



FIRE FOR EFFECT

THE GOD OF WAR

ver. 1.14

by Rydygier

INTRODUCTION

Purpose of „Fire For Effect” addon is to add in simple way smart AI artillery to the battlefield.

Addon supports all generic Arma 3 artillery units, custom artillery may be added via init variables.

INSTALLATION

Addon version is installed by placing folder “@Ryd_FFE” in the same place, where are situated other addon folders. Script version is installed by placing contents of folder “Script version” folder in a mission folder.

USAGE

To use the FFE addon version in your missions launch the game with FFE addon, place on map some artillery pieces, any units of same side acting as FO, some OPFOR as targets and FFE module. That’s all.

To init FFE script version, place following eg in the init field of any unit or activation field of trigger or in init.sqf etc.: **nul = [] execVM "RYD_FFE\FFE.sqf";**

User can also call smoke or illumination fire mission directly via:

_CFF = [reference position, available artillery groups array, ammo type, amount, FO unit] call RYD_ArtyMission;

For example for placing smoke screen between `_badguy` and player:

_CFF = [position _badguy, [Battery1,Battery2], "SMOKE", 6, player] call RYD_ArtyMission;

`_CFF` is an array, that contains boolean flag indicating, if mission can be conducted, chosen battery group(s) leader(s) of that artillery group(s) and used magazines.

CONFIGURATION

FFE is ready to use without any initial configuration, since 1.14 should also auto-detect and use properly any custom artillery assets, but still offers several config options to customize its work by defining listed variables in init.sqf or any act field in editor (eg **RydFFE_Active = false**; values in braces are default):

RydFFE_Active (true)- setting this as false will terminate FAW;

RydFFE_Manual (false)- setting this as true will make FAW waiting with calculating each new fire mission for another variable: **RydFFE_Fire** (false) must be set as true (eg with trigger activated via radio) each time available arty groups should seek for new targets;

RydFFE_NoControl ([])- each arty group (battery) held inside this array will be excluded from FAW control;

RydFFE_ArtyShells (1)- positive integer. Multiplier of default magazines loadout per kind per each artillery piece;

RydFFE_Interval (10)- time gap (in seconds) between each “seek for targets” cycle (each cycle each not busy battery on map looks for new fire mission opportunity);

RydFFE_Debug (false)- if set as true, will be shown map markers that allows user to watch, what is going on. See DEBUG MARKERS chapter for details;

RydFFE_FO ([])- if this array is set as not empty (even with **objNull**), limited spotting mode becomes active, so only members of groups, which names are inside this array or which leaders are of proper class (see below), will have ability of spotting targets for batteries. By default, in unlimited spotting mode, each unit of same side, as battery, or allied excluding civilians, will transmit known enemy coordinates to the battery;

RydFFE_FOClass

```
(["i_spotter_f", "o_spotter_f", "b_spotter_f", "o_recon_jtac_f", "b_recon_jtac_f", "i_sniper_f", "o_sniper_f", "b_sniper_f",  
"i_soldier_m_f", "o_soldier_m_f", "b_g_soldier_m_f", "b_soldier_m_f", "o_recon_m_f", "b_recon_m_f", "o_soldieru_m_f", "i_uav_01_f",  
"i_uav_02_cas_f", "i_uav_02_f", "o_uav_01_f", "o_uav_02_cas_f", "o_uav_02_f", "b_uav_01_f", "b_uav_02_cas_f", "b_uav_02_f"])
```

- this array holds class names (**lowercase only!**) of leaders of groups, that will be automatically added to the **RydFFE_FO** array, if that array already is not empty (means if limited spotting is active);

RydFFE_OnePhase (false) - by default, if limited spotting is active, spotters will guide fire in two steps: 1/6 of salvo (rounded up) will be used as adjustment fire, and rest, with better accuracy, in next salvo as “fire for effect” (FFE). In unlimited spotting or if this variable is set as true, whole salvo will be fired at once as FFE;

RydFFE_2PhWithoutFO (false) – with this set as true also in unlimited spotting mode artillery fire will be conducted in two steps, as described above;

RydFFE_Amount (6) – this holds number of shells, that in summary should be fired in each fire mission. CLUSTER and GUIDED salvo amount is always divided by 3 (rounded up);

RydFFE_Acc (2) – multiplier of whole salvo drift radius. The bigger value, the bigger radius;

RydFFE_Safe (100) – salvo will be not planned for coordinates located within this radius (in meters) around any allied group leader;

RydFFE_Monogamy (true) – by default each enemy group can be a target for only one battery at the time. If set to false – there is no such limitation, so one target can be shelled by any number of batteries at the time;

RydFFE_ShellView (false) – if set to true, in debug mode there is available a tool, that allows to watch shells during their flight marked with orange “drops” in chosen meter radius around position clicked (LMB) on map. Shift + LMB removes that watching circle;

RydFFE_SVRange (3000) – range of ShellView tool in meters;

RydFFE_FOAccGain (1) – additional accuracy multiplier in second, FFE step of fire mission due to adjustments after first step. The lesser value, the greater accuracy, must be not negative. It is multiplied by spotter adjustments factor for FFE stage: $(0.2 + (\text{random } 0.2))$ that changes salvo drift radius;

RydFFE_IowaMode (false) – FFE auto-detects Iowa battleship asset and will use dedicated for it mode of fire execution. If for any reason that mode should be used for any other asset (eg. another custom battleship), user can activate this mode manually, forcing it via this variable. Arty working fine in default mode should work also in “Iowa mode”;

Since 1.14 below four “Add” variables are optional, to use if for any reason auto-detection doesn't work as desired.

RydFFE_Add_SPMortar ([]) – here you can list classnames of custom SP mortar units, that should be controlled by “FFE”;

RydFFE_Add_Mortar ([]) – here you can list classnames of custom mortar units, that should be controlled by “FFE”;

RydFFE_Add_Rocket ([]) – here you can list classnames of custom rocket artillery units, that should be controlled by “FFE”;

RydFFE_Add_Other ([]) – here you can list classnames of other custom artillery units (**lowercase only!**), that should be controlled by “FFE”, if are using custom magazines (classes added here shouldn’t be added to any other “add” array). Format:

[[[list of arty pieces classnames (lowercase)],[list of magazines for these classes in order: primary (HE), rare (WP, Cluster...),secondary (used instead of HE – guided, laser...),smoke, illum]], ...];

Example:

```
RydFFE_Add_Other =  
[  
  [ "rds_d30_fia", "rds_d30_aaf", "rds_d30_csat" ], [ "RDS_30Rnd_122mmHE_D30",  
  "RDS_30Rnd_122mmWP_D30", "RDS_30Rnd_122mmLASER_D30", "RDS_30Rnd_122mm  
  SMOKE_D30", "RDS_30Rnd_122mmILLUM_D30" ] ],  
  [ [ "rds_m119_aaf" ], [ "RDS_30Rnd_105mmHE_M119", "RDS_30Rnd_105mmWP_M119",  
  "RDS_30Rnd_105mmLASER_M119", "RDS_30Rnd_105mmSMOKE_M119", "RDS_30Rnd_10  
  5mmILLUM_M119" ] ]  
];
```

Same magazine class may be used for primary, rare and secondary ammunition. If some category isn't or shouldn't be used use "" (empty string) instead of classname.

HOW IT WORKS

Artillery of given side will shoot at enemy group (chosen with some kind of „temptation” system - targets become primarily probably more valuable enemy groups) with an amendment for its leader motion vector if:

- it is in range;
- it is spotted by any soldier of same or allied faction (except for civilian,s still spotters and other units with binoculars or sniper rifles are valuable FOs because they see and identify the enemy as potential target at a much greater distance);
- it is not captive or defenseless (leader has no weapon, no firing at harmless civilians even when they are hostile);
- it moves with speed lesser than 50 km/h because of keeping chance for hit on reasonable level;
- it is not more than 20 meters above ground (air units may become targets if are landed, or hover very low);
- there is not friendly units in „danger zone” radius (default is 100, multiplied by (1 + overcast)).

If there is allied leader in „danger zone” battery try to correct impact coordinates in attempt to avoid friendly fire by moving impact point in direction of actual target position at the cost of the amendment to the motion vector, if still there is FF-danger, battery try to fire actual target position. If still there is FF-danger, battery will call off fire mission an wait for next chance. However „friendly fire” is still possible due to salvo drift (targeting error) and single rounds impact dispersion or units movement);

Drift calculation is based on factors: distance from the target point, health status of battery leader, weather (worse weather means less predictable atmospheric conditions) and battery leader's aiming skill; Maximal drift is halved when battery shoots to the same target as before;

Dispersion calculation is currently done by Arma 3 engine.

DEBUG MARKERS

In debug mode on map will appear and disappear some markers:

- **Black circle** icon on every battery actual position with its name;
- **black dot** on planned impact position for every battery on firing mission;
- **blue circle** around planned impact position that indicates drift radius;
- **red dot** on a blue circle that indicates central point of salvo impact with text specifying: drift radius in meters, ETA, Time of Flight (TOF) and ammo type; for example: „70m – ETA: 20.5 - TOF: 10.2” - „HE”;
- there are also some **auxiliary lines** between dots.

Available is also „tracking tool” that allow to see rounds on the fly in 1500 meters radius. Projectiles are marked by **orange dots**, and radius of tracking by „**khaki**” circle.