DRACO7RT · OPERATOR GUIDE · 25 OCTOBER 2026

Yamcs Operator Guide

Fly the whole mission from Yamcs, the second ground station: what to open, what to send, and what you should see at every stage

Same vehicle and same stages as the COSMOS Operator Guide, with every command sent from Yamcs. No terminal is needed: everything starts at **https://draco.efsi.com** in Chrome or Edge.

Slides 2 to 5: the ideas, signing in, the page map, how to send. Slides 6 to 15: the pad check, stages 1 to 8 one at a time, the hand-back to COSMOS. Slides 16 to 20: the mission test in one go, charts, the COSMOS screens in Yamcs, events and alarms, troubleshooting. Slides 21 to 23: what was proved, what Yamcs does not do, the bring-up for engineers. Bold = a button or page name exactly as on screen; › = click through, e.g. Yamcs › Stacks = Yamcs's left sidebar, Stacks.

IF YAMCS IS NEW TO YOU

Five ideas, then the rest of this guide reads itself

What Yamcs is

- A second ground station: open-source Yamcs 5.13.7, receiving the **same telemetry** as COSMOS
- Same vehicle, same commands, same command counters. COSMOS stays the primary station
- Yamcs sends commands only while someone has switched its commanding **on** (front door › Demo control › **Switch to Yamcs**)
- NASA 42 feeds GPS and gyro data; without it NAV and IMU read NO_FIX and NO_DATA

How things are named

- **draco**: the vehicle's instance. **Open Yamcs** on the front door opens it directly, on its **Home** page
- **Parameter**: one value, written as a path. /ACS/HK_TLM_PKT/H_MAG = target ACS, packet HK_TLM_PKT, item H_MAG: the same names as COSMOS
- **Parameter list**: a saved table of values. One per stage: DRACO 0 to DRACO 8
- **Stack**: a saved list of commands you send one at a time. One per stage, folder DRACO

The pages you will use (left sidebar)

- **Telemetry › Parameter lists**: the stage tables, live
- **Stacks**: send the commands
- **Events**: the vehicle narrates; errors in red
- **Alarms**: what is out of limits now
- **Links**: are telemetry and commanding flowing?
- **Telemetry › Parameters** › a parameter › tab **Chart**: plot it

How to read the stage slides: **Open** = where to click, **Send** = the stack and its steps, **You should see** = the values and events that mean it worked, **If not** = what to try.

START HERE

Sign in, open Yamcs, make Yamcs the commanding station

1. Sign in

- Open the front door, **https://draco.efsi.com**, in Chrome or Edge
- On the sign-in card: the blue **Sign in with Microsoft**, or type the shared COSMOS password in the box under it and press **Sign in with password**
- The status strip across the top fills in. Read **VEHICLE** (first cell): FRESH · launch-ready, and **42 VIEW**: running

2. Open Yamcs

- **Ground stations** panel (under the strip) › **Yamcs** card (second) › **Open Yamcs**: a new tab
- It opens straight on the **draco** instance's **Home** page: the live packet table
- The left sidebar reaches every page; the top bar reads **draco ▾**
- No second password: the front-door sign-in is Yamcs's only lock

3. Commanding on, and off

- Agree with anyone on COSMOS that you are commanding now
- Bottom panel **Demo control** › COMMANDING STATION › **Switch to Yamcs**, then **Run** in the dialog
- Strip cell **COMMANDING** reads YAMCS (under it: COSMOS hold); the badge reads **YAMCS commanding**; the COSMOS card: HOLD commands
- **LAUNCH (RTS-005)** and **Mission test (COSMOS)** grey out meanwhile
- When done: **Switch to COSMOS**, **Run**. The badge reads **COSMOS commanding** again

If not: Yamcs card dot red, “Not answering”: Yamcs is down, tell an engineer (slide 23). Amber, “Up, no telemetry arriving”: Demo control › HEALTH › **Downlink / map**.

GETTING AROUND

The sidebar, and one list and one stack per stage

Yamcs left sidebar	Use it for
Links	udp-in **In** rising = telemetry arriving. tc-out: DISABLED, or OK, sending
Telemetry › Parameter lists	The stage tables below, live
Telemetry › Displays	Folder **DRACO**: 14 COSMOS screens (slide 18)
Telemetry › Parameters	Search any value, e.g. H_MAG; tab **Chart**
Events	The vehicle's messages; **Start streaming** (top bar) for live
Alarms	Out of limits now; **History** tab
Commanding › Command history	Every command sent and its acks
Stacks	Folder **DRACO**: the stage commands
Archive browser	What was received, by time

Stage	Parameter list	Stack
0 Pad, link	DRACO 0 Pad check	0-check
1 Launch	DRACO 1 Launch and rates	1-launch
2 Deploy	DRACO 2 Deployments	none
3 Detumble	DRACO 1 Launch and rates	none
4 Dump	DRACO 4 Momentum and autonomy	none
5 Burn	DRACO 5 Propulsion	5-stationkeep
6 Fuel low	DRACO 5 Propulsion	6-fuel-demo
7 GPS spoof	DRACO 7 GPS gate	7-gps-spoof
8 IMU fault	DRACO 8 IMU	8-imu-fault

"none": the vehicle does it by itself. Stacks › DRACO also holds 7b-gps-after-42-restart (slide 20), 9-evs-mute, mission-test (slide 16).

HOW TO SEND · 1 MINUTE

Sending a command from a stack

Open

- Yamcs left sidebar › **Stacks** › folder **DRACO** › the stack, e.g. **1-launch.ycs**; tab **Steps**
- Plain lines are notes. Boxes are commands (**Command:** /SC/START_RTS) with their values and an **Info** line saying what should happen

Do

1. Click the command box: a green bar marks it
2. Press ▶ in the toolbar at the top, **Run the selected step**
3. Read the next note, wait as it says, then select and run the next box
4. Avoid **Run all from selected step**: the stages need waits between steps

You should see

- Under the box: **Queued**, **Released**, **Sent**, each OK with a green tick
- Sent means it left Yamcs. The proof is on the vehicle: the change the Info line promises, or the app's CMD_COUNT going up by one
- The refresh button, **Clear all outputs**, wipes the ticks before the next run

If not: **Sent NOK** with a red cross: commanding is off (slide 3, **Switch to Yamcs**); nothing reached the vehicle. A red bar "No processor 'undefined'": the page came from an old link; open it again from the sidebar.

BEFORE YOU LAUNCH · ABOUT 3 MINUTES

Stage 0: a fresh vehicle on the pad, and a working link

Open

- Front door: the status strip (top) and the **Demo control** panel (bottom)
- Yamcs sidebar: **Telemetry › Parameter lists › DRACO 0 Pad check**
- Yamcs sidebar: **Links** (rows tc-out, udp-in)

Send

1. VEHICLE not FRESH: Demo control › STACK › **Fresh vehicle**, **Run**; about 90 s until JOB reads ok and VEHICLE FRESH. 42 VIEW stopped: 3D VIEW › **42 start**
2. In **Links**, the udp-in **In** count grows each time you look
3. **Switch to Yamcs** on the front door (slide 3)
4. Stack **0-check.ycs**: run the four NOOPs, one at a time (slide 5)

You should see

- MODE SAFE, FUEL_KG 12, FUEL_STATE OK, SA_STATE and ANT_STATE STOWED, DEPLOYED_BITS 0
- IMU STATE VALID at about 9 Hz; NAV STATE FIX, FIX_COUNT up about 3 a second; RATE_SOURCE IMU
- Each NOOP: that app's CMD_COUNT +1, and an event such as ACS: NOOP (sim-grade ACS v1.0)

If not: values blank or never change: front door › Demo control › HEALTH › **Downlink / map**. A NOOP shows Sent NOK: commanding is still off.

T+0 TO T+15 S

Stage 1: launch and separation

Open

- Yamcs sidebar › **Telemetry › Parameter lists › DRACO 1 Launch and rates**
- A second tab: sidebar › **Events**, then **Start streaming** in the page's top bar

Send

1. Stacks › DRACO › **1-launch.ycs**: click the box **Command: /SC/START_RTS**, RTS_ID 5, press ▶
2. Send nothing more until Stage 5: from here the vehicle flies itself

You should see

- Events: ACS: LIFTOFF, separation in 10 s; then ACS: SEPARATION confirmed, tipoff rates applied
- MODE goes SAFE to DETUMBLE about 10 s after the command
- WX, WY, WZ jump to a few hundredths of a rad/s; RATE_MAG peaked at 0.060 on 25 October

If not: Yamcs does not check that the vehicle is fresh (the front-door **LAUNCH (RTS-005)** button does), so send it only after Stage 0. Nothing after 15 s: sidebar › **Commanding › Command history**.

T+15 TO T+40 S

Stage 2: the solar array and antenna deploy

Open

- Yamcs sidebar › **Telemetry › Parameter lists › DRACO 2 Deployments**
- Optional, second tab: sidebar › **Events**, **Start streaming**

Do

- Nothing. Two stored sequences arm and fire each deployment by themselves
- Watch each channel climb its ladder

You should see

- SA_STATE, then ANT_STATE, step through ARMED, FIRING, DEPLOYING to DEPLOYED (a step can pass between refreshes)
- Events: ACS: solar array ARMED (window 60 s), PYRO FIRED, DEPLOYING, DEPLOYED (microswitch closed); then the same for the antenna
- At the end: DEPLOYED_BITS 3, PYRO_COUNT 2. On 25 October both were out 30 s after launch

If not: a channel stops at ARMED: on Events type FIRE rejected in **Search events**. Then front door **Fresh vehicle** and launch again.

T+20 TO T+60 S

Stage 3: detumble, then point

Open

- Yamcs sidebar › **Telemetry › Parameter lists › DRACO 1 Launch and rates**
- Optional: sidebar › **Telemetry › Parameters**, search RATE_MAG, click /ACS/HK_TLM_PKT/RATE_MAG, tab **Chart**

Do

- Nothing. The wheels absorb the spin from separation by themselves
- Watch RATE_MAG come down

You should see

- RATE_MAG decays toward zero
- Events: ACS: detumble complete, rates < 0.005 rad/s
- MODE goes DETUMBLE to POINT; on 25 October 20 s after separation
- H_MAG (wheel momentum) starts to climb

If not: MODE stays DETUMBLE for more than 2 minutes: note the rates, front door › Demo control › STACK › **Fresh vehicle**, and tell an engineer.

ABOUT 1.5 MINUTES AFTER POINT, THEN REPEATING

Stage 4: the autonomous momentum dump

Open

- Yamcs sidebar › **Telemetry › Parameter lists › DRACO 4 Momentum and autonomy**
- A second tab: **Telemetry › Parameters**, search H_MAG, click /ACS/HK_TLM_PKT/H_MAG, tab **Chart**, button **15 m** in the Latest: row

Do

- Nothing. This is the point of the demo: nobody touches anything
- Watch H_MAG climb to 4.0
- AP_SAMPLE_COUNT ticking shows the onboard limit checker is watching

You should see

- At 4.0: Events: LC: ACS momentum high: DELTA_H ... RTS = 4, then ACS: momentum dump initiated and, 8 s later, momentum dump complete, |H|=0.29 N m s
- MODE POINT to DELTA_H and back; DUMP_COUNT +1; FUEL_KG down a little
- Then again every 95 to 100 s: a sawtooth on the chart

If not: H_MAG passes 4.0 and nothing happens: the actionpoint may be off. Tell an engineer; COSMOS's page Viewer › Mission › **Limits & autonomy** shows it plainly.

ANY TIME IN POINT · OPERATOR-COMMANDED

Stage 5: a station-keeping burn

Open

- Yamcs sidebar › **Telemetry › Parameter lists › DRACO 5 Propulsion**
- Optional, second tab: sidebar › **Events**, **Start streaming**

Send

1. Stacks › DRACO › **5-stationkeep.ycs**: select **/ACS/STATIONKEEP**, TARGET_DV_CMS 50, press ▶
2. The burn stops itself at 0.5 m/s
3. Optional, later: the box **/ACS/DUMP_MOMENTUM**, a dump by ground command

You should see

- MODE POINT to STATIONKEEP and back to POINT by itself, 24 s on 25 October
- DV_MS ramps to 0.5; THR0 and THR1 on-times grow; FUEL_KG down 0.1 kg
- Events: ACS: station-keeping burn complete, dV=0.500 m/s; BURN_COUNT +1
- The dump: MODE DELTA_H for about 3 s, DUMP_COUNT +1

If not: Events says burn REJECTED - propellant exhausted: in Stacks › DRACO › **6-fuel-demo.ycs** run the box SET_FUEL 12000 (refuel), then SET_MODE MODE 2, and try again.

OPTIONAL · FAULT MANAGEMENT

Stage 6: running the fuel low

Open

- Yamcs sidebar › **Telemetry › Parameter lists › DRACO 5 Propulsion**
- A second tab: sidebar › **Events**, **Start streaming**

Send

1. Stacks › DRACO › **6-fuel-demo.ycs**, one box at a time with ▶: **/ACS/SET_FUEL** 2000, then 400, then 1
2. Then **/ACS/STATIONKEEP**: it burns the last gram
3. **SET_FUEL 12000** refuels
4. Last box, **/ACS/SET_MODE** MODE 2: back to POINT

You should see

- FUEL_STATE OK, LOW, CRITICAL, then EXHAUSTED
- Events: ACS: propellant LOW (2.00 kg, < 20% of load); propellant CRITICAL in red; propellant EXHAUSTED - thrusters ineffective
- MODE goes to SAFE by itself and stays SAFE after the refuel until SET_MODE
- After refuel: FUEL_STATE OK, FUEL_KG 12

If not: no events appear: on Events press **Start streaming** (top bar), or set the time picker at the left of the filter row to **Last hour**.

NEEDS 42 RUNNING

Stage 7: GPS, and the spoof response

Open

- Yamcs sidebar › **Telemetry › Parameter lists › DRACO 7 GPS gate**
- A second tab: sidebar › **Events**, **Start streaming**

Send

1. Check in the list: NAV STATE FIX, REJECT_LATCHED CLEAR
2. Stacks › DRACO › **7-gps-spoof.ycs**: select **/NAV/GPS_FIX**, a forged fix, press ▶
3. Wait for STATE FIX again, then **/NAV/CLEAR_REJECT**
4. Only when REJECT_LATCHED reads CLEAR: **/LC/SET_AP_STATE** AP_ID 1, NEW_STATE 1

You should see

- Events in red: NAV: GPS fix REJECTED, flags 0x16; LC: NAV GPS fix rejected : AP = 1 … RTS = 6
- REJECT_LATCHED SET, REJECT_COUNT +1, LC RTS_EXEC_COUNT +1
- The next genuine fix is accepted: STATE stays FIX
- After recovery: REJECT_LATCHED CLEAR; Events: Set AP state command: AP = 1

If not: REJECT_COUNT climbs on every fix after 42 was restarted: run Stacks › DRACO › **7b-gps-after-42-restart.ycs** (slide 20).

NEEDS 42 RUNNING

Stage 8: the IMU (gyros) and the fallback

Open

- Yamcs sidebar › **Telemetry › Parameter lists › DRACO 8 IMU**
- A second tab: sidebar › **Events**, **Start streaming**

Send

1. Check in the list: IMU STATE VALID, RATE_SOURCE IMU
2. Stacks › DRACO › **8-imu-fault.ycs**: select **/IMU/SAMPLE**, a forged gyro sample (GYRO_X 5 rad/s), press ▶
3. Recover with the next three boxes, in order: **/IMU/CLEAR_FAULT**, **/LC/SET_AP_STATE** AP_ID 2, **/ACS/SET_RATE_SOURCE** AUTO

You should see

- Events: IMU: sample REJECTED, flags 0x06 in red; ACS: rate source -> INTERNAL (fault)
- FAULT_LATCHED SET; LC starts RTS-007, which pins RATE_SOURCE_CMD to INTERNAL
- The mission carries on on the internal estimate
- After recovery: FAULT_LATCHED CLEAR, RATE_SOURCE IMU within a second

If not: STATE NO_DATA or STALE: 42 is not running: front door › Demo control › 3D VIEW › **42 start**. A brief STALE that returns to VALID by itself is harmless.

AFTER THE FLIGHT · 1 MINUTE

Hand back to COSMOS and leave the vehicle ready

Do

1. Front door › Demo control › COMMANDING STATION › **Switch to COSMOS**, then **Run**
2. Tell the COSMOS users they have the vehicle again
3. Next person flies a new mission? Demo control › STACK › **Fresh vehicle** (about 90 s)

Check

- Strip cell **COMMANDING**: COSMOS; badge: **COSMOS commanding**; Switch to COSMOS greys out
- Yamcs sidebar › **Links**: tc-out Detail DISABLED; its **Out** count is what you sent since Yamcs started
- Sidebar › **Commanding › Command history**: every command you sent, with its time and acks

Good to know

- Commands queued in Yamcs while it is disabled fail at once; nothing is held back
- A Yamcs restart always comes back disabled
- COSMOS saw the whole flight too: its pages and the Grafana wall match

THE WHOLE MISSION IN ONE GO · 3 TO 4 MINUTES

Run the mission test from Yamcs: the same nine phases as COSMOS

Script: the exact copy

- Front door › Demo control › YAMCS › **Mission test (Yamcs)**, then **Run**; the log streams in the job pane, tab **Job output**
- It switches commanding on, flies P0 to P8, switches it off
- Or Yamcs sidebar › **Scripts**: Script draco_mission_test.py, Script arguments **--manage-link**, **Run**. Its log: sidebar › **Activities**

Stack: watch each check

- **Switch to Yamcs** first, then sidebar › **Stacks** › DRACO › **mission-test.ycs**
- Click the first box, press **Run all from selected step** in the toolbar
- Each check waits for its value, then ticks green; a red cross stops the run there

You should see

- Script: the log ends **MISSION TEST COMPLETE**; the job badge reads **OK**; Activities shows a green tick
- Stack: every box green down to the last text box
- Both need a **fresh vehicle** and 42 running; they launch it in P1

If not: **Mission test (Yamcs)** greyed, hover says flown this boot: **Fresh vehicle** first. A box says Sent NOK: commanding is off. A red check: note its line, see slide 20.

GRAPHING

Charts and history: the whole mission in one picture

Set it up

1. Yamcs sidebar › **Telemetry › Parameters**, search **H_MAG**, click /ACS/HK_TLM_PKT/H_MAG
2. Tab **Chart**; in the **Latest:** row above the plot pick a range: **15 m**, **1 h** … (1 m to 14 d)
3. Panel **Chart settings** on the right › **Trace 1**: the ⊕ adds another parameter, e.g. FUEL_KG
4. Any parameter list: its **Historical data** tab (beside Realtime) shows the same values over time

What the shapes mean

- H_MAG: a sawtooth. Momentum builds; the autonomous dump cuts it back to about 0.3, every 95 to 100 s
- RATE_MAG: one spike at separation, decays through detumble, flat in POINT
- FUEL_KG: steps down at each dump, slopes during a burn
- FIX_COUNT: a straight ramp while 42 feeds

Where the history is

- Yamcs keeps 30 days; a nightly job deletes older data
- Sidebar › **Archive browser** (bottom): which packets arrived, by time
- Times are when Yamcs received each packet: the vehicle clock is not UTC

SCREENS

The COSMOS screens, in Yamcs: same pages, same values

Open

1. Yamcs sidebar › **Telemetry › Displays** › folder **DRACO**
2. **home.opi**: the mission home page. Its buttons open the others
3. 14 pages: home, overview, deploy, gnc, prop, sequences, events, limits, apps, acs, gnc_sensors, imu, nav, cfdp

You should see

- Live values with units; a value out of limits turns orange (warning) or red (critical)
- Graphs against time, filled from the archive for the range at the top right (15 m by default)
- sequences: each RTS reads **READY**, **RUNNING** or **DISABLED**, as in COSMOS

Command buttons

- Same buttons as COSMOS, e.g. Dump Momentum, Gate ON
- Each asks **Send ...?** first, with the reminder that Yamcs must be the commanding station (**Switch to Yamcs**), then sends through Yamcs
- They only reach the vehicle while the front door shows **YAMCS commanding**

If not: a page stays blank: reload it. A value reads "n/a in Yamcs": COSMOS computes it itself (event time stamps). A button does nothing: front door › Demo control › **Switch to Yamcs** (slide 3).

EVENTS AND ALARMS

What the vehicle says, and what is out of limits

Events

- Yamcs sidebar › **Events**: the flight software's messages, newest first
- **Start streaming** (top bar) follows live; the filter row: **Last hour** (window), **Search events**, **Info level**, **Any source**
- Red = error or critical; black = routine. Source = the app (ACS, NAV, LC, SC ...)
- Type in **Search events**: e.g. REJECTED

Alarms

- Sidebar › **Alarms**, tab **Active**: values out of limits right now, e.g. FUEL_STATE LOW, REJECT_LATCHED SET, FAULT_LATCHED SET, RATE_SOURCE INTERNAL
- They clear themselves when the value returns: nobody needs to acknowledge them
- Tab **History** (beside Active, Pending): every alarm, how long it lasted, how it cleared

Expect, and ignore

- SCH_APP Slots skipped in red, a few a minute: the emulated computer pausing, not the mission
- A brief IMU STALE and RATE_SOURCE INTERNAL that clear within a second or two
- Needs attention: an alarm that stays, or an error from ACS, NAV, IMU, LC or SC you did not cause

WHEN IT DOES NOT LOOK RIGHT

Troubleshooting: what you see, what to do

What you see	What to do
Sent NOK with a red cross under a command	Yamcs is not the commanding station: front door › Demo control › COMMANDING STATION › **Switch to Yamcs**
Red bar: No processor 'undefined' within instance 'draco'	Old link. Open the page again from the left sidebar (or add ?c=draco__realtime to the address)
Values never change; Links udp-in **In** stays put	Front door › Demo control › HEALTH › **Downlink / map**. Still stuck: HEALTH › **Verify**, then STACK › **Fresh vehicle**
IMU NO_DATA or STALE, NAV NO_FIX or STALE	42 is off: Demo control › 3D VIEW › **42 start**
NAV rejects every fix after 42 restarted	Stacks › DRACO › **7b-gps-after-42-restart.ycs**: Gate OFF, wait 3 s, Gate ON, clear latch, re-arm AP 1
RATE_SOURCE stuck at INTERNAL after a fault	Stack **8-imu-fault.ycs**, the three recovery boxes, in order
Stuck in SAFE after a fuel demo	Stack **6-fuel-demo.ycs**, last box: **/ACS/SET_MODE** MODE 2
Events page empty	Events top bar **Start streaming**, or widen **Last hour** in the filter row
Anything else odd	Front door: **Switch to COSMOS**, **Fresh vehicle**; tell an engineer what you saw

THE RECORD

Flown from Yamcs on 25 and 26 October 2026: every stage passed

Stage	Sent from Yamcs	Checked in telemetry	Result
0	4 NOOPs	ACS, LC, NAV, IMU CMD_COUNT each +1	PASS
1-4	SC START_RTS 5	DETUMBLE 10 s; both deployed; POINT; dump at H_MAG 4.0 to 0.33	PASS
5	STATIONKEEP 50, DUMP_MOMENTUM	dV 0.500 m/s in 24 s; BURN_COUNT and DUMP_COUNT +1	PASS
6	SET_FUEL x4, STATIONKEEP, SET_MODE	LOW, CRITICAL, EXHAUSTED, SAFE; refuel OK; POINT	PASS
7	forged GPS_FIX, recovery	latch SET, RTS-006 ran, latch CLEAR, AP 1 armed 5 s later	PASS
7b	gate OFF/ON, clear, re-arm	NAV CMD_COUNT +3; still FIX	PASS
8	forged IMU sample, recovery	latch SET, INTERNAL, RTS-007 ran; back to IMU	PASS
Test	Mission test script, front-door action	P0 to P8, every check, 177 s; evidence record ground_20261025_205629	PASS
Test	mission-test stack, Yamcs UI	42 of 42 checks green, 177 s	PASS
26 Oct	script again: 2 runs, a deliberate failure, 42 stopped	9/9 in 177 and 179 s; on a launched vehicle it stopped at P0 in 30 s and switched commanding off; without 42, P7/P8 skipped, the rest passed	PASS
26 Oct	13 screens, Gate ON button	live values, RTS states as in COSMOS; command sent, NAV CMD_COUNT +1	PASS

How each row was flown and read back: in the speaker notes.

HONEST LIMITS

What Yamcs does not do here, and what rests on people

Not built

- The mission-test stack cannot calculate: it checks a fresh vehicle's starting values and spoofs with a fixed fix. The script is the exact copy of COSMOS's test
- No launch-readiness check inside Yamcs: the front-door buttons check the vehicle is fresh, a Yamcs stack or a script started in Yamcs does not
- Every boot is aimed at the ground by COSMOS (Fresh vehicle does it), so COSMOS must be running
- The COSMOS wall screens are not in Telemetry › Displays yet; 14 other COSMOS pages are (slide 18)

Rules people keep

- One commanding station at a time. Nothing stops two people commanding at once
- Yamcs has no login of its own: anyone past the front door can switch commanding on in Yamcs › Links (tc-out, **Enable**) and edit the shared lists and stacks
- So the front-door sign-in is the only lock. Share it only with people you would let command the vehicle

The vehicle and its autonomy are the same whichever station watches; only the ground side differs.

FOR ENGINEERS

Bring-up: starting Yamcs from nothing

Step	Command (from /home/efsi/draco7_0)	Check
1 Vehicle and COSMOS up	run/bootstrap-draco.sh --verify	no FAIL lines
2 Packet database	python3 tools/ground/cosmos2xtce.py --check	exit 0; if stale, run it without --check and rebuild (step 3)
3 Start Yamcs	cd run/yamcs && docker compose -p yamcs up -d --build	Links: udp-in In rising
4 Pin its CPUs	docker update --cpuset-cpus 0-29 draco-yamcs	done again after every recreate
5 Lists and stacks	python3 run/yamcs/operator/load_operator.py	7 list, 9 stack and 14 display lines printed
6 Telemetry to Yamcs	run/bootstrap-draco.sh --per-boot	tee sends to 10.5.5.200 and .202
7 Health	run/bootstrap-draco.sh --verify	Yamcs receiving; tc-out disabled; every "Yamcs:" line OK (run/yamcs/selfcheck.py)

Lists and stacks live in Yamcs's data volume and survive a restart or rebuild; step 5 is for a new volume or to restore the originals.