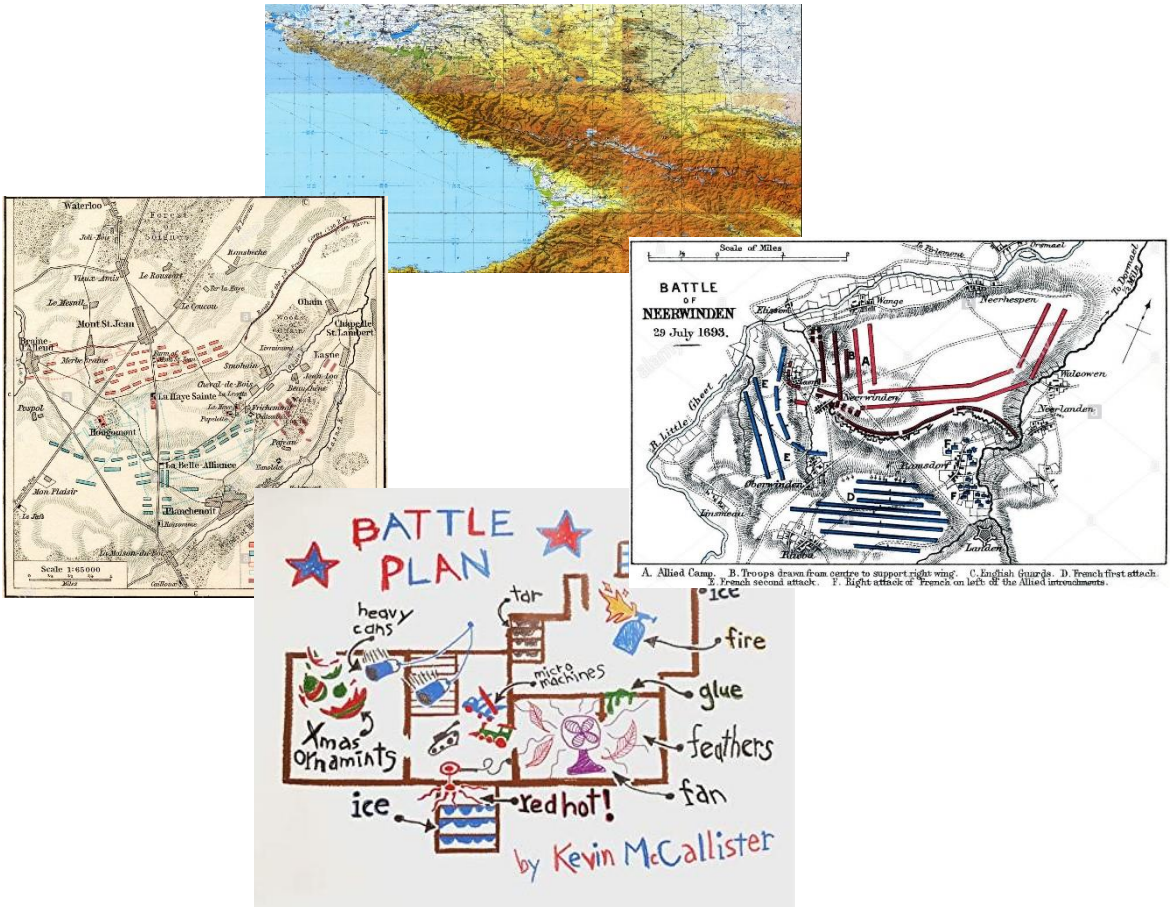


Enhanced Gamemaster Script

15.06.2021

Gamemaster_Functions v. 2.3

Gamemaster_Templates v. 2.1

Functions tested on DCS 2.7.1.7139

Templates up to date with DCS 2.7.1.7139

Requires MOOSE v. 2.7.4

Table of Contents

1. Loading the script into missions

In order to have access to the advanced gamemaster functions in your mission you must **first** load the "Moose.lua" included in the download. MOOSE is a framework for the DCS scipting engine that provides a whole bunch of functions that my script relies on quite heavily. It is being developed by a group of very devoted and talented people to whom I want to address my sincerest thanks! To learn more about MOOSE, head over to their Discord.

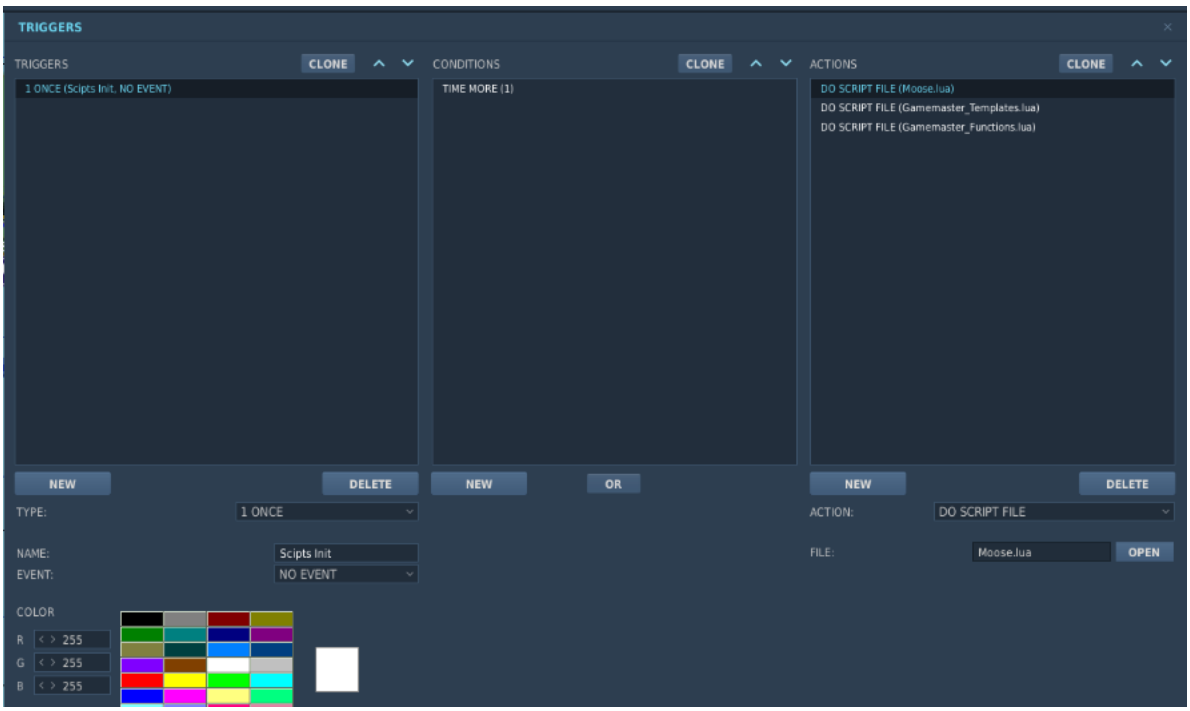
This script utilizes MOOSE version 2.7.4!

Once the "Moose.lua" has been loaded you can load the "Gamemaster_Functions.lua". This sets up the commands for ingame use.

To load the script files into your mission you can either use a trigger "MISSION START" or a „ONCE" trigger combined with a „TIME MORE" condition. In the actions tab select "DO SCRIPT FILE" and select the file to be loaded. The picture below shows how the trigger page should look.

The provided "Gamemaster_Templates.lua" can be loaded as well. This file sets up a whole lot of group templates that can be spawned with the "-s"-command (see 5.1). Loading the "Gamemaster_Templates.lua" is purely optional, the main script works just as fine without it. You must setup every spawnable group manually in your mission then, though.

The successful loading of the scripts will be shown by status messages in the top right corner of your screen.



Pic. 1: Trigger-setup for loading "Moose.lua", "Gamemaster_Functions.lua" and "Gamemaster_Templates.lua"

2. Configuration options in the script file

2.1 Gamemaster_Functions.lua

You can adjust some basic settings within the script. To do so you must open the “Gamemaster_Functions.lua“ with a text editor (best use Notepad++). The config-section is located at the beginning of the file. The options are explained there in detail, which is why I’m only giving an overview over the adjustable settings here:

- Limit access to the gamemaster functions to a specified coalition
- Set a password that has to be entered before each command, for the commands to be recognized
- Change the symbol that the script uses to recognize commands and parameters in the marker text (Default symbol is a hyphen)
- Change the default skill with which all new groups are spawned (“-s“-command)
- Change the default country to which all newly spawned groups belong (“-s“-command)
- Turn EPLRS on or off for newly spawned groups (“-s“-command
- Set a default sound and borders for messages sent with the “-text“ command
- Multiple default settings for the draw commands

Changes to the config only apply once the script has again been loaded into the mission file. To do so you need to reselect the “Gamemaster_Functions.lua“ in the actions tab of the trigger that loads the script at the beginning of the mission and save the mission afterwards.

2.2 Gamemaster_Templates.lua

The “Gamemaster_Templates.lua” file comes with its own configuration options at the top of the script file. By setting the entries for the different template tables to “true” or “false” you can control which templates get loaded into your mission when the script is executed. See section 6 of this document for an overview of the available template tables and the template groups contained within them.

Note that, depending on how many template tables you have selected, loading “Gamemaster_Templates.lua” may cause DCS to freeze for a time.

3. Sending commands

The script uses map markers as its input method. To send a command you need to create a new marker on the F10 map and enter your command and the required parameters into the marker text field. The command is then sent to the script by deleting the created marker.

Pic. 2: Steps required to send a command to the script: 1. Activate marker mode -> 2. Create marker by clicking on the map -> 3. Enter command and parameters as marker text -> 4. Delete marker to send the command

4. Formatting of commands

Commands always start with a hyphen or whatever you have specified in the config options of the script file, followed by the command string and the required parameters. All following examples assume that the default hyphen is set as command symbol.

A complete command should look something like this when entered into the marker text field:

-command string-parameter1-parameter2-parameter3-...-parameter6

Some commands require the entry of group names as a parameter, names containing the same symbol that is used to specify commands and parameters will not be recognized. Make sure that you avoid using your command symbol when naming groups that you want to spawn/control with the script later on.

The next section lists and explains all the commands that are provided by the script. To do so the following symbology is being used:

The command string is shown in red. It must always be entered, otherwise no action will be performed.

[Necessary parameters are shown in orange writing and square brackets. They must be specified or no action will be performed. Note that necessary parameters must always be entered in the order shown in this documentation!]

(Optional parameters are shown in blue writing and round brackets. They can be entered in any order, but always behind any necessary parameters. They are not required for the command to be performed.)

5. List of available commands

5.1 Spawn groups

Spawns a new group at the location of the map marker, requires a late activated group set up in the mission editor as a template. If "Gamemaster_Templates.lua" is loaded, all groups listed in section 6 can be spawned with this command.

Newly spawned planes/helos will orbit around the location of the marker while ground units and ships stay stationary. Ground units can't be spawned on water nor ships on land.

Groups can be spawned an infinite number of times. Note that the group names are modified by the script. Groups spawned with this method won't trigger any triggers directly linked to them in the mission editor.

Command structure: -s-[group name]-(altitude/heading/ground start)-(country)-(skill)-(loadable)-(spawn at original position)-(keep tasking)

Parameter	Explanation	Possible values
group name	the exact name of the group to spawn as specified in the mission editor or in section 6	text
altitude/heading/ground start (optional)	Planes/Helos: Altitude in meters above MSL, if left clear the group will spawn at 1000 m AGL Ground units: Heading in degrees the group will be facing once spawned, if left clear the group will spawn with the heading it has been set up with in the ME. Plane groups can be spawned on the ground if "ground" is entered instead of an altitude. They will spawn an the airbase closest to the marker. Note that the planes will spawn in an uncontrolled state (no pilot)	headings from 0 to 359 altitude in m MSL ground
country (optional)	Country that the group will belong to once spawned, only works if the type of unit is available to the specified country This parameter allows changing the coalition a group belongs to. If left unspecified the group will belong to the default country that is specified in the config section of "Gamemaster_Functions.lua". If that is unspecified as well, the country the group is set up with in the ME will be used.	RUSSIA UKRAINE USA TURKEY UK FRANCE GERMANY AGGRESSORS CANADA SPAIN THE_NETHERLANDS BELGIUM NORWAY DENMARK ISRAEL GEORGIA INSURGENTS ABKHAZIA SOUTH_OSETIA ITALY

		AUSTRALIA SWITZERLAND AUSTRIA BELARUS BULGARIA CHEZH_REPUBLIC CHINA CROATIA EGYPT FINLAND GREECE HUNGARY INDIA IRAN IRAQ JAPAN KAZAKHSTAN NORTH_KOREA PAKISTAN POLAND ROMANIA SAUDI_ARABIA SERBIA SLOVAKIA SOUTH_KOREA SWEDEN SYRIA YEMEN VIETNAM VENEZUELA TUNISIA THAILAND SUDAN PHILIPPINES MOROCCO MEXICO MALAYSIA LIBYA JORDAN INDONESIA HONDURAS ETHIOPIA CHILE BRAZIL BAHRAIN THIRDREICH YUGOSLAVIA USSR ITALIAN_SOCIAL_REPUBLIC ALGERIA KUWAIT QATAR OMAN UNITED_ARAB_EMIRATES SOUTH_AFRICA CUBA PORTUGAL

		GDR LEBANON CJTF_BLUE CJTF_RED UN_PEACEKEEPERS
skill (optional)	The group will spawn with the specified skill level. If left blank, the skill level specified in the config section of "Gamemaster_Functions.lua" will be used. If that too is left blank the skill level the group has been set up with in the ME will be used instead.	a = average g = good h = high e = excellent r = random
loadable (optional)	Groups spawned with this parameter can be loaded into planes and helos with the "-board" command (see 5.13). If CTLD is active, loadable groups can also be transported by units listed in **ctld**.transportPilotNames. If left blank the spawned group cannot be transported, it's impossible to change this after the group has already been spawned.	cargo
spawn at original position (optional)	If this parameter is set the group will not spawn at the marker location but at the location it has been set up at in the ME	op
keep tasking (optional)	If this parameter is set the spawned group keeps the route and tasks it has been set up with in the ME. Useful for respawning tanker aircraft.	kt

5.2 Spawn statics

Spawns a new static at the location of the map marker, needs a static already placed in the mission as template. In a future release static templates will be added to "Gamemaster_Templates.lua" for more convenient spawning. For now, you must place all static types you want to have available for spawning somewhere in your mission.

Statics can be spawned an infinite number of times. Note that the names of the statics are modified by the script. Statics spawned with this method won't trigger any triggers directly linked to them in the mission editor.

Command structure: -sta-[static name] -(heading)-(country)

Parameter	Explanation	Possible values
static name	The exact name of the static. Sadly the names of statics are not shown when clicking them in the F10 map. Instead, use the new query function (see 5.7) to find out the name of the static you want to replicate.	text
heading (optional)	Heading the static will face after spawn.	headings from 0 to 359
country (optional)	Country that the static will belong to once spawned, only works if the type of static is available to the specified country This parameter allows changing the coalition a static belongs to. If left unspecified the static will belong to the default country that is specified in the config section of "Gamemaster_Functions.lua". If that is unspecified as well, the country the static is set up with in the ME will be used.	RUSSIA UKRAINE USA TURKEY UK FRANCE GERMANY AGGRESSORS CANADA SPAIN THE_NETHERLANDS BELGIUM NORWAY DENMARK ISRAEL GEORGIA INSURGENTS ABKHAZIA SOUTH_OSETIA ITALY AUSTRALIA SWITZERLAND AUSTRIA BELARUS BULGARIA CHEZH_REPUBLIC CHINA CROATIA EGYPT FINLAND GREECE HUNGARY INDIA

		IRAN IRAQ JAPAN KAZAKHSTAN NORTH_KOREA PAKISTAN POLAND ROMANIA SAUDI_ARABIA SERBIA SLOVAKIA SOUTH_KOREA SWEDEN SYRIA YEMEN VIETNAM VENEZUELA TUNISIA THAILAND SUDAN PHILIPPINES MOROCCO MEXICO MALAYSIA LIBYA JORDAN INDONESIA HONDURAS ETHIOPIA CHILE BRAZIL BAHRAIN THIRDREICH YUGOSLAVIA USSR ITALIAN_SOCIAL_REPUBLIC ALGERIA KUWAIT QATAR OMAN UNITED_ARAB_EMIRATES SOUTH_AFRICA CUBA PORTUGAL GDR LEBANON CJTF_BLUE CJTF_RED UN_PEACEKEEPERS

5.3 Spawn CTLD-Crates

Only works when CTLD is active in the mission. Allows you to spawn crates predefined in **ctld**.spawnableCrates. Different types of crates are called by the weight parameter.

Command structure: -ctldcr-[coalition] -[weight]

Parameter	Explanation	Possible values
coalition	Coalition that the crate will belong to. Note: CTLD assumes that Russia is always on the RED side and USA always on the BLUE side. If your coalitions are setup differently, crates may spawn for the wrong coalition.	blue -> crate belongs to USA red -> crate belongs to Russia
weight	Does not affect the simulated weight of the crate. Only determines what type of crate will be spawned. For now you need to look up the different weights and their corresponding crates directly in the CTLD script (**ctld**.spawnableCrates). In a future release I will provide a better method of selection.	number

5.4 Spawn CTLD-Infantry

Only works when CTLD is active in the mission. Allows you to spawn infantry groups of varying sizes that can be loaded into helicopters through options CTLD adds to the radio menu. The composition of those groups is dependent on how CTLD is configured.

Command structure: -ctldgr-[coalition] -[quantity]-[search radius]

Parameter	Explanation	Possible values
coalition	Coalition that the group will belong to. Note: CTLD assumes that Russia is always on the RED side and USA always on the BLUE side. If your coalitions are setup differently, groups may spawn for the wrong coalition.	blue -> group belongs to USA red -> group belongs to Russia
quantity	Determines how many soldiers the infantry group is comprised of. CTLD offers configuration options in the script file that further determine the composition of spawned groups.	number
search radius	Radius from the marker position inside of which the spawned group will move randomly and engage any enemies it finds.	number

5.5 Activate groups

Activates a group that has been set up as late activated in the mission editor. The group name is not changed when "spawning" the group with this method. This means that the group will trigger all triggers that are tied to it directly. The downside is that each group can only be activated once.

Command structure: -act-[group name]

Parameter	Explanation	Possible values
group name	Exact name of the group that is to be activated	text

5.6 Delete groups/units/objects

Deletes all units and static objects in a defined radius around the map marker. Alternatively you can also specify a group to be deleted. The radius method won't affect FARPS (the static object itself) and planes/helos that are controlled by players.

Command structure: -del-(group name)-(radius in m)

Parameter	Explanation	Possible values
group name (optional)	Exact name of the group to be deleted. Note: this will delete ALL units in the group.	text
radius (optional)	Radius (m) around the map marker. Everything inside will be deleted. Doesn't discriminate between coalitions. Will be set to 100 m if left blank.	numbers, 1-infinite large radii eat into system performance

Example: -del-1000

Deletes everything within 1 km of the map marker.

5.7 Show name of nearest group/unit

Returns the unit name and, if possible, the group name of the object closest to the map marker. Output is returned as a new map marker that has the query results written into its description. From there it can be selected and copied.

Mainly introduced as a workaround to determine the name of statics ingame. Needed because the names of statics are not shown when clicking them in the F10 map. It works on all types of objects though, not only statics.

Command structure: -?-(radius in m)

Parameter	Explanation	Possible values
radius (optional)	Radius (m) around the map marker that is searched for objects. Defaults to 500 m if left blank.	numbers, 1-infinite large radii eat into system performance

Example: -?-1000

Looks for objects in a radius of 1000 m around the map marker and returns the name of the first object it finds.

5.8 Assign waypoints for ships/ground units

Makes the AI move to the position of the map marker. Movement speed and a formation can be specified. Ground units can be ordered to stick to roads only.

Command structure: -wp-[group name]-(speed)-(formation)-(road use)

Parameter	Explanation	Possible values
group name	Exact name of the group that is meant to move to the map marker	text
speed (optional)	Speed (kph) at which the group moves to the waypoint Defaults to 20 kph if left blank	number
Formation (optional)	formation that the group will hold on the way to the waypoint	v = vee c = cone d = diamond r = rank el = echelon left er = echelon right
road use (optional)	If this parameter is set, the group will use roads to drive to the waypoint (as far as this is possible) If left blank the group will drive to the waypoint in a straight line	road

Example: -wp-T90_1-50-road

Orders the group "T90_1" to drive to the marker position with a speed of 50 kph and to use roads as much as possible.

5.9 Assign orbits for planes and helicopters

The specified group will move to and then fly an orbit above the position of the map marker. You must specify a speed and an altitude for the group. Optionally you can also let the AI fly a racetrack pattern between its current location and the position of the map marker.

Command structure: -orbit-[group name]-[altitude]-[speed]-(racetrack)

Parameter	Explanation	Possible values
group name	Exact name of the group that is meant to perform the orbit	text
altitude	Altitude at which the orbit is to be flown in m above MSL	number
speed	Groundspeed that is to be held while in orbit/racetrack	number
racetrack (optional)	If this parameter is set the group will fly a racetrack pattern between its current position and the position of the map marker.	r

Example 1: -orbit-AH64_1-500-90-r

Orders the group "AH64_1" to fly a racetrack pattern at 500 ft MSL with a speed of 90 kn

Example 2: -orbit-SU33_6-25000-450

Orders the group "SU33_6" to orbit above the map marker at an altitude of 25000 ft AGL and a speed of 450 kn

5.10 Assign escorts to planes and helicopters

Orders a group of aircraft to escort and protect another group of aircraft.

Command structure: -esc-[group name of the escort]-[name of the group to be escorted]-(engage distance)-(position front/back)-(position left/right)-(position above/under)

Parameter	Explanation	Possible values
group name of the escort	Exact name of the group that will provide the escort	text
name of the group to be escorted	Exact name of the group that will be guarded by the escort	text
engage distance	Distance in nm from the protected group at which the escort will start to engage approaching enemy planes. Defaults to 45 nm.	numbers, 1-infinite
position front/back	Position that the escorting group will take in front or behind the escorted group	f for front, b for behind, followed by distance in m i.e.: b50 -> escorting group will stay 50 m behind the escorted group
position left/right	Position that the escorting group will take left or right of escorted group	l for left, r for right, followed by distance in m i.e.: l200 -> escorting group will fly 200 m left of the escorted group
position above/under	Position that the escorting group will take left or right of escorted group	a for above, u for under, followed by distance in m i.e.: u200 -> escorting group will fly 200 m under the escorted group

Example: -esc-Cap_F15_1-B52#1-60-b200-r200

Group “CAP_F15_1” is ordered to protect group “B52#1” and to engage all enemy aircraft that approach to within 60 nm. The F-15s will take a position 200 m right and 200 m behind the B-52s (4 o’clock).

5.11 Order planes and helicopters to follow another aircraft

Orders a group of aircraft to follow another group of aircraft.

Command structure: -fol-[group that leads]-[group that follows]-(position front/back)-(position left/right)-(position above/under)

Parameter	Explanation	Possible values
group that leads	Exact name of the group that will lead	text
group that follows	Exact name of the group that will follow	text
position front/back	Position that the following group will take in front or behind the leading group	f for front, b for behind, followed by distance in m i.e.: b50 -> following group will stay 50 m behind leader
position left/right	Position that the following group will take left or right of leading group	l for left, r for right, followed by distance in m i.e.: l200 -> following group will fly 200 m left of leader
position above/under	Position that the following group will take left or right of leading group	a for above, u for under, followed by distance in m i.e.: u200 -> following group will fly 200 m under the leader

Example: -fol-SA342M-UH60-b500

Group “UH60” is ordered to follow group “SA342”. The UH-60 will take position 200 m behind the SA342.

5.12 Make planes land at specific airbases

Orders a group of airplanes to land at the airbase closest to the map marker.

Note: This command makes use of the MOOSE function GROUP:RouteRTB(), all affected groups will respawn before going RTB. If a group has lost planes to enemy fire, these will respawn as well. I tried and failed at writing my own function for making planes land where I want them to, so you will just have to accept this quirk for now…

Command structure: -rtb-[group name]-(speed)

Parameter	Explanation	Possible values
group name	Exact name of the group that is ordered to land	text
speed (optional)	Ground speed (kn) at which the plane(s) will fly until turning final. If left blank the plane(s) will continue at their current speed	number

Example: -rtb-F16_1-400

The group named “F16_1” is ordered to go RTB to the airbase closest to the map marker and to fly there at a groundspeed of 400 kn.

5.13 Make helicopters land at a map marker

Orders a helicopter to land at the marked location on the map and to stay there for a specified amount of time. After said time has passed, the helicopter will resume its flight.

Command structure: -lz-[group name]-(stay duration)

Parameter	Explanation	Possible Values
group name	Name of the helicopter group that will perform the landing	text
stay duration	Amount of time the helicopter will remain on the ground in seconds. If left blank the helicopter will resume its flight after 120 s.	number (seconds)

Example: -lz-HueyTransport-30

Orders the group „HueyTransport“ to land at the map and to remain on the ground there for 30 seconds.

5.14 Load groups as cargo

Orders a group to enter a plane/helicopter/vehicle as cargo. Can only be performed with groups that have been specified as cargo on spawn (see 5.1). It is possible to set up groups as cargo directly in the mission editor as well, for that check out the documentation on the MOOSE Cargo Module (Section 5.2 of it tells what you need to do).

Command structure: -board-[group that is ordered to board]-[group that will perform the transport]

Parameter	Explanation	Possible values
group that is ordered to board	Exact name of the group	text
group that will perform the transport	Exact name of the group	text

Example: -board-InfSQD1-HueyTransport

Orders the group “InfSQD1” to enter the helicopters of the group “HueyTransport”.

5.15 Unload groups

Orders a goup that has been loaded into a plane/helicopter/vehicle to exit the carrier. Can only be performed if the carrier is stationary.

At this time the script doesn't offer any way to check which groups are loaded into which carriers. You have to keep track yourself.

After unboarding from the carrier the groups will automatically move to the location of the map marker that was used to issue the "-unboard" command.

Command structure: -unboard-[group name]

Erläuterungen zu den Parametern:

Parameter	Explanation	Akzeptierte Werte/
group name	Exact name of the group that is ordered to unboard from its carrier.	text

Example: -unboard-InfSQD1

Orders the group "InfSQD1" to unboard from the carrier into which it has been loaded.

5.16 Toggle immortality for groups

Turns immortality on or off for all units of a specified group.

Command structure: -imm-[goup name]-[status]

Parameter	Explanation	Possible values
group name	Exact name of the group to be affected by the command	Text
status	Specify here whether immortality is to be switched on or off	on off

Example: -imm-Tunguska-on

Turns the group "Tunguska" immortal.

5.17 Toggle invisibility for groups

Turns invisibility on or off for all units of a specified group. Note: Invisible units are only undetectable to the AI, they are still being rendered and can still be seen by human players!

Command structure: -inv-[group name]-[status]

Parameter	Explanation	Possible values
group name	Exact name of the group that will be affected by the command	text
status	Specify here whether invisibility is to be switched on or off	on off

Example: -inv-Tunguska-on

Turns the group "Tunguska" invisible.

5.18 Activate uncontrolled aircraft groups

This command allows you to switch the state of airplanes and helicopters from uncontrolled to controlled. Also works for aircraft that have been spawned on a parking spot with the "-s" command and the "-ground" parameter.

Note that this command doesn't work as a toggle. Once they have been switched to the controlled state, the groups will remain in that state.

Command structure: -ctrlon-[group name]

Parameter	Explanation	Possible values
group name	Exact name of the group that is supposed to be switched "on"	text

Example: -ctrlon-A10_1

The group "A10_1" will switch its state to controlled.

5.19 Toggle AI on/off

Toggles the AI on and off for groups. Only works with ships and ground units.

Groups with deactivated AI continue to exist in the game world, but they won't perform any actions or react to enemy detection/fire.

Command structure: -ai-[group name]-[status]

Parameter	Explanation	Possible Values
group name	Exact name of the group whose AI will be switched on/off.	text
status	Specify here wether the AI is to be switched on or off	on off

Example: -ai-SA15_3-off

Deactivates the AI of the group "SA15_3".

5.20 Shoot flares at marker

This command allows you to deploy a specified amount of coloured flares at the map marker.

Command structure: -flare-[colour]-(direction)-(amount)

Parameter	Explanation	Possible Values
colour	Flare colour	g = Green r = Red w = White y = Yellow
direction (optional)	Direction into which the flare will be shot, defaults to North	ne e se s sw w nw
amount (optional)	Amount of flares that will be shot. Interval between shots is 1 second. Only specify when more than one flare is to be shot.	2-infinity

Example: -flare-g-s-10

Shoots 10 green flares to the south of the map marker.

5.21 Place coloured smoke at marker

Places coloured smoke at the location of the map marker. You can specify an amount of time that the smoke stays active.

Command: -smoke-[colour]-(duration)

Parameter	Erläuterung	Akzeptierte Werte/
colour	Colour of the smoke	b = Blue g = Green r = Red w = White o = Orange
duration (optional)	Duration for which the smoke stays active, default is five minutes. Each smoke event in DCS stays active for 5 minutes and can't be stopped prematurely. Because of that input is automatically rounded to the nearest multiple of 5.	number

Example: -smoke-o-23

Places orange smoke at the location of the map marker. The smoke will stay active for 25 minutes, because 25 is the nearest multiple of 5 from 23.

5.22 Spawn battlefield illumination at marker

Spawns an illumination round at a specified height above the map marker, that will slowly sink to the ground and provide illumination while underway.

Command structure: -illum-(altitude)-(illumination strength)

Parameter	Explanation	Possible Values
altitude (optional)	Altitude in m AGL at which the round will spawn. Defaults to 650 m.	number, 1-infinity
illumination strength (optional)	Power of the illumination in candela (cd). Defaults to 10000 cd.	number, 1-1000000

Example: -illum-1000-20000

Spawns an illumination round 1000 m above the map marker which will shine at 20000 cd.

5.23 Place smoke and fire effects at marker

Places smoke and fire effects at the map marker. Note: These effects do not disappear with time and can't be removed by command either. They will stay active as long as the mission runs.

Command structure: -sf-[effect type]-[intensity]

Parameter	Explanation	Possible values
effect type	Specify one of eight different preset effect types.	ssf = Smoke + Fire, small msf = Smoke + Fire, medium lsf = Smoke + Fire, large hsf = Smoke + Fire, huge ss = Smoke, small ms = Smoke, medium ls = Smoke, large hs = Smoke, huge
intensity	Percentage value, determines the height of the smoke plume.	1-100

Example: -sf-ms-45

Deploys a medium sized smoke effect with a smoke plume height percentage of 45 %.

5.24 Trigger an explosion at marker/group

Triggers a bomb detonation at the location of the map marker that will damage or destroy nearby units and statics. The area of effect is dependent on the specified yield.

You can also specify a group name. **All** units within that group will be detonated.

Command structure: -exp-(group name)-(yield)-(delay)

Erläuterungen zu den Parametern:

Parameter	Explanation	Possible values
group name (optional)	Exact name of the goup whose units shall die a fiery death. Also works on groups that are controlled by players… If left blank the detonation will occur at the location of the map marker.	text
yield (optional)	Strength of the explosion in kg TNT. Defaults to 100 kg TNT.	number, 1-infinity? Have fun! 😉
delay (optional)	A delay in seconds before the explosion is triggered	d + number Exp.: “d5” for a delay of 5s

Example: -exp-30000-d5

Will trigger an explosion with a yield equivalent to 30 tons of TNT at the map marker. The explosion will happen five seconds after the command is sent.

5.25 Play sound files

Plays a sound file to various recipients.

Note: This command only can be used with files that have already been loaded into the mission. The easiest way to load a sound file into your mission is to create a trigger that is activated at mission start and that executes one of the “SOUND TO” actions with the sound file you want to have available for use with this command.

Command structure: -sound-[file name]-(recipient)

Parameter	Explanation	Possible values
file name	Exact name of the sound file to be played, **including the file ending!**	text
recipient (optional)	Group or coalition that the sound shall be played to exclusively. If left blank the sound will be played to all players on the server.	text (group name) b = blue coalition r = red coalition

Example: -sound-Intro.ogg-b

Plays the sound file „Intro.ogg“ to the blue coalition.

5.26 Set flag values

Sets the value of a specified flag. This enables you to trigger actions that you have set up in the mission editor or to fix stuck triggers.

Command structure: -flag-[flag number]-[flag value]

Parameter	Explanation	Possible Values
flag number	Number of the flag that will be affected by the command	1-999
flag value	Value the flag will be set to. Can be a number or a Boolean.	1-999 true = „FLAG ON“ false = „FLAG OFF“

Example: -flag-10-true

Sets the value of flag 10 to true (FLAG ON).

5.27 Display a message

Shows a text message in the top left corner of the screen. You can specify the recipients for who the text will be displayed and the amount of time for which the message remains visible. You can also specify if previous messages that are still on display will be deleted once a new message is sent.

The config section of “Gamemaster_Functions.lua” allows you to specify a sound file that will be played every time a message is sent (see Section 2).

Command structure: -text-[message]-(recipient)-(display time)-(clear screen)

Parameter	Explanation	Possible values
message	Here you can enter the text of the message that is to be shown. **Must not contain any hyphens!**	text
recipient (optional)	Group or coalition the message will be displayed to. If left blank the message will be shown to all players on the server.	text (group name) b = blue coalition r = red coalition
display time (optional)	Time (in seconds) the message will remain on screen. Defaults to 15 seconds.	number, 1-infinite
clear screen (optional)	If this parameter is set all previous messages that might still be on screen will be deleted and only the new message is shown. Note: Does not work for messages that are sent to coalitions (DCS Bug).	c

Example: -text-Hello World-TransportHeli1-30-c

Sends the message „Hello World“ to the group “TransportHeli1” and removes all older messages. The message will be shown for 30 seconds.

5.28 Return coordinates of map marker

Returns the coordinates of the map marker used to send the command. The coordinates are returned as text in a new map marker (allows copy-paste) that gets created at the same spot where the command marker was placed. Coordinates are returned in the following formats: Lat Long, Lat Long with decimal minutes, Lat Long Precise, MGRS.

Command structure: -coord

No parameters required.

5.29 Call external functions

Allows you to call an external Lua function and give it up to five arguments. Intended to be used to call custom functions set up in another script file, but can also call functions from the DCS scripting engine.

Command structure: -func-[function]-(1st argument)-(2nd argument)-(3rd argument)-(4th argument)-(5th argument)

Parameter	Explanation	Possible values
function	Name of the function to be called. Fields in a global function table get accepted down to the second level of subtables. I.e.: trigger.action.outText → callable! someTable.firstSubtable.secondSubtable.thirdSubtable →not callable!	text
1st argument	1st argument that will get passed to the called function.	anything
2nd argument	2nd argument that will get passed to the called function.	anything
3rd argument	3rd argument that will get passed to the called function.	anything
4th argument	4th argument that will get passed to the called function.	anything
5th argument	5th argument that will get passed to the called function.	anything
6th argument	6th argument that will get passed to the called function.	anything

5.30 Draw markings on the F10 map

5.30.1 Textboxes

Draws a textbox, with its the top left corner defined by the marker position.

Command structure: -drawtext-[text]-(coalition)-(font size)-(text color)-(background color)-(text transparency)-(background transparency)

Parameter	Explanation	Possible values
text	Text to be displayed in the textbox	text
coalition	Coalition to whom the textbox is visible, defaults to all when left unspecified	red blue neutral
font size	Letter size used to display the text. Defaults to 14, default value	t+Number Exp.: “t17” = font size 17

	can be adjusted in the config section of the script file	
text color	Color of the text in the textbox The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	r = red b = blue g = green bl = black w = white y = yellow o = orange p = purple c1 = custom color 1 c2 = custom color 2 c3 = custom color 3
background color	Background color of the textbox **Note the added "f"! f = filler** The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	fr = red fb = blue fg = green fbl = black fw = white fy = yellow fo = orange fp = purple fc1 = custom color 1 fc2 = custom color 2 fc3 = custom color 3
text transparency	Transparency setting for the text in the textbox	numbers between 0 and 1 1 = fully opaque 0 = invisible
background transparency	Transparency setting for the background of the textbox **Note the added "f"! f = filler**	f+numbers between 0 and 1 1 = fully opaque 0 = invisible Example: -f0.5

Example: -drawtext-Hello World-red-t20-bl-fp-1-f0.5

Creates a textbox saying „Hello World" with 20p black text over a semi-transparent purple background. Textbox is only visible to the red coalition.

5.30.2 Lines

Draws a line on the F10 map. Requires a second marker labeled "end" to be present on the map. The line is then drawn from the marker used to issue the command to the end marker. Optionally you can place more markers labelled "c1", "c2", "c3" …etc. The line will then be drawn from the command marker, to c1, and from there to c2, and so on, until the end marker has been reached. You can place an unlimited amount of corner markers.

Command structure: -drawline-(coalition)-(color)-(transparency)-(line type)-(label text)

Parameter	Explanation	Possible values
coalition	Coalition to whom the line is visible, defaults to all when left unspecified	red blue neutral
color	color of the line	r = red

	The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	b = blue g = green bl = black w = white y = yellow o = orange p = purple c1 = custom color 1 c2 = custom color 2 c3 = custom color 3
transparency	Transparency setting for the line	numbers between 0 and 1 1 = fully opaque 0 = invisible
line type		s = solid d = dash ld = long dash 2d = double dash dd = dot dash dot = well, take a guess…
label text	Creates a textfield adjacent to the line, the relative position must be specified by shorthand parameters. The text entered after these gets displayed in the text field. Text color and transparency for the textfield are the same as specified for the line itself. Text size can be changed in the config section. The offsets used by the position shorthands can be adjusted in the config section as well.	shorthand position params: ctn = center, offset north (halfway along a direct line between the command marker and the end marker) cts = center, offset south cte = center, offset east ctw = center, offset west end = at the end of the line beg = at the start of the line The script expects a space to be left between the shorthand parameters and the label text, e.g.: -ctn Hello World!

Example: -drawline-neutral-y-1-s-end Phase Line SIERRA

Draws a solid yellow line, labelled “Phase Line SIERRA”. The label will be displayed at the end point of the line. The line is only visible to players of the neutral coalition.

5.30.3 Arrows

Draws an arrow on the F10 map. Requires a second marker labeled “end” to be present on the map. The arrow is then drawn from the marker used to issue the command to the end marker. Optionally you can place more markers labelled “c1”, “c2”, “c3” …etc. Arrows will then be drawn from the command marker, to c1, and from there to c2, and so on, until the end marker has been reached. You can place an unlimited amount of corner markers.

Command structure: -drawarrow-(coalition)-(color)-(fill color)-(transparency)-(fill transparency)-(line type)-(label text)

Parameter	Explanation	Possible values
coalition	Coalition to whom the arrow is visible, defaults to all when left unspecified	red blue neutral

color	color of the arrow outline The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	r = red b = blue g = green bl = black w = white y = yellow o = orange p = purple c1 = custom color 1 c2 = custom color 2 c3 = custom color 3
fill color	Fill color the arrow shape **Note the added "f"! f = filler** The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	fr = red fb = blue fg = green fbl = black fw = white fy = yellow fo = orange fp = purple fc1 = custom color 1 fc2 = custom color 2 fc3 = custom color 3
transparency	Transparency setting for the outline of the arrow	numbers between 0 and 1 1 = fully opaque 0 = invisible
fill transparency	Transparency setting for the inside area of the arrow **Note the added "f"! f = filler**	f+numbers between 0 and 1 1 = fully opaque 0 = invisible Example: -f0.5
line type	Determines the kind of line that will be used to outline the arrow	n = none (no visible outline) s = solid d = dash ld = long dash 2d = double dash dd = dot dash dot = well, take a guess…
label text	Creates a textfield adjacent to the arrow, the relative position must be specified by shorthand parameters. The text entered after these gets displayed in the text field. Text color and transparency for the textfield are the same as specified for the line itself. Text size can be changed in the config section. The offsets used by the position shorthands can be adjusted in the config section as well.	shorthand position params: ctn = center, offset north (halfway along a direct line between the command marker and the end marker) cts = center, offset south cte = center, offset east ctw = center, offset west end = at the end of the arrow beg = at the start of the arrow The script expects a space to be left between the shorthand parameters and the label text, e.g.: -ctn Hello World!

Example: -drawarrow-r-1-f0-d-beg INGRESS

Draws a red hollow arrow with a dashed outline, labelled “INGRESS”. The label will be displayed at the beginning of the arrow. The arrow will be visible to players from all coalitions.

5.30.4 Circles

Draws a circle, using the command marker as the center point. The radius of the circle has to be defined with a second marker labelled “rad”.

Command structure: -drawcircle-(coalition)-(color)-(fill color)-(transparency)-(fill transparency)-(line type)-(label text)

Parameter	Explanation	Possible values
coalition	Coalition to whom the circle is visible, defaults to all when left unspecified	red blue neutral
color	Color of the circle’s outline The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	r = red b = blue g = green bl = black w = white y = yellow o = orange p = purple c1 = custom color 1 c2 = custom color 2 c3 = custom color 3
fill color	Fill color the circle **Note the added “f”! f = filler** The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	fr = red fb = blue fg = green fbl = black fw = white fy = yellow fo = orange fp = purple fc1 = custom color 1 fc2 = custom color 2 fc3 = custom color 3
transparency	Transparency setting for the outline of the circle	numbers between 0 and 1 1 = fully opaque 0 = invisible
fill transparency	Transparency setting for the inside area of the circle **Note the added “f”! f = filler**	f+numbers between 0 and 1 1 = fully opaque 0 = invisible Example: -f0.5
line type	Determines the kind of line that will be used to outline the circle	n = none (no visible outline) s = solid d = dash ld = long dash 2d = double dash

		dd = dot dash dot = well, take a guess…
label text	Creates a textfield in the center of the circle. It's possible to define position offsets in the config section of the script file. Text color and transparency for the textfield are the same as specified for the circle's outline. Text size can be changed in the config section.	t + text The script expects a space to be left between the t and the label text, e.g.: -t Hello World!

Example: -drawcircle-blue-r-fr-1-f0.5-s-t 5 Mile Exclusion Zone

Draws a circle with a solid outline and a semi-transparent red filling, labelled "5 Mile Exclusion Zone". The circle is only visible to players of the blue coalition.

5.30.5 Rectangles

Draws a rectangle, using the command marker as the top left corner. The opposite corner of the rectangle has to be defined with a second marker labelled "c".

Command structure: -drawrect-(coalition)-(color)-(fill color)-(transparency)-(fill transparency)-(line type)-(label text)

Parameter	Explanation	Possible values
coalition	Coalition to whom the rectangle is visible, defaults to all when left unspecified	red blue neutral
color	Color of the rectangle's outline The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	r = red b = blue g = green bl = black w = white y = yellow o = orange p = purple c1 = custom color 1 c2 = custom color 2 c3 = custom color 3
fill color	Fill color of the rectangle **Note the added "f"! f = filler** The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	fr = red fb = blue fg = green fbl = black fw = white fy = yellow fo = orange fp = purple fc1 = custom color 1 fc2 = custom color 2 fc3 = custom color 3
transparency	Transparency setting for the outline of the rectangle	numbers between 0 and 1 1 = fully opaque 0 = invisible

fill transparency	Transparency setting for the inside area of the rectangle **Note the added "f"! f = filler**	f+numbers between 0 and 1 1 = fully opaque 0 = invisible Example: -f0.5
line type	Determines what kind of line will be used to outline the rectangle	n = none (no visible outline) s = solid d = dash ld = long dash 2d = double dash dd = dot dash dot = well, take a guess…
label text	Creates a textfield in the center of the rectangle. It's possible to define position offsets in the config section of the script file. Text color and transparency for the textfield are the same as specified for the rectangle's outline. Text size can be changed in the config section.	t + text The script expects a space to be left between the t and the label text, e.g.: -t Hello World!

Example: -drawrect-g-fg-1-f0.5-s-t Holding Area

Draws a rectangle with a solid green outline and a semi-transparent green filling, labelled "Holding Area". The circle is visible to players from all coalitions.

5.30.6 Polygons

Draws polygons with 3-11 corners. The command marker is always the first corner, further corners have to be defined by markers labelled "c1" to "c10".

Command structure: -drawpoly-(coalition)-(color)-(fill color)-(transparency)-(fill transparency)-(line type)-(label text)

Parameter	Explanation	Possible values
coalition	Coalition to whom the polygon is visible, defaults to all when left unspecified	red blue neutral
color	Color of the polygon's outline The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	r = red b = blue g = green bl = black w = white y = yellow o = orange p = purple c1 = custom color 1 c2 = custom color 2 c3 = custom color 3
fill color	Fill color of the polygon **Note the added "f"! f = filler**	fr = red fb = blue fg = green fbl = black fw = white

	The RGB values for both the custom and predefined colors can be adjusted in the config section of the script file. If the color is not specified, the default color defined in the config section of the script file is used instead.	fy = yellow fo = orange fp = purple fc1 = custom color 1 fc2 = custom color 2 fc3 = custom color 3
transparency	Transparency setting for the outline of the polygon	numbers between 0 and 1 1 = fully opaque 0 = invisible
fill transparency	Transparency setting for the inside area of the polygon **Note the added "f"! f = filler**	f+numbers between 0 and 1 1 = fully opaque 0 = invisible Example: -f0.5
line type	Determines what kind of line will be used to outline the polygon	n = none (no visible outline) s = solid d = dash ld = long dash 2d = double dash dd = dot dash dot = well, take a guess…
label text	Creates a textfield inside the polygon. The position is determined by the mean coordinates of all corner points and deviates from the center of the polygon when corners are spaced unevenly. It's possible to define position offsets in the config section of the script file. Text color and transparency for the textfield are the same as specified for the polygon's outline. Text size can be changed in the config section.	t + text The script expects a space to be left between the t and the label text, e.g.: -t Hello World!

Example: -drawpoly-blue-b-1-f0-dot-t Patrol Area

Draws a polygon with a dotted blue outline and no filling, labelled "Patrol Area". The circle is only visible to players of the blue coalition.

5.31 Delete markings from the F10 map

Removes all drawings and map markers within a specified radius from the command marker.

Command structure: -drawdel-(radius)

Parameter	Explanation	Possible values
radius	Radius in m around the command marker. All drawings and map markers within the radius will be deleted.	numbers, 1-infinity

	Note that the reference position for map drawings is always the position of the command marker by which the drawing was created. Defaults to 500 m if left unspecified.	

Example: -drawdel-5000

Removes all drawings and map markers whose reference position is within 5000 m of the command marker.

6. List of group templates

The following table lists all groups that are included in the "Gamemaster_Templates.lua". All these groups can be spawned with the "-s" command (see 5.1), **but only if "Gamemaster_Templates.lua" has been loaded at mission start and if the table they are included in has been selected for loading in the config section of the script file.** The table name shown in square brackets is the name under which the table can be found in the config section of "Gamemaster_Templates.lua".

Eastern Main Battle Tanks [MBTEast]	
Add sqd directly behind group name to spawn a squad of four units (t55sqd → 4 T-55s get spawned)	
t55	
t72b	
t72b3	
t80	
t90	
ztz	
Western Main Battle Tanks [MBTWest]	
Add sqd directly behind group name to spawn a squad of four units (leo1sqd → 4 Leopard 1s get spawned)	
leo1	
leo2a4	
leo2a4trs	
leo2a5	
leo2a6	
challenger2	
chieftainmk3	
leclerc	
merkava	
patton	
abrams	
Eastern Infantry Combat Vehicles and Troop Transports [ICVEast]	
Add sqd directly behind group name to spawn a squad of four units (bmd1sqd → 4 BMD-1s get spawned)	
bmd1	
bmp1	
bmp2	
bmp3	
pt76	
btrrd	
fddm	
mtlb	
btr80	

btr82	
zbd	
Western Infantry Combat Vehicles and Troop Transports [ICVWest]	
Add sqd directly behind group name to spawn a squad of four units (fuchssqd → 4 TPZ Fuchs get spawned)	
fuchs	
aav7	
m113	
mephisto	
stryker	
strykeratgm	
strykermgs	
strykericv	
strykeratgm	
marder	
mcv80	
lav25	
Bradley	
Eastern Air Defences [ADEast]	
sborka	
ewr1l13	
ewr55g6	
p19sr	
hq7	
tunguska	
osa	
tor	
sa9	
sa13	
sa2l	low strength
sa2m	medium strength
sa2h	high strength
sa3l	low strength
sa3m	medium strength
sa3h	high strength
sa6l	low strength
sa6m	medium strength
sa6h	high strength
sa10l	low strength
sa10m	medium strength
sa10h	high strength
sa11l	low strength
sa11m	medium strength
sa11h	high strength
igla	
shilka	
zsu57	
zu23	
zu23ins	
zu23closed	
zu23closedins	
zu23ural	
zu23uralins	
Western Air Defences [ADWest]	
rolandewr	
roland	

avenger	
chaparral	
linebacker	
rapierl	low strength
rapierm	medium strength
rapierh	high strength
hawkl	low strength
hawkm	medium strength
hawkh	high strength
patriotl	low strength
patriotm	medium strength
patrioth	high strength
stinger	
gepard	
vulcan	
Eastern Artillery [ArtyEast]	
Add sqd directly behind group name to spawn a squad of four units (except mortar)	
mortar	
nona	
gvozdika	
akatsia	
msta	
grad	
smerch	
uragan	
silkworm	
scud	
Western Artillery [ArtyWest]	
Add sqd directly behind group name to spawn a squad of four units (danasqd → 4 Danas get spawned)	
dana	
firtina	
paladin	
m270	
Eastern trucks and Armed/Unarmed Vehicles [TrucksEast]	
Add sqd directly behind group name to spawn a squad of four units (tigrsqd → 4 Tigr get spawned)	
uaz469	
tigr	
atz5	
atz10	
atmz	
5i57	no sqd available
apa5d	
apa80	
uralarmor	
ural4320	
ural375	
gaz66	
kamaztruck	
kraz6322	
zil131	
cobra	
brdm	
Western Trucks and Armed/Unarmed Vehicles [TrucksWest]	
Add sqd directly behind group name to spawn a squad of four units (m818sqd → 4 M818 get spawned)	
landrover109	

hmmwv	
hmmwvm2	
hmmwvtow	
m818	
hemtt	
hemtttanker	
cppredator	
cptrojan	
Civilian Trucks and Vehicles [TrucksCiv]	
civtruckblue	
civtruckred	
civtruckcamo	
civcarwhite	
bluebus	
yellowbus	
whitebus	
uralfiretruck	
aa7firetruck	
Eastern Infantry [InfEast]	
solrus	soldier (russia)
paraaks	paratrooper AKS (russia, blue barret)
pararpg	paratrooper RPG (russia, blue barret)
rforinfgr	3*AK, 1*RPG
rforinfsqd	10*AK, 2*RPG
Western Infantry [InfWest]	
solm4	NATO-soldier with M4
solm249	NATO-soldier with M249
natoinfgr	3*M4, 1*M249
natoinfsqd	10*M4, 2*M249
Insurgent Infantry [InfIns]	
solins	soldier (insurgents)
solak	soldier AK (bearded guy with beanie)
solrpg	soldier RPG (bearded guy with beanie)
insinfgr	3*AK, 1*RPG
insinfsqd	10*AK, 2*RPG
Eastern Convois [ConvoisEast]	
rforuaconv	
rforaconv	
rforsamconv	
Western Convois [ConvoisWest]	
natouaconv	
natoaconv	
natosamconv	
Eastern Ships [ShipsEast]	
neustra	
type52b	
type52c	
type54a	
type71	
grisha	
molniya	
rezky	
moskva	
pyotr	

kuz	
kuzsc	
ssk641	
ssk877	
type93	
Western ships [ShipsWest]	
combattante2	
tico	
perry	
burke	
tarawa	
cvn70	Vinson
cvn71	Roosevelt
cvn72	Lincoln
cvn73	Washington
cvn74	Stennis (not Supercarrier)
cvn75	Truman
Civilian Ships [ShipsCiv]	
cargoyak	
cargoivan	
elnya	
seawise	
zvezdny	
handywind	
Eastern CAP Planes [CAP1sEast, CAP2sEast, CAP3sEast, CAP4sEast]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
capmig15	
capmig19	
capmig21	
capmig23	
capmig25	
capmig29a	
capmig29s	
capmig31	
capsu27	
capsu30	
capsu33	
capjf17	
capl39za	
Western CAP Planes [CAP1sEast, CAP2sEast, CAP3sEast, CAP4sEast]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
capf86f	
capf4e	
capf5e	
capf14a	
capf14b	
capf15c	
capf16a	
capf16c	
capf18a	
capf18c	
capajs37	
capc101	

capm2000c	
capm2000c5	
WW2 Era CAP Planes [CAP1sWW2, CAP2sWW2, CAP3sWW2, CAP4sWW2]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
capp47	
capp51	
capspitfire	
capbf109	
capfw190a	
capfw190d	
capi16	
Eastern CAS planes [CAS1sEast, CAS2sEast, CAS3sEast, CAS4sEast]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
casmig15	
casmig19	
casmig21	
casmig23	
casmig27	
casmig29a	
casmig29s	
cassu17	
cassu24	
cassu25	
cassu25t	
cassu27	
cassu30	
cassu33	
cassu34	
casl39za	
casjf17	
Western CAS Planes [CAS1sWest, CAS2sWest, CAS3sWest, CAS4sWest]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
casa10a	
casa10c	
casa10cii	
casharrier	
casf86	
casf5e	
casf14b	
casf16c	
casf18c	
casajs37	
casc101	
WW2 Era CAS Planes [CAS1sWW2, CAS2sWW2, CAS3sWW2, CAS4sWW2]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
casp47	
casp51	
casfw190d	
casi16	
Eastern Ground Attack Planes [GA1sEast, GA2sEast, GA3sEast, GA4sEast]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	

gamig19	
gamig21	
gasu17	
gasu24	
gasu25t	
gasu27	
gasu30	
gasu33	
gasu34	
gatu22	
gatu95	
gatu160	
gajf17	
Western Ground Attack Planes [GA1sWest, GA2sWest, GA3sWest, GA4sWest]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
gaa10a	
gaa10c	
gaa10cii	A-10C II Tankkiller
gaharrier	
gab1	
gab52	
gas3b	
gaf86	
gaf4	
gaf5	
gaf14	F-14B
gaf15e	
gaf16a	
gaf16c	
gaf18a	
gaf18c	
gatornadogr	
gatornadoids	
gaajs37	
WW2 Era Ground Attack Planes [GA1sWest, GA2sWest, GA3sWest, GA4sWest]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
gap47	
gap51	
gafw190a	
gafw190d	
Eastern SEAD Planes [SEAD1sEast, SEAD2sEast, SEAD3sEast, SEAD4sEast]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
seadmig27	
seadsu17	
seadsu24	
seadsu25	Su-25T
seadsu30	
seadsu34	
seadjf17	
Western SEAD Planes [SEAD1sWest, SEAD2sWest, SEAD3sWest, SEAD4sWest]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	

seadharrier	
seadf16a	
seadf16c	
seadf18a	
seadf18c	
seadtornadogr	
seadtornadoids	
Eastern Anti-Ship Planes [AS1sEast, AS2sEast, AS3sEast, AS4sEast]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
asmig27	
assu17	
assu24	
assu25t	
assu30	
assu34	
astu22	
astu142	
asjf17	
Western Anti-Ship Planes [AS1sWest, AS2sWest, AS3sWest, AS4sWest]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
asb52	
asf16a	
asf18a	
asf18c	
asajs37	
asc101	
astornadogr	
astornadoids	
Eastern Transport Planes [Trans1sEast, Trans2sEast, Trans3sEast, Trans4sEast]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
tan26	
tan30	
til76	
tyak40	
Western Transport Planes [Trans1sWest, Trans2sWest, Trans3sWest, Trans4sWest]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
tc130	
tc17	
Eastern CAS Helicopters [HCAS1sEast, HCAS2sEast, HCAS3sEast, HCAS4sEast]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
caska50	
casmi8	
casmi24	
casmi28	
Western CAS Helicopters [HCAS1sWest, HCAS2sWest, HCAS3sWest, HCAS4sWest]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
casuh1	
casah1	
casah64a	

casah64d	
casoh58	
cassa342l	
cassa342m	
cassa342mini	
assh60	SH-60 in anti-ship configuration
capsa342	SA342 Mistral
Eastern Transport Helicopters [HTrans1sEast, HTrans2sEast, HTrans3sEast, HTrans4sEast]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
tka27	
tmi8	
tmi26	
Western Transport Helicopters [HTrans1sWest, HTrans2sWest, HTrans3sWest, HTrans4sWest]	
Add 2, 3 or 4 directly behind group name to spawn flights of 2, 3 or 4 aircraft (only works if multiplane group tables have been selected for loading)	
tuh1	
tch47	
tch53	
tuh60	