cf/x

Dynamic Mission Library
for DCS

DML
QUICK REFERENCE

ME MODULE INTEGRATION

Table Of Contents

1 ME FLAG INTEGRATION 4
2 DEBUGGER 9
3 Artillery Zones 11
4 Artillery UI 13
5 baseCaptured 14
6 Cargo Receiver 15
7 Changer 16
8 CivAir 18
9 Clone Zone 20
10 Count Down 23
11 Counter 24
12 CSAR Manager 25
13 Delay Flags (“Timer”) 27
14 Delicates 29
15 FARP Zones 31
16 FireFX 32
17 Group Tracker 34
18 Guardian Angel 36
19 Helo Troops 38
20 Impostors 39
21 Limited Airframes 41
22 LZ 43
23 Map Markers 45
24 Messenger 46
25 NDB 50
26 Object Destruct Detector 52
27 Object Spawn Zones 53
28 Owned Zone 55
29 Persistence 57
30 Player Score 60
31 Pulse Flags 61
32 Radio Menu 63
33 Radio Trigger 66
34 Raise Flag 67
35 Recon Mode 68

36 rnd Flags 71
37 Sequencer 73
38 Smoke Zones 75
39 Spawn Zones 76
40 ssbClient 79
41 unGrief 81
42 Unit Persistence 83
43 Unit Zone 84
44 Willie Pete 86
45 Wiper 88
46 xFlags (Flag Testing) 90
47 Module Name 92

1 ME FLAG INTEGRATION

1.1 Core Abilities (all DML Zones)

All Modules automatically support some attributes when they are present. They can be added to any Trigger Zone and their functionality is always present for all modules

1.1.1 Dependencies

All DML modules require dcsCommon and cfxZones.

1.1.2 ME Integration

Name	Value
linkedUnit	Moves the zone's center with the unit whose name exactly matches the value of this attribute. That unit must exist at the beginning of the mission, or the linked zone remains at its ME location until the unit exists and becomes linked (at which point it moves to the unit's location). If *linkedUnit* ceases to exist after the zone has followed it, the zone remains at the last location it moved to. If neither *useOffset* nor *useHeading* are set to true, the zone centers on *linkedUnit*. **Note:** be advised that all player and client units do not exist at the beginning of a mission and spawn only when a player occupies that slot.
useOffset	Must be set to "yes" or "true" to have this effect, ignored otherwise. Only has an effect if the zone is linked. Keeps the offset between the *linkedUnit* and zone constant. Note that the zone's center remains the same in relation to the unit's center. If the unit turns, the offset does not change with the unit's heading. **Requires *linkedUnit* be set** Not compatible with *useHeading* (below) – use either Defaults to <not set>
useHeading	When set to true, the zone moves and turns in synch with and relative to the *linkedUnit*. **Requires *linkedUnit* be set** Not compatible with *useOffset* (above) – use either Defaults to <not set>
owner	The coalition that owns this zone. Used with some zone enhancements. Do not set this attribute unless you know what you are doing, or a module's documentation requests you to do so.

Name	Value
	Defaults to neutral

1.2 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.

1.3 Bang! Methods (output, sending signals)

DML understands the following methods, each identified by a keyword or expression:

- ‘`on`’ **[set to 1]**
  Sets the flag’s value to one, no matter what it was before. Same as using the number 1 (one)

- ‘`off`’ **[set to zero]**
  Sets the flag’s value to 0 (zero), no matter what the value was before. Same as using the number 0 (zero)

- ‘`inc`’ **[increase by 1]**
  Increases the flag’s value by 1 (one). Same as ‘+1’. If, for example, the flag’s value was previously 10, that is increased to 11

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

- ‘`flip`’ **[alternate between 0 and 1]**
  This is one of the most effective methods 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

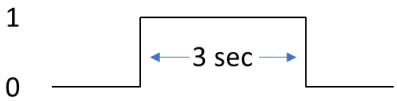
can flip-flop flags, turning them on and off repeatedly.

- (a number) **[set to absolute value]**
  Sets the flag to the fixed value (number), **no parentheses**. Example: 33 – sets the flag's value to (the number) 33

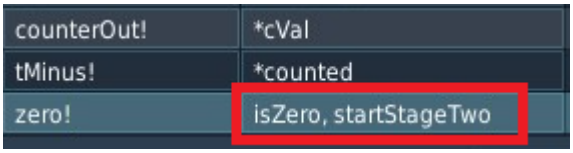
- +(number or flagName) **[add amount or another flag's value**]
  Adds the number give (no parantheses!) or the current value of the flag flagName to the flag. Examples: "+3" adds 3 to the current flag value, while "+killScore" adds the current value of flag "killScore" to the flag, and "+(22)" adds the current value of numbered flag "22" to the current value

- -(number or flagName) **[subtract amount or another flag's value]**
  Subtracts the number given (no parantheses!) the or current value of the flag flagName to the flag. Examples: "-3" subtracts 3 from the current flag value, while "-penalty" subtracts the current value of flag "killScore" from, and "-(22)" subtracts the current value of numbered flag "22" from the current value

- `'pulse'` or `'pulse, <number>'` **[set to 1 and reset automatically]**
  "pulses" the flag by setting its value to 1 (one) for some time, and then re-setting it to 0 (zero) some time later. If you do not specify any time, the flag is reset after three (3) seconds. You can supply your own pulse time by adding a comma and a number, e.g. `"pulse, 4"` will keep the pulse up for four (4) seconds before dropping back to zero.



1.4 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.

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.

1.5 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 flags's value changes to exactly the flag's previous value plus one

- `'dec'` or `'-1'`
  triggers when the flag's value to exactly the flag's previous value minus one

- `'lohi'`
  triggers when the Watchflag's previous value was zero (0) or less and the new value is greater than zero. Often used with the pulse method.

- `'hilo'`
  triggers when the Watchflag's previous value was greater than zero (0) and the new value is zero or less. Often used to detect a countdown reaching zero.

- `'>(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. This is why I recommend you never use number-only flags with DML, and change old-school flag-names to names leading with a letter (e.g. "123" to "F123")

1.6 DML Flag Naming Rules

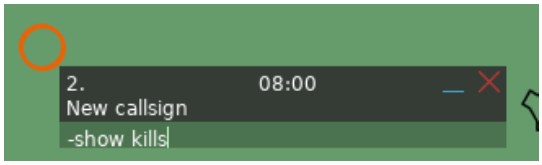
	DML	**DML Zone-Local**
Format	Name must not contain comma ',', **must not start with asterisk '*', double quote '"' nor digit ('0'...'9', recommendation only)**	**Starts with asterisk** '*', must not contain comma ',' must not start with a digit
Examples	A12 With blank F***d up Yup "quotes" too	*1 *A12 *fireCloner *ok multi **
Scope / Visibility	DML modules, Lua Scripts	All DML modules in same zone
Invisible to	n/a	Everyone outside Zone

2 DEBUGGER

2.1 Summary

Interactive (in-mission) debugger utilizing a 'console' via the Add Map Mark Label' mission feature to give commands. Commands start with a hyphen.

"Mark Label"



and will elicit a response when accepted

2.2 Dependencies

The debugger requires dcsCommon and cfxZones

2.3 ME Integration

Name	Description
debug?	List the flag names that the debugger is to observe. All flags listed here are accessible from the debugger under the observer with the same name as the trigger zone **MANDATORY**
triggerMethod debugTriggerMethod inputMethod sayWhen	Trigger condition for the flags (the observer's "condition" that triggers a report for the flag) Defaults to 'change'
method outputMethod debugMethod	DML Method for the debugger's output flags. Rarely used. Defaults to "inc"
notify!	DML flag to bang! when a flag listed in debug? triggers
debugMsg	Message to output when a flag listed in debug? triggers. Supports wildcards, including <f> for the flag name that triggered, and <z> for the zone name. **Note that this allows you to provide individual message formatting per observer**, a feature that is not available for the interactive debugger. Defaults to "`---debug: <t> -- Flag <f> changed from <p> to <c> [<z>]`" which results in a message similar to `---debug: 08:00:12 -- Flag t1 changed from 2 to 4 [many flags]`

Supports DML Flags
Supports Watchflags
Supports zone-local verbosity

2.4 Demos

- Bug Hunt

3 Artillery Zones

3.1 Summary

Simulates an artillery barrage inside the zone.

3.2 Dependencies

dcsCommon, cfxZones

3.3 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? in? artillery?	**DML Watchflag**. 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

3.4 Demos

- Artillery with UI
- Artillery zones triggered
- Pulsing Fun

4 Artillery UI

4.1 Summary

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

4.2 Dependencies

Tcb

4.3 ME Integration

Tbc

4.4 Demos

- Artillery with UI

5 baseCaptured

5.1 Summary

This module generates a signal on the output flags when a base (Airfield, FARP, Ship) is captured (note that currently, ships cannot be captured).

5.2 Dependencies

dcsCommon, cfxZones

5.3 ME Integration

baseCaptured!	Marks this zone as a baseCaptured zone. It lists the flags that should be banged! when the closest base (FARP, Airfield, Ship with Helipad) to this zone is captured by another faction. **MANDATORY**
method captureMethod	DML method for output flags Defaults to 'inc'
blueCaptured! blue!	Flags to bang! when blue faction captures the closest base to this zone Defaults to <none>
redCaptured! red!	Flags to bang! when blue faction captures the closest base to this zone Defaults to <none>
contested!	Flags to bang! when closest base becomes contested and belongs to neither blue nor red Requires handleContested be true in configuration (default) Defaults to <none>
baseOwner	Flag that is set by the module to the current faction (0 = neutral, 1 = red, 2 = blue, 3 = contested) that currently holds this base.

Supports DML Flags
Supports zone-local verbosity

5.4 Demos

- Count Base's Blues

6 Cargo Receiver

6.1 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

6.2 Dependencies

dcsCommon, cfxZones, cfxPlayer, cfxCargoManager.

6.3 ME Integration

cargoReceiver	Marks this zone as a cargo receiver zone. Value is ignored **MANDATORY**
autoRemove	Delete any object <removeDelay> seconds 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)
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.
removeDelay	The delay (in seconds) after which after a successful delivery an object should be removed. Requires that autoRemove be set to true. Defaults to 1, Minimum is 1

Supports DML Flags

6.4 Demos

- Helo Cargo

7 Changer

7.1 Summary

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

7.2 Dependencies

dcsCommon, cfxZones

7.3 ME Integration

Name	Description
change?	The input flag whose value is used to create the output signal **MANDATORY**
out! changeOut!	The output flag.
triggerMethod triggerChangeMethod	Watchflag method that is used to interpret change? when inEval is set to true Defaults to 'change'
inEval	When set to true, the input flag change? is interpreted as a watchflag under triggerChangeMethod's rules Defaults to false
changeTo to	Operation to apply to the input flag to create the ouput value. The following operations are defined: • `'bool'` Output value is 0 if input is 0, 1 otherwise (conversion to bool) • `'not'` Output value is 1 if input is 0, 0 otherwise • `'sign'` Output value is -1 if input is <0, 1 otherwise • `'abs'` Output value is the absolute of input value (e.g. outputs '3' for '-3' and '3') • `'negative'` Output value is input value multiplied by -1, turning the sign. • `'direct'` Output value is the same a input value. Used primarily with when changer is functioning as a gated switch Defaults to "direct"
min	When defined, ensures that the output value has this value at minimum
max	When defined, ensures that the output value has this value. If max is less than min, output is always set to max

Name	Description
paused changePaused	When set to true, the changer starts in paused/off mode and the output flag is not changed. The only way to turn a paused changer on when off/paused is via a signal on the changeOn? input Defaults to false.
on? changeOn?	Turns a paused changer on, enabling transmission of the input signal (after processing) to output, opening the 'gate'. Triggers on "change" Defaults to <none>
off? changeOff?	Turns a changer off, pausing it. Input signals are no longer processed nor propagated to output, closing the 'gate'. Triggers on "change". Defaults to <none>
On/Off? changeOn/Off?	Flag that when defined controls transmission of input to output when the changer in running (i.e. not paused). The value of this input controls the gate as follows: • 0: gate closed, no transmission • Anything else: gate open. Defaults to <none>

Supports DML Flags
Supports Watchflags
Supports zone-local verbosity

7.4 Demos

- Gate and Switch

8 CivAir

8.1 Summary

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

8.2 Dependencies

dcsCommon, cfxZones

8.3 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	A comma-separated list of Types (as defined in https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Aircraft) that define the aircrafts used for civilian flights. These must be fixed wing aircraft (i.e not helicopters). 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. civAir defaults to the following list: • Yak-40, Yak-40, C-130, C-17A, IL-76MD, An-30M, An-26B 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
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	Number of seconds of an aircraft idling that can elapse before it is removed. 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. Defaults to 480 (seconds = 8 minutes)
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	When present, the airfield closest to this trigger zone is added to either the inclusion or exclusion set. Additional keywords are supported:

Name	Description
	• 'exclude' 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. • 'departure' or 'depart only' In this case, the airfield is added to the inclusion set only when looking for airfields to depart from • 'arrival' or 'arrive only' In this case, the airfield is added to the inclusion set only when looking for an airfield to land on. 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 Note also that only functioning airfields are chosen. FARPS or ships are disregarded. **MANDATORY**

8.4 Demos

- Virgin (Civ) Air
- Air Caucasus II
- One-Way Air

9 Clone Zone

9.1 Summary

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

9.2 Dependencies

dcsCommon, cfxMX, cfxZones

9.3 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 a 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. When the clone zone is using *linkedUnit* (moving DML zone, a Core Attribute of all DML zones) in conjunction with *useHeading* (another Core Attribute), the entire route is also rotated to coincide with the linked master unit's heading differential to its original ME heading. 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.

Name	Description
	Defaults to false (**no spawn on start** of mission). **To spawn units at mission start, set this attribute to true.**
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. Defaults to <none, retain template ownership>
spawn? f? in? clone?	**DML Watchflag**. 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!	The value of this flag is changed according to method when all units from the last spawn have been destroyed, including all static objects.
method cloneMethod	Standard DML method for output flags. Defaults to "inc"
deSpawn? deClone? wipe?	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:	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.
randomizedLoc rndLoc	Upon cloning, each cloned unit/object is displaced in a random direction by a random distance from 0 to the target zone's radius. Defaults to false
rndHeading	Upon cloning, each cloned unit/object assumes a random heading. Defaults to false
triggerMethod cloneTriggerMethod	Watchflag method for input flags. Defaults to 'change'
onRoad	Set to true to have any cloned unit deploy on the nearest road to the location it would spawn otherwise. This is very likely to change the template's formation, and can be a long way from the clone zone's intended spawn location, so use carefully. May also have other effects, such as bunching up the spawned units in unrealistically close proximity to each other. Defaults to false

Name	Description
wholeGroups	A modifier for the rndHeading, onRoad and rndLoc attributes. When set to true the clones are affected as follows: • rndLoc - randomizes the location of unit 1 of the group, and keeps all other vehicles in formation relative to that group. In fact, it only randomizes the group's position, not individual units • onRoad - now only the first unit is placed on the center of a road, all other units remain in location • rndHeading - with centerOnly, all units rotate randomly around unit 1's location • Defaults to false (all units individually)

Supports DML flags
Supports Watchflags
Supports zone-local verbosity

9.4 Demos

- Attack of the CloneZ
- Once, twice, three times a maybe
- Clone Relations
- Frog Men Training
- Flag Fun
- Track This!
- Gate and Switch
- Reinforcements A La Carte
- Forever-looping Spawners
- Recon Mode – Reloaded
- Willie Nillie
- BFM Combat Trainer

10 Count Down

10.1 Summary

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

10.2 Dependencies

dcsCommon, cfxZones

10.3 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 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>

10.4 Demos

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

11 Counter

11.1 Summary

Counter simply changes the output as often as it triggers on the input. This can be a simple counter, but if you choose the correct input and output methods, this can be pure magic.

11.2 Dependencies

Counter requires dcsCommon, cfxZones.

11.3 ME Integration

Name	Description
count	Tells DML that this is a counter that is triggered by this input **MANDATORY**
triggerMethod triggerCountMethod	Watchflag – the condition that triggers the input. Defaults to “change”
method countMethod	Output method. Defaults to “+1”
out! countOut!	The output to change when the counter is triggered. Defaults to <none>

Supports DML Flags
Supports Watchflags
Supports zone-local verbosity

11.4 Demos

- Formation Trainer

12 CSAR Manager

12.1 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.

12.2 Dependencies

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

Optional: cfxCommander (when using CSAR Zones)

12.3 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 based on this Zone's attributes. Defaults to <none>
score	Number of points to award when rescued successfully. Only relevant when PlayerScore is installed. If not present, csarManager's default score is used.

12.4 Demos

- CSAR of Georgia

13 Delay Flags (“Timer”)

13.1 Summary

Change a flag on the output side after a timer runs down. Can be paused, continued, reset.

13.2 Dependencies

dcsCommon, cfxZones

13.3 ME Integration

Name	Description
timeDelay	Marks this as a delayFlag/Timer 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 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?	Watchflag that stops a running delay. Has no effect on a stopped delay
pauseDelay?	Watchflag that when triggered pauses the delay to be continued later. Has no effect when triggered when the timer isn’t running or is already paused. Defaults to <none>
continueDelay?	Watchflag that when triggered continues a paused delay. Has no effect when triggered when the timer isn’t running or not paused. Defaults to <none>
delayLeft	Flag that carries the number of seconds left on the timer (including when delay timer is paused). Carries -1 when timer is not running.

Supports DML flags
Supports Watchflags
Supports zone-local verbosity

13.4 Demos

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

14 Delicates

14.1 Summary

Makes units/objects/cargos brittle: they immediately explode when they receive some damage.

14.2 Dependencies

Delicates requires dcsCommon and cfxZones.

14.3 ME Integration

Name	Description
delicates	Marks all static objects and units that are inside this zone when the mission starts up as 'delicate'. If they get damaged at all, they immediately explode. Note that if these units move later outside this zone, they remain delicates, while units that move inside this zone later do NOT become delicates. Value of this attribute is ignored **MANDATORY**
power	Strength of the explosion when the delicate object is triggered. Defaults to 10
f! out! delicatesHit!	Flag to bang! when one of the units/objects defined by this zone explodes. Defaults to <none>
method delicatesMethod	DML Method for output flags Defaults to 'inc'
remove	When set to true, the delicate object/unit is removed from the game when it explodes. Defaults to true
triggerMethod delicateTriggerMethod	Watchflag method for all inputs
blowAll?	Sending a signal on this input causes all surviving delicates defined by this zone to blow up. When you feed back the delicatesHit! Signal into this input, one hit kills all.
safetyMargin	A number that defines how much damage the object can sustain (relative to its initial life) before it explodes. Expressed as a fraction from 0 (0%) to 1 (100%). For example, 0.1 means that the object can sustain 10% damage before it explodes. Defaults to 0 (any damage makes it blow up)

Supports DML Flags
Supports Watchflags
Supports zone-local verbosity

14.4 Demos

- Delicate Subjects

15 FARP Zones

15.1 Summary

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

15.2 Dependencies

dcsCommon, cfxZones

15.3 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.

15.4 Demos

- FARP and away

16 FireFX

16.1 Summary

This module is similar to ‘smoke zone’, except that it creates a fire/black smoke effect that can be turned off much quicker and that can be controlled in size and visual intensity.

16.2 Dependencies

fireFx requires dcsCommon, cfxZones.

16.3 ME Integration

Name	Description
fireFX	Tells DML that you want a fire/smoke effect at the center of the zone. The value of this field specifies the size of the effect. Currently the following values are recognized: • “small” or “S” • “medium” or “M” • “large” or “L” • “Huge” or “H” or “XL” Smoke is always colored black. Note that this effect is visual only. Vehicles or troops inside the fire aren’t damaged by the fire at all. Defaults to “small” (if none of the given values is recognized) **MANDATORY**
flames	If you supply “false” for this attribute, only smoke (no fire) is displayed. Defaults to true (flames and smoke)
density	“Thickness” or visibility” of the smoke produced by this effect.
start?	DML watchflag (input) for when the effect should start. Defaults to <none>
stop?	DML watchflag (input) for then the effect should stop (Note: unlike the smoke zone module, which can take several minutes for the smoke to stop, this fireFX’s smoke stops within seconds) Defaults to <none>
onStart	If set to true, the mission starts with the effect on Defaults to false (no effect on start)
triggerMethod fxTriggerMethod	DML method for inputs Defaults to change

Supports Watchflags
Supports zone-local verbosity

16.4 Demos

17 Group Tracker

17.1 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.

17.2 Dependencies

dcsCommon, cfxZones

17.3 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:	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. If you have stacked the tracker on the same zone, you can use a single asterisk '*' as zone name. Supports a comma-separated list of trackers if you simultaneously want to pass the groups to multiple trackers, e.g. "GroundTrack, HeloTrack" This is useful if the zone contains more than one group, and your trackers use filtering Add all groups that have at least one unit in this zone to the tracker whose zone name is given in the Value field.

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

Name	Description
	Defaults to <none>
numGroups!	The value of this flag always represents the number of groups currently watched by this tracker. **This value is updated 1/ups times per second.** Defaults to <none>
numUnits	The value of this flag always represents the total number of units currently watched by this tracker. **This value is updated 1/ups times per second**. Defaults to <none>
groupFilter	Which unit categories to track. If no attribute is given, **all** categories are tracked. When you supply a groupFilter attribute**, only that category is accepted** when attempting to add to a tracker. Currently supported are • 0 (zero) or “aircraft” or “air” • 1 or “helo” or “heli” or “helicopter” • 2 or “ground” • 3 or “ship” • 4 or “train” Defaults to no filtering
triggerMethod trackerTriggerMethod	Watchflag method for inputs Defaults to ‘change’
destroy?	Watchflag that when triggered destroys all groups that are currently being watched. If any groups are destroyed, the removeGroup output is increased by the number of groups that were removed. numGroup is set to 0 Defaults to <none>
method trackerMethod	Method to bang! on output flags Defaults to “inc”
allGone!	Flag to bang! when the number of tracked groups falls to zero. If it was zero before, no output signal is generated. Defaults to <none>

17.4 Demos

- Track This!
- Moving Spawners II
- Impossible Impostors

18 Guardian Angel

18.1 Summary

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

18.2 Dependencies

Guardian Angel requires dcsCommon and cfxZones

18.3 ME Integration

Guardian Angel uses a config zone “guardianAngelConfig“ to control all settings

Name	Description
verbose	A value of “true” turns on debugging messages. Default is “false”
autoAddPlayer	When set to true, player planes are automatically added to Guardian Angel’s watchlist. Default is true
launchWarning	If true, Guardian Angel announces a missile launch. Default is true
intervention	If true, Guardian Angel destroys a missile before it destroys a watched aircraft. Default is true
announcer	If set to false, Guardian Angel suppresses all announcements. Defaults to true
private	If set to true, all announcements are only made to the group that a missile was fired at. Set to false (everyone can see)
launchSound	Name of the sound file to be played when a missile is launched. Respects ‘private’ attribute
interventionSound	Name of the sound file to be played when Guadian Angel saves an aircraft. Respects the ‘private’ attribute.
explosion	Guardian Angel can add a mostly harmless explosion when a missile is removed due to an intervention. If this value is smaller than one (e.g., -1) this feature is turned off. If you enter a value > 0 (zero), an explosion with a magnitude of this value is placed in direction of that missile’s last location, 500m from the aircraft. A mostly harmless value is 1.0 (one point zero) **WARNING I** Even though this explosion is usually harmless for the protected plane, it can pose lethal to any other plane (wingmen). **WARNING II** The explosive effect is only harmless to the protected plane if the explosion value is small (e.g., 1). If you enter sufficiently larger values, the shock wave can destroy even the protected plane. If you set this value to see explosions, make the value 1.0

Name	Description
	Defaults to -1 (off)
fxDistance	When using explosions, this is the distance (in meters) away from the aircraft where the (real) explosion is going to take place. Defaults to 500
active	The state that Guardian Angel starts up in. True means that it is active, false that it is turned off. Defaults to 'false'
activate? on?	Watchflag to turn on (activate) guardian angel. Defaults to <none>
deactivate? off?	Watchflag to turn off (deactivate) guardian angel. Defaults to <none>

You can selectively add and remove aircraft from protection by Guardian Angel by placing them in a zone with the "guardian" attribute:

Name	Description
guardian	**MANDATORY** Tells Guardian Angel how to treat aircraft inside the trigger zone: • *true* All aircraft inside this zone are protected by Guardian Angel • *false* All aircraft inside this zone will not receive protection from Guardian Angel. Defaults to 'true' (all aircraft inside are protected)

18.4 Demos

- Missile Evasion (Guardian Angel)
- Guardian Angel Reloaded

19 Helo Troops

19.1 Summary

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

19.2 Dependencies

Tbc

19.3 ME Integration

Tbc

19.4 Demos

- Helo Trooper

20 Impostors

20.1 Summary

Impostors allow you to switch AI-controlled units to their static object eqivalents (“impostor”). Impostors will not react to enemies, and consume less CPU. You can use flags to control if a group is in impostor or controlled state.

20.2 Dependencies

Impostors require dcsCommon, cfxMX and cfxZones.

20.3 ME Integration

Name	Description
impostor?	Marks all groups that have at least one unit inside this zone as potential impostor, giving them the ability to change between static object and AI-controlled unit. The value of this attribute is a Watchflag that triggers transition of all surviving units from AI-controlled to static object. **MANDATORY**
reanimate?	Watchflag that triggers transition of all surviving static units to AI-controlled units. They immediately restart any waypoint actions or route orders (if given). Ground units will start moving their first (not initial) waypoint.
triggerMethod impostorTriggerMethod	Method that triggers inputs (DML Watchflags) Defaults to “change”
onStart	If the value of this attribute is true, all units are turned into imposters at the start of the mission. Default is false
blink	The value of this attribute specifies the brief interval (in seconds) between removing the impostor and spawning of the AI-controlled units. Only required for AI aircraft. A good value is 0.1 – 0.2; a value of zero or negative value means no blinking. Deafults to -1 (no blink interval)
trackWith:	Name of a zone with a groupTracker attached. All units are added to that tracker when they are AI-controlled, and removed when they are turned into impostors. Reanimating them subsequently will again add them to the tracker etc.
allDead!	DML flag to bang! when all groups that are managed by this impostor zone have been destroyed
method impostorMethod	DML method for output Defaults to “inc”

Supports DML Flags
Supports Watchflags
Supports zone-local verbosity

20.4 Demos

- Impossible Impostors

21 Limited Airframes

21.1 Summary

Set a maximum per side on the number of player pilots that can be lost.

21.2 Dependencies

dcsCommon, cfxZones, cfxPlayer

21.3 ME Integration

Most features of Limited Airframes are controlled via config zones:

Name	Description
verbose	A value of “true” turns on debugging messages. Default is “false”
enabled	Controls whether or not Limite Airfames is in effect. Defaults to “true”
userCanToggle	Controls whether players can turn Limited Airframes on and off during the mission. Defaults to “true”
maxRed	Maximum (and starting) number of pilots for the red coalition. Set to -1 to make the number unlimited. Defaults to -1 (unlimited)
maxBlue	Maximum (and starting) number of pilots for the red coalition. Set to -1 to make the number unlimited. Defaults to -1 (unlimited)
#red	Flag that continuously is set to the current number of pilots remaining for Red
#blue	Flag that continuously is set to the current number of pilots remaining for blue
redWins! redWinsFlag!	Flag to bang! when blue has lost all pilots and red wins. Defaults to <none>
blueWins! blueWinsFlag!	Flag to bang! when red has lost all pilots and blue wins. Defaults to <none>
method	DML bang! method. Defaults to ‘inc’
warningSound	Name of sound file to play when limited airframes is displaying a message. Defaults to <none>
winSound	Name of sound file to play for winning side when other side has lost all pilots. Defaults to <none>
loseSound	Name of sound file to play for the side that has lost all pilots and therefore lost the engagement. Defaults to <none>
announcer	When set to false, the are no announcements for change on air frames, Defaults to true (airframe changes are announced)

On the map, you should place safe zones (one per side that has limits on player pilots) using the following attributes

Name	Description
pilotsafe	Marks this zone as safe for pilots to change into other airframes when landed. • If the value to this attribute contains neither the word ‘red’ nor ‘blue’, it is safe for all coalitions

Name	Description
	• If the value of this attribute contains the word ‘red’ this zone is safe for the red coalition • If the value of this attribute contains the word ‘blue’ this zone is safe for the blue coalition Note that safe state may be contingent on ownership of the zone. **MANDATORY**

21.4 Demos

- Pilots at their limit

22 LZ

22.1 Summary

A module that can create trigger events for landings and take-offs inside the zone

22.2 Dependencies

dcsCommon, cfxZones

22.3 ME Integration

Name	Description
LZ	Tells DML that this is an LZ that detects landing and take-off of aircraft inside it. The value of this attribute is ignored **MANDATORY**
landed!	Flags to bang! when a plane matching the criteria lands inside the zone Defaults to <none>
departed!	Flags to bang! when a plane matching the criteria lands inside the zone Defaults to <none>
coalition	When given, the coalition that aircraft must match. Supported are “red”, 1, “blue”, 2, “neutral” 0 When you specify “neutral” or 0, all coalitions match Defaults to 0 (any coalition)
playerOnly	When set to true, all AI planes are ignored Defaults to false (all players and AI are considered)
unit units	A comma-separated list of all unit names (case **in**sensitive) that should be considered. Supports wildcard “*” for the last character, in which case all groups that match that everything before the asterisk are considered: “Hog-*” will match “HOG-”, “hog-1-1” and “HoG-5-2”, but not “Hogger” When matching units, any ‘coalition’ setting is ignored Defaults to <none>
group groups	A comma-separated list of all group names (case **in**sensitive) that should be considered. Supports wildcard “*” for the last character, in which case all groups that match that everything before the asterisk are considered: “He*” will match “He”, “HELLO” and “heinkel-1-1”, but not “Hans Heinkel” Defaults to <none>
type types	List of types, separated by comma, that a unit must match at least one of (e.g. “A-10A”). Types must match exactly Additionally, the following special types are also supported • “ALL” or “ANY” – all aircraft match • “HELO” – all helicopters match • “PLANE” – all fixed-wing planes match

Name	Description
	Defaults to “ALL”
isPaused	When set to true, the LZ is paused at mission start Defaults to false (LZ is active on start)
pause?	DML Watchflag to pause the LZ Defaults to <none>
continue?	DML Watchflag to continue a paused LZ Defaults to <none>
method outputMethod	Method to the outputs Defaults to ‘inc’
triggerMethod lzTriggerMethod	Method that triggers inputs. Defaults to “change”

Supports DML Flags
Supports Watchflags
Supports zone-local verbosity

22.4 Demos

- Departures and Landings

23 Map Markers

23.1 Summary

Allows you to place markers on the F10 map.

23.2 Dependencies

dcsCommon, cfxZones

23.3 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”

23.4 Demos

24 Messenger

24.1 Summary

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

24.2 Dependencies

dcsCommon, cfxZones

24.3 ME Integration

Name	Description
messenger? **messenger**	Watchflag. When triggered, the module will display the message and/or play sound. **MANDATORY**
message	The text of the message to be displayed **FORMATTING WILDCARDS** • <n> creates a new line • <z> is replaced with zone's name • <t> is current time in the format as defined with the timeFormat attribute **RESPONSE-SELECTION WILDCARDS** • <rsp: flag name> looks up the value of <flag name> and uses that value and an offset into the responses given with the 'responses' attribute. The first response has an index of 1. If the flag's value is less than 1, the first response is returned, if the value is higher than the number of responses, the last response is returned. • <rrnd> randomly selects one of the possible responses and returns that response • <rhdg: unit/zone> wraps responses around the compass, and then uses the unit/zone's heading as offset. Zones only have a heading if they are linked to a unit. [not yet implemented] • <rbea: unit/zone> selects and returns one of possible responses using the bearing (in degrees) from unit/group to unit/zone as offset into *responses.* [requires group or unit attribute be set] **DATA ACCESS WILDCARDS** • <lat> the latitude of the zone's current position • <lon> the longitude of the zone's current position • <mgrs> the zone's current position in MGRS coordinates • <v: flagName> is replaced with the value currently held by the flag *flagName* • <t: flag name> uses the value from flag *flag name* and interprets it as a time value, formatted according to the timeFormat attribute

	• <lat: unit/zone name> outputs the latitude of the zone or unit that matches the name *unit/zone name*, or “messageError” (usually an empty string) if neither can be found • <lon: unit/zone> outputs the longitude of the zone or unit that matches the name *unit/zone name*, or “messageError” (usually an empty string) if neither can be found • <ele: flagName> outputs the elevation of the zone or unit that matches the name *unit/zone name*, or “messageError” (usually an empty string) if neither can be found. Elevation is calculated in meters (default) or feet if the “imperial” attrigute is true • <latlon: unit/zone> outputs the latitude and longitude of the zone or unit that matches the name *unit/zone name*, or “messageError” (usually an empty string) if neither can be found • <lle: unit/zone> outputs the longitude, longitude and elevation of the zone or unit that matches the name *unit/zone name*, or “messageError” (usually an empty string) if neither can be found • <mgrs.: flagName> outputs the mgrs. coordinates of the zone or unit that matches the name *unit/zone name*, or “messageError” (usually an empty string) if neither can be found • <vel : unit/zone> outputs the velocity (in km/h or knots, depending on imperialUnits) of unit. Zones only have a velocity if they are linked to a master unit • <hdg: unit/zone> outputs the direction that the unit/zone referenced is heading. Zones only have a heading if it is linked to a master unit, and the heading then returned is the one of the linked unit. • <alt: unit/zone> outputs the altitude (barometric) of the unit/zone. If the zone is unlinked, it returns the altitude of the land at the zone’s center. If the zone is linked to a unit it returns the altitude of that unit. Altitude is returned in meters or feet, depending on the imperialUnits attribute. • <type: unit/zone> returns the type of the unit (e.g., “A-10C”). For zones, the type is always “Zone” **UNIT/GROUP RELATIVE (requires group or unit)** These wildcards are only available when you have added a ‘group’ or ‘unit’ attribute (i.e., the messages are sent only to specific player groups or units). If you give the ‘group’ attribute, all information is relative to the group’s lead (unit 1): • <bea: units/zone> bearing (in degrees) of the player group/unit to the unit/zone specified • <rng: unit/zone> range (distance) from the player group/unit to the unit/zone given • <clk: unit/zone> direction as “o’clock” to the unit/zone as seen from the player unit/zone’s heading (12 is straight ahead, 6 is behind) • <hnd: unit/zone> direction “which hand” to the unit/zone: “ahead”, “right”, behind”, “left” as seen from the unit/group • <sde: unit/zone> direction “side” to the unit/zone as seen from the unit//group: “ahead”, “starboard”, “aft”, “port”

	• <asp: unit/zone> aspect of unit/zone towards unit/group. Returns “hot” / “beam” / “drag” • <cls: unit/zone> closing velocity of unit/group with unit/zone. A negative closing velocity means that distance is growing. Closing velocity is given in km/h or knots, depending on imperialUnits • <pcls: unit/zone> precision closing velocity of unit/group with unit/zone. Closing velocity is given in m/s or ft/s with up to one decimal, e.g., “1.3”
responses	A list of comma-separated possible responses that can be accessed by various wildcards. Note that the responses themselves must not contain a comma “,”. Example: “good, better, the best” is treated as three separate possible responses: “good”, “better” and “the best”. Defaults to <none>
triggerMethod msgTriggerMethod	Defines the trigger condition for DML Watchflags. Use only one synonym per zone Defaults to “change”
clearScreen	If true, erase all existing messages. Defaults to 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 should receive the message/sound. If no coalition is given, text and sound are played to all. Legal values are “red”, “blue”, “neutral”, 0, 1, 2 Note that if given, the attributes ‘coalition’, ‘group’ and ‘unit’ are mutually exclusive. Defaults to <none>
group msgGroup	The name of the Group that should receive the message/sound. Adding a group attribute enables unit-relative wildcards. Note that if given, the attributes ‘coalition’, ‘group’ and ‘unit’ are mutually exclusive. Defaults to <none>
unit msgUnit	The name of the Unit that should receive the message/sound. Adding a group attribute enables unit-relative wildcards. Note that if given, the attributes ‘coalition’, ‘group’ and ‘unit’ are mutually exclusive. Defaults to <none>
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 their messageOff input to disable. Defaults to <none>
messageOff?	When the value of this flag changes, the messenger is turned off. Any further messages are suppressed. If the messenger was already turned off, nothing happens. Defaults to <none>

mute messageMute	If set to true, the messenger starts muted and requires a signal on messageOn? To activate. Defaults to false
duration messageDuration	Time (in seconds) how long a message should stay on-screen Defaults to 30 seconds
timeFormat	Time format for any time values Defaults to “<:h>:<:m>:<:s>” – standard 24 hour time
imperial imperialUnits	When true, elevation is calculated in feet (imperial units) else meters. Defaults to false (meters)
error messageError	The text to substitute for a unit or zone reference if the units or zone cannot be found. Defaults to “” (empty string)

Supports DML Flags
Supports DML Watchflags
Supports zone-local verbosity

24.4 Demos

- Bottled Messages
- The Zonal Countdown
- Frog Men Training
- Follow Me!
- CSAR of Georgia
- Track This!
- Watchflag Demo
- Radio Go-Go
- xFlags Field Day
- Count Base’s Blues
- Gate and Switch
- Moving Spawners II
- Reinforcements A La Carte
- **Formation Trainer**

25 NDB

25.1 Summary

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

25.2 Dependencies

dcsCommon, cfxZones.

25.3 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?	Watchflag. **Each time the flag triggers, 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?	Watcheflag. **Each time the flag triggers, the NDB is stopped.** paused value is set to true after the NDB has stopped. Defaults to no flag to watch
triggerMethod ndbTriggerMethod	Defines the trigger condition for DML Watchflags. Use only one synonym per zone Defaults to "change"

25.4 Demos

- ADF and NDB Fun

26 Object Destruct Detector

26.1 Summary

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

26.2 Dependencies

dcsCommon, cfxZones

26.3 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**
method oddMethod	DML Method for output flags. Defaults to “inc”
f! destroyed! objectDestroyed!	The flag to bang! when the object is destroyed. Use only one synonym per zone. Defaults to ”*none”

26.4 Demos

- Object Destruct Detection

27 Object Spawn Zones

27.1 Summary

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

27.2 Dependencies

dcsCommon, cfxZones, (cfxCargoManager)

27.3 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
useDelicates	Name of Delictates Zone that is used to assign delicate status when this spawner spawns objects

27.4 Demos

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

28 Owned Zone

28.1 Summary

Zones that can be conquered and generate a signal when conquered

28.2 Dependencies

dcsCommon, cfxZones, (cfxGroundTroops)

28.3 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**
conquered!	Flag to bang! when this zone changes hands 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”
redCap!	Flag to bang! when red side conquers the zone Defaults to <none>
blueCap!	Flag to bang when blue side conquers the zone. Defaults to <none>

28.4 Demos

- Owned Zones ME Integration

29 Persistence

29.1 Summary

This module provides persistence (load and save) functionality to modules. Must load before any mode that is to use persistence. Requires that the DCS instance that is running the mission be de-sanitized for lfs and io.

29.2 Dependencies

Persistence requires dcsCommon and cfxZones.

29.3 ME Integration

You configure the persistence module with a Trigger Zone named ‘persistenceConfig’

Name	Description
verbose	A value of “true” turns on debugging messages. Default is “false”
versionID	If present, this turns on version matching. When a mission starts up, persistence checks the value provided via the Zone with the one saved. If they do not match, the entire save data is discarded, and the mission starts fresh Defaults to <none>
root	Path to the DCS standard directory (usually “C:\userName\saved games\DCS.openbeta\” or “C:\userName\saved games\DCS\”). This value is passed from DCS to persistence. You can change this to adapt your missions to conform with more elaborate server setups. If you change this. Be sure that you know what you are doing, and initially have verbosity set to true, so you can see which directory your mission will save to. Defaults to your currently running DCS instance’s write dir.
serverDir	Path from the root directory (see above) to the Missions directory. Use this if you set up your DCS different (usually important for dedicated servers). Defaults to “Missions\”
saveDir	Name for the mission’s data directory. Defaults to “<mission name> (data)”. This directory is created in the serverDir automatically if it does not exist If you set saveDir to “”, the mission saves its data directly into serverDir Defaults to “<mission name> (data)” if a configuration zone is present, none without configuration zone (i.e. the data is written into serverDir)
saveFileName	Name for the file that persistence uses to write mission data. Defaults to “<mission name> Data.txt”
saveInterval	Controls auto-save. Any value larger than zero will turn on auto save. The value you give here is the number of minutes between auto saves. Auto-saves co-operate with manual saves, so you can use both methods in your mission

Name	Description
	Defaults to -1 (auto-save off)
cleanRestart?	DML Watchflag. A change signal on this input triggers a “fresh start” request: next time the mission starts up, it won’t load mission data. Defaults to <none>
saveMission?	DML Watchflag. A change signal on this input triggers a ‘manual’ save. Defaults to <none>

You control where persistence saves your mission’s data with the “root”, “serverDir”, “saveDir” and “saveFileName” attributes. Assuming a mission named “coolMission.miz” running from a standard DCS Install, persistence saves data as follows:

In a fully defaulted configuration (i.e. out-of-the-box), persistence uses the following defaults:

attribute	Default
root	The directory that you configured DCS to be the ‘home’ directory. In a freshly DCS install that would usually be C:\Users\(user name)\Saved Games\DCS\ Defaults to what DCS tells persistence is the current home directory
serverDir	The directory name inside “root” that contains all missions. It’s usually called “Missions\” and that is what persistence defaults to
saveDir	A folder (allocated if it doesn’t exist) inside serverDir where persistence saves the mission data as a separate file. You can use this to pool multiple missions’ data into the same folder. Defaults to “<mission name> (Data)” (see note below) **IMPORTANT NOTE** If you completely omit persistence’s config zone, it reverts to simplified save mode, and defaults saveDir to “” (empty string), saving the mission’s data file directly into the serverDir.
saveFileName	The name for the data (plain text in JSON format, can be edited with any text editor) file inside saveDir. Defaults to “<mission name> Data.txt”

You can use Trigger Zones with the ‘saveFlags’ attribute to list the flags that persistence should save.

Name	Description
saveFlags	A list of flags that you want to be saved with the mission. Supports local flags (e.g., “*go”) and numbered ranges (e.g. “3-17”). **MANDATORY**

29.4 Demos

- Being persistent

30 Player Score

30.1 Summary

A module to keep award and tabulate kill score for players

30.2 Dependencies

Tbc

30.3 ME Integration

Tbc

30.4 Demos

- Keeping The score
- More Score

31 Pulse Flags

31.1 Summary

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

31.2 Dependencies

dcsCommon, cfxZones

31.3 ME Integration

Name	Description
pulse!	Marks this as a pulser. The value describes the flags to change on each pulse. The flags are changed according to the method attribute **MANDATORY**
method pulseMethod outputMethod	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 *pulses* 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 a trigger on activate? flag.~~ ~~Defaults to false~~
onStart	When set to false, a pulser starts paused, else active. **Defaults to true** (pulser starts automatically)
pulses	The number of pulses to complete. • If set to -1, the pulser runs until the mission ends or the pause?-flag changes • If set to a number, the pulser will generate that many pulses. • If set to a range (e.g. “3-5”) the pulser will generate a random number of pulses within that range. Defaults to -1 (endless)

Name	Description
time pulseInterval	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)

Supports DML Flags
Support Watchflags
Supports zone-local verbosity

31.4 Demos

- Pulsing Fun
- Frog Men Training
- The Zonal Countdown
- Watchflag Demo
- Forever-looping Spawners

32 Radio Menu

32.1 Summary

Adds a configurable player menu to the Communications→F10 Other menu, with up to four menu items (commands). Supports cooldown.

32.2 Dependencies

Radio Menu requires dcsCommon and cfxZones.

32.3 ME Integration

Name	Description
radioMenu	Name of the menu to install in the Communications→F10 Other menu. **MANDATORY**
coalition	The coalition that has access to this menu. If omitted or set to 'neutral', *all* coalitions have access. 'blue' or 'red' restricts access to that coalition. Defaults to <no coalition>
group groups	Restricts this menu only to the groups of that name. Supports comma separated groups names, e.g., "Eagles 5, Uzi One, Aleph" will make this menu available to all members of those groups **Overrides any 'coalition' and 'type' attribute**; when you add a 'group' attribute, those other attributes are ignored. Defaults to <no group restriction> **NOTE:** requires module cfxMX to load before radioMenus
type types	Restricts access to this menu to player units who control a unit that matches one of the listed types. You can list multiple types, separated by a comma (e.g. "F-15C, A-10A"). Also takes into account information given with the coalition attribute. Supports the class-wildcards 'plane' (all fixed-wing players) and 'helo' (all rotor-wings controlled by players). When you also supply a 'coalition' attribute, access to this menu is restricted to those players who match both. Defaults to <no type restriction> **NOTE:** requires module cfxMX to load before radioMenus
itemA	Name of itemA in this menu. If this attribute is omitted, no menu item appears

Name	Description
itemB	Name of itemB in this menu. If this attribute is omitted, no menu item appears
itemC	Name of itemC in this menu. If this attribute is omitted, no menu item appears
itemD	Name of itemD in this menu. If this attribute is omitted, no menu item appears
A!	DML flag to bang when itemA is chosen. Defaults to <none>
B!	DML flag to bang when itemB is chosen. Defaults to <none>
C!	DML flag to bang when itemC is chosen. Defaults to <none>
D!	DML flag to bang when itemD is chosen. Defaults to <none>
cooldownA	Cooldown (in seconds) after itemA is chosen before it becomes available again. Defaults to 0 (immediately available again)
cooldownB	Cooldown (in seconds) after itemA is chosen before it becomes available again. Defaults to 0 (immediately available again)
cooldownC	Cooldown (in seconds) after itemA is chosen before it becomes available again. Defaults to 0 (immediately available again)
cooldownD	Cooldown (in seconds) after itemA is chosen before it becomes available again. Defaults to 0 (immediately available again)
busyA	Message to display when itemA is chosen while cooldown is still active. Defaults to "Please stand by (<s> seconds)". Supports Time Wildcards <s>, <m>, <h>, <:s>, <:m> and <:h>
busyB	Message to display when itemB is chosen while cooldown is still active. Defaults to "Please stand by (<s> seconds)". Supports Time Wildcards <s>, <m>, <h>, <:s>, <:m> and <:h>
busyC	Message to display when itemC is chosen while cooldown is still active. Defaults to "Please stand by (<s> seconds)". Supports Time Wildcards <s>, <m>, <h>, <:s>, <:m> and <:h>
busyD	Message to display when itemD is chosen while cooldown is still active. Defaults to "Please stand by (<s> seconds)". Supports Time Wildcards <s>, <m>, <h>, <:s>, <:m> and <:h>
method radioMethod	DML method to bang flags. Defaults to ‘inc’, meaning that each time that a menu item is chosen, the flag’s number is increased, generating a signal. To emulate ME’s native menu method that sets a flag to a value <v>, use that number <v> as method
radioTriggerMethod	Watchflag method for inputs Defaults to “change”
removeMenu?	Watchflag that triggers removal of entire menu
addMenu?	Watchflag that triggers adding the menu if it wasn’t shown or removed previously
menuVisible	When set (as per default) the menu is shown at the start of the mission. When set to false, the mission starts up with the menu hidden and requires a signal on addMenu? to appear Default to true (menu is visible on mission start)

Supports DML Flags
Supports Watchflags
Supports zone-local verbose

Use the following wildcards (note that they are identical to messenger’s ‘display as time wildcards) in busyX:

- <s>
  remaining cooldown in seconds. E.g., if remaining cooldown is 254, <s> is replaced with ‘254’

- <m>
  remaining cooldown as whole minutes. E.g., if remaining cooldown is 254, <m> is replaced with ‘4’, while 3891 returns “64”

- <h>
  remaining cooldown as whole hours. E.g., if remaining cooldown is 3891, <m> is replaced with ‘1’

- <:s>
  remaining cooldown converted to a seconds time value (0-60) and formatted with leading zero. E.g., if remaining cooldown is 64, <:s> is replaced with ‘04’

- <:m>
  remaining cooldown converted to a minutes time value (0-60) and formatted with leading zero. E.g., remaining cooldown is 64, <:m> is replaced with ‘01’, and 803 returns “13”

- <:h>
  remaining cooldown converted to an hours time value and formatted with leading zero. E.g., if remaining cooldown is 3764, <:h> is replaced with ‘01’

32.4 Demos

- Reinforcements A La Carte
- Recon Mode reloaded
- Guardian Angel reloaded
- Sequencing Fun
- Slot Blocking and You
- BFM Combat Trainer

33 Radio Trigger

33.1 Summary

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

33.2 Dependencies

dcsCommon, cfxZones

33.3 ME Integration

Name	Description
radio?	Watchflag. Triggers a radio cycle, then resets this flag to its pervious value **MANDATORY**
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

33.4 Demos

- Radio go go

34 Raise Flag

34.1 Summary

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

34.2 Dependencies

dcsCommon, cfxZones

34.3 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	Method or value to set the flag to. Supports • `'inc'` increment the flag's current value by one • `'dec'` decrements the flag's current value by one • `'flip'` sets the flag's current value to 0 if it was anything but zero, and to 1 if it was zero. • (number) – set the flag to this number 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. **If this attribute is omitted, the flag is set 0.5 seconds after the mission starts.**
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

34.4 Demos

- Flag Fun
- Attack of the CloneZ

35 Recon Mode

35.1 Summary

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

35.2 Dependencies

Tbc

35.3 ME Integration

Zone to make aircraft scouts or remove scout abilities:

Name	Description
scout	Marks all aircraft (fixed- and rotor-wing) inside the zone. If the attribute's value is 'true', the aircraft have recon ability. If the value is false, they are 'blind', i.e. they have no recon abilities. Defaults to 'true' **MANDATORY**
dynamic	Controls if all units that start with the same name are automatically included. This is helpful for clone zones that base all names for clones on the name of the unit in the template. Defaults to false

Zone to mark priority and blacklisted ground forces:

Name	Description
recon	Marks all ground groups inside this zone as recon relevant. If the value is "black" all groups that have at least one inside this zone are added to the blacklist, otherwise they are added to the priority target list Defaults to "prio" **MANDATORY**
dynamic	Controls if all groups that start with the same name are automatically included. This is helpful for clone zones that base all names for clones on the name of the unit in the template. Defaults to true
prioMessage	A message that is displayed to the coalition that the scout belongs to. Can contain most of the text wildcards that the messenger module provides: • <n> creates a new line • <z> is replaced with zone's name • <t> is current time in HH:MM:SS format • <lat> the latitude of the discovered group's current position • <lon> the longitude of the discovered group's current position • <ele> the elevation (in feet or meters, as determined by the imperialUnits attribute in reconModeConfig) • <mgrs> the discovered group's current position in MGRS coordinates

Name	Description
	Only applicable for priority targets (i.e. recon's value is something other than "black"), ignored otherwise Default is <none>, i.e.., no message is displayed
spotted!	DML output flags to bang when a group is spotted. Flag is banged *in addition* to the module's global "prio!" flag. Uses the config zone's method. *Only applies to prio zones* (ignored when blacklist zone) Defaults to <none>
silent	Only applicable to priority groups. When present and set to true, the recon report and map mark are suppressed for any groups defined with this zone. Note that this does NOT apply to any prioMessage you have defined for this zone, as that will still be displayed. Defaults to false

Supports DML Flags

Config Zone Settings

Name	Description
verbose	A value of "true" turns on debugging messages. Default is "false"
autoRecon	If true, all planes are automatically treated as actively reconnoitering. **NOTE** 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
redScouts	If true, all red planes are included as scouts when autoRRecon is true. Default is false
blueScouts	If true, all blue planes are included as scouts when autoRecon is true. Default is **true**
greyScouts	If true, all neutral planes are included as scouts when autoRecon is true. Default is false
playerOnlyRecon	If true, only player aircraft are included as scouts when autoRecon is true. AI planes will not be automatically included as scouts. **IMPORTANT** This condition is applied **in addition** to blueScouts and redScouts. If you disallow red scouts, red players will not automatically be added to the list of scouts. Defaults to false
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)

Name	Description
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)
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
mgrs	Defines if the location of the group that is detected is given in Lat/Lon (default) or MGRS. Set to true to enable MGRS. Default is false (coordinates are displayed in Lat/Lon)
imperial imperialUnits	When set to true, the value given in <ele> (elevation) is in feet, otherwise in meters Defaults to false (ele is returned in meters)
activate? on?	Watchflag to monitor for a change. If a change is detected, Recon Mode goes into active state. Defaults to <none>
deactivate? off?	Watchflag to monitor for a change. If a change is detected, Recon Mode turns off Defaults to <none>
active	Set to false to start Recon Mode in disabled (off) state. Defaults to true (Recon Mode enabled)
reportTime	Number of seconds that a recon message stays on the screen. Defaults to 30.

Supports Watchflags
Supports DML Flags
Supports zone-local verbosity

35.4 Demos

- Recon Mode
- Recon Mode - Reloaded

36 rnd Flags

36.1 Summary

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

36.2 Dependencies

dcsCommon, cfxZones

36.3 ME Integration

Name	Description
RND!	Marks this as a randomizer. 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, samAttack, 11-11” **MANDATORY**
method rndMethod	Describes how flags are to be set. Follows standard DML flag method. **Defaults to “inc”**
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 **NOTE**: if no f? (or synonym) is specified, and onStart is false, the randomizer will never activate. You’ll receive a warning.

Name	Description
~~done+1~~ done! rndDone!	The flag number to bang! when the randomizer has run out of flags to change, and reshuffle is false (randomizer did nothing) Is banged! every time that the randomizer runs a cycle on an empty flag set

Supports DML Flags
Supports DML Watchflags
Supports zone-local verbosity

36.4 Demos

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

37 Sequencer

37.1 Summary

Creates signals in a pre-determined sequence of flags

37.2 Dependencies

Sequencer requires dcsCommon, cfxZones.

37.3 ME Integration

Name	Description
sequence!	A comma-separated list of flags that should be banged! in exactly that order. Example: “startGround, startSAM, startSirens” Value of this attribute is ignored **MANDATORY**
interval invervals	A comma-separated list of durations (in seconds) between the flag banging. The number of intervals does not have to match the number of flags given under sequence!, the values simply repeat. A single value means that the same interval is applied between all stages. You can supply value ranges instead of a single value, and the sequencer picks a random value from that range. Example: “10, 12-17, 33” Defaults to 86400 (24 hours)
next?	A signal on this input causes a running sequencer to end the current count-down to the next stage, and immediately start the next stage. Defaults to <none>
onStart	If set to true, the sequencer starts the first sequence 0.25 seconds into the mission (this delay allows all other modules to load and initialize before the first signal is sent from the sequence) Defaults to false (sequencer requires a startSeq? signal.
zeroSequence	Controls if the sequencer operates in “signal-wait” or “wait-signal” mode: if it starts with a signal (zeroSequence is true) or a wait (zeroSequence is false) Defaults to true (start with signal, “signal-wait” mode)
loop	If true, the sequence repeats endlessly or until stopped with ‘stopSeq?’ or ‘reset’. If false, the sequence ends after the last stage, and a signal is sent on done! Defaults to false
done! seqDone!	The signal to send when the sequencer has ended the sequence Defaults to <none>
startSeq?	A signal on this input starts a paused or un-started (when onStart is false) sequencer. If the sequence was paused, the count-down resumes at the moment it was paused.

Name	Description
	If the sequence is already running it has no effect. Note that a sequence that has ended (run through all stages) cannot be restarted with startSeq? but must be reset first. Defaults to <none>
stopSeq?	A signal on this input pauses a running sequence. The current state of the stage's timer is preserved. If the sequencer is already paused, this has no effect. Defaults to <none>
resetSeq	A signal on this input resets the sequencer to the initial state: paused (if onStart is false), sequence stage one, duration one. A sequence that has ended (run out of stages) can be reset and then started again. Defaults to <none>
method seqMethod	The method to use for all outputs. Defaults to 'inc'
triggerMethod seqTriggerMethod	The trigger method for all inputs. Defaults to "change"

Supports DML Flags
Supports Watchflags
Supports zone-local verbosity

37.4 Demos

- Sequencing Fun

38 Smoke Zones

38.1 Summary

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

38.2 Dependencies

dcsCommon, cfxZones

38.3 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?	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

38.4 Demos

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

39 Spawn Zones

39.1 Summary

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

39.2 Dependencies

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

39.3 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
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
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 spawner, you can use a single asterisk ‘*’ as zone name. Supports a comma-separated list of trackers if you simultaneously want to pass the spawned groups to multiple trackers

Name	Description
	Defaults to <None>

39.4 Demos

- Random Death
- Moving Spawners
- Moving Spawners II
- Helo Troops

40 ssbClient

REQUIRES SSB INBSTALLED ON HOSTING MP SERVER AND MISSION RUN AS MP

40.1 Summary

This module allows intelligent slot-blocking of aircraft based on airfield/FARP ownership, and single-use aircraft ("crash them and lose them").

40.2 Dependencies

ssbClient requires dcsCommon, cfxGroups and cfxZones

40.3 ME Integration

Most of ssbClient is controlled via the config zone that must be named "cfxSSBClientConfig"

Name	Description
verbose	A value of "true" turns on debugging messages. Default is "false"
singleUse	A value of "true" turns on single-use: an airframe is blocked after crashing it. Note that this requires that the server's SSB setup sets kickReset to false in SSB. Defaults to false
reUseAfter	If singleUse is **en**abled, this optional attribute controls after how long a delay (in seconds) the slot may be re-used. This can simulate replacements arriving after some time. Setting this value to -1 blocks the slot for the remainder of the mission. Defaults to -1 (remain blocked)
allowNeutralFields	If set to "true", aircraft can spawn on neutral airfields (otherwise they are blocked). Defaults to false (neutral fields do not allow blue nor red aircraft to spawn
maxAirfieldRange	Maximum range in meters to find an airfield/FARP for a 'from ground' start. If no airfield is found that slot will be permanently open. Defaults to 3000 meters
keepInAirGroups	For performance reasons, ssbClient strips all slots for air-starting aircraft from its observation list. In some cases (e.g. when you want to bind the availability of an air-starting aircraft slot to the ownership of an airfield) ssbClient must also manage air-starts. Set this value to true to also retain air-starting slots. Defaults to false
enabledFlagValue	This reflects SSB's flag value of that same name. **DO NOT CHANGE THIS UNLESS YOU ARE ABSOLUTERLY SURE YOU KNOW WHAT YOU ARE DOING**. Defaults to 0

Name	Description
enabledFlagValue	This reflects SSB's flag value of that same name. **DO NOT CHANGE THIS UNLESS YOU ARE ABSOLUTERLY SURE YOU KNOW WHAT YOU ARE DOING**. Defaults to enabledFlagValue + 100

Additionally, ssbClient supports ssbClient control zones that attach themselves to the closest airfield/FARP and can be used to open and close them with DML flags:

Name	Description
ssbClient	Marks this zone as a control zone **for the airfield/FARP that is closest** to this zone. **MANDATORY**
open?	DML Input Watchflag. When triggered, this airfield "opens", allowing planes spawn from this airfield. Note that ownership rules still apply. Defaults to <none>
close?	EML input Watchflag. When triggered, this airfield/FARP "closes". Aircraft can still land and depart from a closed airfield, but player aircraft that originate from there will no longer be able to spawn, their slots are blocked. Defaults to <none>
openOnStart	When set to false, the airfield is closed on mission start, all slots for aircraft spawning here are blocked Defaults to true (airfield is open, allowing slots to spawn according to faction ownership of airfield/FARP)
ssbTriggerMethod	Method for Watchflags. Defaults to "Change"

40.4 Demos

- Slot-Blocking and You

41 unGrief

41.1 Summary

Module to deter griefers and enforce PVE-only rules

41.2 Dependencies

dcsCommon, cfxZones

(SSB on server when using SSB as retaliation)

41.3 ME Integration

unGrief uses a config zone to control the module's main functionality

Name	Description
verbose	A value of "true" turns on debugging messages. Default is "false"
graceKills	Number of own faction ("friendly") kills that are permissible before unGrief retaliates. Set to 0 to disallow (and immediately punish) any friendly fire kills. Default is 1
retaliation	How unGrief retaliates towards the player when their graceKills are exceeded. The following options are supported: • 'boom' Place a small explosive inside the plane, grin, ignite. • 'ssb' Use server-side SSB to kick the player from the plane Default is 'boom'
wrathful	Sometimes griefers are slow learners. When set to true, unGrief disallows a repeat offender to re-slot after their second transgression. Their plane is destroyed every time they try to slot until the mission ends.
pve pveOnly	When set to true, PVE rules are in force for this mission. Player versus Player kills count like friendly kills Defaults to FALSE (PVP killing is allowed)
ignoreAI	Ignores friendly kills if the killed unit was AI-controlled (i.e. not a player unit). Useful in conjunction with PVE rules. Defaults to false.
warnings	When set to true, players entering and leaving a PvP zone receive a notification. Only works when PvE is set to true Defaults to true (notifications when entering/leaving a PVP zone)

You can designate areas as PvP combat zones (requires that pve is enabled)

Name	Description
pvp	The mere presence of this attribute marks this zone as a PVP combat zone. Killing players of another faction in this zone is legal. Same-faction kills are still illegal and will be punished.
strict	When set to true enforces ‘strict’ pvp rules, requiring both planes be inside the PvP zone at the time of the kill. When false, only the killed player plane must be inside the PvP zone. Defaults to false

41.4 Demos

- Good Grief
- The Danger Zone

42 Unit Persistence

42.1 Summary

Module that provides persistence (load & save) ability for ground units and static objects that are placed with ME (i.e. it does not supports units that are spawned dynamically)

42.2 Dependencies

Requires dcsCommon, cfxZones, persistence and cfxMX.

42.3 ME Integration

42.4 Demos

- Being Persistent

43 Unit Zone

43.1 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.

43.2 Dependencies

dcsCommon, cfxZones

43.3 ME Integration

Name	Description
unitZone	Marks this ME Zone as an anchor for unitZone **MANDATORY**
lookFor	Name for the AI group or player unit to check zone status. If the last character in the name is an asterisk "*", exact matches *and* 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: • 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
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	When present, this flag (or flags) is always set to the current state of the unit zone: • 1 when one or more units in the zone • 0 when none of the indicated units in the zone. Default is <none>
uzDirectInv	Like uzDirect, just inverted, provided since unitZones is often used to control many functions, and this saves an inverter (changer) module: • 0 when one or more units in the zone • 1 when none of the indicated units in the zone. Default is <none>

Supports DML Flags
Supports Watchflags
Supports zone-local verbose

43.4 Demos

- Follow Me!
- xFlags – Field Day
- Gate and Switch
- Forever-looping Spawners

44 Willie Pete

44.1 Summary

A module to create artillery fire on locations marked with smoke munitions (WP – white phosphorous)

44.2 Dependencies

WilliePete requires dcsCommon, cfxZones and cfxMX.

Optional: PlayerScore (no load time limitation)

44.3 ME Integration

Name	Description
wpTarget	Marks this zone as a WP Zone. The value of this attribute defines which coalition uses this zone. Valid values are • RED or 1 • BLUE or 2 **MANDATORY**
shellStrength	Explosive power of shell hitting the ground. Defaults to 500
shellNum	Number of shells per artillery salvo Defaults to 17
transitionTime	The time (in seconds) it takes for shells to arrive after the 'fire' command is given Defaults to 20
coolDown	Number of seconds after which the artillery can fire again. Defaults to 180 (3 minutes)
baseAccuracy	Radius (in m) around the wp where the shells hit in. Defaults to 50
method wpMethod	DML method for the outputs
wpFired!	Output that is signaled when a successfully Fire command was received by the WP zone Defaults to <none>
checkInRange	Distance (from the zone's border) from which a plane can check into the zone. Example: if a WP zone has a radius of 3 km, and a checkInRange of 2 km, a plane can check into the zone when it's 5km (= 3 + 2) or less away from the zone's center. Be careful with check-in ranges when they overlap, as this may lead to confusing situations for pilots who may inadvertently sign into the wrong zone.
ackSound	Sound file to play after a player has successfully transmitted target coordinates. Overrides any setting of the wpConfig Defaults to what you defined in wpConfig

Name	Description
guiSound	Sound file to play whenever a player uses a menu item from the WP "FAC" menu. Overrides any setting of the wpConfig Defaults to what you defined in wpConfig

44.4 Demos

- Willie Nillie

45 Wiper

45.1 Summary

Removes objects inside a zone whenever you trigger it

45.2 Dependencies

dcsCommon, cfxZones

45.3 ME Integration

Name	Description
wipe?	Watchflag. Triggers a wipe cycle **MANDATORY**
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 • "unit" or 1 • "weapon" or 2 • "static" or 3 • "base" or 4 • "scenery" or 5 • "cargo" or 6 Defaults to 'static'
wipeNamed	*Optional* 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" **Examples:** • "Ba*" – all objects inside the zone whose name starts with "Ba" • "Grou*, Commander Kirk, He*" – all objects whose name starts with "Grou" or "He", and the object whose name exactly matches "Commander Kirk" 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

45.4 Demos

- Viper with a double you

46 xFlags (Flag Testing)

46.1 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 requries 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)

46.2 Dependencies

dcsCommon, cfxZones

46.3 ME Integration

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

Name	Description
	• 'most' triggers when more than half of the input flags have triggered. Will not trigger is exactly have have triggered, so be careful if the number of input flags is even. • '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 • 'never' used when you are using xFlag's "direct outputs" (xDirect, xCount, xChange) and want it to operate during the entire mission. xFlags will never trigger xSuccess, and keep evaluating, setting xDirect, xCount and xChange accordingly Defaults to 'some'
#hits	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)
xFlagMethod	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"
xSuccess! out!	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 <none>
xChange!	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 <none>
xDirect	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. **This flags value is set directly, not via DML method.** Defaults to <none>
xCount	Each time xFlags evaluates the input flags, it directly sets the xDirect flag to the number of hits (positive test results from the individual flags tests). For example, if three tests of the input flags are successful, xFlags sets the value of this output to the number 3 **This flags value is set directly, not via DML method.** Defaults to <none>
xReset?	When the value of this input changes, the zone's xFlag module is reset, and evaluation starts afresh.

Name	Description
	Note: this input always reacts to a change in the flag's value Defaults to <none>
method xMethod	DML method for output flags Defaults to "inc"
oneShot	When the value if this attribute is false, that zone's xFlag module will not stop evaluating after it triggers xSuccess.
xOff?	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. **Note: this input always reacts to a change in the flag's value** Defaults to <none>
xOn?	Turns a suspended xFlag back on to resume processing. It resumes processing where it left off **Note: this input always reacts to a change in the flag's value** Defaults to <none>
xSuspended	Sets the initial state of xFlag. Setting it to true suspends the xFlag at mission start Defaults to false

Supports DML Flags
Supports DML Watchflags
Supports zone-local verbose

46.4 Demos

- xFlags – Field Day
- Count Base's Blues

47 Module Name

47.1 Summary

47.2 Dependencies

47.3 ME Integration

47.4 Demos