M, An-26B</li></ul> <p>Note that by default, the Yak-40 is twice as likely to be picked for a flight as an aircraft because its type is listed twice</p>
ups	Number of updates per second that civAir checks on its flights. By default, this is 0.05, or once every 20 seconds.
maxTraffic	Maximum number of civilian flights at the same time. Defaults to 10
maxIdle	<p>Number of seconds of an aircraft idling that can elapse before it is removed.</p> <p>CivAir determines that an aircraft is idling by checking if it is moving. If you set this number too low, a cold-starting aircraft may be removed before it can move.</p> <p>Defaults to 480 (seconds = 8 minutes)</p>
initialAirSpawns	Controls if at mission start half of maxTraffic immediately spawn in mid-air to start a mission with planes in the air.

You can add airfields to the inclusion / exclusion set with trigger zones as follows:

Name	Description
civAir	<p>When present, the airfield closest to this trigger zone is added to either the inclusion or exclusion set.</p> <p>If the attribute’s value is ‘exclude’ the closest airfield is added to the exclusion list. In all other cases, it is added to the inclusion list.</p>

Name	Description
	<p>Note that you can add the same airfield multiple times by adding multiple trigger zones with a civAir attribute close to the same airfield. This will proportionally increase the likelihood of the airfield being picked for either destination or departure</p> <p>Note also that only functioning airfields are chosen. FARPS or ships are disregarded.</p> <p><b>MANDATORY</b></p>

## 6.4 Demos

- Virgin (Civ) Air
- Air Caucasus II

## 7 Clone Zone

### 7.1 Summary

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

### 7.2 Dependencies

dcsCommon, cfxMX, cfxZones

### 7.3 ME Integration

Name	Description
cloner	Marks this ME Zone as a clone zone. <b>Value of this attribute is ignored</b> , use it to describe this cloner's function. <b>MANDATORY</b>
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 <b>more than one template zone names</b> , 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 ( <b>no spawn on start</b> of mission). <b>To spawn units at mission start, set this attribute to true.</b>  <b>Deprecated. Use a raiseFlag module instead</b>
masterOwner	If not present, all cloned units retain the exact ownership of the units that the template was created from.

Name	Description
	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. Defaults to <none, retain template ownership>
spawn? f? in? clone?	<b>DML Watchflag.</b> When triggered, the cloner starts a new clone cycle. Defaults to <none> You can use any synonym, but only one per zone
triggerMethod cloneTriggerMethod	Defines the trigger condition for the DML Watchflag. Defaults to "change"
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	<del>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 &lt;none&gt;</del>  <b>Deprecated. Use empty! instead</b>
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. <b>Note:</b> use this in favor of empty+1 <b>Note:</b> use either empty+1 or empty! but <b>not both</b> .
method	Standard DML 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 <b>not</b> trigger subsequently. Defaults to <none>
trackWith:	List of groupTracker zones. All spawned groups are added to these groupTrackers. If you have stacked the tracker on the same zone as the cloner, you can use a single asterisk "*" as zone name. Supports a comma-separated list of trackers if you simultaneously want to pass the cloned groups to multiple trackers, e.g. "GroundTrack, HeloTrack" This is useful if your cloner clones more than one group, and your trackers use filtering.

**Supports DML flags**  
**Supports Watchflags**

## 7.4 Demos

- Attack of the CloneZ
- Once, twice, three times a maybe
- Clone Relations
- Frog Men Training



- Flag Fun
- Track This!

## 8 Count Down

### 8.1 Summary

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

### 8.2 Dependencies

dcsCommon, cfxZones

### 8.3 ME Integration

Name	Description
<b>countDown</b>	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) <b>MANDATORY</b>
loop	If this attribute is true, a count down restarts after reaching zero. If the count down is given as a range, a new random start value is taken from that range (including upper and lower limit)
method ctdwnMethod	DML flag method for output. Use only one synonym per zone. Defaults to "flip"
count? in?	Watchflag. Triggering this flag constitutes a signal to count down by one. Use only one synonym per zone. Defaults to <none>
triggerMethod ctdwnTriggerMethod	Watchflag trigger conditions. Defaults to "change"
zero! out!	DML Output flag to bang when countdown reaches zero. Use only one synonym per zone. Defaults to <none>
tMinus!	DML Output flag to bang when the count value is lowered, and has not yet reached zero. Defaults to <none>
counterOut!	A flag, when given, is set to the current count value. The flag is updated each time that the counter is triggered via count?
disableCounter?	A Watchflag that when triggered, turns off the counter for good. Defaults to <none>

### 8.4 Demos

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



## 9 CSAR Manager

### 9.1 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.

### 9.2 Dependencies

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

Optional: cfxCommander (when using CSAR Zones)

### 9.3 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
<b>CSARBASE</b>	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. <b>MANDATORY</b>
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
<b>CSAR</b>	Identifies this as CSAR Zone that is converted into a CSAR mission upon mission start or when the startCSAR? flag is changed. <b>MANDATORY</b>
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>

## 9.4 Demos

- CSAR of Georgia

## 10 Delay Flags

### 10.1 Summary

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

### 10.2 Dependencies

dcsCommon, cfxZones

### 10.3 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 <b>MANDATORY</b>
out! delayDone!	The flag to bang! after the delay has passed. Use only one synonym per Zone
method delayMethod	DML Flag method for output Defaults to "inc"
f? in? startDelay?	Watchflag for a change that starts the delay. Use any synonym, but only one per zone.
triggerMethod delayTriggerMethod	DML Method for Watchflags
stopDelay?	Stops a running delay. Has no effect on a stopped delay

**Supports DML flags**

**Supports Watchflags**

**Supports zone-local verbosity**

### 10.4 Demos

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

## 11 FARP Zones

### 11.1 Summary

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

### 11.2 Dependencies

dcsCommon, cfxZones

### 11.3 ME Integration

Name	Description
<b>FARP</b>	Indicates that this zone is a FARP zone. Value is ignored. <b>MANDATORY</b>
rPhiHDef	Radius (in m), Phi (degrees) and Heading (degrees) of the center point around which the <b>defenders</b> deploy. Defaults to 0, 0, 0
rPhiHRes	Radius (in m), Phi (degrees) and Heading (degrees) of the center point around which the <b>resource</b> 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.

### 11.4 Demos

- FARP and away

## 12 Group Tracker

### 12.1 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.

### 12.2 Dependencies

dcsCommon, cfxZones

### 12.3 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:	<p>List of groupTracker zones. All groups that have at least one unit inside this zone are added to these groupTrackers. 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.</p> <p>If you have stacked the tracker on the same zone, you can use a single asterisk "*" as zone name.</p> <p>Supports a comma-separated list of trackers if you simultaneously want to pass the groups to multiple trackers, e.g. "GroundTrack, HeloTrack"</p> <p>This is useful if the zone contains more than one group, and your trackers use filtering</p> <p>Add all groups that have at least one unit in this zone to the tracker whose zone name is given in the Value field.</p>

To add a groupTracker to a zone

Name	Description
tracker	<p>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</p> <p><b>MANDATORY</b></p>
addGroup!	<p>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</p> <p>Defaults to &lt;none&gt;</p>
removeGroup!	<p>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</p>



Name	Description
	Defaults to <none>
numGroups!	The value of this flag always represents the number of groups currently watched by this tracker. <b>This value is updated 1/ups times per second.</b> Defaults to <none>
groupFilter	Which unit categories to track. If no attribute is given, <b>all</b> categories are tracked. When you supply a groupFilter attribute, <b>only that category is accepted</b> when attempting to add to a tracker. Currently supported are <ul style="list-style-type: none"> <li>• 0 (zero) or "aircraft" or "air"</li> <li>• 1 or "helo" or "heli" or "helicopter"</li> <li>• 2 or "ground"</li> <li>• 3 or "ship"</li> <li>• 4 or "train"</li> </ul> Defaults to no filtering

## 12.4 Demos

- Track This!

## **13 Guardian Angel**

### **13.1 Summary**

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

### **13.2 Dependencies**

Tbc

### **13.3 ME Integration**

Tbc

### **13.4 Demos**

- Missile Evasion
- Guardian Angel

## **14 Helo Troops**

### **14.1 Summary**

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

### **14.2 Dependencies**

Tbc

### **14.3 ME Integration**

Tbc

### **14.4 Demos**

- Helo Trooper

## 15 Map Markers

### 15.1 Summary

Allows you to place markers on the F10 map.

### 15.2 Dependencies

dcsCommon, cfxZones

### 15.3 ME Integration

Name	Description
<b>mapMarker</b>	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" <b>MANDATORY</b>
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"

### 15.4 Demos

## 16 Messenger

### 16.1 Summary

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

### 16.2 Dependencies

dcsCommon, cfxZones

### 16.3 ME Integration

Name	Description
<b>messenger?</b>	Watchflag. When triggered, the module will display the message and/or play sound.  <b>MANDATORY</b>
<del>f?</del> <del>in?</del> <b>messageOut?</b>	<del>When triggered, the message is displayed, and sound is played</del> <del>You can use any synonym, but only one per zone</del>  <b>Deprecated. Use messenger? instead</b>
triggerMethod msgTriggerMethod	Defines the trigger condition for DML Watchflags. Use only one synonym per zone Defaults to "change"
message	The text of the message to be displayed  <b>TEXT WILDCARDS</b> <n> creates a new line <z> is replaced with zone's name <v> is replaced with the value currently held by the flag specified in <i>messageValue?</i>
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
<del>spaceBefore</del>	If true, an empty line is issued before the text message. Defaults to false  <b>Deprecated. Use &lt;n&gt; in the message instead</b>
<del>spaceAfter</del>	If true, an empty line is issued after the text message. Default is false  <b>Deprecated. Use &lt;n&gt; in the message instead</b>
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 (I10n/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**  
**Supports DML Watchflags**

## 16.4 Demos

- Bottled Messages
- Frog Men Training
- The Zonal Countdown
- Follow Me!
- CSAR of Georgia
- Track This!
- Watchflag Demo
- Radio Go-Go
- xFlags Field Day

## 17 NDB

### 17.1 Summary

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

### 17.2 Dependencies

dcsCommon, cfxZones.

### 17.3 ME Integration

Name	Description
NDB	<p>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.</p> <p><b>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)</b></p> <p><b>MANDATORY</b></p>
fm	<p>If true, the transmission is in FM, else in AM</p> <p>Defaults to false (AM)</p>
ndbSound	<p>Name of the sound file with extension that is to be transmitted. Defaults to '&lt;none&gt;'. Note that the sound file's name must be specified relative to the missions default location for sound files (I10n/DEFAULT/). If you use ME to import the sound files, you do not have to specify the location.</p> <p>Remember to import the sound file into the mission else no sound will play.</p>
watts	<p>Transmission power (in watts) for the NDB. 100 Watts usually has a range of some 150 km.</p> <p>Defaults to 100 Watts</p>
paused	<p>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.</p> <p>Defaults to false</p>
on?	<p>Watchflag. <b>Each time the flag triggers, the NDB is started (will also cause the transmission sound to rewind).</b> The current paused value is ignored, and then set to false after the NDB has started.</p> <p>Defaults to no flag to watch</p>
off?	<p>Watcheflag. <b>Each time the flag triggers, the NDB is stopped.</b> paused value is set to true after the NDB has stopped.</p> <p>Defaults to no flag to watch</p>
triggerMethod ndbTriggerMethod	<p>Defines the trigger condition for DML Watchflags. Use only one synonym per zone</p> <p>Defaults to "change"</p>

## **17.4 Demos**

- ADF and NDB Fun



## 18 Object Destruct Detector

### 18.1 Summary

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

### 18.2 Dependencies

dcsCommon, cfxZones

### 18.3 ME Integration

Name	Description
<b>OBJECT ID</b>	THIS ATTRIBUTE IS FILLED BY ME AND MUST NOT BE CHANGED <b>MANDATORY</b>
<b>NAME</b>	THIS ATTRIBUTE IS FILLED BY ME AND MUST NOT BE CHANGED <b>MANDATORY</b>
f=1	Sets the flag specified in Value to 1 when object is destroyed <b>(deprecated – use oddMethod and f!)</b>
f=0	Sets the flag specified in Value to 0 when object is destroyed <b>(deprecated – use oddMethod and f!)</b>
f++	Increases the value of the flag specified in Value by 1 when object is destroyed <b>(deprecated – use oddMethod and f!)</b>
f--	Decreases the value of the flag specified in Value by 1 when object is destroyed <b>(deprecated – use oddMethod and f!)</b>
method oddMethod	DML Method for output flags. Defaults to “flip”
f! destroyed! objectDestroyed!	The flag to bang! when the object is destroyed. Use only one synonym per zone. Defaults to “*none”

### 18.4 Demos

- Object Destruct Detection

## 19 Object Spawn Zones

### 19.1 Summary

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

### 19.2 Dependencies

dcsCommon, cfxZones, (cfxCargoManager)

### 19.3 ME Integration

Name	Description
objectSpawner	Marks this ME Zone as a spawn zone. <b>Value of this attribute is ignored</b> , use it to describe what it spawns to make mission editor easier for you <b>MANDATORY</b>
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 spawner
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.  <b>WARNING:</b> 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) <b>MANDATORY</b>
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. <b>If provided, MUST BE UNIQUE</b> 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

## 19.4 Demos

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

## 20 Owned Zone

### 20.1 Summary

Zones that can be conquered and generate a signal when conquered

### 20.2 Dependencies

dcsCommon, cfxZones, (cfxGroundTroops)

### 20.3 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) <b>MANDATORY</b>
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. <b>Warning:</b> these types need to <i>exactly</i> 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. <b>Warning:</b> these types need to <i>exactly</i> 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. <b>Warning:</b> these types need to <i>exactly</i> 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. <b>Warning:</b> these types need to <i>exactly</i> 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"

## 20.4 Demos

- Owned Zones ME Integration

## **21 Player Score**

### **21.1 Summary**

A module to keep award and tabulate kill score for players

### **21.2 Dependencies**

Tbc

### **21.3 ME Integration**

Tbc

### **21.4 Demos**

- Keeping The score
- More Score

## 22 Pulse Flags

### 22.1 Summary

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

### 22.2 Dependencies

dcsCommon, cfxZones

### 22.3 ME Integration

Name	Description
<b>pulse</b> <b>pulse!</b>	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  <b>MANDATORY</b>
method	DML Flag output method Defaults to “flip”
done! pulsesDone!	This flag’s value is changed when the pulser completes a fully run of pulses. Can only happen when the <i>pulses</i> attribute supplies a positive number. Use only one synonym per zone. Defaults to <none>
triggerMethod pulseTriggerMethod	Watchflag condition for input flags. Use only one synonym per zone
activate? startPulse?	Watchflag. When triggered, a paused pulser is reset and then restarted. Use only one synonym per zone. Defaults to <none>
pause? pausePulse?	Watchflag. When triggered, a pulser is paused. Use only one synonym per zone. Defaults to <none>
paused pulseStopped	When true, the pulser does not start at the beginning of a mission but waits for trigger on activate?-flag. Use only one synonym per zone. 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. Defaults to -1 (endless)
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 <i>time</i> by setting zeroPulse to false The effect is that the initial pulse happens after the first delay Default is true (initial pulse immediately)

## **22.4 Demos**

- Pulsing Fun
- Frog Men Training
- The Zonal Countdown
- Watchflag Demo



## 23 Radio Trigger

### 23.1 Summary

Provides an interface for ME-based Communication→Other Radio Items, allowing multiple uses by re-setting the flag after it has been triggered.

### 23.2 Dependencies

dcsCommon, cfxZones

### 23.3 ME Integration

Name	Description
radio?	Watchflag. Triggers a radio cycle, then resets this flag to its pervious value  <b>MANDATORY</b>
triggerMethod radioTriggerMethod	Method that triggers the Watchflag Defaults to 'change'
method rtMethod	Method how the output flag should be triggered. Defaults to 'inc'
out! rtOut!	DML Flag to set when the module triggers Defaults to <none>

**Supports DML Flags**

**Supports Watchflags**

**Supports zone-local verbose**

### 23.4 Demos

- Radio go go

## 24 Raise Flag

### 24.1 Summary

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

### 24.2 Dependencies

dcsCommon, cfxZones

### 24.3 ME Integration

Name	Description
<b>raiseFlag</b> <b>raiseFlag!</b>	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 <b>MANDATORY</b>
value	Method or value to set the flag to. Supports <ul style="list-style-type: none"><li>• 'inc' increment the flag's current value by one</li><li>• 'dec' decrements the flag's current value by one</li><li>• 'flip' sets the flag's current value to 0 if it was anything but zero, and to 1 if it was zero.</li><li>• (number) – set the flag to this number</li></ul> Default is 'inc'
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. <b>If this attribute is omitted, the flag is set 0.5 seconds after the mission starts.</b>
stopFlag?	Only useful in conjunction with afterTime. A Watchflag. When triggered and raiseFlag is still waiting for afterTime, raiseFlag is 'disarmed' and raising the flag is cancelled. It will not be raised later. Once stopped, it cannot be re-started.
triggerMethod raiseTriggerMethod	Watchflag condition for stopFlag?

**Supports DML Flags**

**Supports Watchflags**

### 24.4 Demos

- Flag Fun
- Attack of the CloneZ



## 25 Recon Mode

### 25.1 Summary

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

### 25.2 Dependencies

Tbc

### 25.3 ME Integration

Name	Description
verbose	A value of “true” turns on debugging messages. Default is “false”
autoRecon	<p>If true, all planes are automatically treated as actively reconnoitering.</p> <p><b>NOTE</b> This is on by default. To avoid excessive scouting activity, you should reduce the number of active scout planes with enabling or disabling one of the following attributes: redScouts (off), blueScouts (off), greyScouts (off), playerOnlyRecon (on) Default: true</p>
redScouts	If true, all red planes are included as scouts when autoRecon is true. Default is false
blueScouts	If true, all blue planes are included as scouts when autoRecon is true. Default is <b>true</b>
greyScouts	If true, all neutral planes are included as scouts when autoRecon is true. Default is false
playerOnlyRecon	<p>If true, only player aircraft are included as scouts when autoRecon is true. AI planes will not be automatically included as scouts.</p> <p><b>IMPORTANT</b> This condition is applied <b>in addition</b> to blueScouts and redScouts. If you disallow red scouts, red players will not automatically be added to the list of scouts. Defaults to false</p>
reportNumbers	If true, the F10 map markings include a unit count of the group at the time the group was discovered. Default is true
applyMarks	If true, discovered groups are marked on the F10 map. Default is true
announcer	If true, discovered groups are announced via text. Default is true
detectionMinRange	The detection range of a recon plane under worst conditions (low-level flying). Default is 3000 (3 km)
detectionMaxRange	The detection range of a recon plane under best conditions (high-altitude). Default is 12000 (12 km)
maxAlt	The altitude at which a plane achieves maxDetectionRange. Default is 9000 (9 km, 27'000 ft)

<b>Name</b>	<b>Description</b>
prio+	A flag in ME that is increased every time that a priority unit is detected
detect+	A flag in ME that is increased every time that a normal (non-priority) is detected
reconSound	The name of the sound file to play when e recon event occurs. Defaults to <nosound>, which will not play a sound
autoRemove	When a detected group is destroyed, that group's mark is immediately removed from the map if this attribute is set to true Defaults to true

## 25.4 Demos

- Recon Mode

## 26 rnd Flags

### 26.1 Summary

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

### 26.2 Dependencies

dcsCommon, cfxZones

### 26.3 ME Integration

Name	Description
<b>RND</b>	Marks this as a randomizer. The value of this attribute is ignored <b>MANDATORY</b>
<b>flags!</b>	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" <b>MANDATORY</b>
method rndMethod	Describes how flags are to be set. Follows standard DML flag method. <b>Defaults to "on"</b>
pollSize	Number of items to choose from the set of flags during a cycle. 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. Defaults to 1
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. Defaults to false
f? in? rndPoll?	DML Watchflag to start a random cycle. Defaults to <none set> You can use any synonym, but only one per Zone
triggerMethod rndTriggerMethod	Watchflag condition when to trigger. Defaults to "change"
onStart	If true, a cycle is run for this randomizer 0.25 seconds after the mission starts. Defaults to false  <b>NOTE:</b> if no f? (or synonym) is specified, and onStart is false, onStart is automatically set to true so the randomizer runs once

Name	Description
done+1	<p>The flag number to increase when the randomizer has run out of flags to change, and reshuffle is false (randomizer did nothing)</p> <p>Is increased every time that the randomizer runs a cycle on an empty flag set</p>

**Supports DML Flags**

**Supports DML Watchflags**

## 26.4 Demos

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

## 27 Smoke Zones

### 27.1 Summary

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

### 27.2 Dependencies

dcsCommon, cfxZones

### 27.3 ME Integration

Name	Description
smoke	Adds a permanent smoke affect to the center of the zone. Possible values for the smoke effect are: <ul style="list-style-type: none"><li>• “green” or “0”</li><li>• “red” or “1”</li><li>• “white” or “2”</li><li>• “orange” or “3”</li><li>• “blue” or “4”</li><li>• “random”, “?” or “rnd” (random color from above)</li></ul> <b>MANDATORY</b>
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?	Watchflag. When this flag triggers, smoke starts. Defaults to <none>
altitude	Altitude (in meters) above ground that the smoke should be created. Defaults to 1m
triggerMethod smokeTriggerMethod	Conditions when the DML Watchflag should trigger Defaults to ‘change’

### Supports Watchlags

### 27.4 Demos

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



## 28 Spawn Zones

### 28.1 Summary

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

### 28.2 Dependencies

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

### 28.3 ME Integration

Name	Description
spawner	Marks this ME Zone as a spawn zone. <b>Value of this attribute is ignored</b> , use it to describe what it spawns to make mission editor easier for you <b>MANDATORY</b>
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 spawn
types	Type string array for the ground units that are spawned. Example "Roland ADS, Roland Radar, Roland ADS" or "Soldier M4" – <b>WARNING:</b> 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 <a href="https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB">https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB</a> 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 ( <b>Warning:</b> 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: <a href="https://wiki.hoggitworld.com/view/DCS_enum_country">https://wiki.hoggitworld.com/view/DCS_enum_country</a> ).
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. <b>If provided, baseName MUST BE UNIQUE. If you do not assign a base name, a unique one will be generated for you.</b> 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 is 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
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 times. 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 <code>cfxSpawnZones.spawnWithSpawner()</code> . See the API section on how to get a list of eligible spawners. Automatically interfaces with HeloTroops and other enhancements

## 28.4 Demos

- Random Death

- Moving Spawners
- Helo Troops

## 29 Unit Zone

### 29.1 Summary

Tests if a player unit or any member of an AI group is inside/outside a zone and can change flags when the status changes.

### 29.2 Dependencies

dcsCommon, cfxZones

### 29.3 ME Integration

Name	Description
unitZone	Marks this ME Zone as an anchor for unitZone <b>MANDATORY</b>
lookFor	<p>Name for the AI group or player unit to check zone status. If the last character in the name is an asterisk "*", exact matches <i>and</i> all group/unit names that start with that string (minus asterisk) are accepted, e.g. if you supply "Hel*" all of the following would be accepted:</p> <ul style="list-style-type: none"><li>• Hel</li><li>• Hello World</li><li>• Helo Rescue-1</li><li>• Hellfire</li></ul> <p>You can use this feature to your advantage in conjunction with cloners or spawners, as these all produce groups with a known base name.</p> <p>If you only supplied "Hel",only (without the asterisk "*") only the group whose name exactly matches "Hel" is checked.</p>
matching	<p>What type of units to match. Currently supported are</p> <ul style="list-style-type: none"><li>• group (default): look for group names</li><li>• player – look only at player units and match their unit's (not group's) name against lookFor</li></ul> <p>Default: group</p>
coalition uzCoalition	<p>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</p> <p>Defaults to 0 (both)</p>
filterFor	<p>Which categories to look for. If no attribute is given, <b>all</b> categories are checked against the zone (when their name pattern matches). When you supply a filterFor attribute, only that category is considered.</p> <p>Currently supported are</p> <ul style="list-style-type: none"><li>• 0 (zero) or "aircraft" or "air"</li><li>• 1 or "helo" or "heli2" or "helicopter"</li><li>• 2 or "ground"</li></ul>

Name	Description
	<ul style="list-style-type: none"> <li>• 3 or “ship”</li> <li>• 4 or “train”</li> </ul> <p>Defaults to no filtering</p>
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
method uzMethod	DML Flag method for output. Use only one synonym per zone
uzOff?	Watchflag. When triggered, this zone will no longer perform checks. When already off, nothing happens
uzOn?	Watchflag. When triggered, this zone will resume checks. When already on, nothing happens
triggerMethod uzTriggerMethod	Method that determines when the watchflags should trigger. Default is “change”
uzDirect	<p>When present, this flag (or flags) is always set to the current state of the unit zone:</p> <ul style="list-style-type: none"> <li>• 1 when one or more units in the zone</li> <li>• 0 when none of the indicated units in the zone.</li> </ul> <p>Default is &lt;none&gt;</p>

**Supports DML Flags**  
**Supports Watchflags**  
**Supports zone-local verbose**

## 29.4 Demos

- Follow Me!
- xFlags – Field Day

## 30 Wiper

### 30.1 Summary

Removes objects inside a zone whenever you trigger it

### 30.2 Dependencies

dcsCommon, cfxZones

### 30.3 ME Integration

Name	Description
wipe?	Watchflag. Triggers a wipe cycle  <b>MANDATORY</b>
triggerMethod triggerWiperMethod	Method that triggers the Watchflag
category wiperCat	Category of the objects that are affected by the wipe (i.e. if they belong to the category they may be wiped). Possible values are <ul style="list-style-type: none"><li>• “unit” or 1</li><li>• “weapon” or 2</li><li>• “static” or 3</li><li>• “base” or 4</li><li>• “scenery” or 5</li><li>• “cargo” or 6</li></ul> Defaults to ‘static’
wipeNamed	<i>Optional</i> comma-separated name list that an object’s name must match in order to be wiped. Supports an asterisk (“*”) as wildcard to match anything. For example, “Ba*” would match “Base”, “Ba”, “Babushka”, and “Bathyscape”  <b>Examples:</b> <ul style="list-style-type: none"><li>• “Ba*” – all objects inside the zone whose name starts with “Ba”</li><li>• “Grou*, Commander Kirk, He*” – all objects whose name starts with “Grou” or “He”, and the object whose name exactly matches “Commander Kirk”</li></ul> Defaults to <option off>, no name filtering
wipeInventory	A Boolean that turns on the wiper’s inventory function. Whenever triggered, the zone lists all objects, it finds inside the zone, sorted by category. Note that there may be objects inside a zone that wiper cannot find, and that it may return objects that are not really inside the zone. Both are a DCS limitation, not a bug in wiper.

**Supports DML Flags**

**Supports DML Watchflags**

## 30.4 Demos

- Viper with a double you

## 31 xFlags (Flag Testing)

### 31.1 Summary

This requires that multiple input flags meet certain conditions to trigger the output flag. xFlags first applies the trigger method to evaluate input flags individually, and then requires that the individual results meet a condition to arrive at a final results. For example, it can be used to determine if ALL flags (requirement) have individual values of ">2" (individual trigger)

### 31.2 Dependencies

dcsCommon, cfxZones

### 31.3 ME Integration

Name	Description
xFlags?	<p>A list (comma-separated) of input flags whose values should be evaluated to form the output signal</p> <p><b>MANDATORY</b></p>
require	<p>Condition/Operation that should apply to the input flags to form the output value. Currently supports the following conditions:</p> <ul style="list-style-type: none"><li>• 'or', 'any', or 'some' triggers if at least one of the input flags has triggered</li><li>• 'and' or 'all' triggers if all the input flags have triggered</li><li>• 'more than' triggers if more than the value given in '#hits' of input flags have triggered</li><li>• 'at least' triggers if #hits or more of the input flags have triggered</li><li>• 'exactly' if triggers if the number of input flags that have triggered is equal to #hits</li><li>• 'none' triggers if none of the input flags have triggered. Requires that you turn off one-shot mode</li><li>• 'not all' or 'nand' triggers when not all input flags have triggered. Requires that you turn off one-shot mode</li></ul>



Name	Description
	<ul style="list-style-type: none"> <li>• 'most' triggers when more than half of the input flags have triggered. Will not trigger if exactly half have triggered, so be careful if the number of input flags is even.</li> <li>• 'half or more' triggers when half or more of the input flags have triggered. Will also trigger if exactly half of all flags have triggered</li> </ul> <p>Defaults to 'some'</p>
#hits	<p>Value used for only some of the require attribute. Can be a value or flag name (in which case the value will be loaded from that flag). Numbered flags must be enclosed in double quotes, e.g. "22" to access flag 22. Defaults to 1 (one)</p>
xFlagMethod	<p>Condition that must be met for individual input flags. Is identical to trigger method for Watchflags except it is applied to each input flag individually. Defaults to "change"</p>
xSuccess! out!	<p>Flag to bang! when xFlags when the evaluation of input flags succeeds (all conditions are met). Once xSuccess is triggered, and unless oneShot is set to false, this zone's xFlag pauses until xReset is triggered. Defaults to &lt;none&gt;</p>
xChange!	<p>Flag to bang! when xFlags detects a change in the input configuration. Merely detects a change in the input configuration, has no relation with xSuccess!, except that xSuccess will also be accompanied by a bang on xChange! Defaults to &lt;none&gt;</p>
xDirect!	<p>Each time xFlags evaluates the input flags, it directly sets the xDirect flag to the evaluation result (0 or 1) – this is different from what xSuccess may output, since that flag's value is dependent on the xMethod attribute. Defaults to &lt;none&gt;</p>
xReset?	<p>When the value of this input changes, the zone's xFlag module is reset, and evaluation starts afresh. <b>Note: this input always reacts to a change in the flag's value</b> Defaults to &lt;none&gt;</p>
method xMethod	<p>DML method for output flags Defaults to "inc"</p>
oneShot	<p>When the value of this attribute is false, that zone's xFlag module will not stop evaluating after it triggers xSuccess.</p>
xOff?	<p>Flag to turn the xFlag off, suspending it. When turned off, no processing of input flag occurs. The xFlag will still respond to xReset by loading a new zero state. <b>Note: this input always reacts to a change in the flag's value</b> Defaults to &lt;none&gt;</p>
xOn?	<p>Turns a suspended xFlag back on to resume processing. It resumes processing where it left off <b>Note: this input always reacts to a change in the flag's value</b> Defaults to &lt;none&gt;</p>
xSuspended	<p>Sets the initial state of xFlag. Setting it to true suspends the xFlag at mission start</p>

Name	Description
	Defaults to false

**Supports DML Flags**  
**Supports DML Watchflags**  
**Supports zone-local verbose**

## **31.4 Demos**

- xFlags – Field Day

## **32 Module Name**

### **32.1 Summary**

### **32.2 Dependencies**

### **32.3 ME Integration**

### **32.4 Demos**