

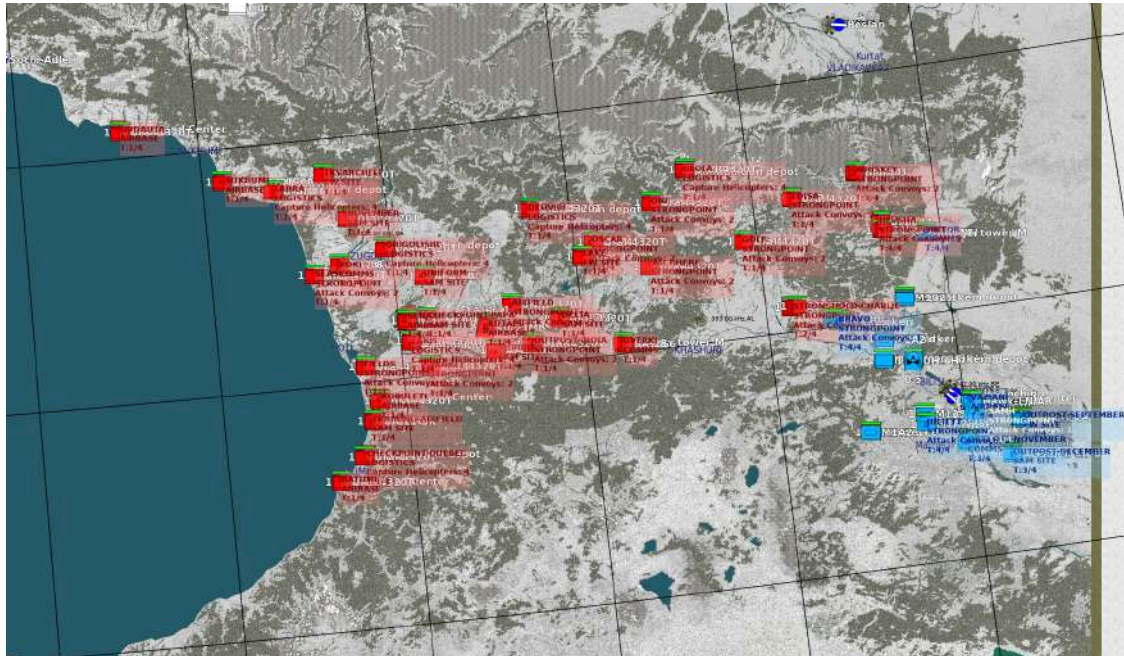
CROSSFIRE

RADIANT
V2



MISSION OBJECTIVE

Your objective, as a pilot, is to neutralize all REDFOR areas. With a custom fog-of-war script, you will only be able to see areas discovered by your coalition. Recon operations will guide you in finding undiscovered areas.



WAREHOUSE

Each coalition airbase has a tier (or level) out of 4. Main airbases are set to tier 4 from the start and have an initial stock of aircraft and equipment. Fuel and liquids are not simulated. Every time equipment and aircraft are delivered to the coalition, they are distributed equally among same-tier airbases as follows:

- 15% of supplies go to airbases tier 1/4
- 25% of supplies go to airbases tier 2/4
- 30% of supplies go to airbases tier 3/4
- 30% of supplies go to airbases tier 4/4

Equipment and aircraft are obtained from: player resupply cargo requests, periodic resupply C-130J-30, and capturing a logistics zone.



EWRS



The EWRS (Early Warning Radar System) fuses data from all units with detection capabilities to create a coalition-wide database of active airborne threats. Even though this system is very reliable, low altitude flying and terrain can prevent detection.

EW areas each contain an EWR unit. Area-level upgrades will respawn the EWR unit if lost.



EWRS is enabled from the F10 menu.

OPERATIONS

Initiated by players from the F10 menu, operations are priority-based. You may choose to use this system or not; keep in mind that completing operations rewards players with XP and notably tokens, which are used for tasking AI.



- CAP (Combat Air Patrol)
- CAS (Close Air Support)
- DEAD (Destruction of Enemy Air Defenses)
- SEAD (Suppression of Enemy Air Defenses)
- Recon

Recon operations will lead you to a set of coordinates, where the coalition believes enemy forces are present.

COMMS ANTENNAS

Situated within COMMS areas, they allow or prevent certain AI missions from being sent. EWRS refresh rate is also affected by how many COMMS Antennas the coalition controls.

They are a key asset for both coalitions, as losing all of them will stop tasking.

Lost COMMS Antennas are brought back up after some time, but a penalty is applied every time one is lost.



LOGISTICS

Logistics areas are made up of ammunition depots and support units; upon capture, the supplies are transferred into the coalition's airbases.



Capture helicopters are stored in logistics areas, and when a zone is neutralized, the coalition will attempt to capture it by sending a helicopter from the closest logistics zone.

Capture helicopters have a limited range (~150km). Users can task capture helicopters that ignore range limits using tokens.

STRONGPOINTS

A well-defended area and commonly found throughout the theatre. Strongpoints have the ability to send attack convoys if an enemy area is nearby.



SAM SITES

Ranging from medium to long range, SAM (Surface to Air Missiles) should be a high cause of concern when approaching the frontline.



XP AND RANKS

As you fly the mission, you are rewarded XP for:

- Completing an operation: **100 XP**
- Aircraft destroyed: **25 XP**
- Helicopter destroyed: **20 XP**
- Infantry kill: **1 XP**
- Vehicle destroyed: **5 XP**
- Intel report (discovering new zone): **25 XP**

For completing an operation, you will be awarded **5 tokens**. Claiming XP and Tokens is done by landing at a coalition airbase (and not crashing).

Ranking up will reward you with **10 to 20 tokens**.

Tokens are used to:

- Request resupply C-130J (**50 tokens**)
- Send JTAC (**5 tokens**)
- Send Capture Helicopter (**10 tokens**)
- Send CAS (**10 tokens**)
- Send SEAD (**15 tokens**)
- Send Strike (**15 tokens**)
- Send CAP (**10 tokens**)
- Send AWACS (**30 tokens**)
- Send Recon (**10 tokens**)

PERFORMANCE OPTIMIZATIONS

Areas that are far from the front line are scaled down (lower tier) to just a few units in order to prevent lag.

The mission itself also handles spawning and AI logic using scripting.
Note that waypoints are not inserted into each aircraft.

Initial testing using a GTX 1080ti, i7-7800X, and 32GB ram, medium settings, 1080p supported the mission with 100+ FPS using a map-wide scenario with over 250 zones.

SCENARIOS

Scenarios determine the layout of the theatre, which airbases are blue, which zone types to attribute, etc. Due to the lack of large-scale testing, a full playthrough of each scenario is required to balance zone distributions, locations, and config settings, which takes a considerable amount of time.

As of now, only one scenario is selected by default; the code is there, but the reliability is not guaranteed.

JTAC

As part of tasking requests, players can request JTACs to already discovered enemy areas. Custom JTAC uses features and scripting from Dzsekeb's well-made JTAC found in Foothold.

JUPITER

Intended for development only, this could also serve users who desire more control over the mission or even resolve issues directly within the mission.

Jupiter is a command handler using the markers in the F10 map (the orange circles). The main commands available are as follows and are enabled by default:

"-smoke" Creates smoke

"-levelup" Increase level/tier of a zone
--

"-discover" and "-discover all" to add undiscovered zones to the F10 map
--

"-dispatch" dispatches AI if possible

"-save" saves the mission manually

“-addtokens (amount)”

“-addxp (amount)”

“-destroy (area of effect in meters)” destroys all units and statics

“-sendresupply” Send a Cargo resupply AI task

“-capture (“blue” or “red”)”

EXTRAS

UPCOMING FEATURES AND IMPROVEMENTS

- C-130J support
 - Operations;
 - MOAB;
 - Ability to upgrade areas by delivering air-dropped cargo;
 - Ability to resupply coalition warehouses
- A couple more scenarios for the Caucasus map
- Improved warehouse equipment
- Rotary-wing support
 - Operations;
 - Logistics;
 - Troops and Cargo
- Crossfire: Syria
- Naval units integration
- Cold War equivalent
- Tanker tasking
- Co-op Operations
- Realistic weather

TECHNICAL NOTES

EDITING CONFIG

Crossfire's `config.lua` is currently embedded in the compiled script. A specific Discord Application (bot) is in the works to provide a hassle-free config.

Crossfire uses two files that define how the mission behaves and how the theatre is set up: `config.lua` and `scenarios.lua`

For performance considerations, these files are compiled into one script. Below are the steps to configure your mission nonetheless.

1. Unzip the crossfire `.miz` file using WinRAR or similar
2. Enter the `I10n` folder, then `DEFAULT`
3. Locate `crossfire_script.lua`
4. Edit the file as you like (edit a copy of the file outside of the mission folder)
5. Save the file
6. Enter the DCS Mission Editor
7. Open the crossfire `.miz` file, go into triggers (a menu with three very large columns)
8. First row, left column, find the second action, in the third column, which is a `DO FILE`, and click `OPEN`
9. Select your edited file by double-clicking it
10. Save the mission and move dirt
11. Note that you will have to repeat steps 8-10 each time you edit the file.

```
102  --==CONFIG==
103
104  --[[
105      DCS CROSSFIRE MISSION CONFIG FILE
106
107      This file allows you to edit how the mission responds, how AI behaved depending on multiple variables.
108
109
110      do not remove the commas at the end of each line
111      Refer to documentation for details on each setting
112  ]]
113
114
115
116  Config = {
117      persistence = {
118          enable = true, -- enables or not persistence, has authority over everything below in this section
119          save_interval = 5*51, -- (seconds) interval at which the mission state is saved
120          save_dir = "Missions/Saves/Crossfire/", -- this is your saves directory in Saved Games
121          save_file = "mission.json", -- this is the name of the mission file
122          user_data_file = "user_data.json", -- this is the name of user data only file, note that this only records user xp, tokens and rank
123      },
124
125      random_scenario_selection = false, -- allows the script to randomly choose a random scenario, authority over scenario selection
126      scenario_selected = "Georgia Liberation", -- subject to the field above, choose your own scenario
127  },
128  operations = {
129      recon_minimum_altitude = 1524, -- (meters)
130      recon_duration = 120, -- (seconds)
```

Concerning scenario edits, make sure that zone names and trigger names are the exact same. You can use the scenario examples in the code or contact me if you would like to create your own with guidance.

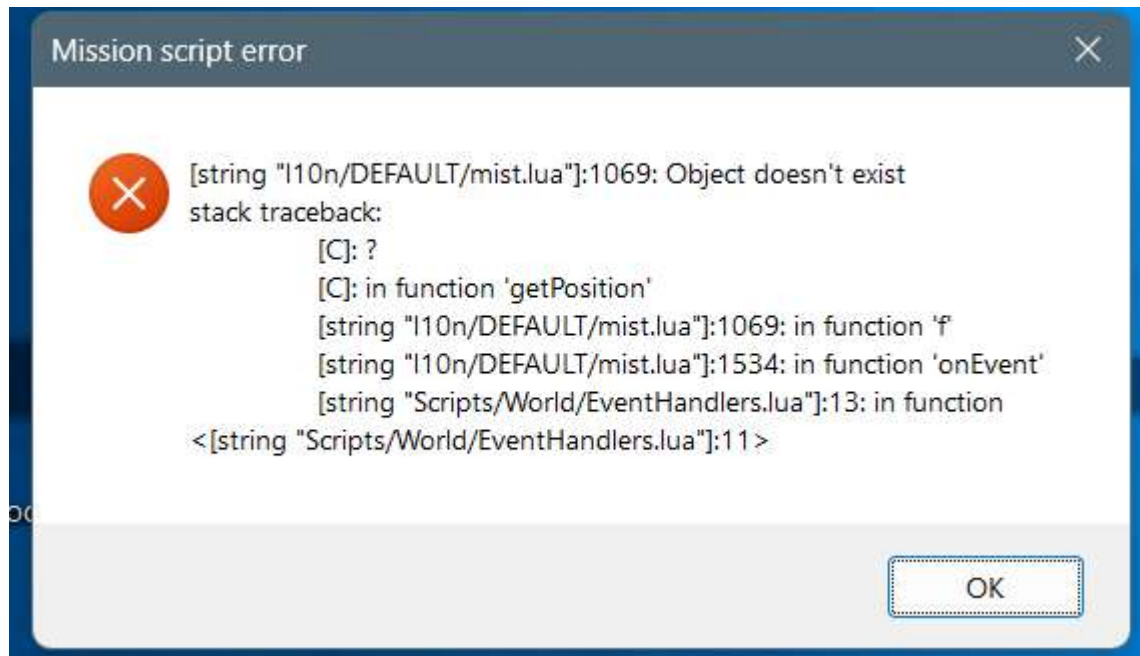
BUG REPORTS AND CONTACT

Crossfire is currently in an early stage of development. This project, which has been under development since November 2024 and currently encompasses 11,000 lines of code, has undergone rigorous testing of every feature. However, anticipating all potential edge cases

users may encounter remains challenging, and reliance on user feedback and suggestions is essential for the continuous improvement of this mission.

If you come across an error like the one below, do not hesitate to contact me on Discord or the DCS user file page.

[Crossfire Discord Server](#)



ENABLE PERSISTENCE

Crossfire uses Lua scripting to save mission-specific data such as zones, users, and scenario data. This is not yet possible using DCS' "Save Mission" feature.

Crossfire requires you to de-sanitize 'io' and 'lfs', which allows the script to write, create, and read files.

Please be aware that this removes a security feature. Proceed at your own risk.

Enabling persistence requires you to find your DCS World folder (not in Saved Games), go to the Scripts folder and `MissionScripting.lua`, from there add two dashes (to comment code) in front of `sanitizeModule('io')` and `sanitizeModule('lfs')`, the result should look like the following:


```

D: > DCS World > Scripts > MissionScripting.lua > ...
1  --Initialization script for the Mission lua Environment (SSE)
2
3  dofile('Scripts/ScriptingSystem.lua')
4
5  --Sanitize Mission Scripting environment
6  --This makes unavailable some unsecure functions.
7  --Mission downloaded from server to client may contain potential
8  --You can remove the code below and make available these functions
9
10 local function sanitizeModule(name)
11     _G[name] = nil
12     package.loaded[name] = nil
13 end
14
15 do
16     sanitizeModule('os')
17     -- sanitizeModule('io')
18     -- sanitizeModule('lfs')
19     _G['require'] = nil
20     _G['loadlib'] = nil
21     _G['package'] = nil
22 end
23

```

Crossfire saves your mission and user save file in your Saved Games/ DCS/ Missions folder, creating a Saves folder

VERSION COMPATIBILITY

As Crossfire is still in early development, new features are rolled out regularly. This means that older save files, when using persistence, will not always be compatible with newer versions. A clear sign of when this happens is when a mission scripting error appears on mission load.

I am working on a way to convert save files into their updated version using Crossfire Config.