

cf/x  
Dynamic Mission Library  
for DCS

# DML

## QUICK REFERENCE

### ME MODULE INTEGRATION

Copyright © 2022 - 2024 by cf/x AG and Christian Franz

# Table Of Contents

|    |  |    |
|----|--|----|
| 1  | ME FLAG INTEGRATION (common abilities) ..... | 4  |
| 2  | DEBUGGER .....                               | 10 |
| 3  | Airfield .....                               | 12 |
| 4  | Artillery Zones .....                        | 16 |
| 5  | Artillery UI .....                           | 18 |
| 6  | ASW (asw, aswZones, aswGUI, aswSubs) .....   | 19 |
| 7  | BaseCaptured .....                           | 22 |
| 8  | BombRange .....                              | 23 |
| 9  | Cargo Receiver .....                         | 26 |
| 10 | Changer .....                                | 27 |
| 11 | CivAir .....                                 | 29 |
| 12 | Clone Zone .....                             | 32 |
| 13 | Count Down .....                             | 38 |
| 14 | Counter .....                                | 40 |
| 15 | CSAR Manager .....                           | 41 |
| 16 | Delay Flags ("Timer") .....                  | 45 |
| 17 | Delicates .....                              | 46 |
| 18 | Factory Zone .....                           | 47 |
| 19 | FARP Zones .....                             | 50 |
| 20 | FireFX .....                                 | 51 |
| 21 | Flare Zone .....                             | 53 |
| 22 | Ground Explosion .....                       | 55 |
| 23 | Group Tracker .....                          | 57 |
| 24 | Guardian Angel .....                         | 59 |
| 25 | Helo Troops .....                            | 61 |
| 26 | Impostors .....                              | 63 |
| 27 | Limited Airframes .....                      | 64 |
| 28 | LZ .....                                     | 66 |
| 29 | Map Markers .....                            | 68 |
| 30 | Messenger .....                              | 69 |
| 31 | (Simple) Mission Restart .....               | 74 |
| 32 | NDB .....                                    | 75 |
| 33 | NoGap / NoGapGUI .....                       | 77 |
| 34 | Object Destruct Detector .....               | 78 |
| 35 | Object Spawn Zones .....                     | 79 |

|    |                             |     |
|----|-----------------------------|-----|
| 36 | OwnAll.....                 | 81  |
| 37 | Owned Zones.....            | 83  |
| 38 | Persistence .....           | 85  |
| 39 | Player Score.....           | 88  |
| 40 | Player Zone.....            | 93  |
| 41 | Pulse Flags .....           | 94  |
| 42 | Radio Menu.....             | 96  |
| 43 | Radio Trigger.....          | 100 |
| 44 | Raise Flag.....             | 101 |
| 45 | Recon Mode.....             | 102 |
| 46 | Rnd Flags.....              | 105 |
| 47 | Scribe.....                 | 107 |
| 48 | Sequencer.....              | 108 |
| 49 | SittingDucks .....          | 110 |
| 50 | Smoke Zones .....           | 111 |
| 51 | Spawn Zones .....           | 112 |
| 52 | ssbClient .....             | 115 |
| 53 | StopGap / StopGapGUI.....   | 117 |
| 54 | TACAN.....                  | 118 |
| 55 | Taxi Police.....            | 120 |
| 56 | TDZ (Touch-Down Zone).....  | 122 |
| 57 | UnGrief.....                | 125 |
| 58 | Unit Persistence .....      | 127 |
| 59 | Unit Zone.....              | 128 |
| 60 | Willie Pete .....           | 130 |
| 61 | Wiper.....                  | 132 |
| 62 | Valet.....                  | 134 |
| 63 | XFlags (Flag Testing) ..... | 137 |
| 64 | Module Name .....           | 140 |

# 1 ME FLAG INTEGRATION (common abilities)

## 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       | Description  |
|------------|--|
| linkedUnit | <p>Moves the zone's center with the unit whose name <i>exactly</i> 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 the <i>linkedUnit</i> ceases to exist after the zone has followed it, the zone remains at the last location it moved to until a new unit with that exact name appears again. This is often the case with player-controlled planes, so expect this to happen and design your mission accordingly.</p> <p>If neither <i>useOffset</i> nor <i>useHeading</i> attributes (see below) are set to true, the zone always centers on <i>linkedUnit's</i> location.</p> <p><b>Note:</b><br/>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. This means that zones linked to player or client unit will only move to those locations after the player has entered the game.</p> <p><b>Note:</b><br/>The 'linkedUnit' attribute is one of two methods to link a zone to a unit. The other method is ME's dedicated 'LINK UNIT' pop-up (see description, above). Use only one of the two methods to link a zone to a unit, else DML will give you an error.</p> <p>The advantage of using a <i>linkedUnit</i> attribute over ME's pop-up method is that the <i>linkedUnit</i> attribute can be used with dynamically spawned units (i.e. units that aren't placed with ME). Its main disadvantage is that it can be prone to spelling mistakes or when you change the name of the linked unit in ME.</p> <p>Defaults to &lt;not linked to any unit&gt;</p> |
| useOffset  | <p>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 linked unit and zone constant.</p> <p>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.</p>   |

| Name       | Description   |
|------------|---|
|            | <p><b>Requires either <i>linkedUnit</i> or LINK UNIT be set</b></p> <p>Not compatible with <i>useHeading</i> (below) – use either<br/>Defaults to &lt;not set&gt;</p>   |
| useHeading | <p>When set to true, the zone moves and turns in synch with and relative to the <i>linkedUnit</i>.</p> <p><b>Requires either <i>linkedUnit</i> or LINK UNIT be set</b></p> <p>Not compatible with <i>useOffset</i> (above) – use either<br/>Defaults to &lt;not set&gt;</p> |
| owner      | <p>The coalition that owns this zone. Used with some zone enhancements.</p> <p>Do not set this attribute unless you know what you are doing, or a module's documentation requests you to do so.</p> <p>Defaults to neutral</p>  |

## 1.2 Understanding DML Flags

A DML module uses “Inputs” and “Outputs” to communicate:

- It *sends* signals (the **output (“!”) attribute**, usually designated by the exclamation point “!” at the end of the attribute's name). When sending a signal, the module changes the value of the flags that are listed after the output attribute. The way that it changes the values is described by the ‘output method’ attribute, which is usually “inc” or “increment the current value by one”.  
Mission designers can change the way a module changes values on a zone-individual basis. **A module sends signals (sets output flags) whenever it sees fit**, usually as a response to something happening in the mission (a group enters a zone, an object is destroyed etc.)
- It *receives* signals: modules periodically look at the flag that is listed for their **input (“?”) flag attribute(s)** (usually an attribute with question mark “?” in their name) and compare that value to what they read before. If the value is the same as before, that input is ignored. If the value did change, they then decide if what they read fulfils the trigger condition (also called ‘input method’). If it does, the module triggers its appropriate action. The most common trigger condition is ‘change’: the input triggers when the flag's current value changes. In DML, trigger conditions are called ‘methods’, and you can tell a module which conditions it should trigger on

### Note:

**Modules periodically look at their input flags**, usually once a second (you can set this interval for the module with the “ups” attribute in the module's config zone). This means that modules do not instantly detect a signal/change, *only the next time they look at their input flags*. This also means that they can miss a signal that was sent if the input flag changes too quickly (a rare occasion).

## 1.3 Bang! Output 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. **This is the most common bang! method and is the default for most DML module outputs**; it co-operates best with input flags that are set to trigger on "change" – which is also the default in DML
- **'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 a very 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. Note that this method can be error prone if one or more modules flip a flag's value multiple times before a module's input can detect a change: if two modules "flip" a flag in rapid succession before an input checks the flag's value, it appears to be the same as before even though it was changed: the second flip returned the flag to its initial value, masking the change. That is why the "inc" method is to be preferred over 'flip' for general flag banging.
- **#(number or flagName) [set to absolute value or that of another flag]**  
Sets the flag to the fixed value (when giving a number) or copies the value from another flag, no parentheses.

### Examples:

- #33 sets the flag to the number 33,
  - #-6 sets the flag to the number -6 (negative six),
  - #kills sets the flag to the value that the flag named 'kills' currently holds,
  - #"122" (note the quotes) sets the flag to the value that the flag named '122' (a legal, old DCS flag name) currently holds
- **+(number or flagName) [add amount or another flag's value]**  
Adds the number give (no parentheses!) or the current value of the flag flagName to the flag.

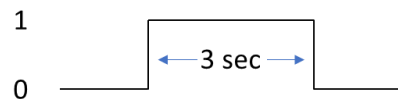
### Examples:

- "+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 DCS flag named 22 to the current value
- `-(number or flagName)` **[subtract amount or another flag’s value]**  
Subtracts the number given (no parentheses!) 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 “penalty” 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,



#### Example:

- ``pulse, 4`` will keep the pulse up for four (4) seconds before dropping back to zero.

## 1.4 Multiple Output Flags

DML can bang! multiple flags at the same time. Unless otherwise specified, all outputs (those attributes that end on an exclamation point “!”) support this ability. To bang! multiple flags, simply list them as the attribute’s value and separate them by comma; leading/trailing blanks are ignored.

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

Please note the following:

- All flags are banged! with the same method.
- All flags are set at the same time, meaning that there is no guaranteed order in which they are changed.
- Should you list the same flag multiple times, it will get set multiple times; this means that the value changes internally multiple times; that does not mean that inputs who watch that flag can detect multiple changes; from an external point of view the value of that flag changes at maximum once between inspections.

## 1.5 Input Methods (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 ``#``  
trigger whenever the watched flag’s value changes. No other conditions need be

fulfilled. **This is the default**

- `'off'` or `'0'` or `'no'` or `'false'`  
triggers when the watched flag's value changes to zero
- `'on'` or `'1'` or `'yes'` or `'true'`  
triggers when the watched flag's value changes from zero to non-zero  
(**Warning:** with this method, 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. To trigger again, the flag must first return to a value of zero)
- `'inc'` or `'+1'`  
triggers when the watched flag's current value is greater than the previous value
- `'dec'` or `'-1'`  
triggers when the watched flag's value is less than the watched flag's previous value
- `'lohi'`  
triggers when the watched flag's previous value was zero (0) or less and the new value is greater than zero. Often used with pulses.
- `'hilo'`  
triggers when the watched flag'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 watched flag's value changes, and the value is larger than the number given or flag identified by name

**Examples:**

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

**Examples:**

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

**Examples:**

- `<4` triggers when the watched flag's value is less than the number 4



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

#### Examples:

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

### Quoting Numbered Flags

Early versions of DCS used flag names that entirely consisted of number. For example, "22" was (and still is) a legal flag name. This can create confusion when using some trigger methods: DML can't tell the difference between a number and a flag whose name happens to be a number.

To allow DML to distinguish between a number and **flags 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 the input to trigger only if the connected flag was equal to flag named 22, that condition would be

`= "22"`

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

## 1.6 DML Flag Naming Rules

|                           | DML   | DML Zone-Local   |
|---------------------------|---|--|
| <b>Format</b>             | <ul style="list-style-type: none"> <li>• Alphanumeric</li> <li>• must not contain comma ','</li> <li>• must not start with asterisk '*'</li> <li>• must not start with double quote '"'</li> </ul> <i>should</i> not start with a digit ('0'...'9') | <b>Starts with asterisk '*'</b> , alphanumeric, must not contain comma ','                                 |
| <b>Examples</b>           | <ul style="list-style-type: none"> <li>• A12</li> <li>• With blank</li> <li>• F***d up</li> </ul> Yup "quotes" too  | <ul style="list-style-type: none"> <li>• *1</li> <li>• *A12</li> <li>• *fireCloner</li> </ul> *ok multi ** |
| <b>Scope / Visibility</b> | DML modules, Entire DCS (newer versions)  | Only DML modules attached to the same zone   |
| <b>Invisible to</b>       | n/a   | Everyone outside Zone  |

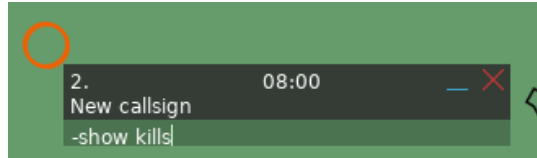
## 2 DEBUGGER

### 2.1 Summary

“The Debugger” is an interactive (in-mission) debugger that sports a ‘console’ via the Add Map Mark Label’ mission feature to give commands. Commands all start with a hyphen.



“Mark Label”



and will elicit a response when accepted

```
[08:00:20] flag <kills> : value <0>
```

### 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<br><br><b>MANDATORY</b>  |
| triggerMethod<br>debugTriggerMethod<br>inputMethod<br>sayWhen | Trigger condition for the flags (the observer’s “condition” that triggers a report for the flag)<br>Defaults to ‘change’   |
| method<br>outputMethod<br>debugMethod                         | DML Method for the debugger’s output flags. Rarely used.<br>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.<br><br><b>Note that this allows you to provide individual message formatting per observer</b> , a feature that is not available for the interactive debugger.<br><br>Defaults to<br>“---debug: <t> -- Flag <f> changed from <p> to <c> [ <z> ]” which results in a message similar to<br><pre>---debug: 08:00:12 -- Flag t1 changed from 2 to 4 [many flags]</pre> |

### 2.4 Demos

- Bug Hunt



## 3 Airfield

### 3.1 Summary

This module provides signals from airfields/FARPS when they are captured, and can provide control over airfield/FARP ownership to assign an airfield to a coalition (RED/BLUE)

### 3.2 Dependencies

Airfield requires the modules dcsCommon and cfxZones.

If you are using persistence and cloners or spawners that take their ownership from airfield zones, airfield should load before cloneZones and spawn zones.

### 3.3 ME Integration

| Name          | Description   |
|---------------|---|
| airfield      | <p>Tells DML that this trigger zone should associate with the closest airfield (or FARP, see config settings)</p> <p><b>The owner of this zone is always the same as the owner of the airfield/FARP that airfield associates with.</b></p> <p>The value of this attribute is ignored.</p> <p><b>MANDATORY</b></p>   |
| farps         | <p>When set to true, airfield zones also associate with FARPs. When set to false, FARPs are ignored.</p> <p>When enabled, care must be taken when placing FARPS close to airfields: an airfield zone associates itself with the <i>closest airfield or FARP</i>, and the in-game location of an airfield may not be where you assume it is. Hint: enable verbosity and see which airfield zone attaches to which object.</p> <p>Defaults to false (associate only with airfields, disregard FARPs)</p>  |
| fixed         | <p>Defines which side holds the associated airfield/FARP <b>at mission start</b>. The airfield/FARP is made <b>unconquerable</b>, and remains in that coalition's possession until the zone receives a signal on one of the following inputs:</p> <ul style="list-style-type: none"><li>• <code>makeRed?</code> – ownership is turned over to RED</li><li>• <code>makeBlue?</code> – ownership is turned over to BLUE</li><li>• <code>makeNeutral?</code> – ownership is turned over to NEUTRAL</li><li>• <code>autoCap?</code> – makes the airfield/FARP capturable by mission ground capture rules.</li></ul> <p>Valid values are 0, 1, 2, red, blue, neutral</p> <p>Defaults to &lt;none&gt;, airfield can be captured normally, and initial ownership is as set up by mission editor.</p> |
| method        | <p>Output method for all outputs.</p> <p>Defaults to 'inc'</p>  |
| triggerMethod | <p>DML input method for all inputs</p>  |

| Name          | Description  |
|---------------|--|
|               | Defaults to 'change'   |
| red!          | Flags that should be sent a signal when the associated airfield/FARP is captured by RED faction<br>Defaults to <none>  |
| blue!         | Flags that should be sent a signal when the associated airfield/FARP is captured by BLUE faction<br>Defaults to <none>   |
| makeRed?      | When a signal is received on the flag that connects to this input, the zone takes control of airfield/FARP ownership (no longer mission ground capture rules), and forces ownership to RED faction. If the airfield/FARP was previously owned by a different faction, a signal is sent to red! Output.<br>To have the airfield return to being able to be captured by other factions, send a signal to the autoCap? input.<br>Defaults to <none>   |
| makeBlue?     | When a signal is received on the flag that connects to this input, the zone takes control of airfield/FARP ownership (no longer mission ground capture rules), and forces ownership to BLUE faction. If the airfield/FARP was previously owned by a different faction, a signal is sent to blue! Output.<br>To have the airfield return to being able to be captured by other factions, send a signal to the autoCap? input.<br>Defaults to <none> |
| makeNeutral?  | When a signal is received on the flag that connects to this input, the zone takes control of airfield/FARP ownership (no longer mission ground capture rules), and forces ownership to NEUTRAL faction.<br><br>To have the airfield return to being able to be captured by other factions, send a signal to the autoCap? input.<br>Defaults to <none>  |
| autoCap?      | When a signal is received on the that connects to this input, the zone relinquishes ownership control over the associated airfield/FARP and ownership is determined by the mission ground capture rules<br>Defaults to <none>  |
| directControl | When set to true, at mission start the zone assumes control over the associated airfield/FARP's ownership: it can no longer be captured by mission ground capture rules. Use the autoCap? input to relinquish control back to the mission, or makeRed?, makeNeutral? and makeBlue? inputs to explicitly set ownership.<br>Defaults to false (at mission start, the associated airfield's ownership is determined by mission ground capture rules)  |
| ownedBy#      | Flags connected to this output are set to the value of the currently owning faction of the associated airfield/FARP: 0 (neutral), 1 (red) or blue (2)<br>Defaults to <none>  |
| redFill       | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the airfield's capture zone when owned by RED coalition. RGBA values each range from 0.0 to 1.0<br>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)  |

| Name        | Description   |
|-------------|---|
|             | Defaults to config zone's "redFill" value   |
| redLine     | <p>Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the airfield capture zone's outline when owned by RED coalition. RGBA values each range from 0.0 to 1.0</p> <p>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)</p> <p>Defaults to config zone's 'redLine' value</p>         |
| blueFill    | <p>Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the airfield's capture zone when owned by BLUE coalition. RGBA values each range from 0.0 to 1.0</p> <p>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)</p> <p>Defaults to config zone's "blueFill" value</p>               |
| blueLine    | <p>Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the airfield capture zone's outline when owned by BLUE coalition. RGBA values each range from 0.0 to 1.0</p> <p>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)</p> <p>Defaults to config zone's 'blueLine' value</p>       |
| neutralFill | <p>Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the airfield's capture zone when owned by NEUTRAL coalition. RGBA values each range from 0.0 to 1.0</p> <p>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)</p> <p>Defaults to config zone's "neutralFill" value</p>         |
| neutralLine | <p>Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the airfield capture zone's outline when owned by NEUTRAL coalition. RGBA values each range from 0.0 to 1.0</p> <p>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)</p> <p>Defaults to config zone's 'neutralLine' value</p> |
| show        | <p>If set to true, the airfield's 2km capture zone is shown on the F10 map in the colors described above, by owning coalition.</p> <p>Defaults to false (capture zone is not shown)</p>   |

### **3.4 Demos**

- Airfield mine
- Send in the Clones

## 4 Artillery Zones

### 4.1 Summary

Simulates an artillery barrage inside the zone.

### 4.2 Dependencies

dcsCommon, cfxZones

### 4.3 ME Integration

| Name                               | Description   |
|------------------------------------|---|
| artilleryTarget                    | Marks this zone as an artillery zone. Value is ignored<br><b>MANDATORY</b>  |
| coalition                          | Used with Artillery UI – the coalition that can give a fire command (the explosions are completely coalition agnostic – they kill anyone). When the artillery zone is marked on the map, only this side will see it. Defaults to 0. Supports “red” and “blue” as values |
| spotRange                          | Used with Artillery UI – the maximum range at which an FO can give a fire command. Measured from center of zone. Defaults to 3000 meters  |
| shellStrength                      | Average power of <b>each</b> exploding shell. Defaults to 500. 3000 is enough to level big buildings, so be conservative.   |
| shellNum                           | Number of shells (salvo) per fire cycle. Defaults to 17 shells per cycle  |
| transitionTime                     | The time (in seconds) the shells take on average to reach the target zone. Note that not all shells arrive at once but are usually spread over a couple of seconds. Defaults to 20  |
| addMark                            | Add the artillery target zone to the F10 map of coalition (see above). Defaults to <b>true</b> .  |
| shellVariance                      | Difference in shell’s explosion power, in percent. Defaults to 0.2 (20%)  |
| f?<br>in?<br>artillery?            | <b>DML Watchflag</b> . When triggered, the artillery bombardment starts. Defaults to <none><br>You can use any synonym, but only one per zone   |
| triggerMethod<br>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  |



## 4.4 Demos

- Artillery with UI
- Artillery zones triggered
- Pulsing Fun

## **5 Artillery UI**

### **5.1 Summary**

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

### **5.2 Dependencies**

Tcb

### **5.3 ME Integration**

Tbc

### **5.4 Demos**

- Artillery with UI

## 6 ASW (asw, aswZones, aswGUI, aswSubs)

### 6.1 Summary

ASW provides Anti-Submarine Warfare mechanics to your mission, allowing players to stock up on ASW munitions, and hunt for submerged aircraft by dropping buoys and torpedoes.

### 6.2 Dependencies

- **asw** requires dcsCommon, cfxZones.
- **aswZones** requires dcsCommon, cfxZones and **asw**
- **aswGUI** requires dcsCommon, cfxZones, **asw** and **aswZones**
- **aswSubs** requires dcsCommon, cfxZones

### 6.3 Configuration

Most ASW\* features are set up via configuration zones:

#### 6.3.1 ASW (main)

| Name           | Description   |
|----------------|---|
| verbose        | A value of "true" turns on debugging for the entire module.<br>Default is "false"   |
| buoyLife       | Duration (in seconds) how long a buoy is active.<br>Defaults to 1800 (=30 minutes)  |
| fixLife        | Number of seconds that a sub fix is reliable and can be used by nearby torpedoes to get a better homing chance.<br>Defaults to 180 (=3 minutes)   |
| detectionRange | Sensor range (in meters) for a buoy. Reliability deteriorates with distance, and beyond this range, no subs are detected.<br>Only enemy subs are detected when in range<br>Does not detect vessels on the surface, even if those vessels are of the submarine class<br>Defaults to 12000 (12km) |
| sureDetect     | "point blank" range for buoys. If a sub is closer than this range they are always detected.<br>Defaults to 1000   |
| detectionDepth | Maximum depth (in meters) in which submarines are detected.<br>Submarines that are deeper than that are invisible and will not be detected.<br>Defaults to 500  |
| fixSound       | Sound effect to play when a new fix on a submarine is made<br>Defaults to "submarine ping.ogg" (file is part of DML's library)  |
| sonarSound     | Sound effect to play when a buoy finds a contact.<br>Defaults to "beacon beep-beep.ogg" (file is part of DML's library)   |
| redKill!       | Output flag to bang! when a red submarine is killed by ASW means.<br>Defaults to <none>   |
| blueKill!      | Output flag to bang! when a blue submarine is killed by ASW means.<br>Defaults to <none>  |
| method         | Method for output flags   |

| Name       | Description   |
|------------|---|
| smokeColor | Color that marks a buoy. Can be a number (0 to 4) or “red”, “green”, “blue”, “white”, “orange.”<br>Defaults to “red”  |
| killScore  | Only relevant when PayerScore is active in mission<br>Score for killing a submarine with ASW munitions<br>Defaults to 0 (no score)                          |
| killFeat   | Only relevant when PlayerScore is active in mission<br>Description of feat when killing a submarine with ASW munitions, supports all PlayerScore wildcards. |

### 6.3.2 aswZones

| Name    | Description   |
|---------|---|
| verbose | A value of “true” turns on debugging for the entire module.<br>Default is “false” |

### 6.3.3 aswGUI

| Name          | Description  |
|---------------|--|
| verbose       | A value of “true” turns on debugging for the entire module.<br>Default is “false”  |
| aswCarriers   | Comma-separated list of types of player aircraft that are allowed to perform ASW.<br>Defaults to DCS Common’s list of troop carriers (Mi-8MT, UH-1H, Mi-24P), and you can provide your own list:<br>Example: “Mi-8MT, UH-1H, C-130J” removes the Hind and adds the Hercules fixed-wing transport to the list of legal ASW carft.<br><br>Supports wildcard type endings: if a type ends on an asterisk (“*”) all types that match whatever precedes the asterisk are accepted. For example, “Mi-*” will match both “Mi-8T” and “Mi-24P”.<br><br>You can supply the type ‘any’ or ‘all’ to allow all aircraft to perform ASW.<br><br>Default <none> (use dcsCommon’s list of troop carriers for ASW) |
| buoyWeight    | Weight (in kg) per ASW buoy<br>Defaults to 50kg  |
| torpedoWeight | Weight (in kg) per ASW torpedo<br>Defaults to 700kg  |

### 6.3.4 aswSubs

| Name            | Description   |
|-----------------|---|
| verbose         | A value of “true” turns on debugging for the entire module.<br>Default is “false”   |
| critDist        | Distance (in meters) to trigger the submarine’s attack on a target vessel (see <i>targets</i> , below). The target vessel will receive multiple (see <i>salvoSize</i> ) hits (see <i>explosionDamage</i> , below) |
| explosionDamage | Damage inflicted by attacking aswSub per hit.<br>Defaults to 1000   |

| Name      | Description   |
|-----------|---|
| salvoSize | A range (e.g., 2-4) that determines how many hits a target receives when attacked. Once a target ship has been hit, no other aswSub will attack that vessel.<br>Defaults to '4-4' (always take four hits) |
| targets   | Comma-separated list of group names that are targets for aswSubs. The group must exist in ME when the mission starts.<br>Defaults to <none>   |

## 6.4 ME Integration

| Name                              | Description  |
|-----------------------------------|--|
| asw                               | Marks this zone as an ASW supply/drop zone<br><br><b>MANDATORY</b>   |
| buoys                             | Number of buoys in stock in this zone. Setting this number to -1 sets the supply to infinite.<br>Defaults to -1 (infinite supply)  |
| torpedoes                         | Number of torpedoes in stock in this zone. Setting this number to -1 sets the supply to infinite.<br>Defaults to -1 (infinite supply)  |
| coalition                         | Coalition which owns the asw that are being dropped by this zone. If this zone is linked to a unit, that unit's coalition is used instead of any value given to this attribute.<br>(only required when dropping ASW and zone is not linked to a unit)<br>Defaults to 'neutral' |
| buoy?                             | Input flag that triggers the drop of an ASW buoy. Will only drop a buoy if there are enough left in the stores.<br>Defaults to <none>  |
| torpedo?                          | Input flag that triggers the drop of an ASW torpedo. Will only drop a torpedo if there are enough left in stores.<br>Defaults to <none>  |
| triggerMethod<br>aswTriggerMethod | DML Method for input flags<br>Defaults to 'change'   |

## 6.5 Demos

- Davy Jones' Rocker

## 7 BaseCaptured

### 7.1 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).

### 7.2 Dependencies

dcsCommon, cfxZones

### 7.3 ME Integration

|                         |  |
|-------------------------|--|
| <b>baseCaptured!</b>    | 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.<br><br><b>MANDATORY</b> |
| method<br>captureMethod | DML method for output flags<br>Defaults to 'inc'   |
| blueCaptured!<br>blue!  | Flags to bang! when blue faction captures the closest base to this zone<br>Defaults to <none>  |
| redCaptured!<br>red!    | Flags to bang! when blue faction captures the closest base to this zone<br>Defaults to <none>  |
| contested!              | Flags to bang! when closest base becomes contested and belongs to neither blue nor red<br>Requires handleContested be true in configuration (default)<br>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.  |

### 7.4 Demos

- Count Base's Blues

## 8 BombRange

### 8.1 Summary

A module that provides bomb range training services to the mission

### 8.2 Dependencies

bombRange requires dcsCommon and cfxZones

### 8.3 ME Integration

| Name             | Description   |
|------------------|---|
| <b>bombRange</b> | Marks this zone as a bombRange target zone.<br><br><b>MANDATORY</b>   |
| percentage       | If set to true, the target zone reports in percent how close to the center the impact was located. 100% means bullseye, 0% means outside.<br>If set to false, any hit inside counts as 100% hit.<br>Only works for circular target zones. If you try to enable percentage for quad-pased target zones, you will receive a warning at mission start, and the zone still reports all hits inside as 100%<br><br>Defaults to true for circular zones, always false for quad-based zones. |
| details          | When set to true, reports of an impact include the details of weapon type, pilot name, airframe type, total travel distance, impact velocity<br><br>Defaults to true  |
| reporter         | When set to true, impacts are reported.<br><br>Defaults to true   |
| reportName       | If true, the target zone's name is reported with "INSIDE target area"<br><br>Defaults to false (zone name not added)  |
| smokeHits        | If set to true, impact locations inside a target area are marked with smoke<br><br>Defaults to false (not marked with smoke)  |
| smokeColor       | Color of the smoke that marks hits inside the zone. Possible values for the smoke effect are: <ul style="list-style-type: none"><li>• "green" or "0"</li><li>• "red" or "1"</li><li>• "white" or "2"</li><li>• "orange" or "3"</li><li>• "blue" or "4"</li></ul> Only relevant if you set smokeHits to true.<br><br>Defaults to "blue"  |

| Name         | Description  |
|--------------|--|
| flagHits     | <p>Similar to 'smokeHits', except that an object is placed on the impact point instead of smoke. Called so because the default object that is placed on the impact location is a flag.</p> <p>Defaults to false (no objects placed on impact point).</p>   |
| flagType     | <p>Type Name of the object that is to be placed on the impact location. See here <a href="https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Statics">https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Statics</a> for a collection of possible objects. Only relevant if you set flagHits to true.</p> <p>Defaults to "Red_Flag"</p>  |
| clipDist     | <p>Maximum distance from center of the trigger zone that a hit is reported.</p> <p>Defaults to 2000 (meters = 2km = 1.2 miles)</p>   |
| method       | <p>DML method to bang! output flags.</p> <p>Defaults to "inc"</p>  |
| hit!         | <p>DML output. List of flags to beng! when a weapon fulfils the hit condition</p> <p>Defaults to &lt;none&gt;</p>  |
| markBoundary | <p>If set to true, the outline of the target zone is marked with a number of objects for visual cues for pilots. All objects belong to Neutral</p> <p>Defaults to false (no marks)</p>   |
| markType     | <p>Type Name of the object that is used along the zone's boundary. See here <a href="https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Statics">https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Statics</a> for a collection of possible objects.</p> <p>Defaults to "Black_Tyre_RF"</p>   |
| markNum      | <p>Number of repeats for the boundary marker per quarter. The number of objects placed is one higher per quarter, so the total numbers of objects placed is <math>(1 + \text{markNum}) * 4</math>. If, for example, you set markNum to 5, a total number of <math>(1 + 5) * 4 = 24</math> objects are places around the target zone's outline.</p> <p>Defaults to 3 (a total of <math>(1 + 3) * 4 = 16</math> objects along the outline)</p> |
| markCenter   | <p>If set to true, the center of the bomb zone is marked by an object. The object belongs to Neutral.</p> <p>Default to false (no object to mark center)</p>   |
| centerType   | <p>Type Name of the object that marks the center. See here <a href="https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Statics">https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Statics</a> for a collection of possible objects.</p> <p>Defaults to "house2arm"</p>  |
| markOnMap    | <p>If set to true, the bombZone is marked on the F-10 map.</p> <p>Default to false (no mark in F-10 Map)</p>   |
| mapColor     | <p>Four numbers, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to</p>   |



| Name         | Description  |
|--------------|--|
|              | <p>draw the zone's outline on the map. RGBA values each range from 0.0 to 1.0</p> <p>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)</p> <p>Defaults to "0.8, 0.8, 0.8 1.0" or #CCCCCFF – a light gray, fully opaque</p>  |
| mapFillColor | <p>Four numbers, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the zone on the map. RGBA values each range from 0.0 to 1.0</p> <p>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)</p> <p>Defaults to "0.8, 0.8, 0.8 0.2" or #CCCCC33 – a light gray, 80% transparent</p> |

## 8.4 Demos

- Bombs Away

## 9 Cargo Receiver

### 9.1 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

### 9.2 Dependencies

dcsCommon, cfxZones, cfxPlayer, cfxCargoManager.

### 9.3 ME Integration

|                       |   |
|-----------------------|---|
| cargoReceiver         | Marks this zone as a cargo receiver zone. Value is ignored<br><b>MANDATORY</b>  |
| 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<br>cargoMethod | Standard DML flag method for output.<br>Defaults to "inc"   |
| f!<br>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.<br>Defaults to 1, Minimum is 1 |

### 9.4 Demos

- Helo Cargo

## 10 Changer

### 10.1 Summary

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

### 10.2 Dependencies

dcsCommon, cfxZones

### 10.3 ME Integration

| Name                                 | Description   |
|--------------------------------------|---|
| change?                              | The input flag whose value is used to create the output signal<br><br><b>MANDATORY</b>  |
| out!<br>changeOut!                   | The output flag.  |
| triggerMethod<br>triggerChangeMethod | Watchflag method that is used to interpret change? when inEval is set to true<br>Defaults to 'change'   |
| inEval                               | When set to true, the input flag change? is interpreted as a watchflag under triggerChangeMethod's rules<br>Defaults to false   |
| changeTo<br>to                       | Operation to apply to the input flag to create the output value. The following operations are defined: <ul style="list-style-type: none"><li>• 'bool'<br/>Output value is 0 if input is 0, 1 otherwise (conversion to bool)</li><li>• 'not'<br/>Output value is 1 if input is 0, 0 otherwise</li><li>• 'sign'<br/>Output value is -1 if input is &lt;0, 1 otherwise</li><li>• 'abs'<br/>Output value is the absolute of input value (e.g. outputs '3' for '-3' and '3')</li><li>• 'negative'<br/>Output value is input value multiplied by -1, turning the sign.</li><li>• 'direct'<br/>Output value is the same as input value. Used primarily with when changer is functioning as a gated switch</li></ul> 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<br>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<br>Defaults to false.  |
| on?<br>changeOn?         | Turns a paused changer on, enabling transmission of the input signal (after processing) to output, opening the 'gate'.<br>Triggers on "change"<br>Defaults to <none>   |
| off?<br>changeOff?       | Turns a changer off, pausing it. Input signals are no longer processed nor propagated to output, closing the 'gate'.<br>Triggers on "change".<br>Defaults to <none>  |
| On/Off?<br>changeOn/Off? | Flag that when defined controls transmission of input to output when the changer is running (i.e. not paused). The value of this input controls the gate as follows: <ul style="list-style-type: none"> <li>• 0: gate closed, no transmission</li> <li>• Anything else: gate open.</li> </ul> Defaults to <none> |

## 10.4 Demos

- Gate and Switch

# 11 CivAir

## 11.1 Summary

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

## 11.2 Dependencies

dcCommon, cfxZones

## 11.3 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”  |
| DCS           | <p>By default, civAir adds a set of standard DCS aircraft types (Yak-40, C-130, C-17A, IL-76MD, An-30M, An-26B) to its list of civil aircraft. You can turn that off by setting the attribute’s value to false</p> <p>Defaults to ‘true’ (add above mentioned standard DCS types)</p>  |
| CAM           | <p>If you set this attribute to true, civAir also adds the following aircraft types to its list of available aircraft: A_320, A_330, A_380, B_727, B_737, B_747, B_757, Cessna_210N, DC_10</p> <p>These aircraft types aren’t included in a standard DCS install, and using such an aircraft type simply results in a non-spawning civil aircraft. The types listed above are aircraft that come with a popular mod called “CAM”. If that mod isn’t installed when DCS runs a mission that tries to access such an aircraft type, the relevant flight simply will not spawn.</p> <p>Defaults to false (no CAM types added tp civAir)</p>   |
| aircraftTypes | <p>A comma-separated list of Types (as defined in <a href="https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Aircraft">https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB/Aircraft</a> ) that define the aircraft types used for civilian flights. These must be fixed wing aircraft (i.e not helicopters).</p> <p><b>When present, all airframe types are picked from this list</b>, and each entry has the same chance to be picked. This means that if you list the same type twice, you increase the chance of that type to be picked.</p> <p>If, for example, you list</p> <p>Yak-40, Yak-40, IL-76MD</p> <p>During the mission two thirds of all civilian flights will be performed by Yak-40, and the remaining third by IL-76MB.</p> <p>The list that you provide here overrides any list that was assembled by the attribute values for “DCS” and “CAM” (see above), or provided in a civil_liveries zone.</p> |

| Name                     | Description  |
|--------------------------|--|
|                          | <p>If you accidentally misspell a type name, or that type does not exist for your DCS installation, DCS will do either of two things:</p> <ul style="list-style-type: none"> <li>• Create no flight at all (some versions of DCS)</li> <li>• Create a civilian flight of a SU-27 with fantasy “placeholder” livery – or some other random airframe/livery combo (other versions of DCS)</li> </ul> <p>Neither will disrupt your mission in any way, so using this feature is safe.</p> <p>Defaults to &lt;none&gt; (aircraft used are taken according to the settings of the attributes DCS and CAM, and whatever you supply in the civil_liveries zone)</p> |
| owner                    | <p>Country ID that all aircraft spawned by civAir belong to.</p> <p>Defaults to 82 (UN Peacekeepers)</p>   |
| ups                      | <p>Number of updates per second that civAir checks on its flights. By default, this is 0.05, or once every 20 seconds.</p>   |
| maxTraffic<br>maxFlights | <p>Maximum number of civilian flights at the same time.</p> <p>Defaults to 10</p>  |
| maxIdle                  | <p>Number of seconds of an aircraft idling that can elapse before it is removed.</p> <p>CivAir determines that an aircraft is idling by checking if it is moving. If you set this number too low, a cold-starting aircraft may be removed before it can move.</p> <p>Defaults to 480 (seconds = 8 minutes)</p>   |
| initialAirSpawns         | <p>Controls if at mission start half of maxTraffic immediately spawn in mid-air to start a mission with planes in the air.</p>   |

### civil\_liveries

With this zone you can add any aircraft type and a list of liveries to be used for that type

| Name      | Description  |
|-----------|--|
| Type_name | List of livery names, separated by comma. Example: |
| Example:  | Aeroflot, Algeria GLAM, Olympic Airways            |
| YAK-40    |  |

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

| Name   | Description   |
|--------|---|
| civAir | <p>When present, the airfield closest to this trigger zone can be added to civ air's airport list. The following values are supported:</p> <ul style="list-style-type: none"> <li>• 'closed' or 'exclude'</li> </ul> <p>No civilian flights originate or land here. This overrides any other settings for this airfield</p> |

| Name | Description   |
|------|---|
|      | <ul style="list-style-type: none"> <li>• 'departure' or 'depart only'<br/>Civilian flights can depart only from this airfield</li> <li>• 'arrival' or 'arrive only'<br/>Civilian flights can only land here</li> <li>• 'in' or 'inbound' – DOES NOT ASSOCIATE AN AIRFIELD<br/>Civilian flights can appear in this trigger zone, in-flight and at altitude. This is intended to be used to simulate off-map long-distance flights that now enter the map arena. Instead of an associated airfield's name, this zone's name is used for the flight's origin</li> <li>• 'out' or 'outbound' – DOES NOT ASSOCIATE AN AIRFIELD<br/>Civilian flights that reach this trigger zone disappear. This is intended to simulate an off-map connection, where a long-distance flight leaves the map. Instead of an associated airfield's name, this zone's name is used for the flight's origin</li> <li>• 'in/out' or 'in/outbound' – DOES NOT ASSOCIATE AN AIRFIELD<br/>Civilian flights can both appear in this trigger zone, and disappear when they reach it as their destination. This is intended as both a source and destination point for flights that have an off-map source and/or destination. Instead of an associated airfield's name, this zone's name is used for the flight's origin or destination</li> <li>• <i>Any other value, or nothing</i><br/>The airfield closest to this zone is an open airfield for departing and arriving civilian flights</li> </ul> <p>Note that you can add the same airfield multiple times by adding multiple trigger zones with a civAir attribute in proximity the same airfield. This will proportionally increase the likelihood that the airfield is picked for destination or departure</p> <p>Note also that only functioning airfields are chosen. FARPS or ships are disregarded.</p> <p><b>MANDATORY</b></p> |

## 11.4 Demos

- Virgin (Civ) Air
- Air Caucasus II
- One-Way Air
- Civ Air International
- Types and civil liveries

## 12 Clone Zone

### 12.1 Summary

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

### 12.2 Dependencies

dcsCommon, cfxMX, cfxZones

### 12.3 ME Integration

| Name      | Description   |
|-----------|---|
| cloner    | Marks this ME Zone as a clone zone. The <b>Value of this attribute is ignored</b> , use it to describe this cloner's function.<br><b>MANDATORY</b>  |
| 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.<br>If this attribute is present, this zone is not scanned for units to create a template from, and no units are removed at start.<br><br>If you supply <b>more than one template zone names</b> , separated by comma (e.g., "SAM 9 small, SAM 9 big"), each time a clone cycle is initiated, the clone zone randomly picks a template from that list.<br><br>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.<br>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.<br><br>When the clone zone is using <i>linkedUnit</i> (moving DML zone, a Core Attribute of all DML zones) in conjunction with <i>useHeading</i> (another Core Attribute), the entire route is also rotated to coincide with the linked master unit's heading differential to its original ME heading.<br><br>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.<br><br>Defaults to false ( <b>no spawn on start</b> of mission). <b>To spawn units at mission start, set this attribute to true.</b>  |



| Name                                | Description  |
|-------------------------------------|--|
| masterOwner                         | <p>If not present, all cloned units retain ownership from the template. This can be problematic if you need to change the ownership of the spawned units (for example when you import templates via source, or the clone zone can be captured).</p> <p>If present, all cloned units are spawned for the faction (red/blue/grey) that owns the zone masterOwner. There's a convenient shortcut to set the masterOwner to this clone zone (see below)</p> <p><b>CONVENIENCE SHORTCUT:</b><br/>When the master owner is set to "*" (Asterisk) wildcard, it is set to the same zone as the cloner. This is a convenience shortcut for cloners that import foreign templates and convert them to their own faction, and to enable a cloner that can be conquered to spawn its own template according to the faction that owns the spawning cloner.</p> <p>Note that when you use the masterOwner attribute, the coalition for which this cloner spawns units can change with the ownership of masterOwner.</p> <p>This can be used with cloners that import templates to easily control spawning the same template that aligns to different factions by setting the spawning cloner's 'owner' to a faction, set masterOwner to itself</p> <p>Defaults to &lt;none, retain <b>template</b> ownership&gt;</p> |
| spawn?<br>f?<br>in?<br>clone?       | <p>Flag to watch for a change. If the value of this flag changes, a new clone cycle is initiated Defaults to &lt;none&gt;</p> <p>This flag has multiple synonyms. Use only one per zone.</p>   |
| triggerMethod<br>cloneTriggerMethod | <p>Watchflag method to tell what to look for in inputs.</p> <p>Defaults to "change"</p>  |
| preWipe                             | <p>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.</p> <p>If units are preWiped, spawning of new units is delayed 0.5 seconds to allow DCS to internally resolve ground height and prevent spawned new units to 'fall to the ground' or spawn suspended in mid-air.</p> <p>Default: false</p>   |
| declutter                           | <p>If this attribute is set to true, all debris and wrecks inside the clone zone are removed before a clone cycle is started. Use this to remove debris from destroyed units before re-spawning units in the same zone.</p> <p>Default: false</p>  |
| empty!                              | <p>The value of this flag is changed according to method when all units from the last spawn have been destroyed, including all static objects.</p> <p>Defaults to &lt;none&gt;</p>   |
| method                              | DML Method for output flags  |

| Name                                | Description  |
|-------------------------------------|--|
| cloneMethod                         | Defaults to "inc"  |
| deSpawn?<br>deClone?<br>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.<br>Note that if you trigger deSpawn?, empty+1 will <b>not</b> trigger subsequently.<br>Defaults to <none>   |
| trackWith:                          | List of groupTracker zones. All spawned groups are added to these groupTrackers.<br>If you have stacked the tracker on the same zone as the cloner, you can use a single asterisk '*' as zone name.<br>Supports a comma-separated list of trackers if you simultaneously want to pass the cloned groups to multiple trackers, e.g. "GroundTrack, HeloTrack"<br>This is useful if your cloner clones more than one group, and your trackers use filtering.  |
| useDelicates                        | Name of a Delicates Zone that is used to assign delicate status when this spawner spawns objects. As with the <i>trackWith:</i> attribute, you can use "*" to refer to this zone.<br>Defaults to <none>  |
| randomizedLoc<br>rndLoc             | Upon cloning, each cloned unit/object is placed randomly inside the zone. Works with moving zones. Works with quad-based zones.<br>Defaults to false   |
| rndHeading                          | Upon cloning, each cloned unit/object assumes a random heading.<br>Defaults to false   |
| triggerMethod<br>cloneTriggerMethod | Watchflag method for input flags.<br>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.<br>Defaults to false  |
| wholeGroups                         | A modifier for the rndHeading, onRoad and rndLoc attributes. When set to true the clones are affected as follows: <ul style="list-style-type: none"> <li>• 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</li> <li>• onRoad - now only the first unit is placed on the center of a road, all other units remain in location</li> <li>• rndHeading - with centerOnly, all units rotate randomly around unit 1's location</li> <li>• </li> </ul> Defaults to false (all units individually) |
| onPerimeter                         | A modifier for rndLoc. If this attribute is set to true, the units are distributed along the edge of the trigger zone. Can be used alongside wholeGroups.<br>Defaults to false.  |
| nameScheme                          | By default, a clone zone uses a simple name scheme to ensure that all cloned units have a mission-unique name. The default name  |

| Name        | Description  |
|-------------|--|
|             | <p>scheme is the original unit's name (as taken from the template), plus a hyphen "-", plus a DML-unique number (e.g. "76791")</p> <p>In certain situations, mission designers need or want a tighter control over how cloners name units: for better (easier to debug) "speaking" unit names, to achieve special effects (intentionally replace existing units), or to be able to interface with more predictably named units (e.g. linkedUnit attribute for moving zones) or other scripts.</p> <p>When you are using a name scheme for a clone zone, the onus to ensure that you don't accidentally replace an existing unit of the same name is on you. See 'Custom naming of cloned units' in this section for details.</p> <p>Clone Zone's name schemes supports wildcards that are filled in at runtime as follows:</p> <ul style="list-style-type: none"> <li>• &lt;o&gt;<br/>the original (as defined in the template) name of the unit</li> <li>• &lt;z&gt;<br/>name of the spawning cloner's zone</li> <li>• &lt;s&gt;<br/>name of clone zone to whom the template belongs that is currently spawning. If the cloner does not use a <i>source</i> attribute, it is the same as &lt;z&gt;</li> <li>• &lt;uid&gt;<br/>creates a DML-wide unique number. It is guaranteed to be unique across all clone zones.</li> <li>• &lt;i&gt;<br/>creates a group-local unique number valid only for this spawn cycle. It increases by one (1) with each use, and reset to 1 whenever a new group is cloned</li> <li>• &lt;lcl&gt;<br/>creates a zone-local unique number. It increases by one (1) with each use. This number is not shared among other clone zones, so it is not guaranteed that this number is unique across multiple cloner zones.</li> <li>• &lt;g&gt;<br/>creates a module-local unique number. It is increased by one (1) with each use. This number is guaranteed to be unique across all clone zones.</li> </ul> <p>Defaults to &lt;none&gt; (this means that clone zones uses its default naming scheme which is equivalent to &lt;o&gt;-&lt;uid&gt;)</p> |
| groupScheme | <p>Like for units, a clone zone uses a simple name scheme to ensure that all cloned <b>groups</b> have a mission-unique name. The default name scheme is the original group's name (as taken from the template), plus a hyphen "-", plus a DML-unique number (e.g. "76791")</p> <p>Like for units, you can change the way how the cloner names groups (this is always independent from how it names units)</p>   |

| Name        | Description   |
|-------------|---|
|             | <p>Clone Zone's name schemes supports wildcards that are filled in at runtime as follows:</p> <ul style="list-style-type: none"> <li>• <code>&lt;o&gt;</code><br/>the original (as defined in the template) name of the unit</li> <li>• <code>&lt;z&gt;</code><br/>name of the spawning cloner's zone</li> <li>• <code>&lt;s&gt;</code><br/>name of clone zone to whom the template belongs that is currently spawning. If the cloner does not use a <i>source</i> attribute, it is the same as <code>&lt;z&gt;</code></li> <li>• <code>&lt;uid&gt;</code><br/>creates a DML-wide unique number. It is guaranteed to be unique across all clone zones.</li> <li>• <code>&lt;i&gt;</code><br/>always returns "1"</li> <li>• <code>&lt;lcl&gt;</code><br/>creates a zone-local unique number. It increases by one (1) with each use (units or groups). This number is not shared among other clone zones, so it is not guaranteed that this number is unique across multiple cloner zones.</li> <li>• <code>&lt;g&gt;</code><br/>creates a module-local unique number. It is increased by one (1) with each use. This number is guaranteed to be unique across all clone zones.</li> </ul> <p>Defaults to <code>&lt;none&gt;</code> (this means that clone zones uses its default naming scheme for groups which is equivalent to <code>&lt;o&gt;-&lt;uid&gt;</code>)</p> |
| identical   | <p>If this attribute is set to true, each time a clone cycle is run, the cloned units receive identical name and ID as the template. This results in clones that</p> <ul style="list-style-type: none"> <li>• Replace any previous clones from this template that were cloned with 'identical = true'</li> <li>• Can still be different in location, coalition etc.</li> </ul> <p><i>identical</i> and <i>nameScheme/groupScheme</i> are mutually exclusive. When identical is true, it overrides any nameScheme settings.</p> <p>Defaults to false</p>   |
| requestable | <p>If set to true, this clone zone can be controlled via other modules, such as HeloTroops. Availability of this cloner to other modules is controlled by those modules (for example, HeloTroops will only control cloners that clone units that can be carried with a helicopter)</p> <p>Note that "requestable:true" and "source:&lt;templates&gt;" are usually mutually exclusive, so you will see a warning when the cloner module starts up and a zone sports both "requestable" and "source" attributes</p> <p>Defaults to <code>&lt;none&gt;</code></p>  |

| Name     | Description   |
|----------|---|
| cooldown | If set to a value greater than zero, a cloner can only start a new clone cycle after <cooldown> seconds have passed since the last clone cycle.<br>Defaults to -1 (no cooldown)                         |
| useAI    | Only works for ground and naval units. If set to false, the spawned groups have their AI disabled and will neither move nor shoot nor otherwise react to enemies.<br>Defaults to true (AI is turned on) |

## 12.4 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
- Cleanup Crew
- Airfield Mine
- Send in the Clones

## 13 Count Down

### 13.1 Summary

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

### 13.2 Dependencies

dcsCommon, cfxZones

### 13.3 ME Integration

| Name                                | Description  |
|-------------------------------------|--|
| <b>countDown</b>                    | Marks this as a count down. The value of this attribute defines the number times until the count reaches zero<br>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.<br>Defaults to 1 (one)<br><b>MANDATORY</b> |
| loop                                | If this attribute is true, a count down restarts after reaching zero. If the count down is given as a range, an new random start value is taken from that range (including upper and lower limit)  |
| method<br>ctdwnMethod               | DML Flag method for output flags. Use only one synonym per zone.<br>Defaults to "flip"   |
| count?<br>in?<br>clock?             | Watchflag that when triggered constitutes a signal to count down by one. Use only one synonym per zone.<br>Defaults to <none>  |
| triggerMethod<br>ctdwnTriggerMethod | Watchflag method for input flags.<br>Defaults to "change"  |
| zero!<br>out!                       | DML Output flag to bang when countdown reaches zero.<br>Use only one synonym per zone.<br>Defaults to <none>   |
| tMinus!                             | DML Output flag to bang when count is lowered, and has not yet reached zero.<br>Defaults to <none>   |
| belowZero!                          | DML Flag. Changes every time count is decreased and is below zero. Note that if the 'loop' attribute is set, this can never happen.<br>Defaults to <none>  |
| counterOut!                         | A flag, when given, is set to the current value of the count. The flag is updated each time that the counter is triggered via count?<br>Defaults to <none>   |
| disableCounter?                     | Watchflag that when triggered turns off (halts) the count down. Signals on clock? are ignored while disabled<br>Defaults to '<none>'   |
| enableCounter?                      | Watchflag that when triggered continues the count down. The count down picks up where it left off when it was disabled.<br>Defaults to '<none>'  |

| Name   | Description  |
|--------|--|
| reset? | Watchflag that when triggered restarts the count down. If the count down is randomized, a new random start amount is selected.<br>Defaults to <none> |

## 13.4 Demos

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

## 14 Counter

### 14.1 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.

### 14.2 Dependencies

Counter requires dcsCommon, cfxZones.

### 14.3 ME Integration

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

### 14.4 Demos

- Formation Trainer



## 15 CSAR Manager

### 15.1 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.

### 15.2 Dependencies

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

Optional: commander (when using CSAR Zones)

### 15.3 ME Integration

ME Integration consists of two parts: placing CSARBASES, which are zones in which a successful landing completes the CSAR mission (and returns a lost pilot when using “limited airframes”), and placing CSAR Zones that start CSAR Missions by placing downed pilots on the map on-demand

| Name            | Description  |
|-----------------|--|
| <b>CSARBASE</b> | <p>Must be present to identify this zone as CSAR Base where CSAR Missions can end. A helicopter that lands inside this zone completes CSAR missions for those pilots that it is carrying. Supports linked zones (for example if the CSAR Base is a ship). Each side that has CSAR Missions must have at least one such zone, or CSAR Missions cannot be completed. There is no upper limit on the number of CSAR Bases a side can have.</p> <p>The value of this attribute defines which sides can complete a CSAR mission here.</p> <ul style="list-style-type: none"><li>• “red” or 1 means that only red pilots can complete their missions here,</li><li>• “blue” or 2 means blue only, and</li><li>• “neutral” or “0” means that all sides can complete CSAR missions in this zone.</li></ul> <p>Defaults to “neutral” – all sides can complete CSAR Missions in this zone.</p> <p><b>MANDATORY</b></p> |
| name            | Optional name for CSARBASE.<br>Defaults to trigger zone’s name   |

| Name        | Description   |
|-------------|---|
| <b>CSAR</b> | Identifies this as CSAR Zone that is converted into a CSAR mission upon mission start or when the value of the startCSAR? flag changes. |

|                             |  |
|-----------------------------|--|
|                             | <p>Contains the name of this mission, recommended is to use a personal name, e.g. "Lt. Wesley Crasher"</p> <p>Names can be randomized by setting the name to "*rnd". In this case, each mission has a random name (first and last name). Using this option requires the <b>"names"</b> module to be present.</p> <p><b>MANDATORY</b></p> |
| coalition                   | Faction (red/blue) for which this mission is generated   |
| 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?<br>start?<br>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.   |
| triggerMethod               | DML Method that triggers inputs.<br>Defaults to "change"   |
| rndLoc                      | If set to false, the evacuee spawns at the CSAR Zone's center, or at a random location inside the CSAR zone otherwise. Note that CSAR Zones support polygonal zone shapes<br>Defaults to TRUE (random spawn location inside the zone)  |
| onRoad                      | If set to true, the evaupee spawns on the nearest road closest to the intended (even when randomized) spawn location. Note that this can cause evacuees outside of the CSAR zone.<br>Defaults to false (units are not moved to the nearest road)   |

## 15.4 Configuration

The lion's share of CSAR manager is controlled via the central configuration zone:

| Name       | Description  |
|------------|--|
| verbose    | Set to true to turn on debugging information. Defaults to false  |
| ups        | Updates per second. Defaults to 1  |
| useSmoke   | When approaching a mission target, activate smoke or not. Smoke can have performance impact when close in a helicopter, so be mindful of this option<br>Defaults to true (activate smoke)                            |
| smokeColor | Color of the smoke to pop when helicopter is in range and enabled. Understands numbers (0-4) and names (green, red, white, orange or blue)<br>Defaults to "blue"   |
| useFlare   | When a player approaches the evacuees, release a flare. This is especially helpful in night-time recovery. The flare is released 5-10 seconds after the player comes into range<br>Defaults to true (launch a flare) |

|                    |  |
|--------------------|--|
| flareColor         | When flares enabled, the color of the flare that is launched. Possible colors are -1, "random" or "rnd" for random, or 0 (zero) to 3 or one of "green", "red", "white", or "yellow" Defaults to "red"  |
| csarRedDelivered!  | Flag to change when a red coalition-flown CSAR mission ends successfully   |
| csarBlueDelivered! | Flag to change when a blue coalition-flown CSAR mission ends successfully  |
| csarDelivered!     | Flag to change when a CSAR mission ends successfully   |
| rescueRadius       | Helicopter must land within this distance (in meters) to the target to pick up. Recommended Value: 70  |
| hoverRadius        | When attempting a hover rescue, helicopter must stay within this range (in meters). Recommended value: 30  |
| hoverAlt           | When attempting a hover rescue, helicopter must stay below this altitude (in meters). Recommended value: 40  |
| rescueTriggerRange | When approaching a mission target, the mission triggers a message from the evacuees at this range. This is also the range at which smoke is triggered if enabled   |
| beaconSound        | Name of sound file (ogg or wav) to play on the ELT frequency. Includes extension. Example: "Radio_beacon_of_distress.ogg"  |
| pilotWeight        | Weight for an evacuee in kg. Recommended Value: 100  |
| hoverDuration      | Time required to hover above pilot to secure winch and complete rescue   |
| rescueScore        | Default number of points awarded for a successful rescue. Awarded upon delivery in CSARBASE. Requires PlayerScore module to have effect  |
| vectoring          | If set to false, the mission report will no longer give range and bearing to the downed pilots. Default is true (range and bearing is provided)<br>Defaults to true (vectoring enabled)  |
| actionSound        | Sound file to play on notifications  |
| troopCarriers      | <p>A list of helicopter types that are allowed to carry/rescue troops in this mission. Defaults to DCS Common's list of troop carriers (which is usually Mi-8MT, UH-1H, Mi-24P), but you can provide your own list (for example to add non-official types).<br/>Example: "Mi-8MT, UH-1H, SA342Minigun" removes the Hind and adds the Gazelle in Minigun configuration to the list of legal troop carriers.</p> <p>Supports wildcard type endings: if a type ends on an asterisk ("*") all types that match whatever precedes the asterisk are accepted. For example, "Mi-*" will match both "Mi-8T" and "Mi-24P".</p> <p>You can supply the type 'any' or 'all' to allow all helicopters to carry troops.</p> <p>Note: all types <b>must</b> be helicopters; non-helicopter types simply are ignored.</p> <p>Default &lt;none&gt; (use dcsCommon's list of troop carriers)</p> |

## **15.5 Demos**

- CSAR of Georgia
- On the record

## 16 Delay Flags (“Timer”)

### 16.1 Summary

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

### 16.2 Dependencies

dcsCommon, cfxZones

### 16.3 ME Integration

| Name                                | Description  |
|-------------------------------------|--|
| <b>timeDelay</b>                    | 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<br><b>MANDATORY</b> |
| out!<br>delayDone!                  | The flag to bang! after the delay has passed. Use only one synonym per Zone  |
| method<br>delayMethod               | DML Flag method for output<br>Defaults to “inc”  |
| f?<br>in?<br>startDelay?            | Watchflag for a change that starts the delay. Use any synonym, but only one per zone.  |
| triggerMethod<br>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.<br>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.<br>Defaults to <none>   |
| delayLeft<br>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.<br>Defaults to <none>  |

### 16.4 Demos

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

## 17 Delicates

### 17.1 Summary

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

### 17.2 Dependencies

Delicates requires dcsCommon and cfxZones.

### 17.3 ME Integration

| Name                                   | Description  |
|--|--|
| <b>delicates</b>                       | 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.<br><br>Value of this attribute is ignored<br><br><b>MANDATORY</b> |
| power                                  | Strength of the explosion when the delicate object is triggered. Defaults to 10  |
| f!<br>out!<br>delicatesHit!            | Flag to bang! when one of the units/objects defined by this zone explodes. Defaults to <none>  |
| method<br>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<br>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)   |

### 17.4 Demos

- Delicate Subjects

## 18 Factory Zone

### 18.1 Summary

This zone 'produces' units to defend the zone and attack other zones.

### 18.2 Dependencies

dcsCommon, cfxZones, cfxOwnedZones, commander, groundTroops

### 18.3 ME Integration

| Name              | Description   |
|-------------------|---|
| <b>factory</b>    | <p>Enables production abilities and attributes for this zone.</p> <p>The value is a string, coma separated, that tells the factory what types of troops to spawn. Can be overridden by other attributes like <i>'production'</i>, <i>'defenders'</i>, <i>'productionXXX'</i> or <i>'defendersXXX'</i></p> <p>Example: "Soldier M4,Soldier M4" produces two Infantry soldiers for defenders and attackers. <b>Warning:</b> these types need to <i>exactly</i> match DCS's types. Be sure not to accidentally insert blanks.</p> <p>Special meta-type: "none" – no troops</p> <p>Defaults to "none" (no troops produced)</p> <p>Note that you must also supply a separate 'owner' attribute (which will also turn this zone into an owned zone); otherwise this factory belongs to the neutral faction and can't be captured</p> <p><b>MANDATORY</b></p> <p><b>Note:</b><br/>In pre-2.x versions of ownedZones/factoryZones, the 'factory' production ability was integrated into ownedZones. If you are upgrading a mission with pre 2.0 version ownedZones to 2.x or later, you gain 99% backward-compatibility simply by adding this 'factory' attribute to the owned zone and leaving all other attributes as they are.</p> |
| <b>owner</b>      | <p><b>MANDATORY</b></p> <p>each Factory must also be an owned zone, the owner attribute must be managed by the ownedZones module</p> <p>For value description, please see "ownedZones"</p>  |
| <b>production</b> | <p>A string, coma separated, that tells the factory what types of troops to spawn. Can be overridden by other attributes like <i>'defenders'</i>, <i>'productionXXX'</i> or <i>'defendersXXX'</i></p>   |

| Name           | Description  |
|----------------|--|
|                | <p>Example: "Soldier M4,Soldier M4" produces two Infantry soldiers for attackers. <b>Warning:</b> these types need to <i>exactly</i> match DCS's types. Be sure not to accidentally insert blanks.</p> <p>Special meta-type: "none" – no troops</p> <p>Defaults to current value of '<i>factory</i>' attribute</p>   |
| defenders      | <p>A string, coma separated, that tells the factory what types of defensive ground units to spawn. Can be overridden by the side-specific attributes 'defendersRED' or 'defendersBLUE'</p> <p>Example: "Soldier M4,Soldier M4" places two red infantry soldiers to defend the factory.</p> <p>Defaults to current value of '<i>factory</i>' attribute</p>  |
| defendersRED   | <p>A string, coma separated, that specifies the types of defensive troops to spawn when the zone is owned by RED. Providing this attribute overrides the settings that you may have supplied in '<i>defenders</i>' or '<i>production</i>'</p> <p>Example: "Soldier M4,Soldier M4" places two red infantry soldiers to defend the factory when it is owned by the red faction.</p> <p>Defaults to current value of '<i>defenders</i>' attribute</p>                           |
| defendersBLUE  | <p>A string, coma separated, that specifies the types of defensive troops to spawn when the zone is owned by BLUE. Providing this attribute overrides the settings that you may have supplied in '<i>defenders</i>' or '<i>production</i>'</p> <p>Example: "Soldier M4,Soldier M4" places two blue infantry soldiers to defend the factory when it is owned by the red faction.</p> <p>Defaults to current value of '<i>defenders</i>' attribute</p>                         |
| productionRED  | <p>A string, coma separated, that specifies the types of offensive troops to spawn when the zone is owned by RED. Offensive troops seek out enemy owned zones. Providing this attribute overrides the settings that you may have supplied in '<i>production</i>'</p> <p>Example: "Soldier M4,Soldier M4" places two blue infantry soldiers to defend the factory when it is owned by the red faction.</p> <p>Defaults to current value of '<i>production</i>' attribute</p>  |
| productionBLUE | <p>A string, coma separated, that specifies the types of offensive troops to spawn when the zone is owned by BLUE. Offensive troops seek out enemy owned zones. Providing this attribute overrides the settings that you may have supplied in '<i>production</i>'</p> <p>Example: "Soldier M4,Soldier M4" places two blue infantry soldiers to defend the factory when it is owned by the red faction.</p> <p>Defaults to current value of '<i>production</i>' attribute</p> |



| Name                                  | Description  |
|---------------------------------------|--|
| redP!                                 | Output to bang! whenever production cycles for “Production” units for the RED faction.<br>Defaults to <none>   |
| redD!                                 | Output to bang! whenever production cycles for “Defender” units for the RED faction.<br>Defaults to <none>   |
| blueP!                                | Output to bang! whenever production cycles for “Production” units for the BLUE faction.<br>Defaults to <none>  |
| blueD!                                | Output to bang! whenever production cycles for “Defender” units for the BLUE faction.<br>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.  |
| attackRadius                          | Radius of circle in which the attackers spawn after they are produced. Defaults to zone radius   |
| attackPhi                             | Angle (direction) in degrees from zone center where attackers are spawning. Defaults to 0.   |
| paused                                | Pauses the zone’s production. “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 not unpause the zone.<br>Since paused zones will signal their capture normally, so you can wire the “conquered!” output into “activate?” to automatically activate paused zones upon capture.<br>Defaults to “no” |
| pause?                                | DML Watchflag to set paused status!<br>Triggers on change<br>Defaults to <none>  |
| activate?                             | DML Watchflag to un-pause a paused owned zone.<br>Defaults to <none>   |
| helpMe?                               | DML Watchflag that triggers a ‘defend’ cycle (will produce new defenders).<br>Defaults to <none>   |
| triggerMethod<br>factoryTriggerMethod | DML method to trigger inputs<br>Defaults to “change”   |
| method<br>factoryMethod               | DML method to bang! on outputs<br>Defaults to “inc”  |

## 18.4 Demos

- My first factory
- Clone Factory

## 19 FARP Zones

### 19.1 Summary

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

### 19.2 Dependencies

dcsCommon, cfxZones

### 19.3 ME Integration

| Name                            | Description  |
|---------------------------------|--|
| <b>FARP</b>                     | Indicates that this zone is a FARP zone. Value is ignored.<br><b>MANDATORY</b>   |
| rPhiHDef                        | Radius (in m), Phi (degrees) and Heading (degrees) of the center point around which the <b>defenders</b> deploy. Defaults to 0, 0, 0   |
| rPhiHRes                        | Radius (in m), Phi (degrees) and Heading (degrees) of the center point around which the <b>resource</b> vehicles deploy as a line. Defaults to 0, 0, 0   |
| redDefenders                    | typeStrings of defender vehicles. Example "ZSU-23-4 Shilka, ZSU-23-4 Shilka". Defaults to "none"<br>Special encoding: "none" – no vehicles   |
| blueDefenders                   | typeStrings of defender vehicles. Example "Roland ADS, Roland Radar, Roland ADS". Defaults to "none"<br>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<br>hideBlue<br>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. |

### 19.4 Demos

- FARP and away

## 20 FireFX

### 20.1 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.

### 20.2 Dependencies

fireFx requires dcsCommon, cfxZones.

### 20.3 ME Integration

| Name    | Description   |
|---------|---|
| fireFX  | <p>Tells DML that you want fire/smoke effects inside the zone. The value of this field specifies the size of the effect itself. Currently DCS recognizes the following:</p> <ul style="list-style-type: none"><li>• "small" or "S"</li><li>• "medium" or "M"</li><li>• "large" or "L"</li><li>• "huge" or "H" or "XL"</li></ul> <p>Smoke is always colored black. Flames are optional (see the 'flames' attribute)</p> <p>Note that <i>this effect is visual only</i>. Vehicles or troops inside the fire aren't damaged by the fire at all.</p> <p>Defaults to "small" (if none of the given values is recognized)</p> <p><b>MANDATORY</b></p> |
| agl     | <p>Altitude <b>in meters</b> above the ground where the effect should start. Often used with oil rigs or chimneys to create torch/smoke effects at their top.</p> <p>Defaults to 0 (directly on the ground)</p>   |
| flames  | <p>If you supply "false" for this attribute, only smoke (no fire) is displayed.</p> <p>Defaults to true (flames and smoke)</p>  |
| density | <p>"Thickness" or visibility" of the smoke produced by this effect.</p>   |
| start?  | <p>DML watchflag (input) for when the effect should start.</p> <p>Defaults to &lt;none&gt;</p>  |
| stop?   | <p>DML watchflag (input) for then the effect should stop</p> <p><b>Note:</b> unlike the smoke zone module, which can take several minutes for the smoke to stop, this fireFX's smoke and flames take roughly 10 seconds to peter out)</p> <p>Defaults to &lt;none&gt;</p>   |
| onStart | <p>If set to true, the mission starts with the effect on</p> <p>Defaults to false (no effect on start)</p>  |

| Name                             | Description  |
|----------------------------------|--|
| triggerMethod<br>fxTriggerMethod | DML method for inputs<br>Defaults to change  |
| num                              | The number (e.g. "3") or range (e.g. "2-7") of fires to start. If you supply a range, each time that you start fires in the zone, a random number from lower to upper range (inclusive) of fires is started.<br>Defaults to 1 (one fire) |
| rndLoc                           | If true, the location(s) of the fire(s) is randomized. If you are allowing more than 1 fires to be started, this value is set to true.<br>Defaults to pause (fire starts at center)  |

## 20.4 Demos

## 21 Flare Zone

### 21.1 Summary

Launches flares from a location on the map on command. Supports heavy randomization.

### 21.2 Dependencies

fireFx requires dcsCommon, cfxZones.

### 21.3 ME Integration

| Name                                | Description  |
|-------------------------------------|--|
| flare                               | <p>Tells DML that this trigger zone is a flare zone.</p> <p>The value of this attribute specifies the flare's color. Valid values are "green", "red", "white", "yellow", "random", "rnd", or "-1" (for random), "0" (for green), "1" (for red), "2" (for white) or "3" (for yellow)</p> <p>Defaults to "green"</p> <p><b>MANDATORY</b></p>   |
| launch?<br>launchFlare?<br>f?       | <p>Watchflag. When the value of this flag changes, the flares are launched. If no flag is defined, no flares can be launched.</p> <p>Defaults to &lt;none&gt;</p>  |
| triggerMethod<br>flareTriggerMethod | <p>DML Method by which the flag is triggered.</p> <p>Defaults to "change"</p>  |
| direction                           | <p>Compass heading in degrees (or a range) in which to launch the flare. The heading must be a positive number. When you supply a range, the direction will be randomized between the lower and upper bounds. Note that both numbers in the range must be positive numbers, and the upper bound must be larger than the lower. So if you want the flares to launch from 270 to 90 degrees, you need to add 260 to the second number.</p> <p>Examples:</p> <ul style="list-style-type: none"><li>• 45 (launch in NE direction)</li><li>• 90-180 (launch somewhere between East and South)</li><li>• 270-450 (launch between West and East)</li></ul> <p>Defaults to 0 (North)</p> |
| altitude<br>flareAlt<br>agl         | <p>Height (in meters) above ground where the flare is launched from.</p> <p>Defaults to 1 (m above ground)</p>   |
| salvo                               | <p>Number of flares to launch when the zone is triggered. Can be range. If you supply a range, a random number is chosen inside the range.</p> <p>Examples:</p>  |

| Name     | Description  |
|----------|--|
|          | <ul style="list-style-type: none"> <li>• 3 (launch 3 flares)</li> <li>• 2-5 (launch from 2 to 5 flares)</li> </ul> <p>Defaults to 1</p>  |
| duration | <p>Only applicable when salvo contains more than 1 flares.<br/>The time (in seconds) over which the salvo is to be launched. If, for example salvo size is 4 flares, and duration is 2 seconds, a flare will be launched every <math>2/4 = 0.5</math> seconds, so that all 4 flares are launched of the span of 2 seconds.</p> <p>Duration can be a range, in which case the duration is randomized to a value inside the range.</p> <p>Defaults to 1 (second)</p> |

## 21.4 Demos

- Effects with a flare

## 22 Ground Explosion

### 22.1 Summary

Module to set off explosions on (and above) the map. Supports multiple explosions, randomized locations, time control.

### 22.2 Dependencies

GroundExplosion requires dcsCommon and cfxZones.

### 22.3 ME Integration

| Name             | Description   |
|------------------|---|
| <b>explosion</b> | <p>Marks this zone as an explosion zone. All explosions triggered will occur within the zone (although their damage/blast may well extend past the zone).</p> <p>The value of this attribute determines the strength of the explosion. A value of 1 may be harmful to individual units, while a value of 3000 is almost sure to level a building (and others close by, depending on their strength)</p> <p>Supports ranges, so if you specify a strength of 50-230, each individual explosion will receive its own randomized strength within that range</p> <p>Defaults to 1</p> <p><b>MANDATORY</b></p> |
| triggerMethod    | <p>DML trigger method for inputs.</p> <p>Defaults to "change"</p>   |
| boom?            | <p>DML watchflag. When triggered, the module runs through an explosion cycle, setting off as many explosions as specified by the attributes</p> <p>Defaults to &lt;none&gt;</p>   |
| num              | <p>Number of explosions to perform every time that the input is triggered.</p> <p>Supports ranges, so setting <i>num</i> to "3-9" will cause groundExplosion to set off a random number between 3 and 9 (inclusive) each time that the input is triggered.</p> <p>Defaults to 1 (a single explosion)</p>  |
| rndLoc           | <p>Controls where the explosion will be located. If set to false (default) the explosion will happen at the exact center of the trigger zone.</p>   |

| Name     | Description   |
|----------|---|
|          | <p>If <i>rndLoc</i> is set to true (or <i>num</i> is greater than 1), the positions of all explosions are always randomized to occur within the area of the trigger zone.</p> <p><b>NOTE:</b><br/>If <i>num</i> is set to a value greater than one (1), the location of each explosion is always randomized inside the zone.</p> <p>Defaults to false (if a single explosion is chosen, it occurs at center of zone)</p>  |
| agl      | <p>Height (in meters) above local ground where the explosion occurs. This allows you to create air bursts, and even Flak effects.</p> <p><b>Important:</b><br/>if your target zone spans large areas with steep inclines, this can result in explosions with great variance in height, as the explosion always occurs <i>agl</i> meters above the local ground.</p> <p>Can be set to a random range (e.g. 20-30) that is applied to each explosion individually</p> <p>Defaults to 1 (m above ground)</p> |
| duration | <p>Only applicable if set to multiple explosions. Sets the time frame (in seconds) for all explosions to occur within, after being triggered. So if you set <i>num</i> to 8 and <i>duration</i> to 2, all 8 explosions occur within a span of 2 seconds.</p> <p>Defaults to 0 (all explosions happen immediately at time of triggering)</p>   |

## 22.4 Demos

- Boom boom



## 23 Group Tracker

### 23.1 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.

### 23.2 Dependencies

dcsCommon, cfxZones

### 23.3 ME Integration

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

| Name          | Description  |
|---------------|--|
| addToTracker: | <p>List of groupTracker zones. All groups that have at least one unit inside this zone are added to these groupTrackers. This happens only at mission start-up, and therefore only work for non-player-controlled planes (since player-controlled planes do not exist at mission start-up). If your player group contains AI planes, place one of those into the zone, and that group can be added to a tracker.</p> <p>If you have stacked the tracker on the same zone, you can use a single asterisk "*" as zone name.</p> <p>Supports a comma-separated list of trackers if you simultaneously want to pass the groups to multiple trackers, e.g.<br/>"GroundTrack, HeloTrack"<br/>This is useful if the zone contains more than one group, and your trackers use filtering</p> <p>Add all groups that have at least one unit in this zone to the tracker whose zone name is given in the Value field.</p> |

To add a groupTracker to a zone

| Name         | Description  |
|--------------|--|
| tracker      | <p>Marks this zone as a groupTracker. It can be referenced by the zone's name passed in the trackWith: and addToTracker: attributes. When referenced locally, a single asterisk "*" can be used as wildcard name for easy copy/paste of the entire stack</p> <p><b>MANDATORY</b></p> |
| addGroup!    | <p>Whenever a group is added to the tracker, the value of this flag is increased. If not changed by other modules, this flag also doubles as a running total of all groups added to the tracker</p> <p>Defaults to &lt;none&gt;</p>  |
| removeGroup! | <p>Whenever a tracked group is destroyed, the value of this flag is increased. If not changed by other modules, this flag also doubles as a running total of all watched groups that have been destroyed while they were tracked</p>   |

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

## 23.4 Demos

- Track This!
- Moving Spawners II
- Impossible Impostors

## 24 Guardian Angel

### 24.1 Summary

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

### 24.2 Dependencies

Guardian Angel requires dcsCommon and cfxZones

### 24.3 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.<br>Default is true  |
| launchWarning     | If true, Guardian Angel announces a missile launch.<br>Default is true   |
| intervention      | If true, Guardian Angel destroys a missile before it destroys a watched aircraft.<br>Default is true   |
| announcer         | If set to false, Guardian Angel suppresses all announcements.<br>Defaults to true  |
| msgTime           | Number of seconds that a warning from Guardian Angel stays on-screen.<br>Defaults to 30 seconds  |
| 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.<br>Respects ‘private’ attribute  |
| interventionSound | Name of the sound file to be played when Guardian Angel saves an aircraft. Respects the ‘private’ attribute.   |
| explosion         | <p>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 &gt; 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)</p> <p><b>WARNING I</b><br/>Even though this explosion is usually harmless for the protected plane, it can pose lethal to any other plane (wingmen).</p> <p><b>WARNING II</b></p> |

| Name                | Description   |
|---------------------|---|
|                     | <p>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.</p> <p>If you set this value to see explosions, make the value 1.0</p> <p>Defaults to -1 (off)</p> |
| fxDistance          | <p>When using explosions, this is the distance (in meters) away from the aircraft where the (real) explosion is going to take place.</p> <p>Defaults to 500</p>   |
| active              | <p>The state that Guardian Angel starts up in. True means that it is active, false that it is turned off.</p> <p>Defaults to 'false'</p>  |
| activate?<br>on?    | <p>Watchflag to turn on (activate) guardian angel.</p> <p>Defaults to &lt;none&gt;</p>  |
| deactivate?<br>off? | <p>Watchflag to turn off (deactivate) guardian angel.</p> <p>Defaults to &lt;none&gt;</p>   |

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

| Name            | Description   |
|-----------------|---|
| <b>guardian</b> | <p><b>MANDATORY</b></p> <p>Tells Guardian Angel how to treat aircraft inside the trigger zone:</p> <ul style="list-style-type: none"> <li><i>true</i><br/>All aircraft inside this zone are protected by Guardian Angel</li> <li><i>false</i><br/>All aircraft inside this zone will not receive protection from Guardian Angel.</li> </ul> <p>Defaults to 'true' (all aircraft inside are protected)</p> |

## 24.4 Demos

- Missile Evasion (Guardian Angel)
- Guardian Angel Reloaded

## 25 Helo Troops

### 25.1 Summary

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

### 25.2 Dependencies

**Required:** dcsCommon, cfxZones, commander, groundTroops

**Optional:** cfxSpawnZones (for requestable troop spawning)

### 25.3 ME Integration

You control Helo Troops via the configuration zone. All attributes you enter here are global to the module.

To configure the HeloTroops module via a configuration zone,

- Place a Trigger Zone anywhere using ME
- Name it "heloTroopsConfig" (note: name must match exactly)
- Add any of the following attributes to this zone:

| Name          | Description   |
|---------------|---|
| legalTroops   | Type list (comma separated) that identifies the unit types that helicopters can load. This is compared against any unit on the ground to determine if the helicopter can load the group. All units in the group must be on that list, or the entire group cannot be loaded.<br>For example, if a group consists of four infantry soldiers, the group can be loaded. If the group also contains a vehicle (e.g. "Hummer"), that group cannot be loaded.<br>Defaults to "Soldier AK, Infantry AK, Infantry AK ver2, Infantry AK ver3, Infantry AK Ins, Soldier M249, Soldier M4 GRG, Soldier M4, Soldier RPG, Paratrooper AKS-74, Paratrooper RPG-16, Stinger comm dsr, Stinger comm, Soldier stinger, SA-18 Igla-S comm, SA-18 Igla-S manpad, Igla manpad INS, SA-18 Igla comm, SA-18 Igla manpad" |
| troopCarriers | A list of helicopter types that are allowed to carry troops in this mission. Defaults to DCS Common's list of troops (which is usually Mi-8MT, UH-1H, Mi-24P), but you can provide your own list (for example to add non-official types).<br>Example: "Mi-8MT, UH-1H, SA342Minigun, C-130J" removes the Hind and adds the Gazelle in Minigun configuration as well as the Hercules fixed-wing transport to the list of legal troop carriers.  |

|                 |   |
|-----------------|---|
|                 | <p>Supports wildcard type endings: if a type ends on an asterisk ("*") all types that match whatever precedes the asterisk are accepted. For example, "Mi-*" will match both "Mi-8T" and "Mi-24P".</p> <p>You can supply the type 'any' or 'all' to allow all helicopters to carry troops.</p> <p>Default &lt;none&gt; (use dcsCommon's list of troop carriers)</p>   |
| troopWeight     | Used to calculate the cargo weight per troop loaded. Currently not used. Defaults to 100 (kg)   |
| autoDrop        | Default setting for helicopter when touching down. Players can change this individually. Defaults to true   |
| autoPickup      | Default setting for helicopter when touching down. Players can change this individually. Defaults to false  |
| pickupRang      | Range in which troops can be picked up, from a helicopter. Defaults to 100 meters   |
| combatDropScore | Score to award when a player unloads troops in a non-aligned owned zone. Requires OwnedZones and PlayerScore to be active. Defaults to 200  |
| actionSound     | Sound file to play to the helicopter group whenever troops are loaded or unloaded   |
| requestRange    | <p>Distance (in meters) that a player helicopter must land within for a spawner or cloner with 'requestable' attribute to be eligible to request troops from.</p> <p>Note that this distance is different (usually greater) than the pick-up range (see above), so helicopters can order a spawner/cloner to produce fresh troops, but may not be close enough to pick them up (i.e., the player must move closer to pick them up).</p> <p>Defaults to 500 (meters)</p> |

## 25.4 Demos

- **Helo Trooper**
- CSAR of Georgia
- Send in the Clones

## 26 Impostors

### 26.1 Summary

Impostors allow you to switch AI-controlled units to their static object equivalents (“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.

### 26.2 Dependencies

Impostors require dcsCommon, cfxMX and cfxZones.

### 26.3 ME Integration

| Name                                   | Description   |
|--|---|
| <b>impostor?</b>                       | 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.<br>The value of this attribute is a Watchflag that triggers transition of all surviving units from AI-controlled to static object.<br><br><b>MANDATORY</b> |
| 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<br>impostorTriggerMethod | Method that triggers inputs (DML Watchflags)<br>Defaults to “change”  |
| onStart                                | If the value of this attribute is true, all units are turned into impostors at the start of the mission.<br>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.<br>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<br>impostorMethod               | DML method for output<br>Defaults to “inc”  |

### 26.4 Demos

- Impossible Impostors

## 27 Limited Airframes

### 27.1 Summary

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

### 27.2 Dependencies

dcsCommon, cfxZones, cfxPlayer

### 27.3 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 Limited Airframes 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 blue coalition. Set to -1 to make the number unlimited. Defaults to -1 (unlimited) |
| red#<br>#red               | Flag that continuously is set to the current number of pilots remaining for Red  |
| blue#<br>#blue             | Flag that continuously is set to the current number of pilots remaining for blue   |
| redWins!<br>redWinsFlag!   | Flag to bang! when blue has lost all pilots and red wins. Defaults to <none>   |
| blueWins!<br>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, there 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. <ul style="list-style-type: none"><li>• If the value to this attribute contains neither the word ‘red’ nor ‘blue’, it is safe for all coalitions</li></ul> |



| Name | Description  |
|------|--|
|      | <ul style="list-style-type: none"> <li>• If the value of this attribute contains the word 'red' this zone is safe for the red coalition</li> <li>• If the value of this attribute contains the word 'blue' this zone is safe for the blue coalition</li> </ul> <p>Note that safe state may be contingent on ownership of the zone.</p> <p><b>MANDATORY</b></p> |

## 27.4 Demos

- Pilots at their limit

## 28 LZ

### 28.1 Summary

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

### 28.2 Dependencies

dcsCommon, cfxZones

### 28.3 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<br><br><b>MANDATORY</b>  |
| landed!         | Flags to bang! when a plane matching the criteria lands inside the zone<br>Defaults to <none>   |
| departed!       | Flags to bang! when a plane matching the criteria lands inside the zone<br>Defaults to <none>   |
| coalition       | When given, the coalition that aircraft must match. Supported are "red", 1, "blue", 2, "neutral" 0<br>When you specify "neutral" or 0, all coalitions match<br>Defaults to 0 (any coalition)  |
| playerOnly      | When set to true, all AI planes are ignored<br>Defaults to false (all players and AI are considered)  |
| unit<br>units   | A comma-separated list of all unit names (case <b>insensitive</b> ) that should be considered. Supports wildcard "*" for the last character, in which case all groups that match that everything before the asterisk are considered:<br>"Hog-*" will match "HOG-", "hog-1-1" and "HoG-5-2", but not "Hogger"<br>When matching units, any 'coalition' setting is ignored<br>Defaults to <none> |
| group<br>groups | A comma-separated list of all group names (case <b>insensitive</b> ) 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"<br>Defaults to <none>  |
| type<br>types   | List of types, separated by comma, that a unit must match at least one of (e.g. "A-10A"). Types must match exactly<br><br>Additionally, the following special types are also supported <ul style="list-style-type: none"><li>• "ALL" or "ANY" – all aircraft match</li><li>• "HELO" – all helicopters match</li><li>• "PLANE" – all fixed-wing planes match</li></ul>                         |

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

## 28.4 Demos

- Departures and Landings

## 29 Map Markers

### 29.1 Summary

Allows you to place markers on the F10 map.

### 29.2 Dependencies

dcsCommon, cfxZones

### 29.3 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"<br><b>MANDATORY</b> |
| coalition | Side that sees this marker. Can be "red", "blue", "neutral", or "all". You can also substitute "1" for red, and "2" for blue. Defaults to "all"                                     |

### 29.4 Demos

## 30 Messenger

### 30.1 Summary

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

### 30.2 Dependencies

dcsCommon, cfxZones

### 30.3 ME Integration

| Name       | Description  |
|------------|--|
| messenger? | Watchflag. When triggered, the module will display the message and/or play sound.<br><br><b>MANDATORY</b>  |
| message    | The text of the message to be displayed<br><br><b>FORMATTING WILDCARDS</b> <ul style="list-style-type: none"><li>• &lt;n&gt; creates a new line</li><li>• &lt;z&gt; is replaced with zone's name</li><li>• &lt;t&gt; is current time in the format as defined with the <i>timeFormat</i> attribute</li></ul> <b>RESPONSE-SELECTION WILDCARDS</b> <ul style="list-style-type: none"><li>• &lt;rsp: flag name&gt; looks up the value of &lt;flag name&gt; 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.</li><li>• &lt;rrnd&gt; randomly selects one of the possible responses and returns that response</li><li>• &lt;rhdg: unit/zone&gt; 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]</li><li>• &lt;rbea: unit/zone&gt; selects and returns one of possible responses using the bearing (in degrees) from unit/group to unit/zone as offset into <i>responses</i>. [requires group or unit attribute be set]</li></ul> <b>DATA ACCESS WILDCARDS</b> <ul style="list-style-type: none"><li>• &lt;lat&gt; the latitude of the zone's current position</li><li>• &lt;lon&gt; the longitude of the zone's current position</li><li>• &lt;mgrs&gt; the zone's current position in MGRS coordinates</li><li>• &lt;v: flagName&gt; is replaced with the value currently held by the flag <i>flagName</i></li><li>• &lt;t: flag name&gt; uses the value from flag <i>flag name</i> and interprets it as a time value, formatted according to the <i>timeFormat</i> attribute</li></ul> |

|  |  |
|--|--|
|  | <ul style="list-style-type: none"> <li>• &lt;lat: unit/zone name&gt; outputs the latitude of the zone or unit that matches the name <i>unit/zone name</i>, or “messageError” (usually an empty string) if neither can be found</li> <li>• &lt;lon: unit/zone&gt; outputs the longitude of the zone or unit that matches the name <i>unit/zone name</i>, or “messageError” (usually an empty string) if neither can be found</li> <li>• &lt;ele: flagName&gt; outputs the elevation of the zone or unit that matches the name <i>unit/zone name</i>, or “messageError” (usually an empty string) if neither can be found. Elevation is calculated in meters (default) or feet if the “imperial” attribute is true</li> <li>• &lt;latlon: unit/zone&gt; outputs the latitude and longitude of the zone or unit that matches the name <i>unit/zone name</i>, or “messageError” (usually an empty string) if neither can be found</li> <li>• &lt;lle: unit/zone&gt; outputs the longitude, longitude and elevation of the zone or unit that matches the name <i>unit/zone name</i>, or “messageError” (usually an empty string) if neither can be found</li> <li>• &lt;mgrs.: flagName&gt; outputs the mgrs. coordinates of the zone or unit that matches the name <i>unit/zone name</i>, or “messageError” (usually an empty string) if neither can be found</li> <li>• &lt;vel : unit/zone&gt; 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</li> <li>• &lt;hdg: unit/zone&gt; 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.</li> <li>• &lt;alt: unit/zone&gt; 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.</li> <li>• &lt;type: unit/zone&gt; returns the type of the unit (e.g., “A-10C”). For zones, the type returned is always “Zone”</li> <li>• &lt;player: unit&gt; outputs the player’s log-in name if that unit is controlled by a player (e.g., “New Callsign”), if the unit isn’t controlled by a player it returns “Unknown”</li> <li>• &lt;coa: flag/unit/zone&gt; outputs the faction (e.g., “RED”) of the flag/unit/zone. It returns the first name match it finds: Unit before zone before flag.</li> </ul> <p><b>“HERE/THERE” RELATIVE (requires group or unit)</b><br/> These wildcards are only available when you have supplied messenger with information about “here” – with a <i>group</i>, <i>unit</i> or <i>linkedUnit</i> attribute. “There” is the unit that is referenced in the wildcard.</p> <ul style="list-style-type: none"> <li>• &lt;bea: units/zone&gt; bearing (in degrees) from “here” to “there”</li> <li>• &lt;rng: unit/zone&gt; range (distance) from “here” to “there”</li> <li>• &lt;clk: unit/zone&gt; direction as “o’clock” to the unit/zone as seen from the player unit/zone’s heading (12 is straight ahead, 6 is behind)</li> </ul> |
|--|--|

|                                   |  |
|-----------------------------------|--|
|                                   | <ul style="list-style-type: none"> <li>• &lt;hnd: unit/zone&gt; direction “which hand” to the unit/zone: “ahead”, “right”, “behind”, “left” as seen from “here”</li> <li>• &lt;sde: unit/zone&gt; direction “side” to the unit/zone as seen from “here”: “ahead”, “starboard”, “aft”, “port”</li> <li>• &lt;asp: unit/zone&gt; aspect of “there” towards “here”. Returns “hot” / “beam” / “drag”</li> <li>• &lt;cls: unit/zone&gt; closing velocity of “here” and “there”. A negative closing velocity means that distance is growing. Closing velocity is given in km/h or knots, depending on imperialUnits</li> <li>• &lt;pcls: unit/zone&gt; precision closing velocity of “here” with “there”. Closing velocity is given in m/s or ft/s with up to one decimal, e.g., “1.3”</li> </ul> <p><b>Interpreted values</b><br/>Return one of two possible values, based on the flag’s value or existence of the unit named</p> <ul style="list-style-type: none"> <li>• &lt;yes: flagOrUnitName&gt; returns “no” if flag’s value is zero and no unit of that name exists, “yes” otherwise</li> <li>• &lt;true: flagOrUnitName&gt; returns “false” if the flag’s value is zero and no such unit exists, “true” otherwise</li> <li>• &lt;in: flagOrUnitName&gt; returns “out” if the flag’s value is zero and no such unit exists, “in” otherwise</li> <li>• &lt;alive: flagOrUnitName&gt; returns “dead” if the flag’s value is zero and no such unit exists, “alive” otherwise</li> <li>• &lt;A/B: flagOrUnitName [zero val   other val]&gt; returns whatever is inside the square brackets “[]” and left of the vertical bar “ ” (“zero val” in this case) when the flag’s value is zero and no such unit exists, and whatever is to the right of the vertical bar “ ” inside the square brackets “[]” otherwise</li> </ul> |
| responses                         | <p>A list of comma-separated possible responses that can be accessed by various wildcards. Note that the responses themselves must not contain a comma “,”.</p> <p>Example: “good, better, the best” is treated as three separate possible responses: “good”, “better” and “the best”.</p> <p>Defaults to &lt;none&gt;</p>   |
| triggerMethod<br>msgTriggerMethod | <p>Defines the trigger condition for DML Watchflags. Use only one synonym per zone</p> <p>Defaults to “change”</p>   |
| clearScreen                       | <p>If true, erase all existing messages. Defaults to false</p>   |
| soundFile                         | <p>Name of the sound file (including extension like ‘.wav’) that is to be played.</p> <p>Defaults to ‘&lt;none&gt;’. Note that the sound file’s name must be specified relative to the mission’s default location for sound files (I10n/DEFAULT/). If you use ME to import the sound files, you do not have to specify the location.</p> <p>Remember to import the sound file into the mission else no sound will play.</p>  |

|                             |   |
|-----------------------------|---|
| coalition<br>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<br>Note that if given, the attributes ‘coalition’, ‘group’ and ‘unit’ are mutually exclusive.<br>Defaults to <none> |
| group<br>msgGroup           | The name of the Groups (separated by comma ‘,’) that should receive the message/sound. Adding a group attribute enables unit-relative wildcards.<br>Note that if given, the attributes ‘coalition’, ‘group’ and ‘unit’ are mutually exclusive.<br>Defaults to <none>                    |
| unit<br>msgUnit             | The name of the Units (separated by comma ‘,’) that should receive the message/sound. Adding a group attribute enables unit-relative wildcards.<br>Note that if given, the attributes ‘coalition’, ‘group’ and ‘unit’ are mutually exclusive.<br>Defaults to <none>                     |
| messageOn?                  | When the value of this flag changes, the messenger is turned on. If it already was on, nothing happens<br>All messengers start in On state and require at least one signal on their messageOff input to disable.<br>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.<br>Defaults to <none>  |
| mute<br>messageMute         | If set to true, the messenger starts muted and requires a signal on messageOn? To activate.<br>Defaults to false  |
| duration<br>messageDuration | Time (in seconds) how long a message should stay on-screen<br>Defaults to 30 seconds  |
| timeFormat                  | Time format for any time values<br>Defaults to “<:h>:<:m>:<:s>” – standard 24 hour time   |
| imperial<br>imperialUnits   | When true, elevation is calculated in feet (imperial units) else meters.<br>Defaults to false (meters)  |
| error<br>messageError       | The text to substitute for a unit or zone reference if the units or zone cannot be found.<br>Defaults to “” (empty string)  |

## 30.4 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**

## **31 (Simple) Mission Restart**

### **31.1 Summary**

Restarts the mission when the flag named 'simpleMissionRestart' changes to any value other than 0 (zero)

### **31.2 Dependencies**

Mission must be run in multiplayer.

No other dependencies

### **31.3 ME Integration**

n/a

### **31.4 Demos**

(none)

## 32 NDB

### 32.1 Summary

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

### 32.2 Dependencies

dcsCommon, cfxZones.

### 32.3 ME Integration

| Name                              | Description  |
|-----------------------------------|--|
| NDB                               | <p>Creates an NDB at the zone's center. If the zone is linked to a unit, this NDB will automatically update to the unit's location.</p> <p>The value of this attribute is the frequency (<b>in MHz</b>) at which the NDB transmits (e.g. 121.5 for 121.5 MHz, 0.42 for 420 kHz)</p> <p>The NDB starts transmitting at mission start unless there is also a paused=true attribute present (see below)</p> <p><b>MANDATORY</b></p> |
| fm                                | <p>If true, the transmission is in FM, else in AM</p> <p>Defaults to false (AM)</p>  |
| soundFile                         | <p>Name of the sound file with extension that is to be transmitted. Defaults to '&lt;none&gt;'. Note that the sound file's name must be specified relative to the missions default location for sound files (I10n/DEFAULT/). If you use ME to import the sound files, you do not have to specify the location.</p> <p>Remember to import the sound file into the mission else no sound will play.</p>                            |
| watts                             | <p>Transmission power (in watts) for the NDB. 100 Watts usually has a range of some 150 km.</p> <p>Defaults to 100 Watts</p>   |
| paused                            | <p>If set to true, on mission start the NDB will not start up. Use the "on?" watch flag attribute or API to turn it on.</p> <p>Defaults to false</p>   |
| on?                               | <p>Watchflag. <b>Each time the flag triggers, the NDB is started (will also cause the transmission sound to rewind).</b> The current paused value is ignored, and then set to false after the NDB has started.</p> <p>Defaults to no flag to watch</p>   |
| off?                              | <p>Watchflag. <b>Each time the flag triggers, the NDB is stopped.</b> paused value is set to true after the NDB has stopped.</p> <p>Defaults to no flag to watch</p>   |
| triggerMethod<br>ndbTriggerMethod | <p>Defines the trigger condition for DML Watchflags. Use only one synonym per zone</p> <p>Defaults to "change"</p>   |

## **32.4 Demos**

- ADF and NDB Fun

## 33 NoGap / NoGapGUI

### 33.1 Summary

Fill empty player ground slots with static aircraft of the same type and livery until players slot into them. Makes airfields look much more alive. This is 'stopGaps' sibling that better resolves multi-unit player groups, but does not work with many popular scripts like SSB (simple slot block)

For multiplayer servers, install noGapGUI to avoid synchronization issues (server only)

### 33.2 Dependencies

dcsCommon, cfxZones, cfxMX

For Multiplayer, you must install the noGapGUI server extension on the server.

### 33.3 ME Integration

To *exclude* aircraft from noGap, use trigger zones and the following attributes:

| Name         | Description   |
|--------------|---|
| <b>noGap</b> | When the value of this attribute is <b>false</b> , any player aircraft inside this zone will not receive a static stand-in, their slot remains empty.<br><br>Defaults to true (unit receives a static stand-in)<br><br><b>MANDATORY</b> |

Alternatively, you can disable noGap when the mission is run in single-player mode (this is useful when, due to some map properties, the aircraft 'falls out of the sky' in single-player, but works well in multiplayer due to noGapGUI's superior synching.

| Name           | Description  |
|----------------|--|
| <b>noGapSP</b> | When the value of this attribute is <b>false</b> , any player aircraft inside this zone will not receive a static stand-in <b>in single-player mode</b> , their slot remains empty. In multiplayer mode, the slot is filled with a static stand-in. Requires noGapGui on the server.<br><br>Defaults to true (unit receives a static stand-in)<br><br><b>MANDATORY</b> |

### 33.4 Demos

- No Gap, No Problem

## 34 Object Destruct Detector

### 34.1 Summary

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

### 34.2 Dependencies

dcsCommon, cfxZones

### 34.3 ME Integration

| Name                                 | Description  |
|--------------------------------------|--|
| <b>OBJECT ID</b>                     | THIS ATTRIBUTE IS FILLED BY ME AND <b>MUST NOT</b> BE CHANGED<br><b>MANDATORY</b>                            |
| <b>NAME</b>                          | THIS ATTRIBUTE IS FILLED BY ME AND <b>MUST NOT</b> BE CHANGED<br><b>MANDATORY</b>                            |
| method<br>oddMethod                  | DML Method to bang! the flags connected to output! when the map object is destroyed.<br>Defaults to "inc"    |
| f!<br>destroyed!<br>objectDestroyed! | The flag to bang! when the map object is destroyed.<br>Use only one synonym per zone.<br>Defaults to "*none" |

### 34.4 Demos

- Object Destruct Detection

## 35 Object Spawn Zones

### 35.1 Summary

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

### 35.2 Dependencies

dcsCommon, cfxZones, (cfxCargoManager)

### 35.3 ME Integration

| Name                          | Description   |
|-------------------------------|---|
| objectSpawner                 | Marks this ME Zone as a spawn zone. <b>Value of this attribute is ignored</b> , use it to describe what it spawns to make mission editor easier for you<br><b>MANDATORY</b>   |
| f?<br>spawn?<br>spawnObjects? | An ME-compatible flag (e.g. 100) that this object spawner monitors for change. Whenever the value of the monitored flag changes, a new set of objects is spawned immediately, ignoring all maxSpawn and cooldown rules.   |
| pause?                        | Flag to observe. Each time the flag's value changes, the spawner's 'paused' setting is forced to 'true'. Used to 'pause' a spawner  |
| activate?                     | Flag to observe. Each time the flag's value changes, the spawner's 'paused' setting is forced to 'false'. Used to 'activate' a paused spawner   |
| types                         | Type string array for the STATIC OBJECTS that are spawned. Example "White_Tyre, Red_Flag". These objects may look like units (if you use the type string for a ground unit or aircraft), but they are static.<br><br><b>WARNING:</b> Blanks are part of the type, and blanks directly before and after the last character are automatically stripped.<br>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)<br><b>MANDATORY</b> |
| count                         | The number of times that the combined object in types is to be repeated. If count equals one (or is omitted), the objects defined in types are assembled in the center of the zone. Otherwise, the objects are distributed over the zone's circumference count times.<br>Defaults to one  |
| country                       | The country for which the static objects are spawned. Examples: 0 = Russia, 1 = Ukraine, 2 = USA etc.<br>Defaults to 2 (USA)  |
| baseName                      | A designation (e.g. "Hill Marines") that is used to name units and groups from during unit spawning.<br><br><b>If provided, you must ensure that resulting unit and group names are UNIQUE.</b> If you do not assign a base name, a safe baseName that is guaranteed to prevent possible name conflicts is generated.   |

| Name         | Description  |
|--------------|--|
|              | <p>A <b>convenience shortcut</b> “*” replaces baseName with the name of the zone. This is a safe baseName and can be used for all spawns.</p> <p>If two spawners have the same baseName, spawned units from one spawner may lead to existing (previously spawned) units being removed from the mission if they have the same name. So if for some reason a spawner appears not spawn units, make it a habit to check for potential name conflicts first.</p> <p>Default: &lt;auto-generated safe baseName&gt;.</p> |
| 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  |
| 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 | <p>Name of a Delicates Zone that is used to assign <i>delicate</i> (brittle, explodes when receiving minimal damage) status when this spawner spawns units</p> <p>You can use an asterisk (“*”) as wildcard to refer to this zone</p> <p>Defaults to &lt;none&gt;</p>  |

## 35.4 Demos

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



## 36 OwnAll

### 36.1 Summary

ownAll is a coordination/helper module that pulls together, and consolidates, information from multiple modules that deal with zone ownership

### 36.2 Dependencies

dcsCommon, cfxZones

### 36.3 ME Integration

| Name   | Description  |
|--------|--|
| ownAll | <p>List of the names of zones, separated by comma “,” that must be owned by the same faction and are monitored throughout the mission.</p> <p>Example: <code>Zone A, Another Zone, abc</code></p> <p>You should include at least two (2) zones to the list, else you will receive a warning</p> <p>Supports all trigger zones, including (especially, really) DML zones with managed ownership like Airfield Zones, Owned Zones and FARP Zones.</p> <p>Since ownAll zones manages its own ownership, you can also include ownAll zones to this value. Do this to create a cascade (or hierarchy) of ownership.</p> <p><b>Mandatory</b></p> |
| red#   | <p>A direct output that represents the number of zones from the list that are currently held by RED faction</p> <p>Defaults to &lt;none&gt;</p>  |
| blue#  | <p>A direct output that represents the number of zones from the list that are currently held by BLUE faction</p> <p>Defaults to &lt;none&gt;</p>   |
| total# | <p>A direct output (set only at the beginning of the mission) that represents the number of zones that are monitored</p>   |

|             |  |
|-------------|--|
| red!        | Output to send a signal on when RED possess <b>ALL</b> zones that are listed<br>Defaults to <none>   |
| blue!       | Output to send a signal on when BLUE possess <b>ALL</b> zones that are listed<br>Defaults to <none>  |
| method      | DML method for outputs (excluding direct outputs)<br>Defaults to "inc"   |
| redLine     | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the zone's outline when it belongs to the RED faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>redline</i>         |
| redFill     | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the zone when it belongs to the RED faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>redFill</i>                   |
| blueLine    | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the zone's outline when it belongs to the BLUE faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>blueLine</i>       |
| blueFill    | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the zone when it belongs to the BLUE faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>blueFill</i>                 |
| neutralLine | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the zone's outline when it belongs to the NEUTRAL faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>neutralLine</i> |
| neutralFill | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the zone when it belongs to the NEUTRAL faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>neutralFill</i>           |

## 36.4 Demos

- All is what I own

## 37 Owned Zones

### 37.1 Summary

Zones that can be conquered and generate signals when conquered

### 37.2 Dependencies

dcsCommon, cfxZones

### 37.3 ME Integration

| Name       | Description   |
|------------|---|
| owner      | <p>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)</p> <p><b>MANDATORY</b></p> <p><b>Note:</b><br/>"owner" is a zone attribute that is implicitly available for <b>all</b> zones in DML. By default, all zones are owned by the neutral faction. All zones support the 'owner' attribute, as it is a core zone ability. By adding the "OwnedZones" module to your mission, you merely change the static ownership of a zone to a dynamic property managed by the OwnedZones module.</p> <p><b>Note II:</b><br/>Originally (pre version 2.x of ownedZones), this module combined the functionality of ownedZones and factoryZones. Production ability has been moved to factoryZone.</p> |
| conquered! | <p>Flag to bang! when this zone changes hands</p> <p>Defaults to &lt;none&gt;</p>   |
| redCap!    | <p>Flag to bang! when red captures this zone</p> <p>Defaults to &lt;none&gt;</p>  |
| redLost!   | <p>Flag to bang! when red loses control over this zone. This includes the zone turning neutral/becoming contested</p> <p>Defaults to &lt;none&gt;</p>   |
| blueCap!   | <p>Flag to bang! when blue captures this zone</p> <p>Defaults to &lt;none&gt;</p>   |
| blueLost!  | <p>Flag to bang! when blue loses control over this zone. This includes the zone turning neutral/becoming contested</p> <p>Defaults to &lt;none&gt;</p>  |
| neutral!   | <p>Flag to bang! when this zone becomes neutral or contested</p> <p>Defaults to &lt;none&gt;</p>  |
| ownedBy#   | <p>When present, the flag specified here is always set to the currently owning faction: 0 (neutral), 1 (red) or 2 (blue). This can be used to create win conditions, and changes on ownership will trigger standard 'change' inputs</p> <p>Defaults to &lt;none&gt;</p>   |

| Name         | Description  |
|--------------|--|
| unbeatable   | “true” or “yes” makes it unbeatable/unconquerable. This zone can't be conquered by the other side, but may still be targeted.<br>Defaults to “no”  |
| untargetable | “true” or “yes” makes it untargetable. Zone will not be targeted by troops with ‘attackOwnedZones’. The zone may still be conquered by the other side.<br>Defaults to “no”   |
| hidden       | “true” or “yes” hides it. Zone is not shown on F10 Map.<br>Defaults to “no”  |
| redLine      | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the zone's outline when it belongs to the RED faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>redline</i>         |
| redFill      | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the zone when it belongs to the RED faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>redFill</i>                   |
| blueLine     | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the zone's outline when it belongs to the BLUE faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>blueLine</i>       |
| blueFill     | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the zone when it belongs to the BLUE faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>blueFill</i>                 |
| neutralLine  | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the zone's outline when it belongs to the NEUTRAL faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>neutralLine</i> |
| neutralFill  | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the zone when it belongs to the NEUTRAL faction. RGBA values each range from 0.0 to 1.0<br>Defaults to config zone's setting of <i>neutralFill</i>           |
| method       | DML output method for all outputs (“!”) that aren't direct (“#”) Defaults to “inc”   |

## 37.4 Demos

- Owned Zones ME Integration
- My first factory

## 38 Persistence

### 38.1 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.

### 38.2 Dependencies

Persistence requires dcsCommon and cfxZones.

### 38.3 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<br>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.<br>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.<br>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).<br>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<br><br>If you set saveDir to "", the mission saves its data directly into serverDir<br><br>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:

C:\Users\xxx\Saved Games\DCS\Missions\coolMission (Data)\coolMission Data.txt

root
serverDir
saveDir
saveFileName

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\<br>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.<br>You can use this to pool multiple missions’ data into the same folder.<br>Defaults to “<mission name> (Data)” (see note below)<br><br><b>IMPORTANT NOTE</b><br>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.<br>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”).<br><b>MANDATORY</b> |

## **38.4 Demos**

- Being persistent

## 39 Player Score

### 39.1 Summary

A module to keep award and tabulate kills and feats for players

### 39.2 Dependencies

Player Score requires the modules dcsCommon and cfxZones.

If you want to save the score table to storage, you must also include the module *persistence*.

If you want to give players access to their score via Communications, add the cfxPlayerScoreUI module

### 39.3 ME Integration

Most of PlayerScore's aspects are controlled via a configuration zone named "playerScoreConfig" (exact match required) with the following attributes

| Name       | Description   |
|------------|---|
| verbose    | A value of "true" turns on debugging messages. Default is "false"   |
| aircraft   | The fallback score to award for killing an aircraft if that unit wasn't found on the score table (name or type). Defaults to 50   |
| helo       | The fallback score to award for killing a helicopter if that unit wasn't found on the score table (name or type). Defaults to 40  |
| ground     | The fallback score to award for killing ground unit if that unit wasn't found on the score table (name or type). Defaults to 10   |
| ship       | The fallback score to award for killing a ship if that unit wasn't found on the score table (name or type). Defaults to 80  |
| train      | The fallback score to award for killing a train if that unit wasn't found on the score table (name or type). Defaults to 5  |
| pkMod      | Factor to multiply score for a player-kill (PvP). Defaults to 1 (same score)  |
| ffMod      | Factor to multiply score with for a friendly kill. Set to 0 to award no points for friendly kills. Defaults to -2 (negative double score for friendly fire kills)   |
| planeLoss  | Score to award when a player loses their airframe. This score is awarded immediately. Set i to a negative value to deduct score from a player. Defaults to 0 (zero, plane loss has no consequence)              |
| landing    | Score for a successful landing, will also award a landing feat. Is only active if the value for landing is greater than zero. Also applies to helicopters. Default: 0 (no score, no feat awarded for landings). |
| announcer  | If false, no kills are announced. Score is still kept. Defaults to true   |
| scoreSound | Name of the sound file to play when a score is announced  |
| badSound   | Name of the sound file to play when killing own troops or being killed in PvP   |
| saveScore? | Watchflag. When triggered, PlayerScore saves the current score table to a plain text file.  |



| Name              | Description   |
|-------------------|---|
|                   | Requires the 'persistence' module be active in the mission, and the hosting DCS installation be de-sanitized.<br>Defaults to <none>   |
| incremental       | When set to true, score save to plain text (via saveScore?) are appended to the save file if it already exists.<br>Defaults to false (overwrite existing player score file)   |
| rankPlayers       | When set to true, players are ranked by their achieved score when saving player score to storage<br>Defaults to false   |
| scoreOnly         | When set to true, a player's score omits all feat details on export to plain text file<br>Defaults to true  |
| deferred          | When set to true, scores and feats are only awarded after the player successfully lands and remains for a certain time in a designated 'safeScore' zone.<br>Defaults to false (immediately awards score)  |
| delayAfterLanding | Number of seconds a player must remain in a safeScore zone after a successful landing before they are rewarded their accumulated score. Should they die in the interim, all accumulated scores are discarded. Only applicable if deferred is true.<br>Default: 10 (seconds)   |
| scoreFileName     | Name of the plain text file (without extension) that scores are exported to when saveScore? is triggered.<br>Defaults to "Player Scores"  |
| reportScore       | When queried to report the player's statistics, should the module include score? Set to false to suppress the player's score.<br>Defaults to true (score is reported)   |
| reportFeats       | When set to true, the module also reports feats when queried for the player's statistics.<br>Defaults to true (feats are reported)  |
| reportCoalition   | When set to true, the module reports the coalition's total with a player's score.<br>Defaults to false (no coalition score reported)  |
| noGrief           | Controls if negative player scores also affect the player's coalition score. By default, to prevent 'griefing' (i.e. childish behavior), when a player is awarded demerits (negative score), that is only subtracted from the player's total, but not the coalition. That way, a griefing player can't intentionally sabotage a coalition's score by committing fratricide or crashing planes.<br>When set to false, this protection is disabled, and griefing would be possible.<br>Defaults to true (griefing protection is on, only positive player scores affect coalition score) |
| redScore#         | Immediate Output flag that always contains the current total score for RED faction<br><br>Defaults to <none>  |
| blueScore#        | Immediate Output flag that always contains the current total score for BLUE faction<br><br>Defaults to <none>   |

You can also set up a mission-specific score table via a zone called “playerScoreTable” (exact match required) that can include named units and unit types as attributes as follows:

| Name                        | Description  |
|-----------------------------|--|
| <type or name>              | <Score as number>  |
| Type Exampe:<br>BTR-80      | Example:<br>15   |
| Name Example:<br>Big Kahuna | Example:<br>130<br>Note: The name is first checked again a unit’s name, and then against the unit’s group. |

If you enable deferred scoring in the config zone, you designate zones in which scores are awarded after landing by adding the following attributes:

| Name             | Description  |
|------------------|--|
| <b>scoreSafe</b> | Designates this entire zone as a zone where players can land inside and, after successfully landing and remaining the required amount of time (see <i>delayAfterLanding</i> config attribute) inside the zone, have their accumulated score and feats awarded.<br>The value of this attribute determines which faction can have their scores awarded after landing as follows: <ul style="list-style-type: none"> <li>• 0 or ‘neutral’: all factions</li> <li>• 1 or ‘red’: only red faction</li> <li>• 2 or ‘blue’: only blue faction</li> </ul> Defaults to ‘neutral’ (both red and blue can land here to have their scores awarded)<br><br><b>Mandatory</b> |

You can place feats in ME with trigger zones that have the following attributes:

| Name        | Description  |
|-------------|--|
| <b>feat</b> | Designates this zone as a zone in which the completion of certain actions leads to a feat being awarded to the player who completes it.<br><br>The value of this attribute can restrict the feat to certain factions: <ul style="list-style-type: none"> <li>• 0 or <i>neutral</i> – all players can achieve this feat</li> <li>• 1 or <i>red</i> – only red players can achieve this feat</li> <li>• 2 or <i>blue</i> – only blue players can achieve this feat</li> </ul> Defaults to 0 (all players can achive this feat)<br><br><b>Mandatory</b> |
| featType    | What a player needs to accomplish in order to have this feat awarded. Currently supports the following <ul style="list-style-type: none"> <li>• <i>land</i> or <i>landing</i> – player lands their unit inside this zone</li> <li>• <i>kill</i> – players kills a unit that is currently inside this zone</li> <li>• <i>pvp</i> – player kills a unit that is currently inside this zone and is controlled by another player.</li> </ul>   |

|             |   |
|-------------|---|
|             | Defaults to <code>kill</code>   |
| description | <p>Text of the feature that is to be awarded to the player. The description supports <b>wildcards</b> as follows.</p> <p>Feat-specific Wildcards:</p> <ul style="list-style-type: none"> <li>• <code>&lt;player&gt;</code> - callsign of the player</li> <li>• <code>&lt;kplayer&gt;</code> - callsign of other player when that plane was controlled by a player, 'unknown AI' else.</li> <li>• <code>&lt;punit&gt;</code> - name of the unit that the player is controlling</li> <li>• <code>&lt;unit&gt;</code> - name of unit that was killed</li> <li>• <code>&lt;punit&gt;</code> - type (e.g. "A-10A") of the unit that the player is controlling</li> <li>• <code>&lt;type&gt;</code> - type (e.g. "A-10A") of the unit that was killed</li> <li>• <code>&lt;pgroup&gt;</code> - name of the group that the player's unit is with</li> <li>• <code>&lt;group&gt;</code> - name of the group that the unit that was killed belonged to</li> </ul> <p>General Wildcards</p> <ul style="list-style-type: none"> <li>• <code>&lt;n&gt;</code> creates a line feed</li> <li>• <code>&lt;z&gt;</code> - feat zone's name as set with ME</li> <li>• <code>&lt;t&gt;</code> - current mission time in the format "HH:MM:SS", e.g. "08:42:11".</li> <li>• <code>&lt;lat&gt;</code> - latitude of the feat zone's current position</li> <li>• <code>&lt;lon&gt;</code> - longitude of the feat zone's current position</li> <li>• <code>&lt;ele&gt;</code> - elevation of the feat zone's current position.</li> <li>• <code>&lt;mgrs&gt;</code> - feat zone's current position in MGRS coordinates</li> <li>• <code>&lt;v: flagName&gt;</code> - value of flag <code>&lt;flagName&gt;</code></li> <li>• <code>&lt;t: flagName&gt;</code> - value of flag <code>&lt;flagName&gt;</code> interpreted as time. Format be further configured with the <i>timeFormat</i> attribute.</li> <li>• <code>&lt;lat: unit/zone&gt;</code> - latitude of unit/zone with that name</li> <li>• <code>&lt;lon: unit/zone&gt;</code> - longitude of unit/zone with that name</li> <li>• <code>&lt;ele: unit/zone&gt;</code> - elevation of unit/zone with that name.</li> <li>• <code>&lt;mgrs: unit/zone&gt;</code> - mgrs of unit/zone with that name</li> <li>• <code>&lt;vel: unit/zone&gt;</code> - velocity of unit/zone with that name.</li> <li>• <code>&lt;alt: unit/zone&gt;</code> - altitude of unit/zone with that name.</li> <li>• <code>&lt;hdg: unit/zone&gt;</code> - heading of unit/zone with that name</li> <li>• <code>&lt;type: unit&gt;</code> - type (e.g. "A-10A") of unit with that name</li> <li>• <code>&lt;player: unit&gt;</code> - player callsign who controls a unit with that name</li> </ul> <p>Defaults to <code>(some feat)</code></p> |
| awardLimit  | Total number of times that this feat can be awarded during the life time of a mission. A negative value means 'infinite number of times'<br>Defaults to -1 (feat can be accomplished an infinite number of times)   |
| awardOnce   | If set to true, this feat can only be awarded once per player. Note that awardLimit and awardOnce work cumulative: if you set awardOnce to true, and awardLimit to 2, the feat can be accomplished twice, but only by different players.<br>Defaults to 'false', a player can achieve this feat multiple times during the same mission  |

You can restrict all scorable kills to kill zones (units can still be killed outside kill zones, but killing them will not award score to players)

| Name     | Description   |
|----------|---|
| killZone | Designates this zone as a zone in which killing other units can earn score or feats. All kills outside this or other kill zones do not award a score or feat.<br>To count, at least the target must be inside a kill zone, optionally (see <i>duet</i> attribute, below) you can also enforce that the victorious player must be inside the same kill zone.<br><br>The value of this attribute is ignored<br><br><b>Mandatory</b> |
| duet     | If set to true, <i>both</i> units (player and target) must be inside the <i>same</i> kill zone.<br>Defaults to 'false' (only killed unit must be inside kill zone)  |

### Flags to trigger coalition score

You can use specially named zones "blueScoreFlags" (for blue coalition) and "redScoreFlags" (red coalition) to award score whenever one of the flags listed as attribute name changes its value. The spelling of the zone's names must match exactly.

| Name            | Description  |
|-----------------|--|
| <any flag name> | Whenever the value of the flag listed here changes, PlayerScore adds the amount given in the Value field to the relevant coalition. Negative values are valid. If you give a value of 0 (zero) or non-numeric value, PlayerScore will give you a warning when it starts the mission.<br><br>The value of this attribute is ignored |

## 39.4 Demos

- PlayerScore (Keeping The score)
- More Score
- Later Score
- Flag Score
- Player Score to Win

## 40 Player Zone

### 40.1 Summary

Counts the number of players currently in a zone, and generates signals when players enter or leave the zone.

### 40.2 Dependencies

Player Zone requires the modules dcsCommon and cfxZones.

### 40.3 ME Integration

| Name               | Description   |
|--------------------|---|
| <b>playerZone</b>  | <p>Tells DML that this trigger zone is a player zone.</p> <p>The value of this attribute specifies which faction's players are counted. Possible values</p> <ul style="list-style-type: none"><li>• "0" or "neutral" – red and blue are counted</li><li>• "1" or "red" – only red players are counted</li><li>• "2" or blue – only blue players are counted</li></ul> <p>Defaults to "neutral" (red and blue players are counted)</p> <p><b>MANDATORY</b></p> |
| method<br>pwMethod | <p>DML output method.<br/>Defaults to "inc"</p>   |
| pNum#              | <p>Direct output: the number of qualifying player units that are inside the zone at this moment.<br/>Defaults to &lt;none&gt;</p>   |
| added!             | <p>Output to bang! when players units have entered the zone since the last time the zone checked. Note that there are edge cases where added! and removed! can signal a change even though the flag pNum remains the same: for example if at the same time one player unit leaves the zone (or is killed), and another player unit enters the zone.<br/>Defaults to &lt;none&gt;</p>  |
| removed!           | <p>Output to bang! when plyer units have left the zone since the last time that the zone has cheked. Note that there are edge cases where added! and removed! can signal a change even though the flag pNum remains the same: for example if at the same time one player unit leaves the zone (or is killed), and another player unit enters the zone.<br/>Defaults to &lt;none&gt;</p>   |

### 40.4 Demos

- Players in the Zone

## 41 Pulse Flags

### 41.1 Summary

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

### 41.2 Dependencies

dcsCommon, cfxZones

### 41.3 ME Integration

| Name                                  | Description  |
|---------------------------------------|--|
| <b>pulse!</b>                         | Marks this as a pulser. The value describes the flags to change on each pulse. The flags are changed according to the method attribute<br><b>MANDATORY</b>   |
| method<br>pulseMethod<br>outputMethod | DML Flag output method<br>Defaults to “flip”   |
| done!<br>pulsesDone!                  | This flag’s value is changed when the pulser completes a fully run of pulses. Can only happen when the <i>pulses</i> attribute supplies a positive number. Use only one synonym per zone.<br>Defaults to <none>  |
| triggerMethod<br>pulseTriggerMethod   | Watchflag condition for input flags. Use only one synonym per zone   |
| activate?<br>startPulse?              | Watchflag. When triggered, a paused pulser is reset and then restarted. Use only one synonym per zone.<br>Defaults to <none>   |
| pause?<br>pausePulse?                 | Watchflag. When triggered, a pulser is paused. Use only one synonym per zone.<br>Defaults to <none>  |
| onStart                               | When set to false, a pulser starts paused, else active.<br><br><b>Defaults to true</b> (pulser starts automatically)   |
| pulses                                | The number of pulses to complete. <ul style="list-style-type: none"><li>• If set to -1, the pulser runs until the mission ends or the pause?-flag changes</li><li>• If set to a number, the pulser will generate that many pulses.</li><li>• If set to a range (e.g. “3-5”) the pulser will generate a random number of pulses within that range.</li></ul> Defaults to -1 (endless) |
| time<br>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.<br>Defaults to 1  |
| zeroPulse                             | Usually, a pulser starts with an initial pulse (“pulse zero”). This initial pulse can be delayed by <i>time</i> by setting zeroPulse to false The effect is that the initial pulse happens after the first delay   |

| Name | Description                                 |
|------|---|
|      | Default is true (initial pulse immediately) |

## 41.4 Demos

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

## 42 Radio Menu

### 42.1 Summary

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

### 42.2 Dependencies

Radio Menu requires dcsCommon and cfxZones.

### 42.3 ME Integration

| Name            | Description   |
|-----------------|---|
| radioMenu       | Name of the menu to install in the Communications→F10 Other menu.<br><br><b>MANDATORY</b>   |
| coalition       | The coalition that has access to this menu. If omitted or set to 'neutral', <i>all</i> coalitions have access. 'blue' or 'red' restricts access to that coalition.<br>Defaults to <no coalition>  |
| group<br>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<br><b>Overrides any 'coalition' and 'type' attribute;</b> when you add a 'group' attribute, those other attributes are ignored.<br><br>Defaults to <no group restriction><br><br><b>NOTE:</b> requires module cfxMX to load before radioMenus  |
| type<br>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")<br><br>Supports the class-wildcards 'plane' (all fixed-wing players) and 'helo' (all rotor-wings controlled by players).<br><br>When you also supply a 'coalition' attribute, access to this menu is restricted to those players who match both.<br><br>Defaults to <no type restriction><br><br><b>NOTE:</b> requires module cfxMX to load before radioMenus |
| itemA           | Name of itemA in this menu. If this attribute is omitted, no menu item appears  |
| itemB           | Name of itemB in this menu. If this attribute is omitted, no menu item appears  |



| Name      | Description   |
|-----------|---|
| 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>         |
| valA      | Overrides radioMethod (see below) when this item is chosen to set flag A! according to this method.<br>Example: #3<br>Defaults to <none> (no override)                                  |
| valB      | Overrides radioMethod (see below) when this item is chosen to set flag B! according to this method.<br>Example: #3<br>Defaults to <none> (no override)                                  |
| valC      | Overrides radioMethod (see below) when this item is chosen to set flag C! according to this method.<br>Example: #3<br>Defaults to <none> (no override)                                  |
| valD      | Overrides radioMethod (see below) when this item is chosen to set flag D! according to this method.<br>Example: #3<br>Defaults to <none> (no override)                                  |
| ackA      | Acknowledge message when itemA was selected. Broadcast to all/coalition when itemA was chosen and not on cooldown. Supports wildcards.<br>Defaults to <none> - no acknowledging message |
| ackB      | Acknowledge message when itemB was selected. Broadcast to all/coalition when itemB was chosen and not on cooldown. Supports wildcards.<br>Defaults to <none> - no acknowledging message |

| Name                  | Description  |
|-----------------------|--|
| ackC                  | Acknowledge message when itemC was selected. Broadcast to all/coalition when itemC was chosen and not on cooldown. Supports wildcards.<br>Defaults to <none> - no acknowledging message  |
| ackD                  | Acknowledge message when itemD was selected. Broadcast to all/coalition when itemD was chosen and not on cooldown. Supports wildcards.<br>Defaults to <none> - no acknowledging message  |
| method<br>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<br>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<br>Default to true (menu is visible on mission start)               |

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'

Additionally, busyX and ackX also support the full gamut of wildcards that valet and messenger support.

## **42.4 Demos**

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

## 43 Radio Trigger

### 43.1 Summary

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

### 43.2 Dependencies

dcsCommon, cfxZones

### 43.3 ME Integration

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

### 43.4 Demos

- Radio go go

## 44 Raise Flag

### 44.1 Summary

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

### 44.2 Dependencies

dcsCommon, cfxZones

### 44.3 ME Integration

| Name                                | Description  |
|-------------------------------------|--|
| <b>raiseFlag!</b>                   | Marks this as a flag raiser. The value of this attribute is the flag that is to be raised after a delay.<br><br><b>MANDATORY</b>   |
| method                              | DML method to set the flag to<br><br><b>Examples:</b> <ul style="list-style-type: none"><li>• inc - increment the flag specified in raiseFlag!</li><li>• #5 – set the flag specified in raiseFlag! to the number 5</li></ul> Defaults to 'inc' – the flag's value will be incremented by one |
| 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.<br><br>Defaults to 0.5 seconds after mission start  |
| stopFlag?                           | Watchflag, only useful in conjunction with afterTime. When triggered and raiseFlag is still waiting for afterTime, raiseFlag is 'disarmed' and no flag will be raised in the future. Once stopped, raiseFlag cannot be re-started.   |
| triggerMethod<br>raiseTriggerMethod | Watchflag condition for stopFlag?  |

### 44.4 Demos

- Flag Fun
- Attack of the CloneZ
- Send in the Clones

## 45 Recon Mode

### 45.1 Summary

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

### 45.2 Dependencies

Tbc

### 45.3 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.<br>Defaults to 'true'<br><b>MANDATORY</b> |
| 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.<br>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<br>Defaults to "prio"<br><b>MANDATORY</b>  |
| 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.<br>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: <ul style="list-style-type: none"><li>• &lt;n&gt; creates a new line</li><li>• &lt;z&gt; is replaced with zone's name</li><li>• &lt;t&gt; is current time in HH:MM:SS format</li><li>• &lt;lat&gt; the latitude of the discovered group's current position</li><li>• &lt;lon&gt; the longitude of the discovered group's current position</li><li>• &lt;ele&gt; the elevation (in feet or meters, as determined by the imperialUnits attribute in reconModeConfig)</li><li>• &lt;mgrs&gt; the discovered group's current position in MGRS coordinates</li></ul> |

| Name     | Description  |
|----------|--|
|          | <i>Only applicable for priority targets</i> (i.e. recon's value is something other than "black"), ignored otherwise<br>Default is <none>, i.e., no message is displayed  |
| spotted! | DML output flags to bang when a group is spotted. Flag is banged <i>in addition</i> to the module's global "prio!" flag. Uses the config zone's method.<br><br><i>Only applies to prio zones</i> (ignored when blacklist zone)<br>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.<br>Defaults to false |

### 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.<br><br><b>NOTE</b><br>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)<br>Default: true |
| redScouts       | If true, all red planes are included as scouts when autoRecon is true. Default is false   |
| blueScouts      | If true, all blue planes are included as scouts when autoRecon is true. Default is <b>true</b>  |
| greyScouts      | If true, all neutral planes are included as scouts when autoRecon is true. Default is false   |
| playerOnlyRecon | If true, only player aircraft are included as scouts when autoRecon is true. All planes will not be automatically included as scouts.<br><br><b>IMPORTANT</b><br>This condition is applied <b>in addition</b> to blueScouts and redScouts. If you disallow red scouts, red players will not automatically be added to the list of scouts.<br>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   |
| marksFadeAfter  | Time (in seconds) after which a mark on the F10 map is automatically removed.<br>Set to a negative value (e.g., -1) for 'eternity' (marks never time out)<br>Defaults to 30*60 = 1800 seconds (= 30 minutes)  |
| marksLocked     | When set to true, marks cannot be removed by players, and will disappear automatically after they time out (see marksFadeAfter, above) or the group is destroyed (when autoRemove is true)  |

| Name                      | Description   |
|---------------------------|---|
|                           | Defaults to false (player can remove marks)   |
| announcer                 | If true, discovered groups are announced via text. Default is true  |
| detectionMinRange         | The detection range of a recon plane under worst conditions (low-level flying). Default is 3000 (3 km)  |
| detectionMaxRange         | The detection range of a recon plane under best conditions (high-altitude). Default is 12000 (12 km)  |
| maxAlt                    | The altitude at which a plane achieves maxDetectionRange. Default is 9000 (9 km, 27'000 ft)   |
| 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 a 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<br>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.<br>Default is false (coordinates are displayed in Lat/Lon) |
| imperial<br>imperialUnits | When set to true, the value given in <ele> (elevation) is in feet, otherwise in meters<br>Defaults to false (ele is returned in meters)   |
| activate?<br>on?          | Watchflag to monitor for a change. If a change is detected, Recon Mode goes into active state.<br>Defaults to <none>  |
| deactivate?<br>off?       | Watchflag to monitor for a change. If a change is detected, Recon Mode turns off<br>Defaults to <none>  |
| active                    | Set to false to start Recon Mode in disabled (off) state.<br>Defaults to true (Recon Mode enabled)  |
| reportTime                | Number of seconds that a recon message stays on the screen.<br>Defaults to 30.  |

## 45.4 Demos

- Recon Mode
- Recon Mode - Reloaded



## 46 Rnd Flags

### 46.1 Summary

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

### 46.2 Dependencies

dcsCommon, cfxZones

### 46.3 ME Integration

| Name                              | Description   |
|-----------------------------------|---|
| <b>RND!</b>                       | Marks this as a randomizer.<br><br>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.<br>Examples:<br>"2, 4, A, A, F, 6"<br>"A9, 3-18, C33, samAttack, 11-11"<br><br><b>MANDATORY</b> |
| method<br>rndMethod               | DML Output flag method.<br><b>Defaults to "inc"</b>   |
| pollSize                          | Number of items to choose from the set of flags during a cycle. Can be a range: two numbers separated by a hyphen, e.g. "2-5". When a range is given, pollSize is randomized each cycle to a number between the lower and upper bounds, inclusive.<br>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<br>Defaults to false  |
| f?<br>in?<br>rndPoll?             | DML Watchflag to start a random cycle. Defaults to <none set><br>You can use any synonym, but only one per Zone   |
| triggerMethod<br>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.<br>Defaults to false<br><br><b>NOTE:</b><br>if no rndPoll? (nor synonym) is specified, and onStart is false, the randomizer will never activate. You'll receive a warning.   |

| Name  | Description   |
|-------|---|
|       | If you specify neither onStart nor an input rndPoll? (or synonym), onStart is automatically set to true, so that the randomizer runs exactly once, 0.25 seconds after mission start.                        |
| done! | The flag number to bang! when the randomizer has run out of flags to change, and reshuffle is false (randomizer did nothing)<br>Is banged! every time that the randomizer runs a cycle on an empty flag set |

## 46.4 Demos

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

## 47 Scribe

### 47.1 Summary

This module **records all players accomplishments** (engine start-ups, time in a type, landings, take-offs, crashes) and can – per player – give them a summary. Supports persistence so a player can see their accumulated stats **across multiple replays**.

### 47.2 Dependencies

Scribe requires dcsCommon, cfxZones and cfxMX

Scribe requires persistence to persist data

Scribe records all successful rescues when csarManager is active.

### 47.3 ME Integration

Scribe currently uses no map-specific zone, all features are controlled through the config zone

### 47.4 Demos

- On the Record

## 48 Sequencer

### 48.1 Summary

Creates signals in a pre-determined sequence of flags

### 48.2 Dependencies

Sequencer requires dcsCommon, cfxZones.

### 48.3 ME Integration

| Name                  | Description  |
|-----------------------|--|
| sequence!             | <p>A comma-separated list of flags that should be banged! in exactly that order. Example: "startGround, startSAM, startSirens"</p> <p>Value of this attribute is ignored</p> <p><b>MANDATORY</b></p>   |
| interval<br>intervals | <p>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"</p> <p>Defaults to 86400 (24 hours)</p> |
| next?                 | <p>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 &lt;none&gt;</p>   |
| onStart               | <p>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).</p>   |
| zeroSequence          | <p>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)</p>   |
| loop                  | <p>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</p>  |
| done!<br>seqDone!     | <p>The signal to send when the sequencer has ended the sequence Defaults to &lt;none&gt;</p>   |
| startSeq?             | <p>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.</p>   |

| 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.<br>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.<br>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.<br>Defaults to <none> |
| method<br>seqMethod               | The method to use for all outputs.<br>Defaults to 'inc'  |
| triggerMethod<br>seqTriggerMethod | The trigger method for all inputs.<br>Defaults to "change"   |

## 48.4 Demos

- Sequencing Fun

## **49 SittingDucks**

### **49.1 Summary**

A module that allows 'destructible player slots' for multiplayer.

### **49.2 Dependencies**

SittingDucks requires dcsCommon, cfxZones. And stopGap

To run, the mission must as multiplayer, and the server must run the following add-ons:

- SSB (simple slot block)
- stopGapGUI

### **49.3 ME Integration**

SittingDucks requires no ME integration past the config zone

### **49.4 Demos**

- Sitting Ducks in a Barrel

## 50 Smoke Zones

### 50.1 Summary

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

### 50.2 Dependencies

dcsCommon, cfxZones

### 50.3 ME Integration

| Name                                | Description   |
|-------------------------------------|---|
| smoke                               | Adds a permanent smoke effect to the center of the zone. Possible values for the smoke effect are: <ul style="list-style-type: none"><li>• “green” or “0”</li><li>• “red” or “1”</li><li>• “white” or “2”</li><li>• “orange” or “3”</li><li>• “blue” or “4”</li><li>• “random”, “?” or “rnd” (random color from above)</li></ul> <b>MANDATORY</b> |
| paused                              | When true, will not start smoke at mission start, but wait for a signal on the <i>startSmoke?</i> flag.<br>Defaults to false (smoke starts at mission beginning). Note that if you set paused to true and omit a <i>startSmoke?</i> attribute, you cannot start the smoke.  |
| startSmoke?<br>f?                   | Watchflag. When this input is triggered via the connected flag, smoke starts.<br>Defaults to <none>   |
| stopSmoke?                          | Watchflag. When this input is triggered via the connected flag, the smoke effect is not renewed when it times out. Note that <b>it can take up to 5 minutes</b> (unlike fireFX, there are no provisions in DCS for smoke to be extinguished immediately, so we have to wait for it to time out).<br>Defaults to <none>                            |
| agl<br>alt<br>altitude              | Altitude (in meters) above ground that the smoke should be created.<br>Defaults to 1 meter  |
| triggerMethod<br>smokeTriggerMethod | Conditions when the DML Watchflag should trigger<br>Defaults to ‘change’  |

### 50.4 Demos

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

## 51 Spawn Zones

### 51.1 Summary

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

### 51.2 Dependencies

dcsCommon, cfxZones, commander, groundTroops, (Helo Troops)

### 51.3 ME Integration

| Name                        | Description  |
|-----------------------------|--|
| spawner                     | <p>Marks this ME Zone as a spawn zone.</p> <p>The value for this attribute is a “type string” list that describes the ground units that are to be spawned. Example “Roland ADS, Roland Radar, Roland ADS” or “Soldier M4” – <b>WARNING</b>: Blanks are part of the type, and blanks before and after the last character are automatically stripped.</p> <p>For a full reference of objects and their types, see here <a href="https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB">https://github.com/mrSkortch/DCS-miscScripts/tree/master/ObjectDB</a> and use whatever is given as value for the “typeName” attribute, e.g. “Soldier M249” for the “INF Soldier M249.lua”</p> <p><b>MANDATORY</b></p> |
| f?<br>spawn?<br>spawnUnits? | <p>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.</p> <p>Defaults to no flag to observe</p> <p>Use only one synonym per zone</p>  |
| pause?                      | <p>Flag to observe. Each time the flag’s value changes, the spawner’s ‘paused’ setting is forced to ‘true’. Used to ‘pause a spawner</p>   |
| activate?                   | <p>Flag to observe. Each time the flag’s value changes, the spawner’s ‘paused’ setting is forced to ‘false’. Used to ‘activate’ a paused spawn</p>   |
| country                     | <p>The country (a number) the units that spawn belong to, e.g. “22” for Switzerland (<b>Warning</b>: unlike many other zone extensions, we use a County, not a Coalition here. The coalition is determined by which Faction the country belongs to as is defined when you create the mission, or by using the faction editor.</p> <p>Common Countries are Russia = 0, Ukraine = 1, USA = 2, UN Peace Keepers = 82</p> <p>You can find a reference of all country codes here: <a href="https://wiki.hoggitworld.com/view/DCS_enum_country">https://wiki.hoggitworld.com/view/DCS_enum_country</a> ).</p>  |
| masterOwner                 | <p>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</p>  |



| Name       | Description   |
|------------|---|
|            | <p>away. You can use this to start spawning reinforcements in a completely unrelated part of the map when units conquer the masterOwner zone.</p> <p>If no masterOwner is specified, the Spawner spawns as directed and disregards any surrounding zones that happen to be owned</p> <p>Optional, defaults to empty</p>   |
| baseName   | <p>A designation (e.g. "Hill Marines") that is used to name units and groups from during unit spawning.</p> <p><b>If provided, you must ensure that resulting unit and group names are UNIQUE.</b> If you do not assign a base name, a safe baseName that is guaranteed to prevent possible name conflicts is generated.</p> <p>A <b>convenience shortcut</b> "*" replaces baseName with the name of the zone. This is a safe baseName and can be used for all spawns.</p> <p>If two spawners have the same baseName, spawned units from one spawner may lead to existing (previously spawned) units being removed from the mission if they have the same name. So if for some reason a spawner appears not spawn units, make it a habit to check for potential name conflicts first.</p> <p>Default: &lt;auto-generated safe baseName&gt;.</p> |
| 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 | <p>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.</p> <p>Defaults to 'false'</p>   |
| heading    | The direction the spawned group is oriented to, from the center of the spawn zone. Defaults to 0  |
| formation  | Formation of the spawned group. See dcsCommon for supported group formations. Defaults to 'circle_out'.   |
| paused     | When paused, a spawner only spawns when other scripts tell it to (e.g. your own scripts, cfxHeloTroops, triggers). Defaults to "no"   |
| orders     | This is an optional interface to other troop-governing modules, e.g. groundTroops. Default is "guard", and spawners support in addition to those that groundTroops support  |
| range      | An attribute used to pass a range value to orders (e.g. JTAC laze range, detection/engage range)  |
| target     | An attribute used to pass a target zone when used in conjunction with the 'attackZone' orders   |
| maxSpawns  | The maximum number of times that this spawner spawns groups. Set it to a positive number (e.g. 3) to spawn that many times. Set it to a   |

| Name         | Description  |
|--------------|--|
|              | negative number for an unlimited number of spawns (default is -1). Set it to zero (0) and the spawner will never spawn.  |
| requestable  | Interfaces with other scripts, if you set this value to true, troops will only spawn on request via <code>cfxSpawnZones.spawnWithSpawner()</code> . See the API section on how to get a list of eligible spawners. Automatically interfaces with HeloTroops and other enhancements   |
| trackWith:   | List of groupTracker zones. All spawned groups are added to these groupTrackers.<br>If you have stacked the tracker on the same zone as the spawner, you can use a single asterisk '*' as zone name.<br>Supports a comma-separated list of trackers if you simultaneously want to pass the spawned groups to multiple trackers<br>Defaults to <None> |
| useDelicates | Name of a Delicates Zone that is used to assign <i>delicate</i> (brittle, explodes when receiving minimal damage) status when this spawner spawns units<br>You can use an asterisk ("*") as wildcard to refer to this zone<br>Defaults to <none>   |

## 51.4 Demos

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

## 52 ssbClient

**REQUIRES THAT SSB INBTALLED ON HOSTING MP SERVER *AND* MISSION TO RUN AS MP**

### 52.1 Summary

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

### 52.2 Dependencies

ssbClient requires dcsCommon, cfxGroups and cfxZones

### 52.3 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.<br>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.<br>Defaults to false   |
| reUseAfter         | If singleUse is <b>enabled</b> , 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.<br>Defaults to -1 (remain blocked)   |
| allowNeutralFields | If set to “true”, aircraft can spawn on neutral airfields (otherwise they are blocked).<br>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.<br>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.<br>Defaults to false |
| enabledFlagValue   | This reflects SSB’s flag value of that same name. <b>DO NOT CHANGE THIS UNLESS YOU ARE ABSOLUTELY SURE YOU KNOW WHAT YOU ARE DOING.</b><br>Defaults to 0  |

| Name             | Description   |
|------------------|---|
| enabledFlagValue | This reflects SSB's flag value of that same name. <b>DO NOT CHANGE THIS UNLESS YOU ARE ABSOLUTELY SURE YOU KNOW WHAT YOU ARE DOING.</b><br>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  |
|------------------|--|
| <b>ssbClient</b> | Marks this zone as a control zone <b>for the airfield/FARP that is closest</b> to this zone.<br><b>MANDATORY</b>   |
| open?            | DML Input Watchflag. When triggered, this airfield "opens", allowing planes spawn from this airfield. Note that ownership rules still apply.<br>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.<br>Defaults to <none> |
| openOnStart      | When set to false, the airfield is closed on mission start, all slots for aircraft spawning here are blocked<br>Defaults to true (airfield is open, allowing slots to spawn according to faction ownership of airfield/FARP)                               |
| ssbTriggerMethod | Method for Watchflags.<br>Defaults to "Change"   |

## 52.4 Demos

- Slot-Blocking and You

## 53 StopGap / StopGapGUI

### 53.1 Summary

Fill empty player ground slots with static aircraft of the same type and livery until players slot into them. Makes airfields look much more alive.

For multiplayer servers, install stopGapGUI to avoid synchronization issues (server only)

### 53.2 Dependencies

dcsCommon, cfxZones, cfxMX

For Multiplayer, you must install the stopGapGUI server extension on the server.

### 53.3 ME Integration

To *exclude* aircraft from stopGap, use trigger zones and the following attributes:

| Name           | Description   |
|----------------|---|
| <b>stopGap</b> | When the value of this attribute is <b>false</b> , any player aircraft inside this zone will not receive a static stand-in, their slot remains empty.<br><br>Defaults to true (unit receives a static stand-in)<br><br><b>MANDATORY</b> |

Alternatively, you can disable StopGap only when the mission is run in single-player mode (this is useful when, due to some map properties, the aircraft 'falls out of the sky' in single-player, but works well in multiplayer due to StopGapGUI's superior syncing.

| Name             | Description  |
|------------------|--|
| <b>stopGapSP</b> | When the value of this attribute is <b>false</b> , any player aircraft inside this zone will not receive a static stand-in <b>in single-player mode</b> , their slot remains empty. In multiplayer mode, the slot is filled with a static stand-in. Requires stopGapGui on the server.<br><br>Defaults to true (unit receives a static stand-in)<br><br><b>MANDATORY</b> |

### 53.4 Demos

- No Gap, No Glory
- Caucasus Hangar
- My Immortal

## 54 TACAN

### 54.1 Summary

Places TACAN nav aids inside trigger zones.

### 54.2 Dependencies

Tacan requires the modules dcsCommon and cfxZones.

Optionally: groupTracker

### 54.3 ME Integration

| Name          | Description  |
|---------------|--|
| tacan         | <p>Tells DML that this trigger zone is a tacan zone.</p> <p>The value of this attribute specifies which faction owns the TACAN.<br/>Possible values</p> <ul style="list-style-type: none"><li>• “0” or “neutral”</li><li>• “1” or “red”</li><li>• “2” or blue</li></ul> <p>Defaults to “neutral” (TACAN is owned by neutral faction, listed for red and blue factions when TACAN GUI is enabled)</p> <p><b>MANDATORY</b></p> |
| channel       | <p>The channel to use for the TACAN, e.g. “72”.</p> <p>For randomization, supports lists and ranges, e.g. “12, 17, 42-47”</p> <p>Defaults to “1” (The channel 1)</p>   |
| mode          | <p>Either “X” or “Y”</p> <p>Mode “Y” usually is reserved for airborne TACAN (currently not supported with tacan zones)</p> <p>Defaults to “X”</p>  |
| callsign      | <p>Three-letter callsign for this TACAN station, e.g. “TCN”.</p> <p>For randomization, supports lists, e.g. “TXN”, “ABC”, “XYZ”</p> <p>Defaults to “TXN”</p>   |
| onStart       | <p>When set to true (default) the TACAN is created on mission start. If set to false, you must provide a “deploy?” attribute to be able to deploy the TACAN while the mission is running.</p> <p>Defaults to true (deploy on mission start)</p>  |
| heading       | <p>Orientation (in degrees) of the physical unit that is placed in game when the TACAN is deployed. Has no influence on the TACAN’s function.</p> <p>Defaults to 0 (oriented North)</p>  |
| rndLoc        | <p>When set to true, the TACAN is placed in a random location inside the trigger zone. Supports polygonal trigger zones. Otherwise, the TACAN is placed at the exact center of the trigger zone.</p> <p>Defaults to false (place at center of zone)</p>  |
| triggerMethod | <p>DML method that tells when an input should trigger.</p>   |

| Name       | Description  |
|------------|--|
|            | Defaults to "change"   |
| deploy?    | Watchflag that triggers on triggerMethod and then initiates a deployment cycle. If preWipe is set to false, the previously deployed TACAN remains, otherwise it is removed before the new TACAN deploys.<br>Defaults to <none>   |
| preWipe    | If set to true (default), the last deployed TACAN from this zone is removed (if it exists) before a new TACAN is deployed.<br>Defaults to true (remove last spawn before deploy)   |
| destroy?   | Watchflag that triggers on triggerMethod and then removes the last previously deployed TACAN if it still exists.<br>Defaults to <none>   |
| c#         | Output (direct). Is always set to the currently selected channel of the last spawned TACAN. When that TACAN is destroyed via destroy? Signal, the channel is set to 0 (zero)<br>Defaults to <none>   |
| announcer  | When set to true, any TACAN created with this tacan zone is announced with callsign, channel and mode to all players (when owning faction of the TACAN is neutral) or all players of the same faction that also owns the TACAN.<br>Note that TACANS that are created at mission startup (onStart = true) are not announced.<br>Defaults to false (no announcement) |
| trackWith: | When a TACAN is created, it is tracked with the groupTracker modules that are listed as value. Supports lists.<br>Note that the groupTracker module must load before the tacan module in order to work correctly.  |

## 54.4 Demos

- Take on TACAN

## 55 Taxi Police

### 55.1 Summary

This module enforces a speed limit for player-controlled planes on tarmac, ramp and taxiways. Repeat offenders are actioned against.

### 55.2 Dependencies

TaxiPolice requires dcsCommon and cfxZones

### 55.3 ME Integration

TaxiPolice uses mainly a config zone named “*taxiPoliceConfig*” for most settings:

| Name        | Description  |
|-------------|--|
| verbose     | A value of “true” turns on debugging messages. Default is “false”  |
| speedLimit  | The speed limit for aircraft on all taxiways and tarmacs, set in m/s (meter per second, a.k.a. ‘civilized units’).<br>Recommended values are 10 (36 km/h) and 14 (50 km/h)<br>Defaults to 14 ( 14m/s = 50 km/h = 27 knots)   |
| triggerTime | Grace period (in second) after which a transgression is registered. If a pilot goes above the speed limit for less than that time, it is forgotten.<br>Defaults to 3 (seconds)   |
| leeway      | The width in meters of the ‘corridor’ along a runway to which the runway ‘no limit’ bounds extend to each side. Thus, exceeding the speed limit just off the runway is allowed – in case the pilots veers off the runway slightly.<br>To see each airfield’s runway corridors, turn on verbosity, and switch to the F10 Map view. All runway corridors are marked on the map with a dotted black line<br>Defaults to 5 (meters)                      |
| extend      | The length in meters of the ‘corridor’ along the runway to with it extends in front of and behind the runway. That way, pilots who touch down too early or too late (and survive) don’t get in more trouble than they already are.<br>To see each airfield’s runway corridors, turn on verbosity, and switch to the F10 Map view. All runway corridors are marked on the map with a dotted black line.<br>Defaults to 500 (meters)                   |
| radius      | The radius (in meters) around an airfield’s location as defined in DCS’s airfield data base. Aircraft inside this radius come under scrutiny if they are on the ground.<br><br>To see each airfield’s center, turn on verbosity, and switch to the F10 Map view. Each airfield’s center is marked on the map with a dotted red line. Note that in many maps (e.g., Caucasus), that airfield’s location coincides with the runup area of a particular |



| Name       | Description  |
|------------|--|
|            | runway, not the center of an airfield as one would have naively assumed.<br><br>Defaults to 3000 (meters)  |
| maxTickets | Number of tickets a pilot (not unit) may receive before TaxiPolice starts taking active retribution for offenses.<br>Defaults to 3 (i.e. the fourth offense will be painful)   |
| active     | Initial state of taxiPolice when the mission starts up. When set to false, TaxiPolice is off duty until enabled.<br>Defaults to true (taxiPolice is active at mission start)   |
| greetings  | When set to true, TaxiPolice will greet every pilot upon slotting or landing on an airfield, unless they are off duty. Pilots are given the current speed limit in km/h and knots. If an airfield is exempt from policing (as set with a 'taxiPolice' attribute at an airfield), that fact is also mentioned.<br>Defaults to true – pilots are updated of the current speed limit and enforcement policies of their current airfield |
| onPatrol   | Watchflag. Whenever the value of this flag changes, TaxiPolice becomes active on all policed airfields. A NOTAM is sent to all current players. Player's transgressions are recorded and sanctioned when they exceed the maxTicket limit. Note that turning taxiPolice on or off does not reset a player's violation count, only restarting the mission will do that.<br>Defaults to <none>  |
| offDuty    | Watchflag. Whenever the value of this flag changes, TaxiPolice steps off the plate and ignores any speed limit violations on the map. A NOTAM explaining that fact is sent to all current players. Note that turning taxiPolice on or off does not reset a player's violation count, only restarting the mission will do that.<br>Defaults to <none>   |

Also, you can selectively exclude airfields from police oversight as follows:

| Name       | Description   |
|------------|---|
| TaxiPolice | When present, <b>the airfield closest to this trigger zone</b> is removed from TaxiPolice's list of airfields to monitor. Speeding on taxiways, ramps or tarmac areas is ignored.<br><br><b>MANDATORY</b> |

## 55.4 Demos

- Taxi Police

## 56 TDZ (Touch-Down Zone)

### 56.1 Summary

A module to rate player's landings with regards to accuracy (touching down inside the TDZ) and giving statistics on landing run etc.

### 56.2 Dependencies

TDZ requires dcsCommon and cfxZones

### 56.3 ME Integration

| Name       | Description  |
|------------|--|
| <b>TDZ</b> | Marks this trigger zone as TDZ and use the center of this zone to determine which runway on the map to attach itself to.<br><br><b>MANDATORY</b>   |
| verbose    | When set to true, TDZ reports debugging information for this zone. This is particularly useful to discover the "Left" direction of an airfield's runways<br><br>Defaults to false (verbosity off)  |
| helos      | When set to true, the Zone also reacts to helicopter landings.<br><br>Defaults to false  |
| extend     | Extend the length of the runway (as reported internally by the map's DB) by this amount of meters. Note that the extension is added to both ends. Negative numbers shorten the runway on both end by this amount.<br><br>Defaults to 0 (meters)                    |
| expand     | Expands the width (i.e. widens) of the runway (as reported internally by the map's DB) by this amount of meters. Note that the width is added to both sides. Negative numbers make the runway narrower on both sides by this amount.<br><br>Defaults to 0 (meters) |
| starts     | Offset (in meters) from the runway threshold (as defined by DCS's airfield DB) where the TDZ starts<br><br>Defaults to 0 (meters, directly at threshold)   |
| ends       | Offset (in meters) from the runway threshold (as defined by DCS's airfield DB) where the TDZ ends.<br><br>Defaults to 610 (meters = 2000 ft)   |
| landed!    | Output to send a signal when an aircraft has successfully landed (touched down, did not leave the runway, and come to a complete stop on the runway (if 'opposite' is true – as per default – the direction of the   |

| Name       | Description  |
|------------|--|
|            | landing is ignored)<br><br>Defaults to <none>  |
| touchdown! | Output to send a signal when an aircraft has touched down on the runway<br><br>Defaults to <none>  |
| failed!    | Output to send a signal when an aircraft has touched down, and subsequently left the runway before coming to a complete stop (for whatever reasons, be they intentionally or an unscheduled rapid disassembly)<br><br>Defaults to <none>   |
| rwFill     | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the runway. RGBA values each range from 0.0 to 1.0<br>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)<br><br>Defaults to "0.0, 0.0, 0.0, 0.0" or "#00000000" transparent black      |
| rwFrame    | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the runway's outline. RGBA values each range from 0.0 to 1.0<br>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)<br><br>Defaults to "0.0, 0.0, 0.0, 1.0" or "#000000FF" opaque black |
| tdzFill    | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to fill the TDZ. RGBA values each range from 0.0 to 1.0<br>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)<br><br>Defaults to "0.0, 1.0, 0.0, 0.25" or "#00FF0040" transparent green        |
| tdzFrame   | Four numeric values, separated by comma (e.g., 1.0, 0, 0, 1.0) that define the RGBA (red, green, blue, alpha = opacity) values for the color used to draw the runway's outline. RGBA values each range from 0.0 to 1.0<br>Alternatively, you can use the #RRGGBBAA format, where RR, GG, BB and AA are each hexadecimals (as is used in HTML/CSS to denote colors)<br><br>Defaults to "0.0, 1.0, 0.0, 1.0" or "#00FF00FF" opaque black |
| visible    | When set to true, TDZ draws a frame and fills the runway with the selected colors and then draws the TDZ(s) on top.<br><br>Defaults to true (draw runway and TDZ)  |

| Name  | Description  |
|-------|--|
| left  | <p>When set to true, the 'left' direction (runway's "main" landing direction, as reported by the map's DB) of this runway is active, landings in this direction are reported, and the TDZ in this direction is active.</p> <p>Defaults to true</p> |
| right | <p>When set to true, the 'right direction (opposite to the airfield's reported 'main' landing direction) of this runway is active, landings in this direction are reported, and the TDZ in this direction is active.</p> <p>Defaults to true</p>   |

## 56.4 Demos

- Landing Lessons

## 57 UnGrief

### 57.1 Summary

Module to deter griefers and to enforce PVE-only rules

### 57.2 Dependencies

dcsCommon, cfxZones

(SSB on server when using SSB as retaliation)

### 57.3 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.<br>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.<br>Default is 1   |
| retaliation    | How unGrief retaliates towards the player when their graceKills are exceeded. The following options are supported: <ul style="list-style-type: none"><li>• 'boom'<br/>Place a small explosive inside the plane, grin, ignite.</li><li>• 'ssb'<br/>Use server-side SSB to kick the player from the plane</li></ul> 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<br>pveOnly | When set to true, PVE rules are in force for this mission. Player versus Player kills count like friendly kills<br>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.<br>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<br>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.<br>Defaults to false |

## 57.4 Demos

- Good Grief
- The Danger Zone

## **58 Unit Persistence**

### **58.1 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)

### **58.2 Dependencies**

Requires dcsCommon, cfxZones, persistence and cfxMX.

### **58.3 ME Integration**

### **58.4 Demos**

- Being Persistent

## 59 Unit Zone

### 59.1 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.

### 59.2 Dependencies

dcsCommon, cfxZones

### 59.3 ME Integration

| Name                | Description  |
|---------------------|--|
| <b>unitZone</b>     | <p>Marks this ME Zone as an anchor for unitZone. The value of this attribute defines which coalition groups/players are checked. Legal values are "0", "1", "2", "red", "blue", "neutral". Note that "<b>0</b>" (<b>zero</b>) and '<b>neutral</b>' means '<b>both</b>': unitZones never considers groups belonging to the neutral coalition.</p> <p>Defaults to 0 (both coalitions red and blue)</p> <p><b>MANDATORY</b></p>   |
| lookFor             | <p>Name for the 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:</p> <ul style="list-style-type: none"><li>• Hel</li><li>• Hello World</li><li>• Helo Rescue-1</li><li>• Hellfire</li></ul> <p>If you want to match all groups or players, simply supply "*" (default) as this will match all names.</p> <p>Use this feature to your advantage in conjunction with cloners or spawners, as these all produce groups with a known base name.</p> <p>If you only supplied "Hel", only (without the asterisk "*") only the group whose name exactly matches "Hel" is checked.</p> <p>Defaults to "*" – meaning all names are matched.</p> |
| matching            | <p>What type of units to match. Currently supported are</p> <ul style="list-style-type: none"><li>• group (default): look for group names</li><li>• player – look only at player units and match their unit's (not group's) name against lookFor</li></ul> <p>Default: group</p>   |
| filterFor<br>filter | <p>Which unit categories to look for. If no attribute is given, <b>all</b> categories are checked against the zone (when their name pattern matches).</p>  |



| Name                             | Description   |
|----------------------------------|---|
|                                  | <p>When you supply a filterFor attribute, only that category is considered. Currently supported are</p> <ul style="list-style-type: none"> <li>• 0 (zero) or “aircraft” or “air”</li> <li>• 1 or “helo” or “heli” or “helicopter”</li> <li>• 2 or “ground”</li> <li>• 3 or “ship”</li> <li>• 4 or “train”</li> </ul> <p>Defaults to no filtering</p>                              |
| enterZone!                       | <p>Change this flag when the first unit (player) or part of all groups that match the criteria enters the zone</p> <p>Defaults to &lt;none&gt;</p>  |
| exitZone!                        | <p>Change this flag when the last unit (player) of all groups that match the criteria have exited the zone (being destroyed counts as leaving)</p> <p>Defaults to &lt;none&gt;</p>  |
| changeZone!                      | <p>Changes this flag whenever enterZone! or exitZone! are triggered</p> <p>Defaults to &lt;none&gt;</p>   |
| method<br>uzMethod               | <p>DML Flag method for output. Use only one synonym per zone</p> <p>Defaults to “inc”</p>   |
| uzOff?                           | <p>Watchflag. When triggered, this zone will no longer perform checks. When already off, nothing happens</p> <p>Defaults to &lt;none&gt;</p>  |
| uzOn?                            | <p>Watchflag. When triggered, this zone will resume checks. When already on, nothing happens</p> <p>Defaults to &lt;none&gt;</p>  |
| triggerMethod<br>uzTriggerMethod | <p>Method that determines when the watchflags should trigger.</p> <p>Default is “change”</p>  |
| uzDirect<br>uzDirect#<br>direct# | <p>Used mainly to control (directly enable/disable modules and open/close gates (changer modules)) When present, this flag (or flags) is always set to the current state of the unit zone:</p> <ul style="list-style-type: none"> <li>• 1 when one or more units in the zone</li> <li>• 0 when none of the indicated units in the zone.</li> </ul> <p>Default is &lt;none&gt;</p> |

## 59.4 Demos

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

## 60 Willie Pete

### 60.1 Summary

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

### 60.2 Dependencies

WilliePete requires dcsCommon, cfxZones and cfxMX.

**Optional:** PlayerScore (no load time limitation)

### 60.3 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 <ul style="list-style-type: none"><li>• RED or 1</li><li>• BLUE or 2</li></ul> <b>MANDATORY</b>  |
| shellStrength      | Explosive power of shell hitting the ground.<br>Defaults to 500   |
| shellNum           | Number of shells per artillery salvo<br>Defaults to 17  |
| transitionTime     | The time (in seconds) it takes for shells to arrive after the 'fire' command is given<br>Defaults to 20   |
| coolDown           | Number of seconds after which the artillery can fire again.<br>Defaults to 180 (3 minutes)  |
| baseAccuracy       | Radius (in m) around the wp where the shells hit in.<br>Defaults to 50  |
| method<br>wpMethod | DML method for the outputs  |
| wpFired!           | Output that is signaled when a successfully Fire command was received by the WP zone<br>Defaults to <none>  |
| checkInRange       | Distance (from the zone's border) from which a plane can check into the zone.<br>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.<br>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<br>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<br>Defaults to what you defined in wpConfig |

## 60.4 Demos

- Willie Nillie

## 61 Wiper

### 61.1 Summary

Removes objects inside a zone whenever you trigger it. Can be used to remove some detonation artifacts like debris, wrecks and craters.

### 61.2 Dependencies

dcsCommon, cfxZones

### 61.3 ME Integration

| Name                                | Description  |
|-------------------------------------|--|
| <b>wipe?</b>                        | Watchflag. Triggers a wipe cycle<br><br><b>MANDATORY</b>   |
| triggerMethod<br>triggerWiperMethod | Method that triggers the Watchflag   |
| category<br>wipeCat<br>wipeCategory | <p>List, separated by comma “,” of object categories that are affected by the wipe (i.e. if they belong to the category they may be wiped). Possible values are</p> <ul style="list-style-type: none"><li>• “unit” or 1</li><li>• “weapon” or 2</li><li>• “static” or 3</li><li>• “base” or 4</li><li>• “scenery” or 5</li><li>• “cargo” or 6</li></ul> <p>If a category is not recognized, it defaults to “3” (static)</p> <p>A special value ‘none’ can be used to skip object removal entirely. This can be useful if you are only interested in the ‘declutter’ (wreck and debris removal) ability.</p> <p><i>Examples</i></p> <ul style="list-style-type: none"><li>• “none” (skips entire object removal step)</li><li>• “unit, 3” (wipes units and static objects)</li><li>• “cargo” (wipes cargo type objects)</li></ul> <p>Defaults to ‘none’</p> |
| wipeNamed                           | <p><i>Optional</i> comma-separated name list that an object’s name must match in order to be wiped. Supports an asterisk (“*”) as wildcard to match anything. For example, “Ba*” would match “Base”, “Ba”, “Babushka”, and “Bathyscape”</p> <p><b>Examples:</b></p> <ul style="list-style-type: none"><li>• “Ba*” – all objects inside the zone whose name starts with “Ba”</li></ul>  |

| Name          | Description  |
|---------------|--|
|               | <ul style="list-style-type: none"> <li>“Grou*, Commander Kirk, He*” – all objects whose name starts with “Grou” or “He”, and the object whose name exactly matches “Commander Kirk”</li> </ul> <p>Defaults to &lt;option off&gt;, no name filtering</p>  |
| declutter     | <p>If set to true will remove all debris, wrecks, craters and similar detonation artifacts from inside the zone.</p> <p><i>Caution:</i><br/>This option invokes DCS’s <code>world.removeJunk()</code> method which is currently suspected of doing some nefarious stuff, including mission crashes in multiplayer. Suspected, not proven.</p> <p>Defaults to false (no debris removal)</p> |
| wipeInventory | <p>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.</p>                                       |

## 61.4 Demos

- Viper with a double youu

## 62 Valet

### 62.1 Summary

Valet is a module that helps you to easily generate events, text message or sound effects for players who enter or leave a zone like airfields, Hospital zones, FARPs, naval groups, destinations etc. Valet has strong built-in abilities to assemble complex text messages and can send these or sound to that player unit.

### 62.2 Dependencies

dcsCommon, cfxZones

### 62.3 ME Integration

| Name     | Description  |
|----------|--|
| valet    | Tells DML that this trigger zone is a valet<br><br><b>MANDATORY</b>  |
| greeting | Message to be displayed if a player unit triggers the valet. The message can be heavily customized with dynamic wildcards. Valet supports the following wildcards in the greeting, which are replaced with current values as the mission runs: <ul style="list-style-type: none"><li>• &lt;player&gt; - callsign of the player</li><li>• &lt;unit&gt; - name of the unit that the player is controlling</li><li>• &lt;type&gt; - type (e.g. "A-10A") of the unit that the player is controlling</li><li>• &lt;group&gt; - name of the group that the player's unit is in</li><li>• &lt;in&gt; - number of greetings that the player has received from this valet</li><li>• &lt;out&gt; - number of goodbyes that the player has received from this valet</li><li>• &lt;n&gt; creates a line feed</li><li>• &lt;z&gt; - valet zone's name as set with ME</li><li>• &lt;t&gt; - current mission time in the format "HH:MM:SS", e.g. "08:42:11". Format be further configured with the <i>timeFormat</i> attribute.</li><li>• &lt;lat&gt; - latitude of the valet zone's current position</li><li>• &lt;lon&gt; - longitude of the valet zone's current position</li><li>• &lt;ele&gt; - elevation of the valet zone's current position. Format be further configured with the <i>imperialUnits</i> attribute.</li><li>• &lt;mgrs&gt; - valet zone's current position in MGRS coordinates</li><li>• &lt;v: flagName&gt; - value of flag &lt;flagName&gt;</li><li>• &lt;t: flagName&gt; - value of flag &lt;flagName&gt; interpreted as time. Format be further configured with the <i>timeFormat</i> attribute.</li></ul> |

| Name                      | Description   |
|---------------------------|---|
|                           | <ul style="list-style-type: none"> <li>• &lt;lat: unit/zone&gt; - latitude of unit/zone with that name.</li> <li>• &lt;lon: unit/zone&gt; - longitude of unit/zone with that name.</li> <li>• &lt;ele: unit/zone&gt; - elevation of unit/zone with that name. Format be further configured with the <i>imperialUnits</i> attribute.</li> <li>• &lt;mgrs: unit/zone&gt; - mgrs of unit/zone with that name</li> <li>• &lt;coa: flag/unit/zone&gt; - faction (e.g., "RED") of zone/unit, or the flag value interpreted as such.</li> <li>• &lt;vel: unit/zone&gt; - velocity of unit/zone with that name. Format be further configured with the <i>imperialUnits</i> attribute.</li> <li>• &lt;alt: unit/zone&gt; - altitude of unit/zone with that name. Format be further configured with the <i>imperialUnits</i> attribute.</li> <li>• &lt;hdg: unit/zone&gt; - heading of unit/zone with that name.</li> <li>• &lt;type: unit&gt; - type (e.g. "A-10A") of unit with that name</li> <li>• &lt;player: unit&gt; - player callsign who controls a unit with that name.</li> </ul> <p>If the message text is empty, no message is displayed.</p> <p>Defaults to &lt;empty message&gt;, i.e., no greeting diosplayed</p> |
| firstGreeting             | <p>When present, the text of this message replaces the message that is displayed when a player enters the valet for the first time since they (re)spawned. If the attribute is present but the text is empty, no message is displayed (i.e., this can be used to suppress the first greeting to a player)</p> <p>Supports all wildcards that <i>greeting</i> does.</p> <p>Defaults to &lt;none, no separate first greeting&gt;</p>  |
| goodbye                   | <p>Message to display when a player leaves the valet zone</p> <p>Supports all wildcards that <i>greeting</i> does.</p> <p>Defaults to &lt;empty&gt;, i.e. no text message.</p>  |
| imperial<br>imperialUnits | <p>When set to true, wildcards return units for velocities, length etc. in "imperial" funny units (knots, feet, tablespoon etc.)</p> <p>Defaults to false (SI units are used: m, km/h)</p>  |
| timeFormat                | <p>The time format in which time values are displayed.</p> <p>Defaults to &lt;:h&gt;:&lt;:m&gt;:&lt;:s&gt;</p>  |
| duration                  | <p>Time duration (in seconds) that a text message is displayed for</p> <p>Defaults to 30</p>  |
| inSoundFile               | <p>Name of the sound file that is to be played when a player enters the valet zone</p> <p>Defaults to &lt;none&gt;</p>  |
| firstInSoundFile          | <p>When present, this audio is played when a player enters the valet for the first time since they (re)spawned.</p> <p>Defaults to &lt;none&gt;, i.e., insoundFile is played</p>  |
| greetSpawns               | <p>Controls if a unit that spawns inside a valet zone should be greeted. If set to false, a unit that spawns within a valet zone will not receive a greeting, and must exit and then re-enter the valet zone for its first greeting.</p>  |

| Name                        | Description   |
|-----------------------------|---|
|                             | Default is false (spawning units inside a valet zone will not be greeted but have to leave zone first)  |
| method<br>valetMethod       | DML method for outputs<br>Defaults to 'inc'   |
| hi!                         | Output flag to set when an eligible player unit enters the valet zone   |
| bye!                        | Output flag to set when an eligible player unit leaves the valet zone   |
| coalition<br>valetCoalition | When given, restricts the valet to players of that coalition (e.g., 'red'). Players of other coalition are ignored.<br>Defaults to <none> - all players are eligible  |
| types<br>valetTypes         | A list of unit types (e.g., "A-10A"), separated by comma. The valet zone is restricted to players who control units of that type. List supports end-wildcards, i.e., when the type ends on an asterisk "*", all types that match everything before the asterisk are matches. Example: "A-10*" matches "A-10A", "A-10C" and "A-10C_2", but not "AV8BNA".<br><br>Types are not case sensitive.<br><br>Defaults to <none>, all types permitted                           |
| groups<br>valetGroups       | A list of group names, separated by comma. The valet zone is restricted to players who control units who are members of the groups on the list. List supports end-wildcards, i.e., when the type ends on an asterisk "*", all group names that match everything before the asterisk are matches. Example: "atta*" matches "Attackers-1" and "atta99" but not "avenger-1"<br><br>Group names are NOT case sensitive.<br><br>Defaults to <none> - no group restrictions |
| units<br>valetUnits         | A list of unit names, separated by comma. The valet zone is restricted to players who control units with names on the list. List supports end-wildcards, i.e., when the type ends on an asterisk "*", all unit names that match everything before the asterisk are matches. Example: "atta*" matches "Attackers-1" and "atta99" but not "avenger-1"<br><br>Unit names are NOT case sensitive.<br><br>Defaults to <none> - no unit restrictions                        |

## 62.4 Demos

- I say hello goodbye



## 63 XFlags (Flag Testing)

### 63.1 Summary

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

### 63.2 Dependencies

dcsCommon, cfxZones

### 63.3 ME Integration

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

| Name              | Description   |
|-------------------|---|
|                   | <ul style="list-style-type: none"> <li>• 'most'<br/>triggers when more than half of the input flags have triggered. Will not trigger if exactly half have triggered, so be careful if the number of input flags is even.</li> <li>• 'half or more'<br/>triggers when half or more of the input flags have triggered. Will also trigger if exactly half of all flags have triggered</li> <li>• 'never'<br/>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</li> </ul> <p>Defaults to 'some'</p> |
| #hits             | <p>Value used for only some of the require attribute.<br/>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.<br/>Defaults to 1 (one)</p>   |
| xFlagMethod       | <p>Condition that must be met for individual input flags. Is identical to trigger method for Watchflags except it is applied to each input flag individually.<br/>Defaults to "change"</p>  |
| xSuccess!<br>out! | <p>Flag to bang! when xFlags when the evaluation of input flags succeeds (all conditions are met). Once xSuccess is triggered, and unless oneShot is set to false, this zone's xFlag pauses until xReset is triggered.<br/>Defaults to &lt;none&gt;</p>   |
| xChange!          | <p>Flag to bang! when xFlags detects a change in the input configuration. Merely detects a change in the input configuration, has no relation with xSuccess!, except that xSuccess will also be accompanied by a bang on xChange!<br/>Defaults to &lt;none&gt;</p>  |
| xDirect           | <p>Each time xFlags evaluates the input flags, it directly sets the xDirect flag to the evaluation result (0 or 1) – this is different from what xSuccess may output, since that flag's value is dependent on the xMethod attribute.</p> <p><b>This flags value is set directly, not via DML method.</b></p> <p>Defaults to &lt;none&gt;</p>  |
| xCount            | <p>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</p> <p><b>This flags value is set directly, not via DML method.</b></p> <p>Defaults to &lt;none&gt;</p>   |
| xReset?           | <p>When the value of this input changes, the zone's xFlag module is reset, and evaluation starts afresh.</p>  |

| Name              | Description  |
|-------------------|--|
|                   | <b>Note: this input always reacts to a change in the flag's value</b><br>Defaults to <none>  |
| method<br>xMethod | DML method for output flags<br>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.<br><b>Note: this input always reacts to a change in the flag's value</b><br>Defaults to <none> |
| xOn?              | Turns a suspended xFlag back on to resume processing. It resumes processing where it left off<br><b>Note: this input always reacts to a change in the flag's value</b><br>Defaults to <none>   |
| xSuspended        | Sets the initial state of xFlag. Setting it to true suspends the xFlag at mission start<br>Defaults to false   |

## 63.4 Demos

- xFlags – Field Day
- Count Base's Blues

## **64 Module Name**

### **64.1 Summary**

### **64.2 Dependencies**

### **64.3 ME Integration**

### **64.4 Demos**