



Script written by Grendel for DCS

# Revision History

Version	Date	Description
1.0	2025-05-09	Initial release.
1.1	2025-05-21	Updated command table with two new commands: direct and strength.  Added example 12 to show usage of these new commands.  Added previously undocumented command "speed" to table.  Reorganized command table to differentiate between persistent and immediate waypoint commands.

# TABLE OF CONTENTS

[Purpose](#)

[DCS Limitations](#)

[The Goal](#)

[Outcomes](#)

[How to Build a Mission](#)

[Formations are Fundamental](#)

[Defining a Formation](#)

[Planning for a GameMaster](#)

[Examples](#)

[Example 1: Move formation to multiple locations using a schedule](#)

[Example 2: Change formation's heading as it moves across the map](#)

[Example 3: Change formation's scale](#)

[Example 4: Designate a "leader" unit in the formation](#)

[Example 5: Keep groups whole on startup](#)

[Example 6: Dismount troops from a vehicle](#)

[Example 7: Formations can use roads if available](#)

[Example 8: Use a flag to order a formation to move](#)

[Example 9: Enable a flag when formation reaches a destination](#)

[Example 10: Change the formation's ROE](#)

[Example 11: Coordinate maneuvers using phases](#)

[Example 12: Setup formation to retreat if taking heavy losses](#)

[Demo 1: A comprehensive example](#)

[Demo 2: Total war!](#)

[Commands](#)

[Flags](#)

[Additional Considerations](#)

[Special Thanks](#)

## Purpose

Designing immersive ground combat scenarios in DCS can be a frustrating task. The default tools often fall short—especially when trying to create prolonged engagements between AI forces. In most cases, battles are over in seconds, thanks to the AI's uncanny, sniper-like accuracy.

Like many mission makers, I find this deeply unsatisfying. I want to see the battlefield *in action*—not arrive just in time to find smoking wrecks and nothing left to destroy. So how do we fix this?

Introducing **Troops in Contact** (TIC)—a script designed to transform your ground battles into dynamic, believable engagements. TIC helps mission creators quickly set up ground fights that look and feel like real combat, where both sides are exchanging fire, and there's still plenty of action left for players to engage with.

Best of all, anyone can do it! **No Lua scripting experience required.**

## DCS Limitations

DCS presents three major limitations when it comes to building large-scale, immersive, and dynamic ground war missions:

- **AI lacks strategy** – no planning, no adapting, no tactical variety.
- **AI has no autonomy** – every move must be pre-scripted by the mission maker.
- **Manual unit control** via Combined Arms is limited, clunky, and far from intuitive.

The **Troops in Contact (TIC)** script doesn't claim to solve the first two problems—but it *does* empower human GameMasters, Ground Force Commanders (GFCs), and mission makers with tools that make it *easier* to do so. TIC streamlines mission creation and provides a smoother control experience, removing many of the headaches associated with Combined Arms.

## The Goal

Easily create a convincing large-scale ground war...

### What do pilots want?

- A battlefield that *feels alive*—not static units frozen in place
- AI that *doesn't insta-wipe everything* with sniper-level precision
- The chance to *engage targets*, not arrive after the action is over
- A balanced challenge for rotary-wing pilots—without getting sniped by APCs from kilometers away
- AI that reacts realistically: retreats, takes cover, and survives long enough to fight back
- All of the above with *minimal FPS impact*

### What do GameMasters want?

- The ability to *manage the flow of battle* without micromanaging individual units
- Smarter unit movement that avoids common terrain traps (e.g. bridges, trees, buildings)
- Tools to define *battle phases* and coordinate large-scale maneuvers
- Ground units that automatically *detect and engage targets*
- A clean way to issue move orders using Combined Arms—via the "SetPath" button

### What do mission makers want?

- A fast way to create exciting ground battles with *minimal setup*
- The option to *schedule maneuvers*—so the mission plays out even without a live GameMaster
- Troops that can *mount and dismount* from APCs realistically
- Hooks to *trigger other mission events* as formations move across the battlefield
- **No Lua scripting required**

## Outcomes

- 800+ ground units active on the map with *minimal to no FPS impact*, even on a server hosting 40+ human players in various aircraft
- Ground units automatically engage in realistic, large-scale firefights—no micromanagement required
- A stuck unit will not prevent the rest of the formation from advancing
- AI accuracy is intentionally reduced, ensuring plenty of viable targets remain for human pilots
- Formations can be:
  - Large and structured
  - Directed to move, rotate, stretch, constrict, or dismount troops
- **Leader-based movement system:** Any unit in a formation can be designated as the "leader," with move orders issued via SetPath button on the CA panel in the F10 map. All other units in the formation will follow intelligently

## How to Build a Mission

1. Create a "MISSION START" trigger to load the latest `Moose.lua`
2. Set any flags to adjust TIC behaviors (see [Flags](#) below)
3. Create a "ONCE" trigger to load `TIC.lua`
4. Define late-activated **formations** (see [Defining a Formation](#) below)
5. Run mission

TRIGGERS		
TRIGGERS	CONDITIONS	ACTIONS
4 MISSION START (Load MOOSE)	TIME MORE (2)	DO SCRIPT FILE (TIC_v1.0.lua)
1 ONCE (Load TIC, NO EVENT)		

An example showing how to load the TIC script.

## Formations are Fundamental

**Formations** in TIC are similar to DCS's concept of a "group," but with a key difference: each unit in a formation is its own group of one.

Why introduce this new concept? Because DCS AI ground movement has a *major flaw*—units often get stuck on terrain features like bridges, trees, and buildings. In a traditional group (e.g. 12 units), if even **one unit gets stuck**, the *entire group grinds to a halt*.

TIC formations solve this. Since each unit is independent, **a stuck unit won't stop the rest**—the formation keeps advancing. Despite being made up of individual groups, formations still maintain their shape and cohesion as they maneuver across the battlefield.

## Defining a Formation

**Formations** are defined by naming convention so that **no LUA knowledge** is required

To create a formation...

1. Define one or more late-activated groups
2. Each group should follow this naming convention **example...**

```
"TIC:Armor Co 1#M1A2"
```

- All formations begin with the prefix "TIC:"
- `Armor Co 1` is the name of the formation. This name can be anything you want.
- `#M1A2` is used to ensure the group name is unique.
- If you want to define a formation using multiple groups, make sure each group has the same formation name in between `:` and `#`

3. Choose one of the groups in a formation to define waypoints
4. Set appropriate initial heading of formation

Set the heading of the group chosen in 3 above that best represents the overall direction the formation as a whole is "facing". For example, if formation appears mostly line abreast oriented north to south, then choose an easterly heading if this formation will be moving east. If you instead choose a northerly heading, then the formation will represent a trail formation

## Planning for a GameMaster

The first question to ask when designing a mission is:

**Will there be a live GameMaster (GM) available to direct ground formations during the mission?**

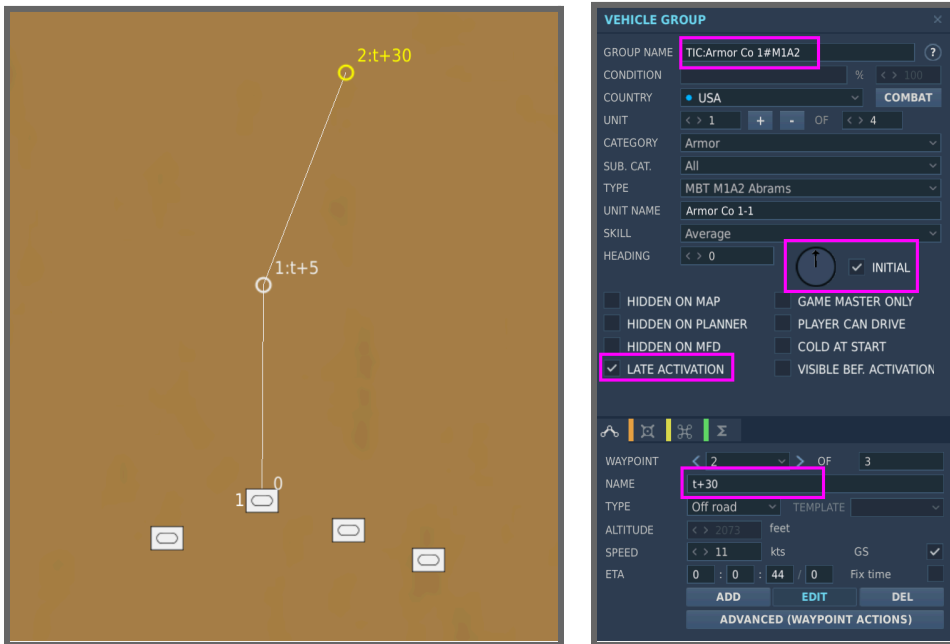
- **If yes:** You can take a more hands-off approach. Simply place the formations on the map and let the GM handle maneuvering in real time. Just make sure to define **formation leaders** (see [Example 4](#)) so the GM can issue movement commands via the F10 map.
- **If not:** You'll need to invest more time up front—planning routes, timing, engagement sequences, and battle progression. Even if a GM *might* be present, it's wise to build in scripted behaviors as a fallback to ensure the mission still plays out meaningfully.

The next section walks through several examples that showcase the flexibility and features of the TIC script. As the mission maker, it's up to you to decide which tools best fit the story and experience you're building.



## Examples

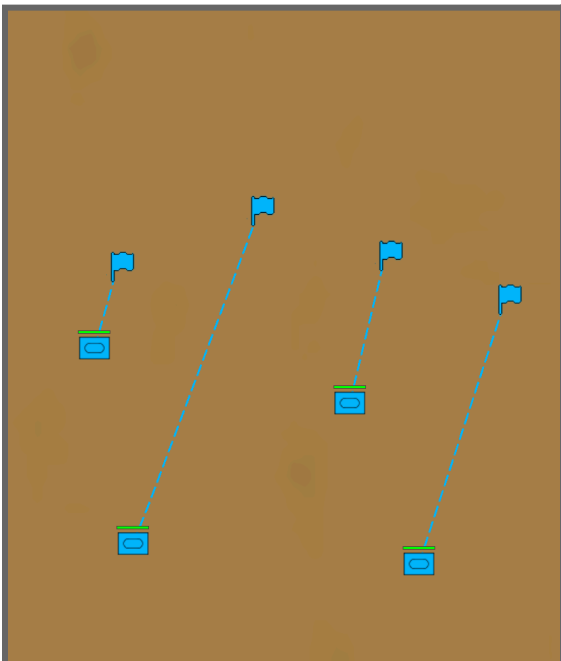
### Example 1: Move formation to multiple locations using a schedule



On mission start, the formation remains at its current location. After 5 minutes has elapsed, formation will move to wp1. Then, 25 minutes later, the formation will move to wp2. This example shows you can define commands at various waypoints, in this case we learned to use the "t+" command.

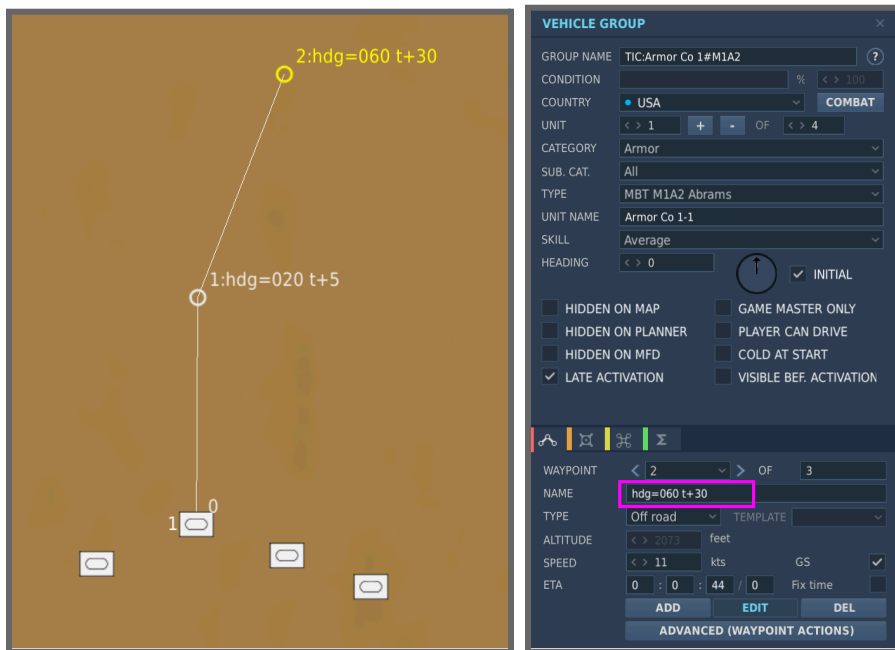
NOTE: as described under the [Flags](#) section of this document, you can override the timing of the "t+" commands. More specifically, the `tic_activate` flag can be set to false on mission start which delays "t+" commands from being processed. Once `tic_activate` is set to true, all "t+" commands will start relative to the mission time when the flag was enabled.

### Result:



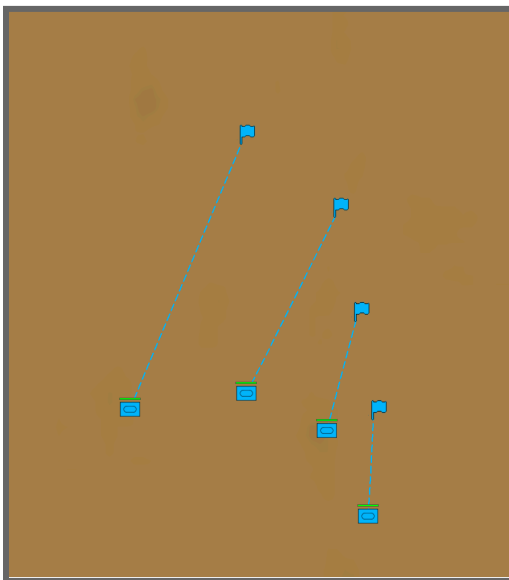
After 30 minutes, the formation begins moving to the final waypoint.

## Example 2: Change formation's heading as it moves across the map



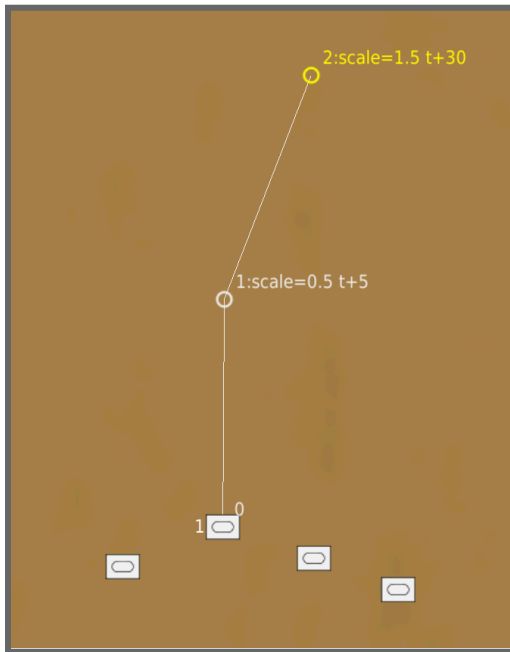
The formation will rotate to heading 020 degrees at wp1 and finally face 060 degrees at wp2. Notice how you can combine multiple commands in the same waypoint (e.g. hdg and t+). This example shows how to define the "hdg=" command. The value is any positive integer between 0 and 360 (leading zeroes are allowed) and represents a true direction on the compass rose.

### Result:



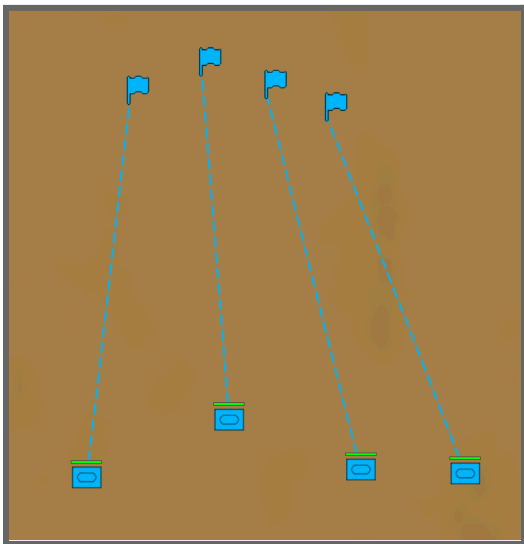
After 30 minutes, the formation begins moving to the final waypoint. Notice how the formation is rotated to face 060 degrees.

### Example 3: Change formation's scale

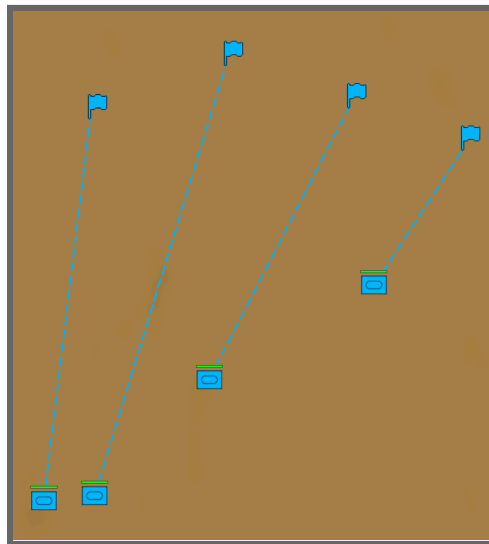


The formation constricts at wp1 and then stretches out at wp2. This example shows how to use the "scale=" command. The value is a positive number. If  $< 1$  then formation will constrict, otherwise if  $> 1$  then formation will stretch. For example, if the value is 1.5, then formation will be 150% stretched. If 0.4, then formation will be constricted to 40% its original size.

#### Results:



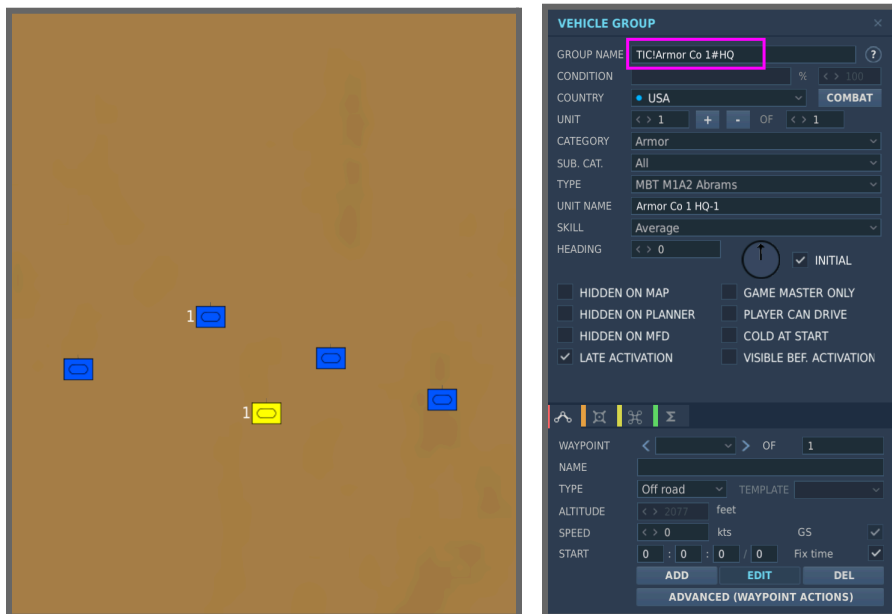
WP1



WP2

At wp1, you can see the formation narrow, and then on the way to wp2 the formation widens.

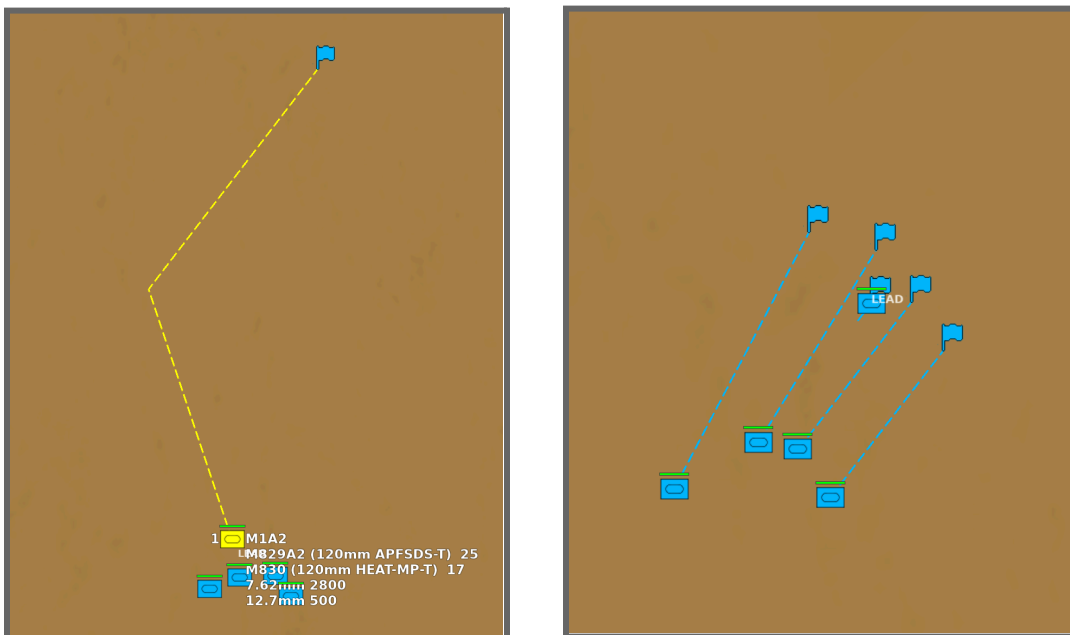
#### Example 4: Designate a "leader" unit in the formation



The yellow unit is designated as leader by simply tweaking the naming convention: instead of a colon after the **TIC** prefix, use an exclamation mark. Ex "TIC!Armor Co 1"

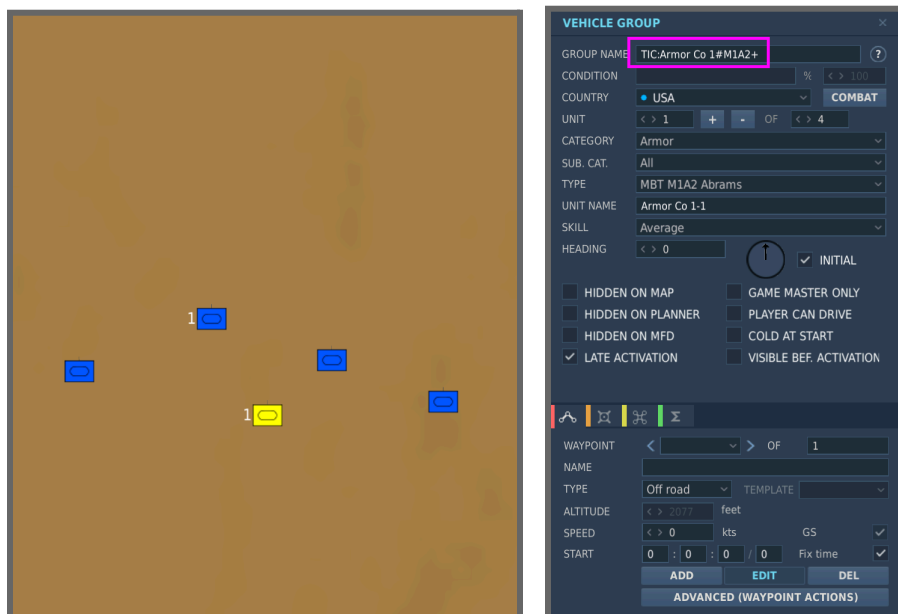
Unfortunately due to limitations with the scripting API, I am unable to detect if someone clicks the "SetPath" button to issue move orders. However, as a workaround, if you designate a unit in a formation as the "leader" and then give that unit a move command, all other units in the same formation will "follow the leader". Though a little clunky, this allows players to directly move formations from the F10 map using a mouse.

#### Results:



Lead units are marked by a label on the F10 map named **LEAD**. Units in the formation will only follow the leader! Notice that the final heading of the formation represents the course of the last 2 move coordinates. Changing the heading of a formation is as simple as defining a small dogleg.

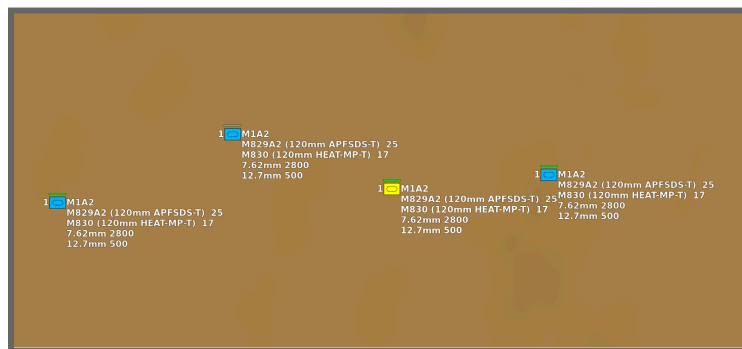
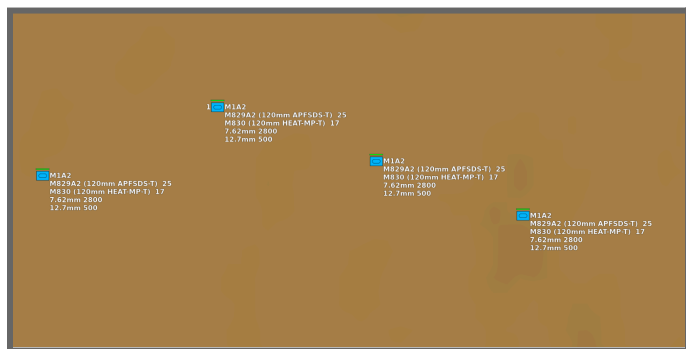
## Example 5: Keep groups whole on startup



By default, when the script activates the units, all groups are broken apart so that all groups left over have only 1 unit. There are some scenarios where you may want to prevent the breakup of a group. To do this, append a "+" to the end of the group's name so it will remain intact. I can think of two scenarios where this is desirable ...

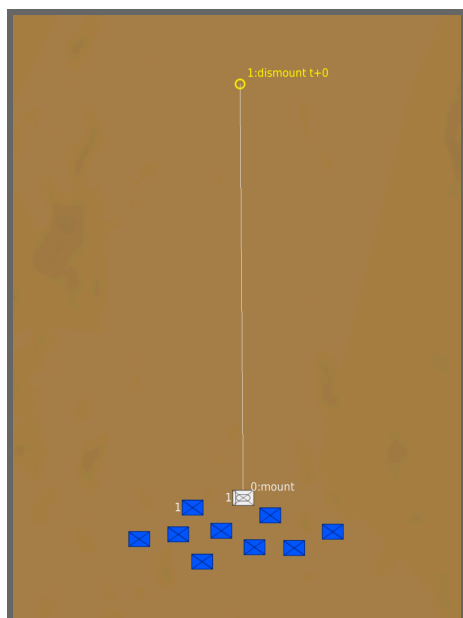
1. A group is a convoy and you want all units to stick together.
2. A player wants to ride around in a vehicle as a passenger. This can be done using the DCS AutoPilot feature. Simply slot into a vehicle and turn on AutoPilot (typical key binding is 'c'). There is one DCS limitation however, for AutoPilot to work, you should not slot into the 1st vehicle of a group. You must slot into the 2nd, 3rd, etc. vehicle. This is only possible if you leave a group whole.

## Results:



Notice that the group on the left remains whole, while the group on the right is broken into four different groups (denoted by the number 1 near the unit icon).

## Example 6: Dismount troops from a vehicle



Infantry units are treated differently than vehicles. If you place infantry near a vehicle that supports carrying troops, the infantry will automatically be assigned as passengers to that vehicle. In this example, troops first mount the vehicle nearest to them. Then the vehicle moves to a new location and dismounts all troops. This example shows how to use the “mount” and “dismount” commands.

### Results:

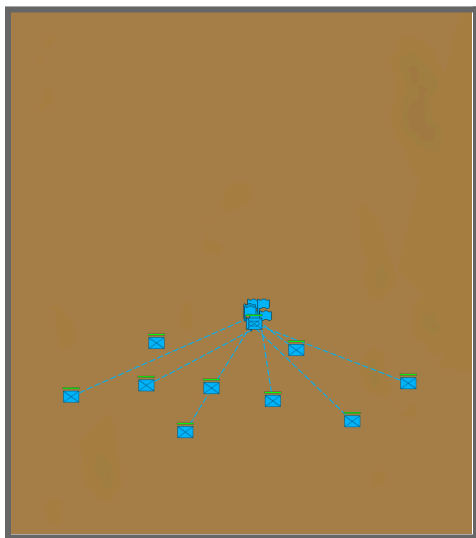


Fig 1: Troops move near IFV.



Fig 2: IFV moves to destination

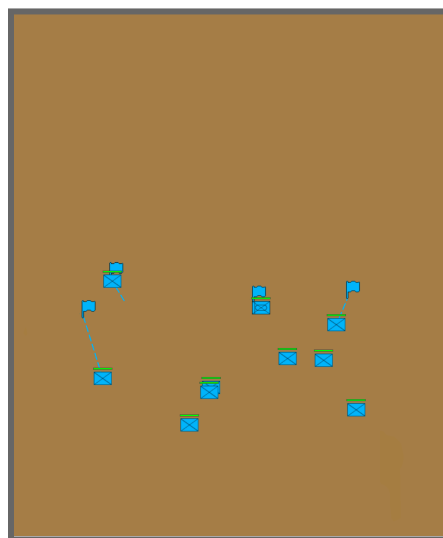
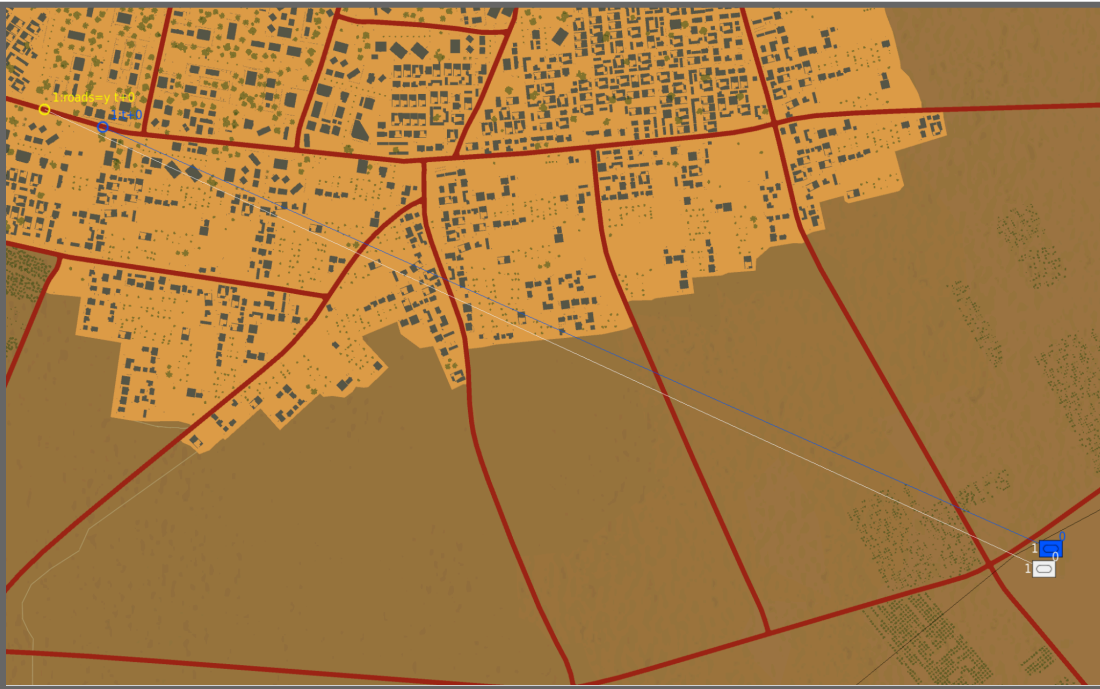


Fig 3: Troops dismount and spread out to original formation

This is all an illusion, in fact, when boarding, troops are despawned. When it is time to dismount, they are respawned at the new location.

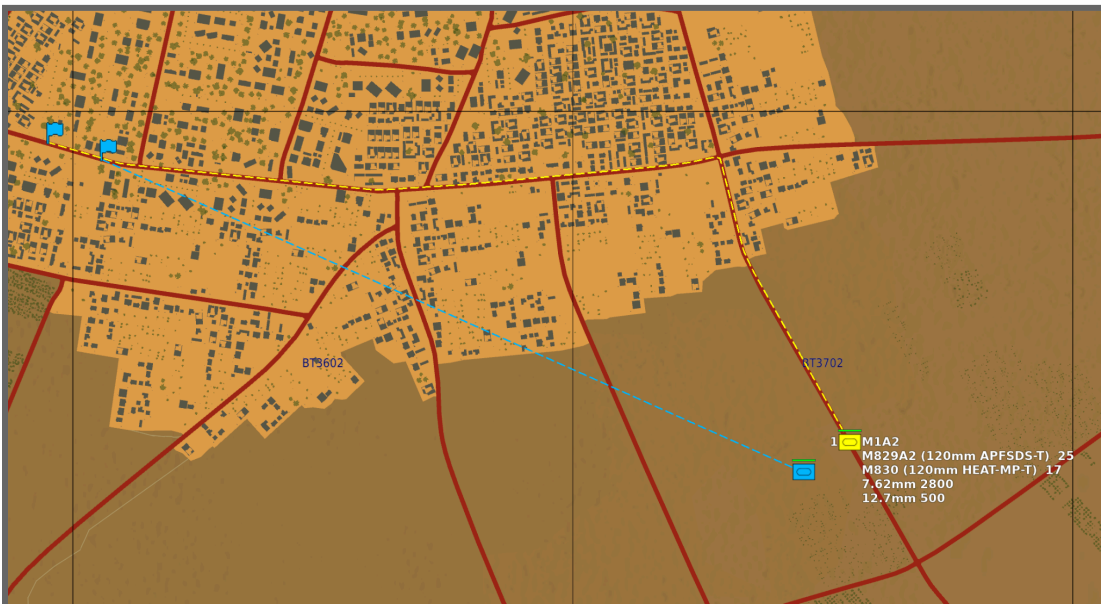
NOTE: Dismounting also works for amphibious assaults.

## Example 7: Formations can use roads if available



When travelling to a new location, you can tell the formation to use roads if available. This example shows how to use the "roads=" command. Possible values are either "y" or "n". Place this command on the destination waypoint and the formation will use roads according to the value. Be sure to set `roads=n` when you don't want the formation to use roads anymore. On mission start, `roads=n` is assumed for all formations.

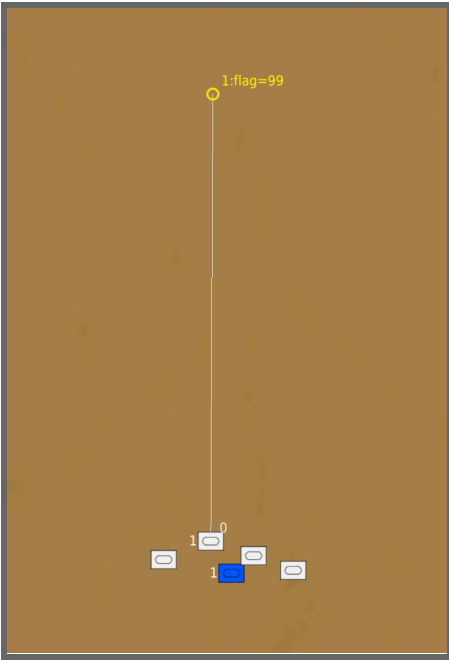
### Results:



In this example, the blue unit has no command for roads and the yellow unit has "roads=y". Therefore, the path in yellow is using roads, while the path in blue is moving directly.

NOTE: I cannot demand a unit to use a road, I can only suggest it. If the AI determines it would be quicker to move off-road, then that is what it will do, so plan and test accordingly.

**Example 8:** Use a flag to order a formation to move

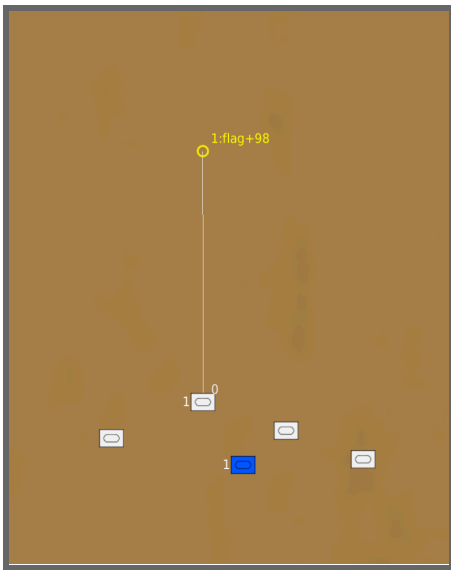


This example shows how to use the "flag=" command. The value is a number representing the flag's name. In this example, when flag 99 is turned on, the formation will automatically move to wp1.

NOTE: "flag=" and "t+" commands can be combined and will not interfere. The formation will either move based on time or by flag, whichever happens first.

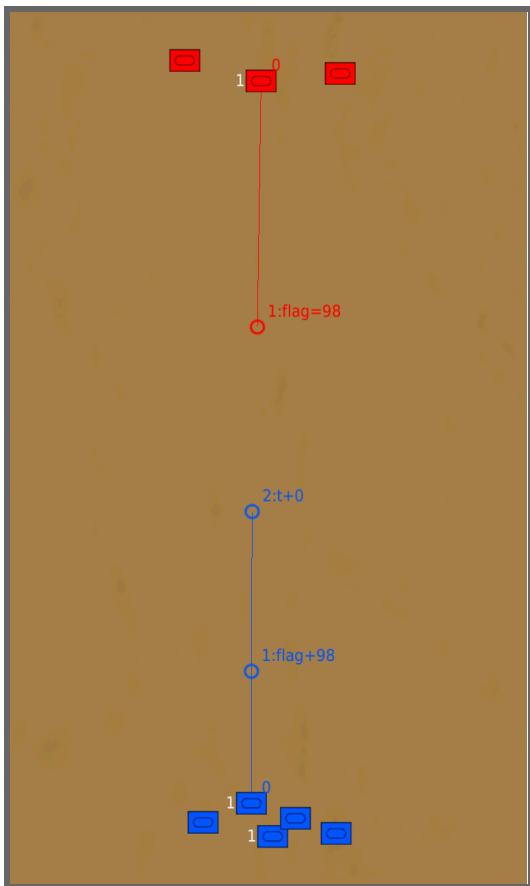


### Example 9: Enable a flag when formation reaches a destination



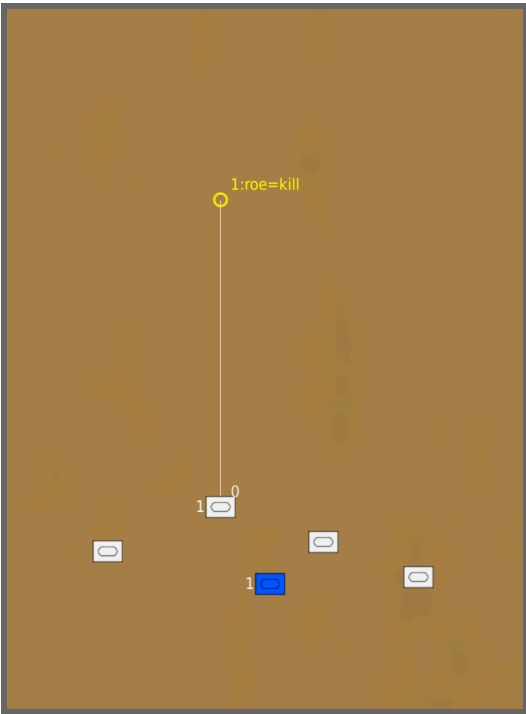
This example shows how to use the "flag+" command. The value is a number representing the flag's name. In this example, when wp1 is reached, the flag 98 is set to true. Can be used in conjunction with the "flag=" command.

### Results:



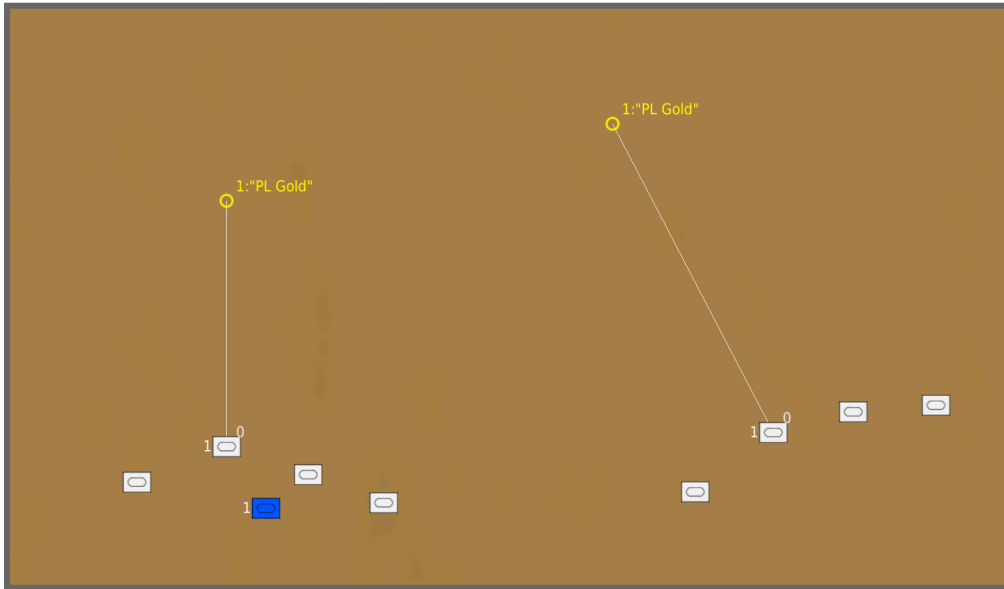
In this example, when blue reaches wp1, flag 98 is turned on. This triggers red to move to its wp1.

### Example 10: Change the formation's ROE



This example shows how to use the "roe=" command. Possible values are: `simulate`, `kill`, or `hold`. This command changes the Rules of Engagement (ROE) for the formation. By default, all formations start with a value of `simulate` (aka "storm trooper" logic). If you want formation to destroy the enemy, set the ROE to `kill`.

### Example 11: Coordinate maneuvers using phases



If the "Troops in Contact" menu is available, you can issue coordinated maneuvers using phase names. A phase name is a piece of text surrounded by double quotes (""). This is a handy way to issue one command that orders multiple formations to move simultaneously to their perspective waypoint with that phase name.

#### Results:

3. Main. Other. Troops in Contact

F1. Activate  
F2. BLUE Coalition...  
F3. RED Coalition...  
F4. DEBUG...

F11. Previous Menu  
F12. Exit

4. Main. Other. Troops in Contact. BLUE Coalition

F1. Start Phase...  
F2. ROE...  
F3. Disable Schedules...  
F4. Halt Movement  
F5. Resume Movement

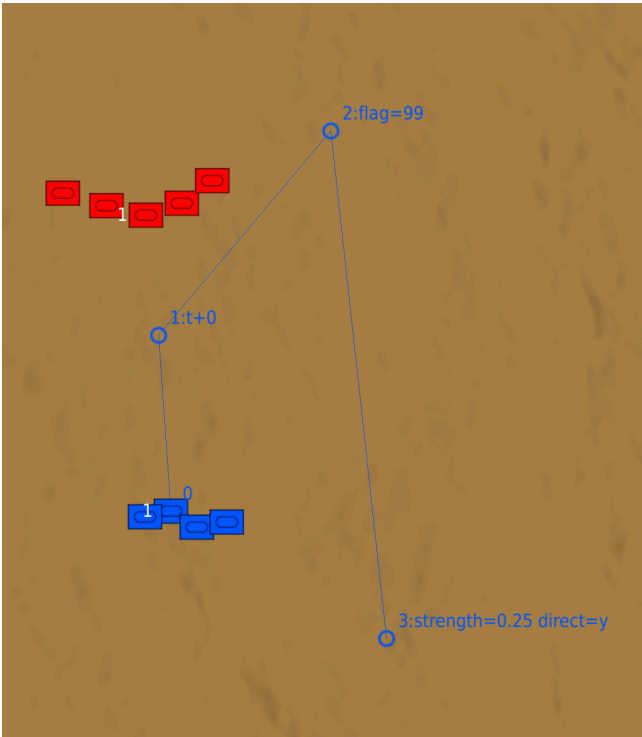
F11. Previous Menu  
F12. Exit

5. Main. Other. Troops in Contact. BLUE Coalition. Start Phase

F1. "PL GOLD"

F11. Previous Menu  
F12. Exit

### Example 12: Setup formation to retreat if taking heavy losses



This example demonstrates how to use the "strength" and "direct" commands to trigger a retreat behavior.

- The command "direct=y" is placed on wp3 to ensure the formation skips all intermediate waypoints and moves directly to wp3 when retreating.
- The command "strength=0.25" is also placed on wp3, which tells the formation to retreat to this waypoint if its remaining combat strength drops below 25%.

In this case, if blue is reduced to one remaining tank (i.e. 25% strength), the formation will bypass wp2 and move straight to wp3 to retreat.

## Demo 1: A comprehensive example

Be sure to check out the `tic_demo.miz` file included with this documentation. This mission provides a concrete example of how mission makers can leverage the features offered by the TIC script and covers all of the examples described above.

In this scenario, Red forces are dug in near an airfield, holding a defensive line, while Blue forces advance from multiple directions, executing a coordinated multi-axis assault.

Use this setup to explore how formations behave under pressure, how scripted movement and engagement logic plays out, and how the battlefield remains dynamic and full of opportunity for player interaction.

## Demo 2: Total war!

Another mission, `tic_total_war.miz`, demonstrates how easy it is to place hundreds of units in close proximity and let them unleash firepower in an all-out brawl. Despite the chaos—explosions, tracer fire, and constant engagement—very few units are actually destroyed.

This behavior is intentional and ideal: it creates a visually intense, immersive battlefield while ensuring that plenty of targets remain for players to engage. Pilots get the thrill of flying into an active combat zone—*and still have things to blow up*.

## Commands

The following table summarizes the **waypoint** commands used in the examples above.

Command	Value	Example	Description
These commands stay in effect for future waypoints			
hdg=270	number between 0 and 360	<a href="#">heading</a>	Rotate formation to this cardinal heading.
scale=1.5	number as a factor	<a href="#">scale</a>	Stretch or constrict formation by this factor. If value < 1, then formation constricts. If value > 1, then formation stretches.
roe=hold	simulate kill hold	<a href="#">roe</a>	Change rules-of-engagement for this formation. Simulate represents "stormtrooper" logic, while kill orders units to use AI logic to destroy enemy units quickly.
roads=y	y n	<a href="#">roads</a>	If y, then formation attempts to use roads to reach this waypoint.
shift=y	y n		If y, then units in the formation will move around near the waypoint so not to appear like statues. Default is y.
speed=30	number		Set speed of formation in km/hour.
These commands have an immediate effect and do not persist			
t+5	number as minutes	<a href="#">time</a>	The formation will move to this waypoint once the number of minutes has elapsed.  Value represents the number of minutes since script was activated, not necessarily when the TIC script was initialized.
flag=99	flag name	<a href="#">flag</a>	The formation will move to this waypoint when the given flag becomes true.
flag+99	flag name	<a href="#">set flag</a>	When formation reaches this waypoint, the flag is set to true.
dismount		<a href="#">dismount</a>	Order infantry units to dismount from troop carrying vehicles.
mount		<a href="#">mount</a>	Order infantry units to load into troop carrying vehicles.

"Phase 1"	text in quotes	<a href="#">phase</a>	Formation will move to this waypoint if a player orders a phase maneuver via the F-10 menu.
direct=y	y n	<a href="#">retreat</a>	Allows a formation to skip all intermediate waypoints and move straight to this one.
strength=0.3	number as a factor  Examples: 0.3 = 30% 0.7 = 70%	<a href="#">retreat</a>	Defines a combat effectiveness threshold for the formation. If the formation's strength drops below the specified value, it will proceed to this waypoint. When used with "direct", this enables retreat logic for formations under heavy losses.

## Flags

Use Mission Editor flags to adjust script behaviors. These flags are available so you are not required to write any LUA.

In DCS, a flag is essentially a named variable that holds a numeric value. While it's common in the community to name flags using just numbers (e.g. 1001), it's important to know that flag names can be descriptive strings, like `start_attack` or `drop_troops`.

The following flags are only processed when the script first loads, so they must be defined **before loading** the `tic.lua` file. After load, to adjust any behaviors, you will need to use the menu system.

Flag name	Values	Default	Description
<code>tic_init</code>	1 - true 2 - false	1	Automatically scans for groups that follow naming conventions. Set to false if you want to write LUA instead.
<code>tic_menu</code>	1 - true 2 - false	1	Indicates if the TIC menu system should be created.  You must slot into an aircraft or other role in order to see the F10 menu.
<code>tic_activate</code>	1 - true 2 - false	1	Formations will do nothing unless they are activated. By default, formations automatically activate on mission start.  Note: t+ commands are executed relative to when units are activated, not when TIC is initialized.
<code>tic_stormtrooper</code>	1 - true 2 - false	1	If true, then units open fire with low accuracy. If false, then units use normal AI logic which is considerably more accurate.
<code>tic_disableT</code>	1 - true 2 - false	2	Indicates if t+ commands should be ignored. If a human GM or GFC is going to issue commands, then disabling timings is probably advised so formations will not move unless directed by a human.

If flag is the default (equals 0), then the TIC script assumes the value in the Default column.



## Additional Considerations

### Visibility and the “Stormtrooper” Logic

To simulate poor accuracy—affectionately referred to as “**stormtrooper logic**”—the script intentionally has units aim near their targets rather than directly at them. However, there's a catch: if a unit *can see* its enemy, DCS's built-in AI will override this behavior and shoot to kill with precision.

To get around this, the script makes all controlled units **invisible** to the DCS AI. This ensures the scripted “near-miss” behavior works as intended.

While this invisibility is a necessary workaround, it comes with a tradeoff: Fog of War becomes ineffective. Because invisible units aren't “seen” in the traditional DCS sense, standard visibility mechanics won't apply.

### Role of the GameMaster

Having human GameMasters dramatically enhances the battlefield experience. A GM can adapt scenarios on the fly, create new challenges, and make unit behavior feel much more natural and reactive—something no script or AI can fully replicate.

Think of it like this:

- **One GM**—controlling both sides is similar to a PvE experience. The GM acts as a “dungeon master,” setting up challenges for the players without actively trying to “win.” The focus is on storytelling, pacing, and creating an engaging environment.
- **Two GMs**—one for Bluefor and one for Redfor—shifts the dynamic closer to PvP. Now, each side has a human actively coordinating forces against the other, making the battle more competitive and unpredictable.

When using two GMs in a PvP-style mission, it's recommended to **disable “Stormtrooper” logic**. This forces both sides to move more cautiously, plan engagements more carefully, and treat ground forces more tactically. Refer to [Flags](#) for instructions to disable “Stormtrooper” logic.

### Air Defense Units

Mobile air defense units like AAA and SAM systems do not adhere to Stormtrooper logic. They still use DCS AI targeting logic to find and engage enemy air threats.

### Manual Human Interaction

If a player issues manual move actions to a **leader** unit via “SetPath” button, then all t+ commands that are pre-defined for that formation are now cancelled.

### Debug Options

Use the DEBUG submenu under “Troops in Contact” to show the various routes for each formation. This can help the mission maker visually confirm the formations will move, scale, and face the correct heading as they move across the battlefield. Use “Smoke Positions” to pop smoke on the current location of all units, very handy to see how well the units are spread out over the 3D terrain.

## Special Thanks

### **MOOSE**

The TIC script relies on several core components from the DCS MOOSE framework. A huge thanks to everyone who has contributed to keeping the MOOSE project alive and evolving—it's no small task to maintain such a powerful toolkit as DCS continues to change over the years.