

```

local function removespaces( _text )
    _text = _text:gsub( " ", "" )
    _text = _text:gsub( "- ", "" )
    return _text
end

-- Function to validate the structure of the MGRS coordinates
local function checkValidMGRS( _mgrs, len )
    if len == 13 then
        Pattern: 2 digits 1 letter UTM Zone, 2 letters 8 digits Easting, 4 digits Northing
        return _mgrs:match( "^[0-9]{2}[A-Z][0-9]{8}[0-9]{4}$" )
    elseif len == 10 then
        Pattern: 2 letters MGRS Gridref, 4 digits Easting, 4 digits Northing
        return _mgrs:match( "^[A-Z]{2}[0-9]{4}[0-9]{4}$" )
    elseif len == 8 then
        Pattern: 4 digits Easting, 4 digits Northing
        return _mgrs:match( "^[0-9]{4}[0-9]{4}$" )
    else
        return false
    end
end

-- Function to validate and complete MGRS coordinates
local function processMGRS( _text, _playerPos )
    local cleanedText = string.upper( removespaces( _text ) )
    local len = #cleanedText
    local MGRS = checkValidMGRS( cleanedText, len )
end

```

Multiplayer version

Introduction

The Arty Spotter Script was inspired by the use of helicopters as airborne artillery spotters. In DCS (SP) we often identify targets, but we have to engage them ourselves, as the AI wingman will only attack predefined targets.

The intention of the script is to give the (single) player a tool, with which he can identify targets on any map, at any location and have them destroyed.

The script is independent from unit names or any AI assets at all, to provide easy accessibility for everyone. It will work for helicopters and jets with TPODs alike, even with aircraft that rely on Mk I eyeballs.

No AI artillery needs to be placed on the map, thus no worrying about minimum and maximum ranges.



How does it work

There are two ways to mark a target location.

Empty map marker

Aircraft like the Gazelle provide you with range and bearing to a target.



Once you acquired a target, switch to the F10 map and use the ruler to find the approximate target position. Alternatively you can use landmarks to identify the target location.



Now place an empty map marker on the F10 map. The script will acknowledge the marker with “Marker added”.



The marker's position will now be interpreted as the target location and you can call the artillery using the F10 Radio commands.

Entering a MGRS coordinate

Aircraft with a TPOD or the Kiowa Warrior with its MMS, provide an exact MGRS coordinate of the target.



Write down the coordinate and open the F10 map. You can now place a map marker anywhere you like and then input the coordinate into the marker's text field.

In the above example, you can input:

36R XA 5818 2679 or *XA 5818 2679* or *5818 2679*

Blanks and dashes will be removed by the script and missing UTM Zone ("36R") and Digraph ("XA") will be added automatically, based on the player aircraft's position. Only convention is that Easting and Northing ("5818 2679") need to be 4 digits each. Click on the map, to log in the input.



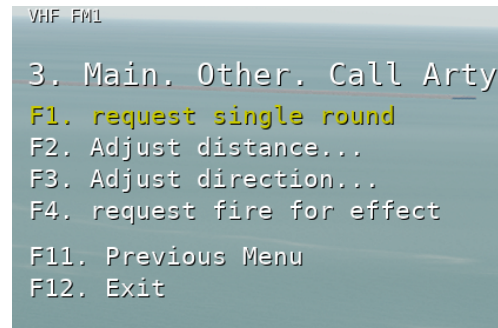
If a valid coordinate is entered, this will override the marker's position as the target location and be used instead. The position of the map marker is thus irrelevant. The script will acknowledge the input with the message "Marker changed".

Note: to prevent exploitation of the artillery, a maximum distance between player aircraft and target is implemented. By default this is 15 kilometers, but can be adapted by the user.

Calling the artillery

Once the map marker has been placed, the player can call the artillery using his F10 radio commands and selecting the sub menu “Call arty”.

The first shot will be a single round to calibrate the fire. In case the fire is off target, the player can adjust the fire through the F10 menu options:



- ***adjust distance***
the distance can be adjusted in steps of 20, 50, 100, 200 and 500 meters.
- ***adjust direction***
the direction can be adjusted as North, North-East, East, South-East, South, South-West, West, North-West

It is recommended to first adjust the distance and then the direction.

The player can now either request another calibration round on the adjusted position or, when happy, he can request fire for effect.

The default setting for “fire for effect” is 20 rounds with an impact radius of 50 meters. Both values can be adjusted to the user’s liking.

Engage a new target

If the player wants to engage a new target, he can either remove the marker and place a new one or simply enter a new coordinate into the textfield (remember to click onto the map, to acknowledge the text input).

User variables

The user can change four variables to adapt the artillery's behavior to his liking.

```
6      -- User configurable variables
7
8      local user_fireDelay = 10
9      local user_quantity = 20
10     local user_spread = 50
11     local user_spottingDistance = 15
12
13     -- end of user block
```

- ***user_fireDelay***: this is the time it takes for the rounds to arrive after an artillery strike was requested (in seconds). With real artillery this can take minutes, so users can adjust this value to have it either more realistic (higher value) or a bit more fun (lower value).
- ***user_quantity***: this the number of rounds that will be fired in a fire for effect request. Rounds will impact with a 1 second interval.
- ***user_spread***: this is the impact radius (in meters) when a fire for effect is requested.
- ***user_spottingDistance***: this is a cheat prevention, to keep players from marking a target far beyond their visual range. It is the maximum allowable distance between player aircraft and target location in kilometers. For jets using TPODs it is recommended to raise this value.

User flags for interaction with triggers in the ME

The script provides two user flag to interact with it:

artyEnabled

artyEnabled is set to true by default, but can be set to *false* (and *true* again) using normal triggers, for instance to restrict the use of artillery by time or make it dependant on the player being within (or outside) a specific zone.

Simply use the trigger action FLAG ON (*artyEnabled*) and FLAG OFF (*artyEnabled*).

artyFired

artyFired is set to true everytime a (valid) firing task was issued. You can use this flag in the ME triggers for instance to create a cool down period and allow usage of the artillery only every 10 minutes or the like.

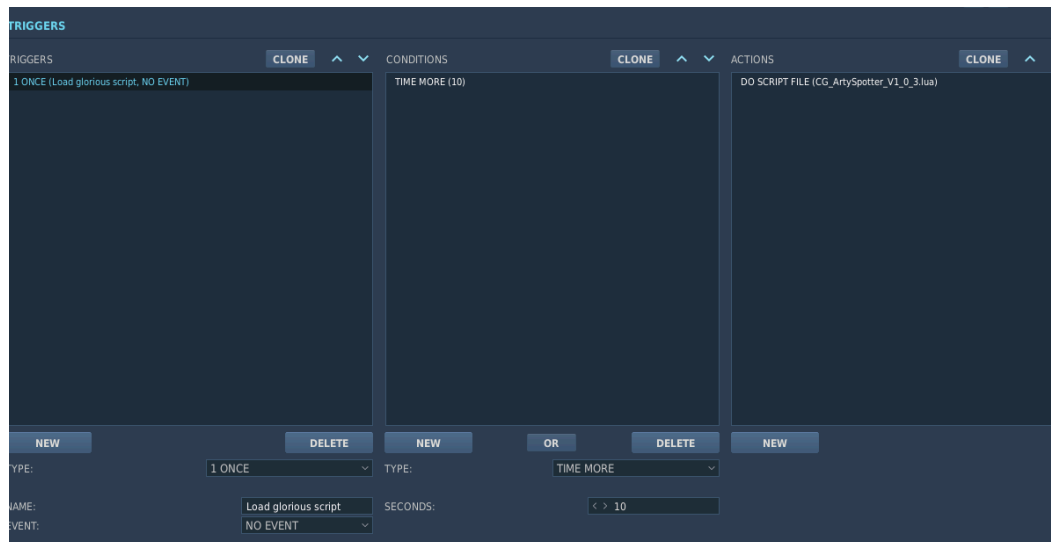
Building the script into a mission

The script needs only one trigger in the mission editor in order to run:

TRIGGER: **ONCE** *(best select a name like "load script")*

CONDITION: **TIME MORE (x)** *with x being seconds after mission start. As usually a lot happens at mission start, best load a script after a little while.*

ACTION: **DO SCRIPT FILE** and browse to your downloaded cg_Arty_Spotter_Script_v1_0_3.lua.



Multiplayer

This version of the script was optimised for multiplayer use. Changes in comparison to the original script include:

- artillery tasks will be stored related to the unit who placed the map marker
- on screen messages will only be displayed to the group to which the marking unit belongs
- addition of three possible restrictions:
 - restrict by type
 - restrict by unit name
 - restrict by markertext prefix

```
12 local user_restrictByType = ""
13 local user_restrictByUnitName = ""
14 local user_markerPrefix = ""
```

user_restrictByType

The MP host can choose to restrict the usage of artillery to helicopters only, in this case please use *"helo"*.

user_restrictByUnitName

The MP host can choose to restrict the usage of artillery based on a predefined string as part of a unit's name, for instance *"spotter"*. In this case all units that have *"spotter"* anywhere in their name, will be allowed to call artillery.

user_markerPrefix

In order to deconflict artillery task map markers from other map markers, for instance used as navigation or target points, the MP host can set a mandatory prefix, that needs to be used in the map marker's text field, like *"#arty"*.

Imprint

cg_Arty_Spotter_Script_v1_2a_MP.miz

version 1.2a MP

06/2024

By Don Rudi

Enjoy!