H145

Hype Performance Group

H145 (BK-117 D2) Helicopter
Base Pack & Action Pack
Version 3.0 (Build .500) Last Updated: 2024/08/30
User Guide V 2.13

This product is compatible only with Microsoft Flight Simulator 2020.
Our H145 helicopter has been rebuilt as a native helicopter. We now leverage the Asobo CFD helicopter flight model, however we also augment the flight controls to realistically simulate the H145 systems. Our helicopter includes unstabilized flight, stabilized flight (by Stability Augmentation System) and fully hands-off Upper Modes, like HDG/ALT/IAS. H145 also supports GPS-referenced modes and the ability to command a precision hands-off auto hover. H145 supports Force Trim (Trim Release) which you won't find in other MSFS helicopters. We also integrate a Garmin GTN750 which enables GPS approaches and much more.
We use an installer/update program called **Hype Operations Center** to manage installation of our products. You will find easy installation, quick updates, downgrades and rollback to earlier versions as you prefer.
H145 has over 600 key bindings which are available for you to bind using Hype Operations Center. You can also directly send events and monitor local variables using programs like SPAD.next or FSUIPC.

This manual consists of:

- Included PDF-Documentation (see `\Community\hpg-airbus-h145\H145 User Guide.pdf`)
- H145 Userguide (Internet)
- Hype Operation Center (Internet)
- Mission System Documentation
- Mission System (Internet)

You can find other manuals and translations on http://dvrgl.georgl.info
With thanks to Dave and the development team for their excellent product, have fun
D-VRGL, Leo Schoonboodt

Content

Historical

The H145, formerly the Eurocopter EC 145, was a further development of the MBB BO 105 helicopter. The first flight was in 1979. Development partners were MBB and the Japanese manufacturer Kawasaki Heavy Industries (KHI) with a contract from 1977. MBB later merged with the French manufacturer Aerospital (as part of Daimler Benz Airospace) to jointly produce this helicopter under the designation EC 145. In 20219, a new rotor system with a 5-blade lifting screw was introduced. It enabled a smaller blade length, a lower empty weight and a higher payload with fewer vibrations. It also made it possible to fold the rotor blades for transportation.
In 2010, flight tests began in Donauwörth with a new airframe and, for the first time, a Fenestron tail rotor, which was first presented in Orlando in 2011 and certified by EASA in 2014.
The helicopter is primarily used in the areas of air rescue and intensive care transport or as a police helicopter. Additional equipment packages from third-party manufacturers, such as a cable winch, enable further areas of use such as SAR services in difficult-to-access terrain.
In Germany, the ADAC air rescue service and the DRF Luftrettung have the largest contingents. In Spain, the Guardia Civil and in France the gendarmerie are equipped with the EC 145. Iraq, South Africa and the USA use military versions.

Specifications

Dimensions

D-Value	13.54 m
Rotor diameter	10.80 m
Internal Volume (cabin & cargo)	6.03 m²

Characteristics

Crew + Passengers	1 or 2 pilots and up to 10 passengers
Empty weight, Baseline Aircraft Def.	1,895 kg
Max takeoff weight (MTOW)	3,800 kg
Useful load	1,905 kg
Engine	2 * Afran Helicopter – Arriel 2E
Standard fuel tank capacity	723 kg
Take off power	667 kW

Performance

Recommended cruise speed	241 km/h
Max range with standard fuel tank	650 km
Max endurance with standard fuel tank	3h 35min
Flight envelope (temperature)	-45°C +50°C
Max altitude for TO and landing	6,096 m
Sound level during level flight	87.5 dB

Getting Started

Installation

Complete all three steps:

1. Download & Install H145
2. Configure Controls & Bindings
3. Learn about Aircraft Settings

If you have already installed the H145, copy the directories from 2020 directly into the community directory of 2024.

- hpg-airbus-h145
- hpg-airbus-h145-usersetup
- hpg-hatws-data

Change your community folder in HYPE Operation Center from 2020 to 2024.
PMS50-gtn750 get the updated MFS24 version directly from pms50's website!

Flying Basics

H145 is similar to something like a mix of an airliner and a traditional helicopter.

Before flight, the pilot should turn on A.TRIM , AP1 , AP2 and BKUP (These are all on the Autopilot control panel or APCP, which is on the center console below the GTN750).

The pilot shall hold Cyclic Trim Release before raising the collective for takeoff. The pilot will lift off from the ground and only once in a hover may the trim be released. Once the trim is released, the aircraft AFCS will continually attempt to maintain the current attitude where the pilot last let go of trim release.

The autopilot has more than just attitude stabilization (which is the normal mode, and not visualized on the MFD), there are also so-called UPPER MODES . These upper modes are traditional flight control modes like HDG (heading hold), ALT (altitude hold), but also ground-based GPS modes likes GTC. You can use the tablet autopilot panel (click the clock at the top of the tablet) to learn more about the modes and to issue commands without setting up buttons on your controller.

Remember, you need to heed the limits on the FLI (the vertical tape to the left side of your pilot MFD). Take-off power (yellow) is allowed for 30 minutes per flight. Once you exceed and go "into the red", the engines won't be able to keep up and your rotor rpm will slow. If the rotor rpm slows below 80%, you will fall out of the sky.

Next you can learn more about the AFCS and other Aircraft Systems and Procedures (Checklists).

Notable Changes since Version 1.0

- The new native flight model is more challenging and also authentic in both AFCS OFF and AFCS ON modes.
- You may use the MSFS Piloting Assistance: Tail Rotor setting to neutralize rotor torque and simplify flying (for twist grip controllers and gamepads), but do not use the MSFS Piloting Assistance: Cyclic Assist in MSFS as it will not work with H145. This replaces the Arcade flight model.
- Stability Level is configurable for your controller, for each axis.
- Trim Release (CYCLIC TRIM RELEASE) is required now. It exists as the same control in 1.0, however it is necessary to use it now, even when follow-up trim is available.
- A.TRIM OFF requires Trim Release, if you preferred this mode, change to the Latching Trim Release bindings.
- New settings are available on the tablet.
- AirlandFS support is deprecated.

Information for Livery Authors

- Paint Kit
- Configure external hardware JSON
- texture.cfg for each variant

See Livery Author Info

Controls & Bindings

Control bindings are now available in either the previous fixed-wing bindings or the new (SU11 and later) native helicopter bindings. You can use either version, but do not use both at once for the same axis as this will result in loss of control.

Primary Flight Controls

Function	MSFS Axis Bindings
Collective	`THROTTLE AXIS` or `COLLECTIVE AXIS`
Cyclic Pitch	`CYCLIC LONGITUDINAL AXIS`
Cyclic Roll	`AILERONS AXIS` or `CYCLIC LATERAL AXIS`
Yaw Pedals	`RUDDER AXIS` or `TAIL ROTOR AXIS` *

* You may also use the split-rudder axis (`RUDDER AXIS LEFT` and `RUDDER AXIS RIGHT`)
Button inputs are also available on all axis.

Beep Trim

Function	MSFS Axis Bindings
Cyclic Pitch	INCREASE ROTOR LONGITUDINAL TRIM and DECREASE ROTOR LONGITUDINAL TRIM
Cyclic Roll	INCREASE ROTOR LATERAL TRIM and DECREASE ROTOR LATERAL TRIM
Collective	INCREASE AUTOPILOT N1 REFERENCE and DECREASE AUTOPILOT N1 REFERENCE
Yaw Pedals *	RUDDER TRIM LEFT and RUDDER TRIM RIGHT

* - Yaw trim is known as Collective Left/Right beep trim.

Other Important Bindings

Function	MSFS Binding	Notes
Cyclic Trim Release	ROTOR TRIM RESET	Hold down anytime you manipulate the cyclic with the AFCS. There are also latching bindings available to avoid holding it.
AP/BKUP ON	AUTOPILOT ON	Press once to engage BKUP/AP1/AP2, press again to engage ALT/HDG/IAS.
AP/UM OFF	AUTOPILOT OFF	Press once to cancel upper modes. Hold for 2 seconds to clear all bugs
AP/BKUP CUT	TOGGLE DISENGAGE AUTOPILOT	Press once to disengage AP1/AP2. Press again to disengage BKUP.
AP/GTC	TOGGLE AUTO HOVER	Press once for GTC. Press twice for GTC.H
RESET (message list)	ANNUNCIATOR SWITCH OFF	Clear message list on FND page
OEI HI/LO	ARM AUTO THROTTLE	Toggle between OEI HIGH and OEI LOW rating
Go Around	AUTO THROTTLE TO GA	Activate Go-Around mode
Bambi Bucket Dump Cargo Attach/Detach Fire Weapons	TOGGLE YAW DAMPER	

Take care to use the **On Release** logic in MSFS bindings, to avoid key repeating. This does not apply to Trim Release as it is configured to be held with repeating.

Continue with the 1.0 documentation for more information on bindings, or proceed to Hype Operations Center to bind others.

Custom Control Bindings

Add custom control bindings through Hype Operations Center.

1. Open `Hype Operations Center` (You must have version 1.0.31 or higher for language selection)
2. Select your language under `Settings` (this ensures you see localized text which will be the same as you see later in MSFS)
3. Visit the H145 page, then click `View/Modify Key Bindings`

4. Scroll to the bottom and click `Add Binding` to define a new control binding
5. Pick a MSFS event (nearly at random) on the left side. You can start with `SET ADF VOLUME`
6. Pick the H145 function on the right side (You may test this in the cockpit using the Event Tester tablet app)
7. Click Save.
8. If MSFS is already running, you must now **restart the flight** (not the full simulator).
9. In MSFS, select the binding you picked in step 5, and the button on your controller.

You're done. Pressing the button should now activate the H145 function. You can repeat this process to create as many custom bindings as needed.

`*` - MSFS `HELD` events are likely to have incorrect/odd behavior with repeating. `*` - Using the 'On Release' logic in MSFS often avoids incorrect key repeating which could break the ability to use a toggle-style button.

Trim Release

Trim Release (Cyclic Trim Release) is a button on the cyclic control which the pilot holds anytime they are manipulating the cyclic. This is a very important aspect of H145 and doing so both pauses the AFCS (so it doesn't fight you) and also gives you maximum precision (no deadzone).

There are multiple bindings you can choose from, the HOLD version (which you hold continually) or the LATCH version (which you simply click), which you may assign to a switch or other logic. When pressing Trim Release or hands on detection is activate, you will see “OVERRIDE” at FND-page.

If you fail to use trim release, you will fight the autoflight systems!

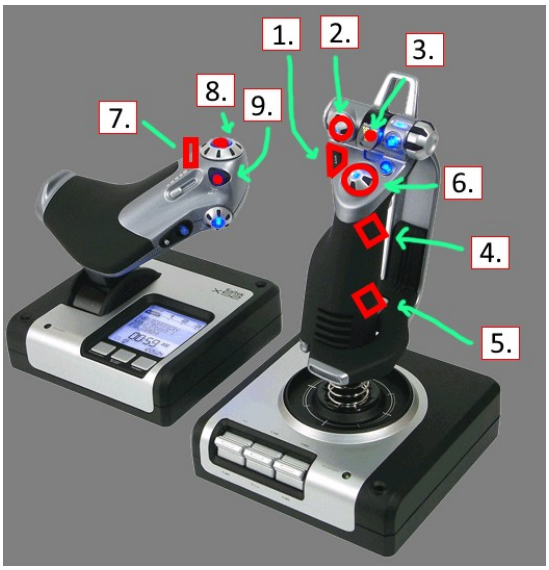
Note: In previous H145 versions, `A.TRIM OFF` could be used similarly, however now you must move to the latching bindings (see below).
Note: If you see OVERRIDE flickering it is possible that your hands on-detection is the problem.

Binding	Notes
Cyclic Control - Trim Release (HOLD)	Recommended. This button will keep the trim release open until released, allowing for precise manual flight
Cyclic Control - Trim Release (Latch: Open)	This will set trim release to the open (pushed by pilot) state
Cyclic Control - Trim Release (Latch: Closed)	This will restore trim release to the closed (not pushed) state
Cyclic Control - Trim Release (Latch: Toggle)	Clicking this button will toggle the trim release state

There is also a lesser-used Collective Trim Release. This command is only needed when the AFCS has a collective mode engaged (like ALT, VS, CR.HT, V.APP etc.). Often it is easier to simply issue AP/UM OFF and cancel the mode instead.

Binding	Notes
Collective Control - Trim Release (HOLD)	Recommended. This button will keep the trim release open until released, enabling override of modes engaged on the collective axis
Collective Control - Trim Release (Latch: Open)	This will set trim release to the open (pushed by pilot) state
Collective Control - Trim Release (Latch: Closed)	This will restore trim release to the closed (not pushed) state
Collective Control - Trim Release (Latch: Toggle)	Clicking this button will toggle the trim release state

Example Control Profile



	H145 Function	Notes
1	AP/GTC	Press once: Engage GTC mode (ground-speed hold), Press twice: Engage GTC.H mode (auto hover)
2	UP: AP/BKUP ON, DOWN: AP/UM OFF	Engage Autopilot, Disengage Upper Modes
3	AP/BKUP CUT	Press once: Disengage AP1 and AP2 system Press twice: Disengage Backup SAS
4	CYCLIC TRIM RELEASE	Press and hold whenever making manual aircraft inputs.
5	RESET	Acknowledge any new messages in the Message List.
6	CYCLIC BEEP TRIM (UP, LEFT, DOWN, RIGHT)	4-way cyclic beep trim. Issue inputs in the respective axis (up and down is pitch, left and right is roll) in AFCS modes and manual trim with the AFCS off.
7	COLLECTIVE BEEP TRIM (LEFT, RIGHT)	4-way collective beep trim. Issue inputs in the respective axis (up and down is collective, left and right is yaw)
8	COLLECTIVE BEEP TRIM	(UP) See above.
9	COLLECTIVE BEEP TRIM	(DOWN) See Above.

Settings for Xbox controller

The default Xbox controller profile will work with H145.

These settings will better accommodate the small throw of the Xbox controller joystick:

H145 Tablet:

- `Cyclic Control` : Centering-Springs
- `Follow-Up Trim` : Both
- `Collective Step Size (Keyboard)` : Large
- Pedal trim system: Software (default)
- FEED On detection strategy: Deadzone (default)

MSFS Piloting Assistance:

- `Tail Rotor` : ON
- `Cyclic` : OFF

Recommended MSFS Settings

Flight Simulator 2020

•General Options – Camera

CAMERA SHAKE: OFF

Camera Shake causes some problems with the helicopter flight models.

•General Options – Data

ONLINE FUNCTIONALITY: ON

Online functionality is required for H145 to activate, as well as for a number of H145 features like online maps and weather.

•General Options – Graphics

Glass Cockpit Refresh Rate: High

Assistance Options - Piloting

AUTO-RUDDER: OFF

ASSISTED YOKE: OFF

ASSISTED LANDING: OFF

ASSISTED TAKEOFF: OFF

AI ANTI-STALL PROTECTION: OFF

AI AUTO-TRIM: OFF

ASSISTED CONTROLLER SENSITIVITY: OFF

Fixed-wing pilot assistance settings cause control problems for helicopter flight models. Unexpected behaviors are very often caused by these settings and it is very important that they are disabled.

Flight Simulator 2024

General Options – Camera

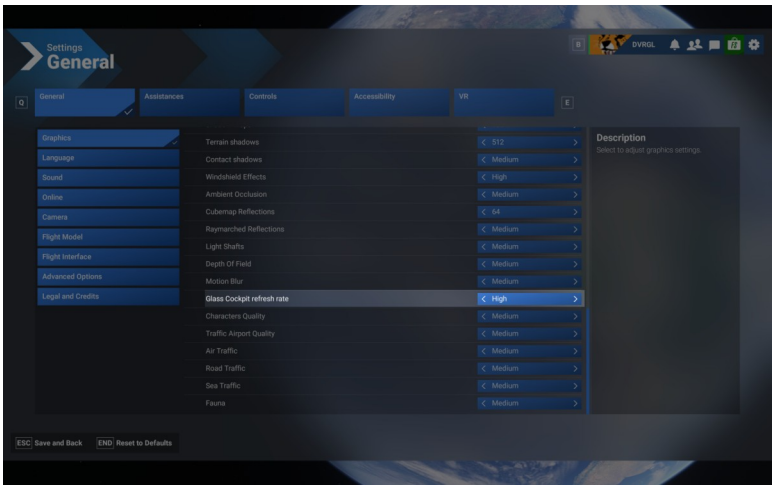
CAMERA SHAKE: OFF

Camera Shake causes some problems with the helicopter flight models.

General Options – Graphics

Glass Cockpit Refresh Rate: High

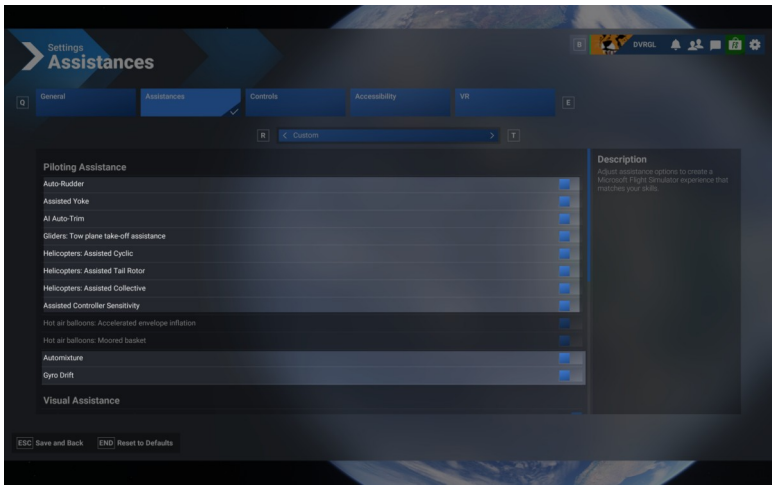
If you fly VR, don't forget to set this in VR as well.



Assistances – Piloting

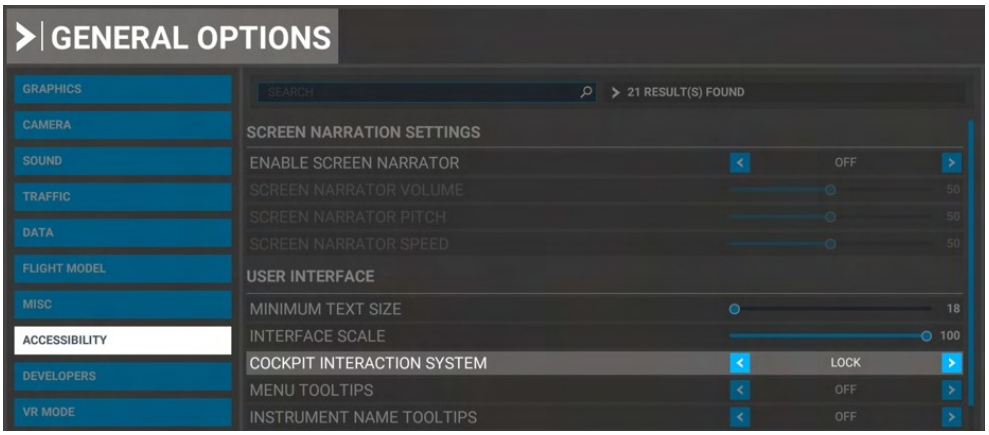
Auto-Rudder:	OFF
Assisted Yoke:	OFF
AI Auto-Trim:	OFF
Helicopters Assisted Cyclic	OFF
Helicopters Assisted Tail Rotor	OFF
Helicopters Assisted Collective	OFF
Assisted Controller Sensitivity	OFF
Automixture	OFF
Gyro Drift	OFF
Gyro Drift	OFF
Gyro Drift (realistic)	ON

(Reason being, these are across the board settings and any physical (spinning gyro) will, in fact, drift a bit and must be accounted for by the pilot.)



MSFS Cockpit Interaction System

Microsoft Flight Simulator has two modes of operating interactive elements within the virtual cockpit. These modes are controlled by the Cockpit Interaction System setting, under the Accessibility category of General Options. You may be using either setting depending on when you first installed the game.



Lock: Xbox-preferred interaction method, new in Sim Update 5. This is the default setting for new players. This mode uses a blue and yellow visual highlight on most (but not all) interactive elements.
Legacy: Classic input method used by FSX and Prepar3D. Moving the mouse over an interactive element will show a cursor but will not change their visual appearance of that element.

Operating knobs with a center push function

Lock:

1. Target a knob and press-and-hold Left mouse button
2. Click the Right mouse button
3. Release the Left mouse button

Legacy:

1. Target the center of the knob (not the left/right or top/bottom sides)
2. Click the Left mouse button

H145 in Multiplayer

Seeing other helicopters
In order to see another H145 in MSFS multiplayer you will need to ensure that both players:
1. Have the same aircraft installed
2. Have the same livery installed
3. Configured the MSFS setting Use Generic Plane Models to OFF
If these three conditions are not met, you will see a hovering fixed-wing plane in place of a player who is actually flying a helicopter

Limitations
Without helicopter support in MSFS the H145 has many custom systems and implementations which are not normal for other aircraft.
The lack of door and helicopter variables means that you will see other players showing the same as your aircraft configuration. This includes doors, the rotor spinning state, WSPS, radome and other external part configurations. Later we may be able to remove these restrictions.

Weapons in multiplayer

Other players will not see your H145M weapon launches, they exist only in your local simulator. Weapons will also not be able to shoot down AI or multiplayer aircraft at this time.

Aircraft Settings

Aircraft settings are generally controlled on the H145 tablet, inside the Aircraft app.

H145 Tablet Setup

Settings have changed starting in Build 450. Skip to the section below.
The tablet has a new set of settings for the native flight model. Open the tablet -> `Aircraft` (app) -> `Setup` (page).

Setting	Description

Aircraft Damage
 	
`On`
or
`Off`
	This setting controls the built-in damage model which impacts the aircraft engines, systems and flight model.

Gameplay Mode
 	
`Realistic`
or
`Arcade`
	The `Arcade` mode will not allow over-torque on the collective, the FLI limit will be automatic.

Vortex Ring State
 	
`On`
or
`Off`
	The `On` mode will enable a realistic VRS simulation which will cause loss of rotor lift.

Cyclic Control Settings

Setting	Description

Cyclic Control
 	
`No-Springs`
or
`Centering-Springs`
	The `Centering-Springs` setting is intended for joysticks with a centering spring. The aircraft trim position will be approximated so the joystick is always centered to maintain the current flight condition.
	
	The `No-Springs` setting requires you to deflect the cyclic as the aircraft moves into cruise, or you may take advantage of the center-displacement binding.

SAS Stability Level `-100 to +100`	This slider controls the relative stability of the SAS. If you have a full cyclic with a long extension, you may prefer values at approximately -75. If you have a joystick, the default value of 0, and for game controllers a value of +100 may be preferred. This setting is mainly to compensate for different controllers. Avoid settings lower than -80 as they are unrealistically destabilized.
Deadzone `1 to 100`	This slider controls the size of the deadzone which is used to determine `HANDS ON` detection. This setting should be as low as possible such that when releasing the controller, `HANDS ON` is reliably **not** displayed.

Follow-Up Trim
 	
`Off`
`Only Hover`
`Only Cruise`
`Both`
	Follow-Up trim will enable you to avoid using trim release for some cyclic manipulations. Hover domain is defined as less than 30 knots airspeed. Trim Release is preferred to depending on follow-up trim.

AFCS Override
 	
`Dual Input`
`Autopilot only`
	Changing to `Autopilot only` will prevent any input from the cyclic when in ATT mode or any upper mode. This setting is intended for users that lack a friction lock and also lack a spring on their cyclic. This setting allows use of the AFCS by using beep trim and Trim Release.

Center Displace Reset Time	This setting applies only to `H:H145_SDK_CYCLIC_FORCE_TRIM_DISPLACE_CENTER Cyclic Control - Displace Cyclic Center (Force Trim)` and controls how long to wait before reconnecting the user flight controls.

Tail Rotor Control Settings

Setting	Description

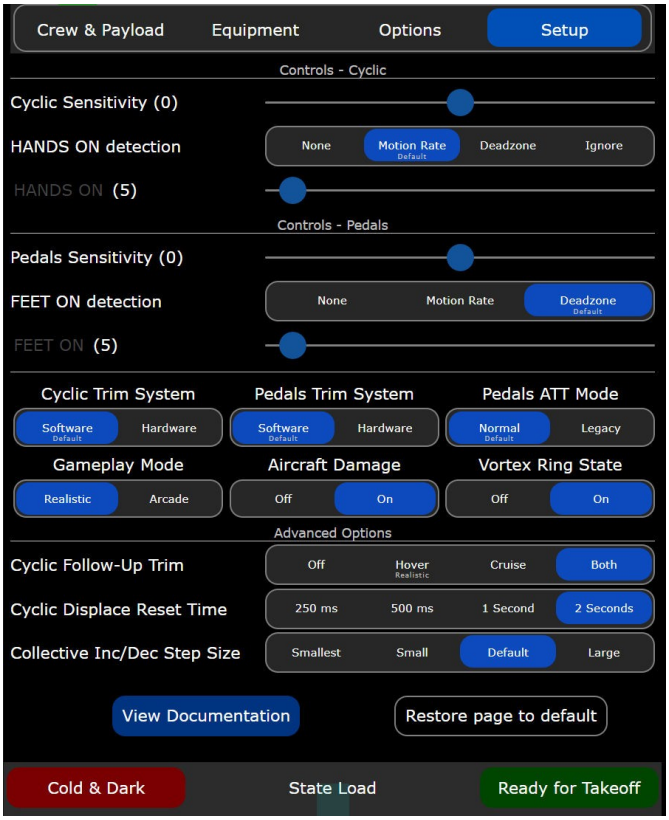
Tail Rotor Control
 	
`Centering-Springs`
or
`Legacy`
	The `Centering-Springs` setting is designed for pedals with and without springs, and enables yaw stabilized when in hover
	
	The `Legacy` setting has free yaw pedals, which is not as realistic.

SAS Stability Level `-100 to +100`	This slider controls the relative stability of the yaw SAS. It will be easier to hold a heading in hover with a higher value. This setting is mainly to compensate for different controllers. Avoid settings lower than -80 as they are unrealistically destabilized.
Deadzone `1 to 100`	This slider controls the size of the deadzone which is used to determine `FEET ON` detection. This setting should be as low as possible such that when releasing the controller, `FEET ON` is reliably **not** displayed.

Collective Control Settings

Setting	Description
SAS Stability Level `-100 to +100`	This slider controls the relative stability of the collective SAS. It will be easier to hold height over the ground in hover with a higher value. This setting is mainly to compensate for different controllers. Avoid settings lower than -80 as they are unrealistically destabilized.
Step Size (Keyboard)	This setting applies only to button/keyboard collective inputs. It will control how large the response of the collective for each button press.

Setup (V2)



The below settings apply starting with build 450.

Please note that there are no general and optimal settings! In order to fly realistically, you would also need force feedback devices for all three inputs (cyclic, collective and pedals). The settings here should help you to find a good solution for your equipment and your flying experience and requirements. Invest a little time here, it's worth it.

Cyclic Sensitivity

This setting is essentially the same as the MSFS Controls Sensitivity setting

HANDS ON detection

This setting controls the `HANDS ON` detection strategy. The slider below corresponds to the threshold relative to the selected strategy.

Strategy	Description
None	Used for Force-Feedback controllers. In this mode there is no `HANDS ON` detection unless `L:FFB_HANDS_ON_CYCLIC`=1.
Motion Rate	Used for any controller. Detection is based on rate of motion. The slider will adjust the relative motion rate which triggers detection.
Deadzone	Used for any controller. Detection is based on the deflection of the stick exceeding a specific threshold, which is set by the slider below.
Ignore	Used for controllers without sufficient ability to hold a position. In this mode the cyclic is

	completely ignored when using AFCS upper modes. Recommended for sticks without feathers.

Pedals Sensitivity

This setting is essentially the same as the MSFS Controls Sensitivity setting.

FEET ON detection

This setting controls the `FEET ON` detection strategy. The slider below corresponds to the threshold relative to the selected strategy.

Strategy	Description
None	Used for Force-Feedback controllers. In this mode there is no `FEET ON` detection unless `L:FFB_FEET_ON_PEDALS`=1.
Motion Rate	Used for any controller. Detection is based on rate of motion. The slider will adjust the relative motion rate which triggers detection.
Deadzone	Used for any controller. Detection is based on the deflection of the stick exceeding a specific threshold, which is set by the slider below.

Cyclic Trim System

This setting controls whether there is a virtual trim (good for controllers with springs) or a hardware trim (good for Force-feedback controllers).

Strategy	Description
Software	The virtual trim point is visible on the VMS CTRL page as a blue cursor. It will move slowly to follow the stick and relax any forces.
Hardware	The virtual trim is disabled, Force-Feedback controllers are likely in use.

Pedals Trim System

This setting controls whether there is a virtual trim (good for controllers with springs) or a hardware trim (good for Force-feedback controllers).

Strategy	Description
Software	The virtual trim point is visible on the VMS CTRL page as a blue cursor. It will move slowly to follow the pedals and relax any forces.
Hardware	The virtual trim is disabled, Force-Feedback controllers are likely in use.

Pedals ATT Mode

This setting is provided for compatibility with previous usage.

Strategy	Description
Normal	ATT mode will stabilize and hold heading in hover.
Legacy	ATT mode will not stabilize and hold heading in hover.

Gameplay Mode

This setting provides an easier flight model for users with less comprehensive hardware.

Strategy	Description
Realistic	Realistic SAS implementation.
Arcade	Heavy-handed SAS implementation.

Aircraft Damage

This setting configures the wear and damage model for the aircraft.

Strategy	Description
Off	No damage. Check that your MSFS damage is also turned off.
On	Engine damage, IBF clogging.

Vortex Ring State

This setting configures the VRS model on top of the MSFS flight dynamics. VRS is a dangerous condition where lift is lost.

Strategy	Description
Off	VRS model is not active.
On	VRS is active, avoid descent below 500fpm when below 20kias.

Cyclic Follow-Up Trim

This setting configures the Follow-Up Trim behavior.

Strategy	Description
Off	Trim Release must be used to update AFCS attitude setpoints.
Hover	Hands On detection in hover domain will result in updating AFCS attitude setpoints.
Cruise	Hands On detection in cruise domain will result in updating AFCS attitude setpoints.
Both	Hands On detection in both hover and cruise domain will result in updating AFCS attitude setpoints.

Cyclic Displace Center-Reset-Time

This setting configures how long the cyclic is disabled after the Displace-Center binding function is used.

Collective Inc/Dec Step Size

This setting configures how much collective is added or remove when clicking increment/decrement buttons. This does not apply when using an axis binding.

View Documentation

Opens a Web-Browser and load the H145-Handbook. Be carefull if in VR.

Restore page to default

Resets the settings to default values.

Cold & Dark

This switch puts the airplane in a switched-off state. It is used, for example, when taking off from a heliport or runway as all systems are always automatically started up here.

Ready for Takeoff

This switch makes the aircraft ready for take-off when the button is pressed.

Checklists

General Limitations

The H145 shall be operated in compliance with the limitations in this section. This helicopter is approved for flight under VFR and IFR flight rules in addition to overwater operation.

The following are prohibited:

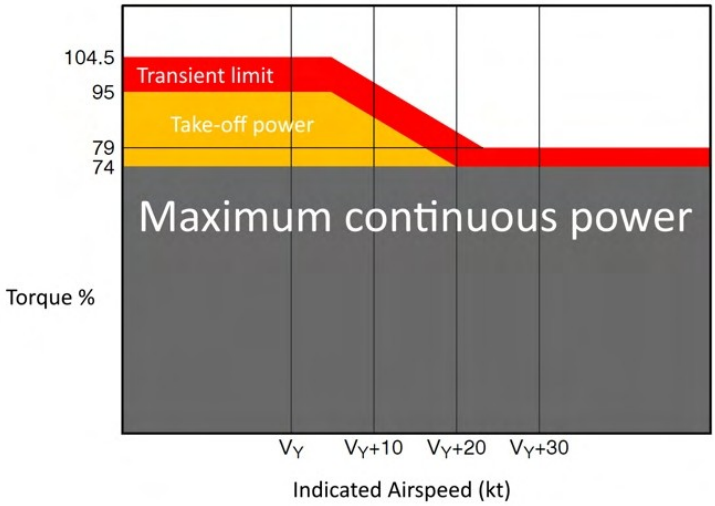
- Aerobatic maneuvers
- Flight into icing conditions. Should icing conditions be encountered unexpectedly, the conditions shall be left immediately

Limitation	
Maximum speed for starting and stopping the rotor	50kt
Maximum relative wind speed from any direction (except headwind)	30kt
Sloped landing limitation	Sloping nose down: 10° Sloping up to the right: 12° Sloping up to the left: 8° Sloping nose up: 8°
Maximum operating altitude	20,000FT PA
Maximum operating altitude for takeoff, hover and landing	20,000FT PA or DA (whichever is less)
Maximum air temperature Minimum air temperature **Or with cold weather kit:** Maximum air temperature Minimum air temperature	ISA +35C (max +50C) -30C +35C -45C
Ground operations duration when > 40C OAT	Ground operations limited to 20 minutes **NOTE:** When >35C OAT, lower cockpit temp by using max ventilation
Maximum gross mass for flight Minimum gross mass for flight	3700kg 2000kg
V[NE]	150kt or less (see Airspeed indicator)

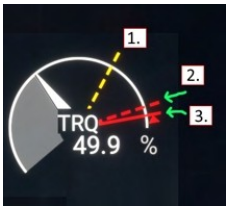
Torque limitations

	AEO Limitation	**Torque**
1	Max continuous power (no limit)	2x74%
2	Take-off power (30mins)	2x95% below Vy+5
3	Transient limit (unintended use)	2x104.5 below Vy+5

Above Vy+5kt, take-off power will gradually disappear. The limits on the engine indications and the FLI will move according to the chart:

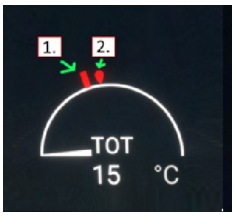


	OEI Limitation	Torque
1	Max continuous power (no limit)	1x100%
2	2-minute power	1x143%
3	30-second power	1x150%

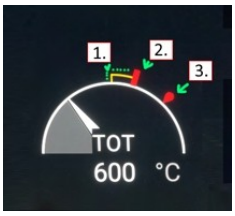


TOT limitations

	Starting Limitation	TOT
1	Continuous starting	840C
2	Transient starting (max 10 seconds)	960C



	AEO Limitation	TOT
1	Max continuous	901C
2	Take-off power (max 30 minutes)	918C
3	Transient limit (unintended use)	945C



	OEI Limitation	TOT
1	Max continuous	945C
2	2-minute power	987C
3	30-second power	1006C

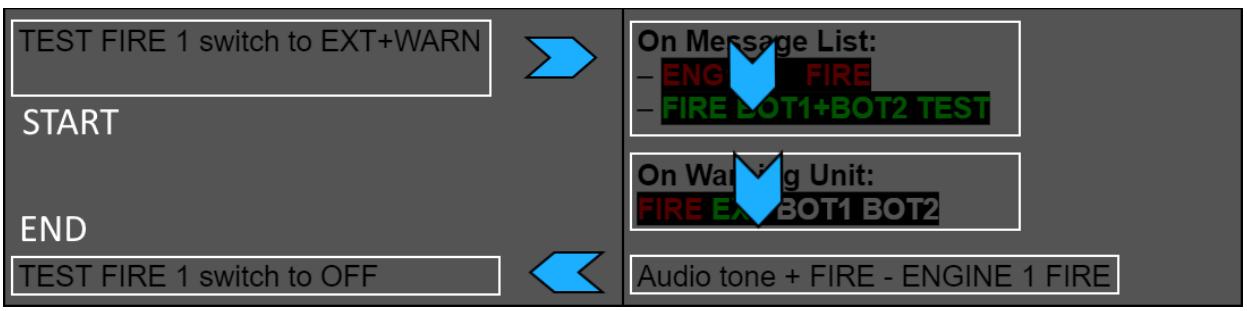
N1 limitations

	AEO Limitation	N1
1	Max continuous	98.5%
2	Take-off power (max 30 minutes)	100.6%
3	Transient limit (unintended use)	101.7%

	OEI Limitation	N1
1	Max continuous	101.7%
2	2-minute power	104.3%
3	30-second power	105.7%

Procedures

Procedures below are realistic, adapted from the BK-117 D2 (H145) flight manual and simplified where necessary. Execute each step starting at the top left, monitoring items on the right, and ending by doing cleanup tasks on the bottom left.



Procedure	Purpose	Notes
Cockpit safety inspection	Confirm cockpit switches are in a safe position before bringing power to the aircraft.	Not included, Cold & Dark startup at a gate has already completed these steps for you.
Power-Up Procedure	Bring electrical power to the aircraft and prepare to start engines.	
Engine Start Procedure	Start the first and then second engine	
Hydraulic check procedure	Verify that both SYSTEM 1 and SYSTEM 2 are functional	To be completed on the first flight of the day
After Engine Start Procedure	Prepare the aircraft for flight after engine start	
Pre-Flight Procedure	Verify the AFCS before flight.	
After Takeoff Procedure	Verification of takeoff steps and preparation to conduct a safe flight.	
After Landing Procedure	Reduce engines to idle and prepare to stop the rotor and shutdown the aircraft	
Engine Shutdown Procedure	Shutdown the engines and remove power from the aircraft.	

Power-Up Procedures

BAT MSTR switch to **ON**, then **ENGAGE**	Allow 20 seconds for system startup – PWR-UP TST OK
TEST FIRE 1 switch to EXT	– FIRE BOT1+BOT2 USED Audio tone
TEST FIRE 1 switch to EXT+WARN TEST FIRE 1 switch to OFF	On Message List: – ENG1 FIRE – FIRE BOT1+BOT2 TEST On Warning Unit: FIRE EXT BOT1 BOT2 Audio tone + FIRE - ENGINE 1 FIRE
Repeat fire test for 2 TEST switch to OFF	 Audio Test – TRAIN (Engine CP) – APCP: all lights on – EXIT lights illuminate
TEST switch to LAMP	
VMS page, press NUM	Check Battery Voltage > 23.5
VMS page, WEIGHT subformat	Adjust Crew, Payload, and Fuel as required. Use the PUSH knob to advance the line
VMS page, press PREV	
FND page	On Message List: – ENG 1 FAIL ENG2 – AFT+FWD FUEL PMP – PITOT1 HEATER OFF PITOT2 – PWR-UP TST OK
Flight instruments, IESI	Wait for alignment, set baro as required
EXT LIGHTS ACOL	Set as required

Engine Start Procedure

FUEL PRIME PUMP 1 switch to ON FUEL PRIME PUMP 2 switch to ON	– **FUEL 1 PRIME PUMP ON FUEL 2**
ENG 1 MAIN switch to IDLE	N1: Monitor, increasing TOT, N2, NR: Monitor, increasing MGB oil pressure: increasing Hydraulic pressure: increasing – **START-UP TEST** At 60% N1, **STARTER** extinguishes. At 78% N1, **IDLE** appears.
Repeat for second engine	– **START-UP TEST OK**

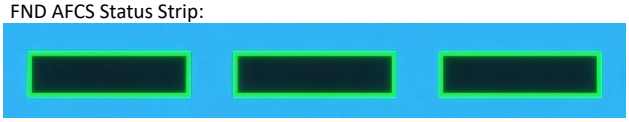
Hydraulic Check Procedure

TEST HYD switch to SYS1	– **LOW PRESS HYD2** Check VMS page HYD2 pressure low caution Check cyclic/collective/pedals normal forces
TEST HYD switch to SYS2 TEST HYD switch to OFF	– **HYD1 LOW PRESS** – HYD1 TR SHUT OFF Check VMS page HYD1 pressure low caution Check cyclic/collective/pedals normal forces

After Engine Start Procedure

PT/STATIC HTG PILOT switch to ON PT/STATIC HTG COPILOT switch to ON AVIO MSTR 1 switch to ON AVIO MSTR 2 switch to ON STBY BAT switch to ON LAVCS switch to PIL	
FUEL PRIME PUMP 1 switch to OFF FUEL PRIME PUMP 2 switch to OFF FUEL XFER PUMP F switch to ON FUEL XFER PUMP A switch to ON	
EXT LIGHTS **POS** and **STROBE**	As required
INT LIGHTS EM/EX switch to ARM	
Cockpit/Pax Ventilation	As required

Pre-Flight Procedure

Collective lever to FULL DOWN TEST switch to LAMP	
TEST switch to PRE-FLIGHT TEST switch to OFF	– **P-FLT TEST** Wait for the test to complete – **P-FLT TEST OK**
AP/BKUP ON - UP button PRESS	FND AFCS Status Strip:  (boxes will disappear after a few seconds)
AP/BKUP CUT button PRESS	**SAS SAS AFCS DISENGAGED**
AP/BKUP CUT button PRESS AP/BKUP ON - UP button PRESS	**SAS SAS AFCS DISENGAGED**
BEEP TRIM	Check all 4 directions
FIXED LANDING and S/L LIGHTS	As required
EMER FLOATS	As required
ENG 1+2 MAIN switch to FLIGHT (and latch)	

After Takeoff Procedure

Perform Hover Flight	N2, NR > 101% FLI: AEO Indication No warnings on message list

Before Landing Procedure

Fixed LANDING and S/L Lights	As required
FND & VMS pages	No messages on master list ENG 1 & ENG 2 N2 > 101% NR > 101%
DH (Decision Height)	As required

Engine Shutdown Procedure

Cyclic to neutral position Collective to FULL DOWN ENG 1 MAIN switch to IDLE ENG 2 MAIN switch to IDLE	Wait 30 seconds for engine cool down
FUEL XFER PUMP F switch to OFF FUEL XFER PUMP A switch to OFF PT/STATIC HTG PILOT switch to OFF PT/STATIC HTG COPILOT switch to OFF AVIO MSTR 1 switch to OFF AVIO MSTR 2 switch to OFF STBY BAT switch to OFF LANDING LIGHT switch to OFF S/L LIGHT switch to OFF Other electrical consumers to OFF	Check electrical load on VMS page
ENG 1 MAIN switch to OFF ENG 2 MAIN switch to OFF	– ENG 1 FAIL ENG2 TOT: Monitor decrease N1: Monitor decrease Wait for rotor to stop
EXT LIGHTS ACOL switch to OFF	Check VMS FLIGHT REPORT – DOWNLOAD IN PROGRESS Download takes approx. 1 minute – DOWNLOAD COMPLETE
BAT MSTR switch to OFF	

Abnormal Procedures (Checklists)

This Section Is Unfinished

Procedure	Condition	Notes
Warning Indications: FIRE	Engine fire or overtempture in the engine compartment	

Warning Indications: FIRE

Conditions: Overtemperature in engine compartment	**On Warning Unit:** FIRE and Audio tone or On Message List: – ENGi FIRE and voice message FIRE - ENGINE i FIRE
FIRE on Warning Unit PRESS	**On Warning Unit:** ACTIVE BOT1
BOT1 on Warning Unit PRESS	After 15 seconds bottle 2 will become active. **On Warning Unit:** BOT2 After 1 minute if the fire is not extinguished, proceed to discharge bottle 2.
BOT2 on Warning Unit PRESS	
	On Message List: FIRE BOT1+BOT2 USED

Systems

Cockpit Arrangement

1. Warning Unit
2. Co-pilot MFD (MFD1)
3. Standby instruments (IESI)
4. Center MFD (MFD4)
5. Pilot MFD (MFD2)
6. Tablet Hinge
7. Engine control panel (ECP)
8. Co-pilot GTN750
9. Pilot GTN750
10. Co-pilot control panel (CCP)
11. Auto-pilot control panel (APCP)
12. Weather Radar control panel (WXRCP)
13. TFM-138B Radio
14. Ground Power control panel (GPCP)
15. Cabin air control panel
16. Data transfer device

Master Warning Unit

The Warning Unit panel displays critical conditions which require immediate pilot action. The Engine shutoff and fire extinguishing controls are also integrated into the panel.

ACTIVE	Emergency Fuel Shut-Off valve is ACTIVE
LOW FUEL 1/2	Fuel in respective supply tank is below 40%
ENG 1/2 FAIL	Engine failure
ROTOR RPM	Rotor RPM is above 109% or below 97%
BAT OVHT	Battery OverheatMain battery over 50C
MGB OIL P	Main Gearbox oil pressure out of limits
AP	Autopilot failure
CARGO SMOKE	Smoke is detected in cargo compartment

EXT	Fire extinguishing system is active (due to emergency fuel shutoff being activated)
BOT1 BOT2	Press to discharge the respective fire bottle. (Bottles 1 and 2 are shared between both engines, and available to discharge on either side)
FIRE	FIRE **indication** (engine fire is detected) FIRE **push button** (lift guard first) activates the Emergency Fuel Shutoff for the respective engine. See the engine fire procedures further in this document

Engine Control Panel

The Engine Control Panel (ECP) is used to start and stop the engines as well as operate additional functions for emergency or abnormal procedures. The Training button between the engines is a function used to simulate One-Engine-Inoperative flight condition.

1. Engine 1 Main switch (with latch)
2. Engine 1 Ventilation Button (inoperative)
3. Engine 1 FADEC Emergency switch (with guard)
4. OEI Training mode (inoperative)

The **Engine Main** switch may be placed into 3 detents: OFF, IDLE and FLIGHT. The IDLE detent is used for starting the engine and for cooldown after flight, it will not raise the rotor RPM to 100%. The FLIGHT detent is to be selected prior to takeoff and until after landing. The OFF detent will close the fuel valve and trigger an engine shutdown. When in the FLIGHT position, the engine main switches should be latched.

The **FADEC EMER** switch is to be used in case of the **FADEC FAIL** indication on the message list. When the FADEC EMER switch is placed to ON it will activate a backup mode which will meter the fuel valve of the engine of the failed side to match the TOT of the working side.
NOTE: FADEC EMER will reduce the rate at which N1 changes. Expect this and avoid large power demand changes.

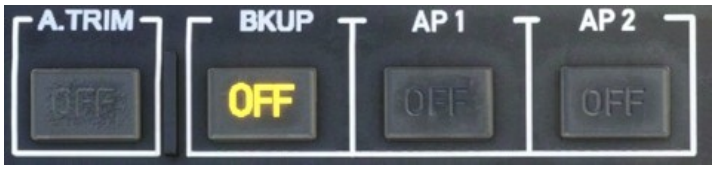
The **Engine Ventilation** button (**inoperative**) is to be used when directed after a failed engine start. The starter motor will run without opening the fuel valve in order to clear the engine for possible restart.

The **Training** button (**inoperative**) simulates One-Engine-Inoperative (OEI) flight condition while retaining the safety margin of the working engine. (For simulation use you may safely take one engine to the IDLE position for a similar effect)

Autopilot Control Panel

The Autopilot Control Panel (APCP) is used to control the autopilot systems. It is similar in nature to a Mode Control Panel from large airliners.

The APCP is logically structured such that major systems are shown along the top row. Each is a push button which toggles the system on or off, as well as an illuminated **OFF** status. When **OFF** is displayed, the system may have been switched off by the pilot or the system may be inoperative due to a failure or lack of electrical power.



A.TRIM or AUTO TRIM is a system which controls the trim and feedback forces of the cyclic control. It allows the pilot to fly the helicopter hands-off and to intervene by using a CYCLIC TRIM RELEASE button or by manually pushing on the springs of the trim system, which either pauses or provides follow-up movement of the trim.

BKUP or BACKUP SAS is an independent system which provides only 3-axis basic stabilization. To remain on always, but only used in the event of AP1 and AP2 failure.

AP1 and **AP2** are redundant autopilot systems, implemented in the aircraft management computers. These systems provide both basic stabilization (SAS) as well as UPPER MODES like HDG, NAV, ALT etc. When both systems are enabled, one will act as the primary and the other will operate in a standby mode, ready to take over if the primary system should fail. AP1 and AP2 require **AVIONICS (AVIO 1 or AVIO 2)** to be switched to ON.

Upper Modes

Each of the upper modes knobs has a push function which toggles the mode, as well as left and right turn which changes the bug or setpoint for that mode.

The VS/HDG butterfly knob switches between traditional (HDG and VS) and GPS (TRK and FPA) modes. When switched to GPS the VS mode will become FPA and the HDG mode will become TRK.

Collective modes below (CRHT, IAS, ALT.A, VS/FPA) will have the 4-axis autopilot use the collective control, so engagement will prevent you controlling your collective axis, unless you hold COLLECTIVE TRIM RELEASE.

Collective Modes

CRHT or Cruise Height works like an altitude hold, but uses the radio altimeter as the reference. This will result in a rough ride but enables the helicopter to traverse hilly terrain. This feature is designed for use over water.

IAS or Indicated Airspeed works the same as a fixed-wing airspeed hold.

ALT.A or Altitude Acquire. When you turn the knob the bug will move, but your aircraft will not change from the current mode (even if **ALT** is selected).This mode works as a way to allow a preselection for a new altitude, and then it will use the **VS** mode to move from your current altitude to the new altitude. Once at the new altitude, **ALT** mode will be automatically engaged and the new altitude will be held.

VS/FPA or Vertical Speed/Flight-Path-Angle works the same as a fixed-wing vertical speed hold. When in FPA mode the aircraft speed is taken into account, allowing a descent angle to be defined. Useful with the FPV (Flight Path Vector) available on the SVS (Synthetic vision) view.

ALT or Altitude Hold works the same as fixed-wing altitude hold mode. It uses the barometric altitude and maintains it. Note that the only way to change the setpoint for ALT mode is by using COLLECTIVE BEEP TRIM or by switching to ALT.A mode.

Roll/Yaw Modes

HDG/TRK or Heading/Track Hold works the same as fixed-wing heading hold, however it will use roll to accomplish the task at higher speeds, and yaw while in hover domain (under 30kt).
Track will consider the current wind and pick a heading that allows a straight line to be flown despite a crosswind.

Modes not on the APCP

NAV or Navigation mode (as well as **APP** and **V.APP**) is engaged by selecting a navigation source on the pilot MFD (use the NAV softkey on the MFD to pick between GPS, NAV1 and NAV2. The CPL softkey to couple the source to the AP)

GTC or Ground Trajectory Command mode is engaged by pressing AP/GTC binding or the Tablet autopilot panel.

GTC.H or Ground Trajectory Command in Hover submode is engaged by double-clicking

AP/GTC binding or the Tablet autopilot panel.

ATT mode is automatically engaged in the absence of an upper mode on the axis.

GA or Go-Around mode is engaged when pressing the COLLECTIVE GA control

Beep Trim

When any of the above modes are active, you will find that context-sensitive “beep trim” control is available. You can assign these to a hat, or click the buttons on the Tablet autopilot panel. These context sensitive modes will intelligently change the correct bug for the modes that are engaged. For example if you have ALT engaged, CYCLIC BEEP TRIM UP (and down) will control the altitude bug. If you were in VS mode, CYCLIC BEEP TRIM UP (and down) will control your vertical speed bug.

AFCS (Autopilot)

See Systems - AFCS.

Helionix Avionics Suite

The Helionix system consists of 3 identical multi-function displays (MFDs), as well as two aircraft management computers and an array of sensors and hardware.

MFDs

Each MFD has 6 buttons on each side (top, right, bottom, left). When the button has a function, text will be shown on the MFD display directly adjacent to the physical button.

The buttons along the top of each MFD operate the main pages which the displaycan show.

Each MFD also has buttons to adjust the intensity of the various layers:

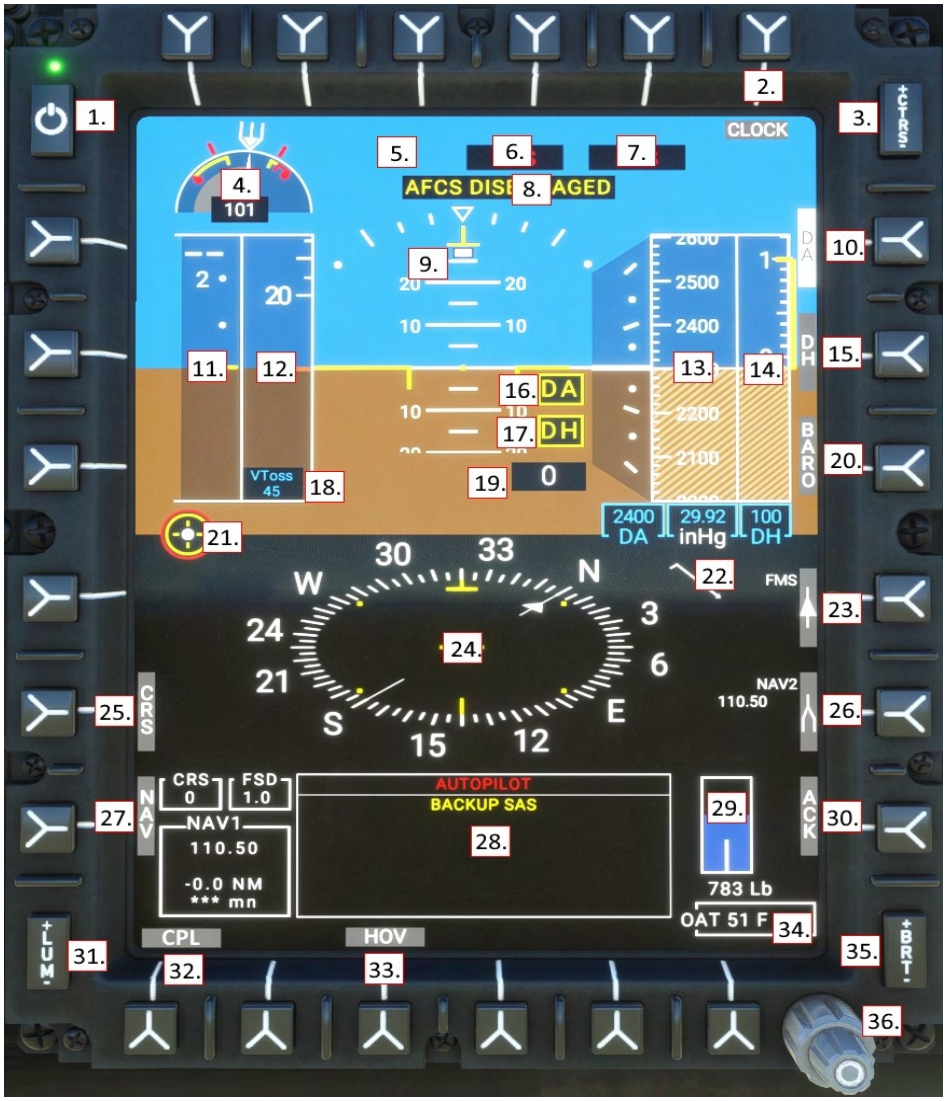
- LUM: Overall display luminance
- CTRS: Weather and Terrain overlay intensity
- BRT: SVS and DMAP underlay intensity

Page Name	Function
FND: Flight and navigation display	Top: PFD (Primary Flight display) Middle: Navigation Bottom: Aircraft (Message list, fuel)
NAVD: Navigation display	Navigation information, route display, map overlays (terrain, weather, topo)
VMS: Vehicle Management Systems	Aircraft and systems information
DMAP: Digital Map System	Map information streamed from online service
MISC: Miscellaneous	Used for display of auxiliary cameras
EFB: Electronic Flight Bag	Used for offline and online EFB functions. **Not Installed**.

Note: When using ConfigTool or H:Events the MFDs are identified by the numbers, 1, 2 and 4. The pilot MFD is 2, the center MFD is 4 and the copilot MFD is 1.

FND Page

Flight and Navigation Display



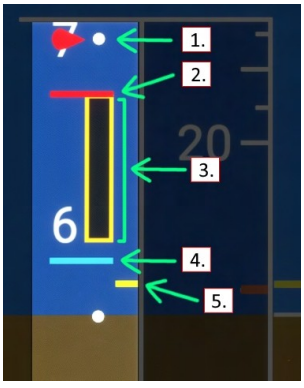
1. MFD Power
2. Stopwatch / Clock
3. CTRS – Overlay intensity (WXR,HTAWS)
4. Rotor gauge
5. AFCS Collective Axis status
6. AFCS Roll/Yaw Axis status
7. AFCS Pitch Axis status
8. AFCS AP Main status
9. Slip/Skid indicator
10. Decision Altitude (DA)
11. First Limit Indicator (FLI)
12. Airspeed indicator
13. Barometric altitude indicator
14. Radio altitude indicator
15. Decision Height (DH)
16. Below DA Alert
17. Below DH Alert
18. Takeoff Safety Speed (VTOSS)
19. Radio altitude (digital)
20. Kohlman Setting (Baro)
21. Mast Moment indicator
22. Wind indicator
23. Bearing 1 selector
24. Horizontal situation indicator
25. NAV Course selection
26. Bearing 2 selector
27. Cycle navigation source
28. Message list
29. Fuel gauge
30. Acknowledge new messages
31. LUM - Overall display brightness
32. Couple/Decouple AP nav source
33. Change views (HSI/SCT/HOV) – HSI, Sector, Hover views
34. Outside air temperature indicator
35. BRT – Underlay brightness (SVS, DMAP)
36. MFD Knob with push function

First Limit Indicator (FLI)

The FLI is presented on the FND page and communicates to the pilot the combined power-plant limits: Engine/MGB torque (TRQ), Engine N1 compressor speed (N1), and Engine turbine output temperature (TOT). The background of the FLI tape is the collective position, from 0 degrees pitch to 10 degrees.

When exceeding limits, you should expect the NR to fall (if N1 capability is exceeded) and excessive torque will cause engine wear and MGB damage. Except in failure, the FADEC will not exceed N1 or TOT limits.

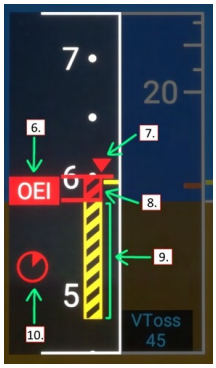
All Engines Operative (AEO)



When all engines are operative, the FLI indicates the power limits, usually based entirely on torque. The pilot should take care to avoid exceeding the transient power limit (1), use of the range between the maximum power limit (2) and transient power limit (1) is permitted only for unintended use, up to 2 seconds.The blue line (4) indicates the amount of power which will be immediately available should an engine fail. If the current
collective position is above the blue line and an engine should fail, the pilot would immediately need to lower the collective otherwise rotor RPM would begin to decay. The takeoff power limit (3) is available for 30 minutes, and then collective power should be reduced to maximum continuous power (bottom of (3)).
The power rating timer (10) will be visible 90 seconds prior to 30 minutes elapsing

1. Transient power limit (teardrop)
2. Maximum power limit
3. Takeoff power (30 minutes)
4. OEI power limit
5. Current power setting

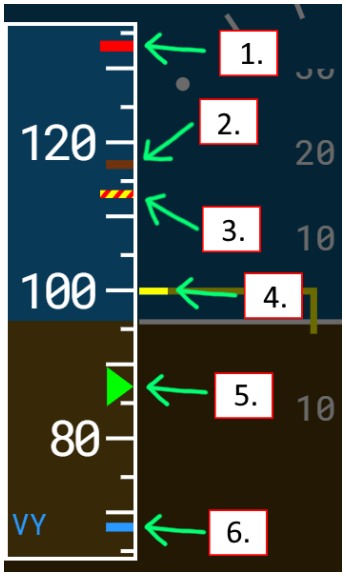
One-Engine-Inoperative (OEI)



The FLI will clearly indicate the OEI status (6) and show the power limits available. Using the OEI HI/LO button, the pilot may switch between use of the 30 second and 2 minute OEI power bands. When collective position exceeds the capability of the engine, the rotor rpm will drop.

6. OEI (One Engine Inoperative) flag
7. Topping signal (FADEC limit)
8. 30 second OEI power rating
9. 2 minute OEI power rating
10. Power rating timer

Airspeed indicator



1. VNE (velocity never-exceed speed, airframe limit). This is calculated based on your weight, outside air temp and pressure altitude. It will slide up and down based on those factors

2. GPS Ground Speed (this is intended to be shown as a digital value if it is off-scale, but I didn't do that)

3. VNE Power Off (velocity never-exceed speed while in autorotation). This is the same as above, calculated from charts and intended to be used should both engines fail

4. Current airspeed (IAS)

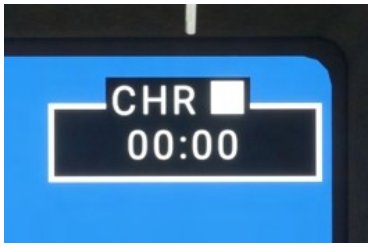
5. Speed bug

6. VY (Best climb speed)

7. Not shown: VTOSS (Takeoff Safety Speed) at 45kt. This is a minimum takeoff speed for engine failure

8. Not shown: CHK VNE (Check VNE speed). Occurs when avionics cannot calculate VNE

Stopwatch (chronometer)

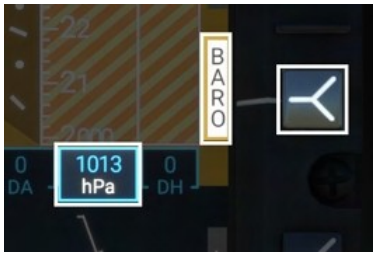


Use the CLOCK sk to cycle between local time, stopwatch and off.

In Stopwatch mode, use the MFD knob push function as follows.

- First short press starts the stopwatch
- Second short press stops the stopwatch
- Long press while stopwatch is running resets and restartsstopwatch
- Long press when stopwatch is not running resets stopwatch

Set altimeter (BARO)



There are 3 altimeters in H145 (pilot, copilot and IESI). Change the MFD altimeter setting (kohlman setting) with this procedure:

1. Select the BARO key, the BARO text will illuminate white in reverse video.

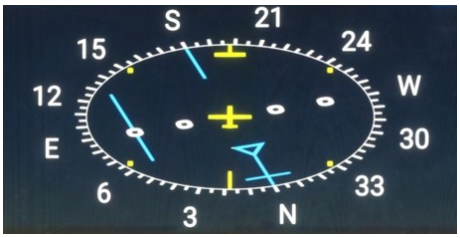
2. Turn the MFD knob to increase or decrease the current altimeter setting. Press the knob for STD/1013/29.92.

3. Select the BARO key again to exit the mode

NOTE: Change units between in/hg and hpa in the tablet aircraft app.

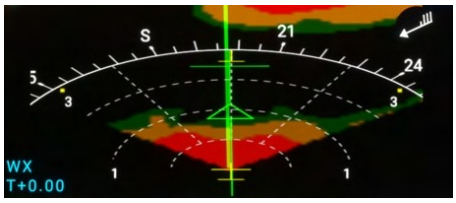
HSI View

HSI mode displays a navigation source and deviation range.



SCT (Sector) View

Sector mode displays a 60 degree situation view with the flight plan line and navigation source as well as weather radar.



HOV (Hover) View

Hover mode provides brown GPS-based ground-speed lines which can be used to hover precisely without suitable ground reference. Hover mode is also useful in GTC mode as it displays the green digits corresponding to the current trends that the GTC mode is maintaining.

SVS (Synthetic vision)

The PFD will operate in both FDS (blue sky, brown ground) and SVS mode which enables a synthetic vision background.
Note that in SVS mode the pitch scale is compressed in FDS mode and expanded to 1:1 scale in SVS mode.
SVS is available on all 3 MFDs and may be combined with all other modes.

Wind indicator

The wind indicator is shown on both the FND and NAVD pages.

Each long feather is 10kt
Each short feather is 5kt
The triangle is 50kt
The circle is wind calm.

	Wind calm
	Wind 25kt
	Wind 50kt
	Wind 75kt

NAVD Page

Navigation Display

1. NAV sk - change nav source (NAV1, NAV2, GPS/FMS) 2. Next waypoint information 3. Ground Speed and True Airspeed 4. MAP sk - toggles the hill shading layer 5. RNG sk - map range (use the knob to control range) 6. Bearing 1 sk - Cycle bearing 1 (NAV1, GPS, hidden) 7. Bearing 2 sk - Cycle bearing 2 (NAV2, GPS, hidden) 8. FMS destination time and fuel estimation	9. Bearing frequency, distance and identifier 10. FMS waypoint and destination information 11. DATA sk - Show information on all nav sources 12. CPL/DCPL couples the AFCS to the selected navigation source 13. SCT/ROS/PLN sk - ROSE or SECTOR or PLAN view 14. WXR sk - Toggle weather overlay. (FAIL and STBY indicate WXR panel switch position) 15. HTAWS sk - Toggle terrain aware overlay (green/red

NAVD Page (DATA subformat)

The DATA select key will reveal information regarding all installed navigation sources.

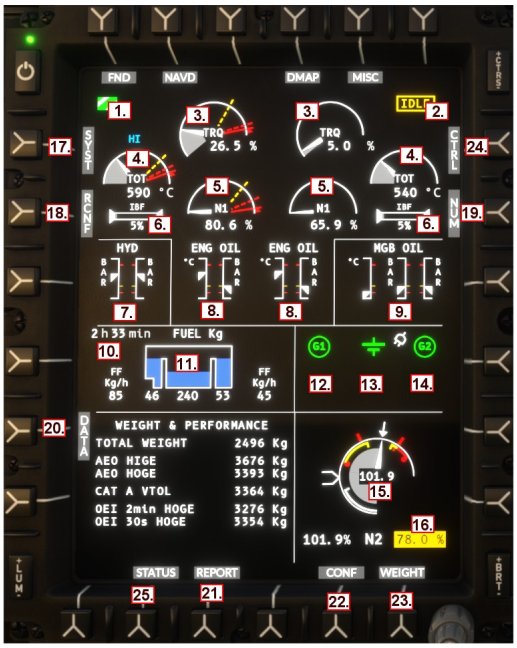
NAVD Page (Sector view)

Sector view provides a 60 degree situation view in front of the aircraft.

The other views are
ROS: (Rose) - A 360 degree situation view with the aircraft heading oriented upward.
PLN: (Plan) - A 360 degree situation view with North oriented upward

VMS Page

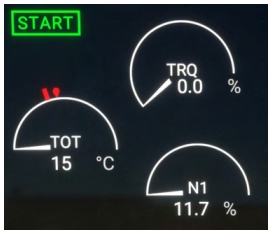
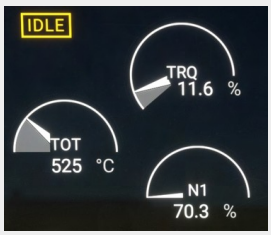
Vehicle Management Systems

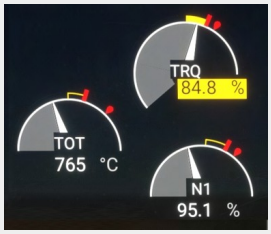
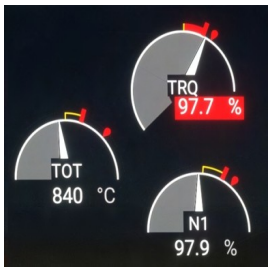
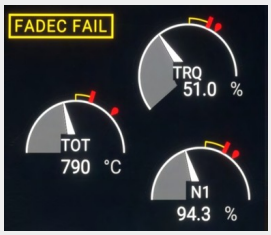
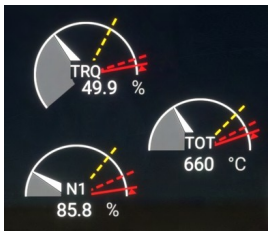
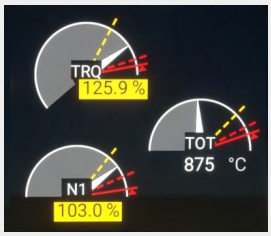
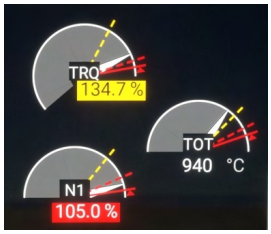


1. Engine low pressure bleed valve status 2. Engine main status (IDLE, START, FAIL) 3. Engine Torque 4. Engine exhaust temperature 5. Engine N1 compressor 6. Inlet Barrier Filter status 7. Hydraulic System 1 and 2 pressure 8. Engine oil temperature and pressure 9. Main Gearbox oil pressure (system 1 and 2) and oil temperature 10. Endurance (based on present fuel flow) 11. Fuel (center feeds into left and right supply tanks) 12. Engine generator 1 status 13. Main battery status	14. Engine generator 2 status 15. Rotor rpm gauge (and N2 for engine 1 and 2) 16. N2 for engines 1 and 2 17. SYST sk - System page 18. RCNF sk - Reconfigure page 19. NUM sk - reveal nominal digital values 20. DATA sk - Switch between timezone and performance, hoist, and cargo hook subformats 21. REPORT sk - Switch to flight report subformat 22. CONF sk - Switch to aircraft config subformat 23. WEIGHT sk - Switch to weight subformat 24. CTRL page – Switch to Flight Control page 25. STATUS page – Switch to System Status subformat

The top section of the VMS page displays the main format, and the bottom section displays the subformat.

VMS Page - Engine indications

Starting engine (starting limits)	TOT limits: Max: 760C Transient: 840C Transient starting TOT is allowable for up to 10 seconds.	
Engine idle (no limits)	IDLE detent is to be used for starting, ground operations, and cool-down after flight. No limits are applied when the engine is in IDLE.	
Two engines in FLIGHT (AEO limits)	Note the presence of the amber takeoff power band on all three Torque, TOT and N1. **NOTE:** Limits are computed and will vary based on pressure altitude and outside air temperature.	

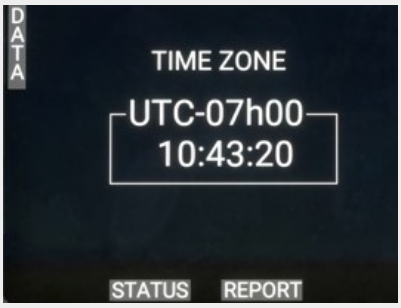
Takeoff power (TOP) (AEO limits)	The amber band is takeoff power and is available for 30 total minutes per flight. A white timer will be displayed when 90 seconds are remaining. After 30 minutes power should be reduced to maximum continuous.	
Transient exceedance (AEO limits)	Transient power is available for **unintended** use for up to 12 seconds. A gong sound will play at the beginning of each exceedance.	
FADEC Failure (no limits)	Level 3 FADEC failure -Fuel valve is frozen and the engine cannot respond to commands to change the N1. Use FADEC EMER to attempt recovery to level 2 FADEC failure.	
One engine in flight (OEI limits)	OEI (one engine operative) limits are displayed as lines. Note that the OEI limits are much higher than AEO limits.	
OEI - 2 minute rating	The amber band is the 2-minute power rating when only one engine is operative.	
OEI - 30 second rating	The red power band is the 30 second power rating. Note the red topping triangle (OEI HI and OEI LO) setting controls whether the FADEC will allow use of the 30 second rating or instead droop the NR when commanding more collective power.	

VMS Main Page (MAIN subformat)

The main subformat contains two panels. The right panel is used for NR and N2 indications, and the left panel has a page controlled by the DATA select key.

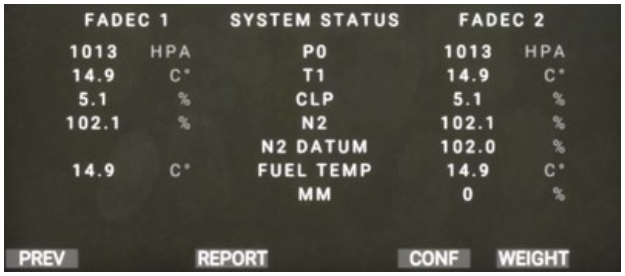
NR is shown for both engines in percent, managed automaticaly. If you run out of engine power, NR will start to drop. To recover NR you should lower the collective, which lowers the angle of attack of your rotor blades, which lowers the air resistance acting on blades, thus making engine power required less, allowing the engine to speed up the rotor back to 100%.

Pressing the DATA key will cycle through

WEIGHT & PERFORMANCE	Aircraft weight and estimations on performance margins	
TIME ZONE	Displays the current sim time zone and current local time	
HOIST CABLE LENGTH	If installed, the hoist length reeled out will be displayed here. **Only available with Medical, Firefighter, Offshore variants.**	
CARGO HOOK WEIGHT	If installed, the cargo hook weight detection is displayed. **Only available with Firefighter, Offshore, Cargo variants**	

VMS Main Page (STATUS subformat)

The Status page displays information about the FADEC which is generally used by maintenance only.



VMS Main Page (REPORT subformat)

The Flight Report page displays information about the most recently conducted flight

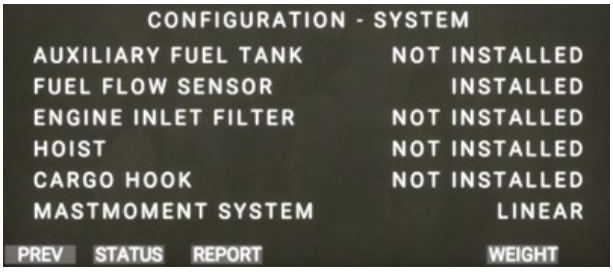
VMS Main Page (EPC subformat)

The Engine Power Check page displays the interface to conduct an on ground or in flight power check.

EPC is not currently implemented.

VMS Main Page (CONF subformat)

The Configuration page displays information about the Helionix configuration which was loaded.



VMS Main Page (WEIGHT subformat)

The Weight page is used to program the aircraft weights before flight.

Press VAL when finished to commit the weights.

Use the **PUSH** function of the MFD knob, to advance the weight entry between PAYLOAD, CREW, and TOTAL FUEL entry lines. All items will be updated immediately when you spin the MFD knob. Both the small and large knobs are active, enabling large and small increments of both payload and fuel.

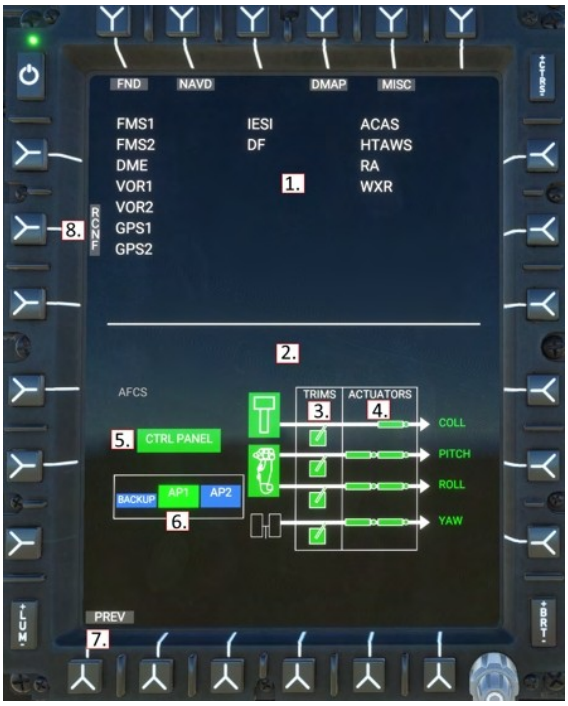
NOTE: Using the in-aircraft WEIGHT tool will not cause the MSFS weight dialog to update. The aircraft will be updated from changes to either location, but the changes are not reflected back in the MSFS weight dialog unless you only use that tool for weight entry. In any case the aircraft will show its accurate weight and fly using those numbers.

VMS Page (System format)

The System page is used to display status on equipment connected to the aircraft (shown at the top) and on the AFCS system (shown at the bottom).

1. Equipment status area
2. AFCS status area
3. AFCS Trim actuator status
4. AFCS SEMA type actuator status
5. APCP (Control Panel), Collective control and Cyclic control status
6. AFCS system status
7. Return to main VMS format
8. Switch to RCNF (Reconfigure) VMS format

AFCS System Status legend:
Green: Active
Cyan: Standby
Red: Inoperative
Gray: Disabled by upstream fault

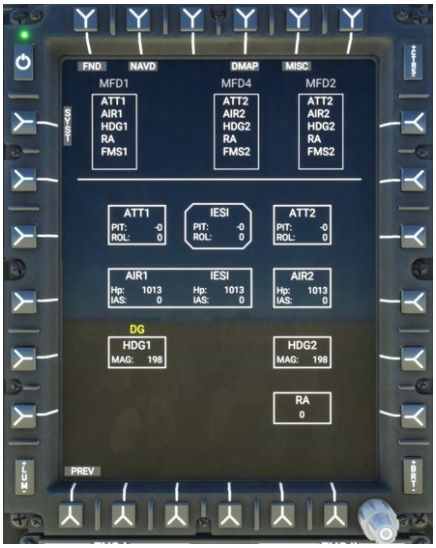


VMS Page (Reconfigure format)

The Reconfigure format is used to display information about the system configuration regarding the AHRS and ADC, Magnetometers and RA. The top of the page shows each MFD and which configuration it is currently using. The bottom of the page shows the individual sensors and their current readings.

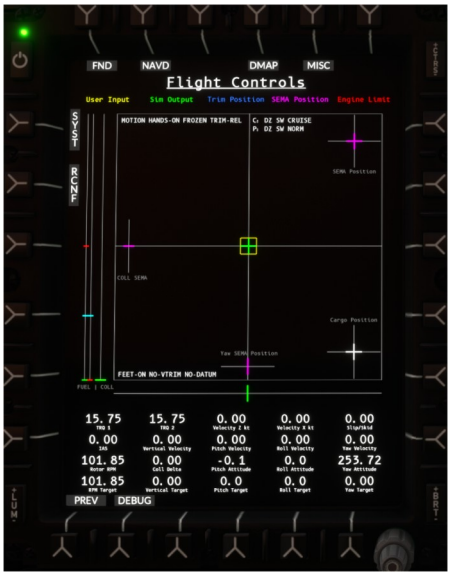
SYST: Switch to the SYST/System VMS format
PREV: Switch to the main VMS format
DG: This DG (Directional Gyro) is currently in free mode. The CCP is used to switch the Dgs from free to slaved (to the magnetometer) and to slew the heading reference when in free mode.

Selective reconfiguration of sensors by the user is currently inoperative.



VMS Page – CTRL (Flight Control)

The Flight Control page displays the current values of the input devices, trim and the SEMA status both numerically and graphically during the flight.
This page can be a great help in resolving errors or issues.



DMAP Page

Digital Map

1. LAYR sk - Toggle airspace visibility
2. Aircraft magnetic heading
3. MAP sk - Toggle between OSM and OpenTopoMap layers
4. PLN/ROS sk – Toggle between NorthUp and HeadingUp views
5. WXR sk - Toggle online weather overlay
6. Inop.
7. HTAWS sk – Toggle HTAWS height above terrain overlay

Use the MFD knob to control the range of the map.

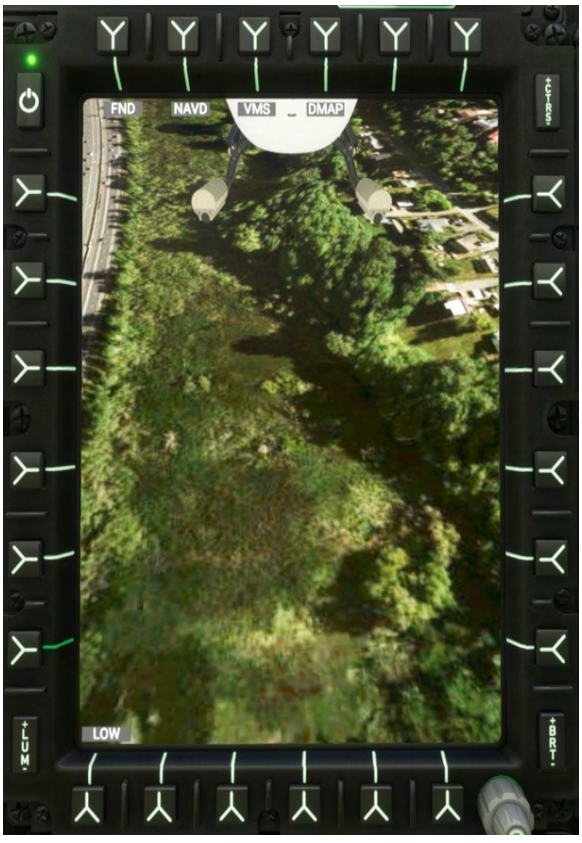
Note: To show traffic on the screen, activate transponder (see activate transponder code)

Data Sources (Advanced)

The map data source is driven by slippy tile xyz sources in:
`Community\hpg-airbus-h145\html_ui\HPGH145-User\MFD\DMAP.json`.

The API format is OpenLayers2 (not leaflet). See here for more providers:
http://leaflet-extras.github.io/leaflet-providers/preview

MISC Page



Miscellaneous, used for auxiliary camera sources. The MISC page is available on MFD1 (copilot) and MFD4 (center).

The tail boom camera is available on all variants. The camera itself is a view provided by MSFS and composed from the satellite imagery overlaid on the 3d synthetic vision viewport.

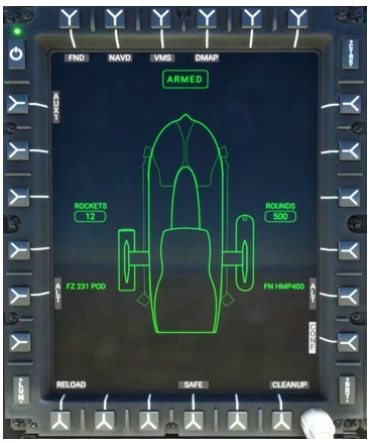
The select keys at the bottom for LOW and HIGH are mutually exclusive.

LOW (Default): Camera will not be updated with aircraft pitch and bank information.

HIGH: Camera will be updated with pitch and bank information, which appears to have a modest cost in sim performance but enables a more realistic view.

When multiple camera sources are available, AUX1 and AUX2 will be displayed in the upper left of the MISC page. Only one source may be visible at a time.

When using the H145M variant, the weapons status page is shown as a camera source. See the Weapons System section for more information



GTN750 Flight Management System

The GTN750 acts as a flight management system, letting you manage the flight plan, locate nearby airports, view maps, and more. While both software integration options (by pms50 and TDSSim) are optional (the aircraft is usable without either option), it is highly recommended to select one.

The pilot GTN750 is powered by both the avionics 2 bus and the essential 2 bus. This means that as long as the battery master is on, the GTN750 on the right side will have power. The copilot GTN750 is powered by only the avionics 1 bus. You will need to use the AVIO 1 switch on the overhead panel to gain access to the left GTN750. Note also that COM2 and NAV2 are on the pilot (right) side of the aircraft, and COM1/NAV1 on the left side.

GTN750 Software Options

Mode	Installation Procedure

Pms50 GTN750
(Recommended)
	Install the pms50 GTN750.
	You will have two folders within Community:
	`hpg-airbus-h145`
	`pms50-instrument-gtn750`

TDSSim GTNXi	Purchase and install the TDSSim GTNXi.
	
	On the H145 tablet, go to the Aircraft app, Options page, and select **GTN Software to TDSSim GTNXi**.
	
	You will have two folders within Community:
	`hpg-airbus-h145`
	`tds-gtnxi-gauge`

No GTN750 **(Not Recommended)**	The GTN750 provides important flight management functions which would not otherwise be available, however a flight plan may still be selected and used from the world map. You will have only one folder within Community: `hpg-airbus-h145`

Pms50 GTN750

The free version is suitable but the premium version adds many features including navigraph charts, checklists and more. The addon is not included in the H145 download, check the installation instructions earlier in this guide for specific installation steps.

A **Not Installed** message indicates that the GTN750 addon was not discovered. Check your Community folder for the pms50-instrument-gtn750 folder

Checklists

To install the checklist follow this procedure: **(Premium GTN750 required)**
1. Unpack the checklist zip file. Rename `HPG_AirbusH145_The1L2P.json` to `import.json` and place the file into `Community\pms50-instrument-gtn750\checklists\gtn750`. Overwrite the existing import.json file.
2. Inside MSFS: In the GTN750 click on: System -> Setup -> Checklists -> Import local file
3. You'll find the checklist under Utilities -> Checklists.

Registration

The GTN750 registration page is for the premium GTN750 license. **Do not enter your H145 license key into GTN750**.

Disable an individual FMS unit

Push buttons may be used to blank the FMS screen and disable processing for that MSFS gauge.

TDSSim GTN750

The TDSSim GTN750 is integrated as 2 units, with unit 1 on the copilot side. You should manually set the navigation source to unit 2 to select the pilot unit. The invisible click spot is not implemented in the VR to switch navigation sources.

The TDSSim GTN750 does not currently save the flight plan into MSFS, so the NAVD and DMAP displays will not show the navigation route

Operating the GTN750

Direct-To airport procedure

If you have an existing flight plan, a Direct-To will replace it with a new flight plan that goes from your current position to the destination. Start from the GTN750 Home page. If you don't see the home page, select the HOME button and you will see a grid of icons. The destination may be found by selecting a nearby airport from a list or by selecting the 4-digit ICAO airport code.

After activating a Direct-To flight plan, you can use the tablet autopilot panel to select the GPS navigation source, or you can use the NAV and CPL softkeys on the pilot MFD to select and couple the navigation source.

Direct-To: Select a nearby airport

1. Select **Nearest** 	2. Select **Airport** 	3. Select an airport 
4. Select the **Direct-To** button 	5. Select **Activate**	

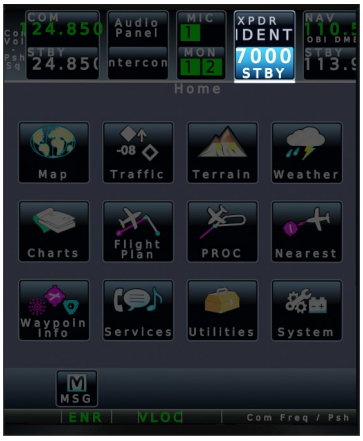
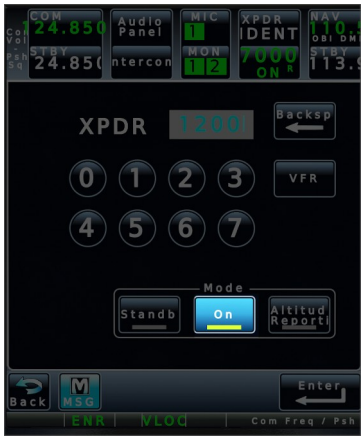
		

Direct-To: Select an airport by name

1. Select **Waypoint Info** 	2. Select **Airport** 	3. Select - - - 
4. Type an ICAO (e.g. **KSEA**) 5. Press **ENTER** 	6. Select the **Direct-To** button 	7. Select **Activate** 

NOTE: WTT Mode is inoperative with version 2.0!

Enter transponder code and turn ON and OFF

1. Select Transponder (in full version on Copilot side) 	2. Enter Code or press VFR 	3. Press **ON** 
4. Press **ENTER** 		

Aircraft Lights

The H145 lights are configurable by the user in the tablet aircraft app. Lights are similar to fixed wing operation, the LDG and S/L will often be used together for any night takeoff and landing.

Lights are primarily controlled by the lighting section on the overhead, as well as controls on the Collective Head.

Lighting Controls

1. High Intensity Search Light - Lamp (if installed)
2. Landing (LDG) Light
3. Strobe Lights
4. Position/Navigation (POS) Lights
5. Anti-collision (ACOL) Beacon Light
6. Cargo/Passenger area Lights (inoperative)
7. Emergency Exit Lights
8. Instrument Panel Light Dimmer switch.
9. Instrument Panel Light Day/Night/Night-Vision-Goggles mode switch.

Instrument Panel lights

Instrument panel lights are available in DAY, NIGHT and NVG modes. DAY mode switches the panel lights off, while NIGHT and NVG enable the integral (green) panel lights. Use the dimmer switch (8, above) to adjust the light intensity.

DAY, NIGHT and NVG will also impact the base brightness for the MFDs.

Cockpit lights

There are two cockpit lights available at the front and rear of the overhead panel. Click the light itself to activate.

Cockpit lights are available when the master battery switch is ON or when the Ground Power switch is set to ON (not to be confused with External Power). When using Ground Power without the aircraft battery, the lights will be switched off after 10 minutes.

Exterior lights

1. Landing (LDG) Light
2. Second Landing Light
3. Search and Landing (S/L) Light
4. Landing (LDG) Light - (Optional if the radome is not installed)
5. Boarding step light
6. Hoist light (automatic activation)
7. Navigation/Position (POS) Lights (left - red, right - green, rear – white)
8. Strobe (white) and Beacon (red) lights

Emergency Exit Lights

Exit lights (4) are installed in the cabin and cockpit. They are powered by both the main battery and the standby battery.

The exit lights will illuminate when
1. EM/EX switch to ON
2. EM/EX switch set to ARM (armed)
   and one of the following
   a. Hard landing
   b. Door opens
   c. Loss of electrical power

Emergency Floats

The emergency flotation system installed permanently on the Luxury and Offshore variants. Floats are optional on the EMS, Civil Cargo and Firefighter variants.

The system must first be armed using the switch on the overhead panel, and then it can be activated either automatically or when the pilot activates the **FILL FLOATS** function.

Access the fill floats and repack functions by clicking the top clock on the tablet and then clicking the button on the Emergency Floats notification in the list.

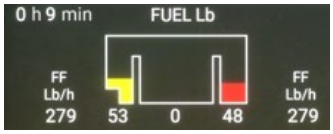
Function	Action	Notes
Arm / Disarm	Use the switch on the overhead panel, EMER FLOATS and set it to ARM or OFF	
Activate	**Automatic** activation upon water landing. **Manual** activation is accomplished by a button on the tablet, or the FILL FLOATS binding.	Automatic activation when landing on water is mandatory.
Repack	Use the Tablet Aircraft app, click the top clock, and select **Repack** on the notification.	Not a realistic function
Test	Set EMER FLOATS to TEST	This test is checking for power from both electrical sources.

Fuel System

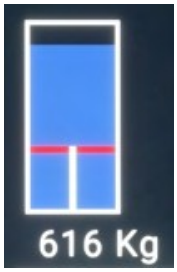
The H145 has a fuel system that consists of a main (center) tank, two supply tanks (right and left, connected only to the respective engine on that side of the aircraft), 2 transfer pumps to move fuel from the main tank into the supply tank, and 2 prime pumps to move fuel from the supply tank to their respective engine.

There are 4 fuel pumps. The two transfer (XFER) pumps move fuel from the main (center) tank into the supply tanks. Both transfer pumps will move fuel independently from the main tank into both supply tanks. The two prime pumps move fuel from the supply tank to the engine on that side of the aircraft. The two prime pumps should both be switched ON during engine start, and OFF after both engines are running. Transfer pumps
should be switched on after engine start and remain on until just before engine shutdown

Information about the fuel system is presented on the FND and VMS pages, as well as critical indications on the message list and on the Warning Unit.



- Even when the main tank is empty, the transfer pumps are to remain ON.
- A red line indicates that the transfer pumps are not active and cannot move fuel from the main tank to either supply tank.
- The fuel system has been designed such that the left engine will shut down from fuel starvation before the right. Should this ever occur an immediate landing shall be conducted.
- Display units and presence of Fuel Flow sensor may be set using the tablet Aircraft app.



The warning unit contains low fuel warnings relating to the supply tanks only. The red guardsmay be lifted to engage an emergency fuel cutoff for the engine should a fire occur.

Electrical System

The H145 has a 28V DC electrical system consisting primarily of:
- Aircraft main battery
- Standby battery
- Bus tie (x2)
- Combined Starter/Generator (x2)

The two sides of the aircraft are redundant and split into system 1 and system 2. Critical equipment is powered by both systems. There are additional shedding buses on each system which are left unpowered unless both electrical generators are running or external power is connected. Each system has a bus tie which allows the system to be isolated (normally they are not isolated to allow sharing of electrical loads) When bus ties are closed, both systems can use power from the main battery or either generator.

BUS TIE (1 and 2)

Bus tie connections are used to isolate a system from a malfunction within another part of the aircraft. They shall remain in the NORM and guarded position unless directed by a checklist.
When switching a bus tie switch to the NORM position from OFF, it should first be moved to the spring-loaded RESET position.

GEN (1 and 2)

Generator switches may be used to isolate a malfunctioning generator from the rest of the aircraft. They shall remain in the NORM and guarded position unless directed by a checklist. When switching a generator switch to the NORM position from OFF, it should first be moved to the spring-loaded RESET position.

EMER SHED BUS

The Emergency Shedding Bus is a third optionally isolated part of the electrical system. The EMER SHED BUS switch shall remain in the NORM (off) position unless directed by a checklist. The Emergency shedding bus provides power to the IESI and emergency exit lights, and also provides an alternate path for the Radar Altimeter. If you should lose both generators in flight you may recover the RA parameter by switching the EMER SHED BUS to the ON position.

BAT MSTR

The BATTERY MASTER switch is the main aircraft power switch. It has 3 positions, OFF, ON and ENGAGE. When switching from OFF to ON, it shall first be moved to the spring-loaded ENGAGE position. This act activates the bus tie system andcloses the BUS TIE 1 and BUS TIE 2 connections.Failure to move the switch to the ENGAGE position will result in BUS TIE OPEN messages on the message list as well as failure to share electrical load between systems 1 and 2.

DC RECEPT

The D.C. Power Receptacles switch controls a utility bus which may be used to charge the onboard tablet.

STBY BAT

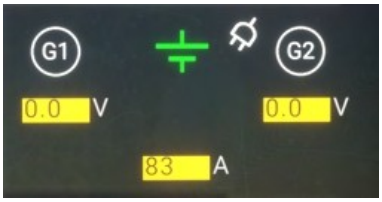
The STANDBY BATTERY powers the IESI, exit lights and the emergency shedding bus when isolated. It should be engaged for flight and shut off after shutdown, otherwise the IESI may remain powered after shutting off the master battery switch. There is no indication on the VMS page for the standby battery. If it is discharging a notification will be present on the message list

AVIO (1 and 2)

Avionics bus 1 and 2 power critical avionics equipment such as the aircraft main computers. AVIO 1 is required for the copilot GTN750 to function. The AP1 and AP2 functions on the APCP depend on AVIO 1 and 2 respectively.

VMS electrical information section

The VMS page contains a section for electrical information, showing the status of both generators and the aircraft main battery. The VMS NUM sk can be used to reveal nominal digital parameters.



WHITE: Source is not active
GREEN: Source is active
YELLOW: Source is over limit

External Power

The tablet may be used to connect and disconnect external power. An icon will be displayed in the notification bar if the external power unit is available or connected. A message in the notification list may be used to CONNECT and DISCONNECT external power.

Note: External power will not power the aircraft unless the BATTERY MASTER switch (and subsequently the bus tie system) is ON.

Weather Radar

Weather radar is available as an option in the civilian variant, and always available in the luxury variant. The exterior radome option and weather radar are installed and uninstalled together. You can select the radar option in the tablet aircraft app.

Weather radar is available on the FND and NAVD pages. The DMAP page and the tablet Maps app use an online weather service which works irrespective of the selected radome option.

The weather radar is controlled via the WXRCP panel on the center console. The main knob has 4 functions (OFF, Standby, TEST, and On). STBY mode is used when on the ground. TEST mode is used to display a test pattern on the FND and NAVD pages when weather has been selected.

Message	Meaning	Notes
FAIL or **WXR FAIL**	Weather radar is not detected electrically. Set the WXRCP knob to ON.	Radome must be installed for WXRCP to be visible on the center console.
STBY or **WXR STBY**	Weather radar is detected but in standby mode. Set the WXRCP knob to ON.	
CHECK RANGE	MFD4 and MFD2 range knobs must match, or WXR should be de-selected.	MFD1 has independent range.
WXR TX INHIB	Weather radar is automatically inhibited when less than 50FT RA to protect personnel.	Weather radar will become available automatically when crossing 50FT RA.

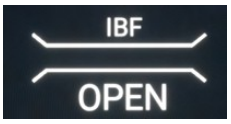
Engine Filters (IBF)

Inlet Barrier Filters protect the engine from abrasive sand causing excessive wear on the engines. The filters block sand and dirt but will eventually become clogged which will reduce engine performance. The pilot may control the operation of the bypass doors, which allow a way for air to directly reach the engine, bypassing the filters. It's best to keep the IBF system in the NORM position unless operations require otherwise.
Clogging over 100% will reduce engine performance.

IBF controls are on the overhead panel, and IBF indications are on the VMS page and on the Message List.

Each IBF switch corresponds to the respective engine, and has 3 positions as explained below. The Recall switch is currently inoperative

CLOSED: Bypass doors are closed and will not open automatically. The filters will protect the engine but in the case of OEI the bypass doors will not open automatically, meaning full power will not be available.
OPEN: Bypass doors are open and the engines are directly exposed to outside conditions.
NORM: Bypass doors will automatically open and close based on the relevant systems logic. Bypass doors will normally stay in the closed position but will open automatically in the case of clogging over 100% or in the case of OEI.

Bypass door open	With the bypass door open (either due to the switch or by automatic logic) the engine has maximum performance but sand will cause excessive wear.	
Clogging	IBF clogging up to 100% will not reduce engine performance. IBF CLOG TREND message will be associated with early clogging.	
Excessive clogging	IBF Clogging over 100% will result in performance degradation, OEI performance margin will not be guaranteed.	

Clogging levels may be controlled in the Failure & Maintenance app. Clogging is disabled when the Aircraft Damage is set to Off in the Aircraft app on the setup page.

Failures & Damage Model

The H145 has a suite of failure modes and also a systems damage model for key aircraft systems. These features can be optionally disabled in the tablet Aircraft app, on the Setup page by setting **Aircraft Damage** to **Off**.

Failure & Maintenance app

The Failures page shows a list of any active failures, and the maintenance page shows damage and degradation status.

Do All Maintenance: This will reset all damage

Aircraft failures are best triggered from within the mission system, as various timers or triggers may be devised to enter or exit the failure condition.

Aircraft Failure Variables

Scope	L:Vars
Engines	**0-100 (percent damage)** L:H145_PERSIST_DAMAGE_ENG1_PCT L:H145_PERSIST_DAMAGE_ENG2_PCT **0 or 1 (boolean logic)** ENG ON FIRE:1 ENG ON FIRE:2 GENERAL ENG FAILED:1 GENERAL ENG FAILED:2 L:H145_FAIL_FADEC1 L:H145_FAIL_FADEC2 **Note**: Using FADEC EMER will recover from level 3 to level 2 failure and thus regain substantial engine control by use of TOT-matching the engines. **Bottles 0: empty, 1: charged** L:H145_SDK_FIREBOTTLE_1 L:H145_SDK_FIREBOTTLE_2 H:H145_SDK_FIREBOTTLE1_EMPTY H:H145_SDK_FIREBOTTLE1_FULL H:H145_SDK_FIREBOTTLE2_EMPTY H:H145_SDK_FIREBOTTLE2_FULL
Hydraulics	**0 or 1 (boolean logic)** L:H145_FAIL_HYD1_LOWPRESS L:H145_FAIL_HYD2_LOWPRESS
AFCS	**0 or 1 (boolean logic)** L:H145_FAIL_AP1 L:H145_FAIL_AP2 L:H145_FAIL_BKUP L:H145_FAIL_APCP L:H145_FAIL_PITCH_SEMA1 L:H145_FAIL_PITCH_SEMA2 L:H145_FAIL_ROLL_SEMA1 L:H145_FAIL_ROLL_SEMA2 L:H145_FAIL_YAW_SEMA1 L:H145_FAIL_YAW_SEMA2 L:H145_FAIL_COLLECTIVE_SEMA1
Fuel	**0 or 1 (boolean logic)** L:H145_FAIL_FUEL_F L:H145_FAIL_FUEL_A
Transmission	**0-100 (percent damage)** L:H145_PERSIST_DAMAGE_MGB_PCT L:H145_PERSIST_DAMAGE_TGB_PCT **0 or 1 (boolean logic)** L:H145_FAIL_MGB_CHIP
IBF System	**0-165 (percent clogging)** L:H145_PERSIST_IBF1_PCT L:H145_PERSIST_IBF2_PCT
Other	**0 or 1 (boolean logic)** L:H145_SDK_MASTMOMENT_EXCEEDED H:H145_SDK_MASTMOMENT_EXCEED_ON H:H145_SDK_MASTMOMENT_EXCEED_OFF

Hype Tablet

The tablet may be opened or closed by clicking the hinge. The home button will close apps until there are no more apps open, then will also close the tablet. If you use the hinge you may keep the app open while the tablet is closed. The tablet battery will drain if you do not turn on the helicopter and also engage the DC RECEPT switch.

Tablet Hinge: Click the hinge along the left to open or close the tablet.
Home Button: The home button will close the current app (or view) until showing the home screen, and then one more press will close the tablet.
Status bar: Action Center is available by clicking the status bar at the top of the screen

Apps

Aircraft	Configure the aircraft here
Failures & Maintenance	Configure aircraft failures and restore aircraft damage here.
Missions	Discover and execute missions here.
Documents	View image charts of your choosing. A copy of the user guide and normal procedures is also included here.
LittleNavMap	View the LittleNavMap.exe view from the tablet.
Sound Mixer	Adjust aircraft sound intensities here.
Maps	View maps.
Alarms	Set alarms and timers.
METAR	Query METARs for airports (live weather).
Web	Access select (very restricted) web pages directly.
EFBConnect	Mirror web pages into the simulator.
Flappy Bird	A simple game (to be used only on the ground).
Neopad	Access the NeoFly app
Direction Finder	Configure the DF hardware with a location.
Event Tester	Test the Home Cockpit SDK events without first binding the controls.

Aircraft (Setup)

The aircraft app combines unrealistic functionality with required setup and equipment and options configuration.

Setup

Setting	Options	Notes
Gameplay Mode (Flight Model) **not used since .500**	Realistic Arcade	**Realistic mode** is the default flight model and is much more advanced. Arcade mode is an older version which some users may prefer but is not recommended. **Arcade mode** will also disable advanced flight model features: - Rotor overspeed and underspeed - Vortex Ring State - Aircraft Damage
Rotor Torque	Off / On	Rotor Torque is the normal tendency of the aircraft to pull to the right unless countered with the left pedal. The real H145 has trimmable pedals which operate in sync with the AFCS, so both settings can be considered reasonable. You may prefer the Off setting if you do not have physical rudder/yaw pedals.
Vortex Ring State	Off / On	VRS is a dangerous condition where the aircraft descends into its own downwash.

		Not available in Arcade mode.
Aircraft Damage	Off / On	Damage to Transmission and Engines, IBF simulation that will restrict engine airflow and subsequent power available Not available in Arcade mode.
AFCS ATT Follow-Up Trim	Only Cruise Only Hover	When in ATT mode (blank on the MFD), this setting controls whether trim release is required at lower or higher speeds.
Cyclic Sensitivity	(-10)-(+10)	This setting is intended to compensate for a cyclic control which is longer or shorter (like an xbox controller or a more professional control setup)
Cyclic Deadzone	1%-100%	This configures the deadzone used by the AP to determine when you are pushing on your joystick controller vs leaving it neutral/at rest. Select the lowest value that still results in your controller being consistently detected correctly (yellow=you are pushing on it, white=at rest)
Pedals Deadzone	1%-100%	Same as above, but for pedals
AirlandFS Flight Model	(not present) Active	This item is shown only when AirlandFS has configured the aircraft for external flight model use. Once AirlandFS is closed, clicking Active will recover the integral flight model.

State Load	Options	Notes
Ready for Takeoff	Configures the aircraft for takeoff (engines, AFCS).	
Cold & Dark	Configures the aircraft for Cold & Dark at the ramp.	

Equipment

Setting	Options	Notes
Radome (Weather Radar)	Installed Not Installed	This option will be set by the livery author and may then be changed at any time.
External Weapon Pods	Installed Not Installed	Weapon pods on the right and left. **Only available on the military variant.**
Fabric Glare Shields	Installed Not Installed	Required for night flight, these fabric curtains block light from shining upwards from the ground onto the canopy window
Emergency Floats	Installed Not Installed	
Wire Strike Protection (Top, Bottom, Skids)	Installed Not Installed	This option will be set by the livery author and may then be changed at any time.
Skid Settling Protectors	Installed Not Installed	This option will be set by the livery author and may then be changed at any time. **Not available on the luxury variant.**
Air Conditioning	Installed Not Installed	This option will be set by the livery author and may then be changed at any time.
Fuel Flow Sensor	Installed Not Installed	Endurance calculation unavailable without fuel flow sensor
Inlet Barrier Filter (IBF)	Installed Not Installed	BF system protects engine inlets from sand ingestion while providing bypass doors for emergency power
ACAS (Airborne anti-collision system)	Installed Not Installed	Traffic alerts unavailable without ACAS
HTAWS (Terrain awareness system)	Installed Not Installed	HAT (height above terrain) database and aurals unavailable without HTAWS.

Options

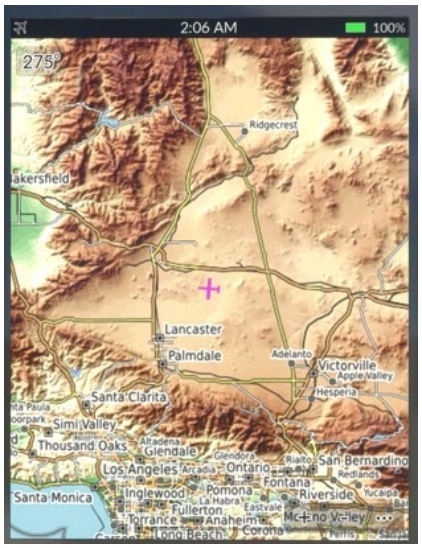
Setting	Options	Notes
GTN750 FMS System Software	Pms50 GTN750 TDSSim GTNXi	See the GTN750 section of the user guide for more information. Restarting the flight is not necessary but it is not recommended to load both GTN750 software options concurrently.
Fuel Units	Lb (Pounds) Kg (Kilograms) L (Liters) Gal (Gallons)	
Weight Units	Kg (Kilograms) Lb (Pounds)	
Barometric Units	In/Hg hPa	Controls the baro readout on the 3 MFDs and IESI.

Temperature Units	C F	Displays the OAT (outside air temperature) value on the FND page in either fahrenheit or celsius
Rotor Downwash Effects	On Off	Particle effects on dirt/grass, sand, snow, water. Has GPU impact
Pilot automatic hide	Head & Body Head Only	This setting controls whether you can slew into the pilot bodies in the cockpit Use the Head Only setting if you experience the pilot flickering based on your movements with TrackIR or VR
Rotor Blur Casts Shadow	On Off	Set to OFF to reduce flicker in cockpit
Show parking brake on tablet	On Off	Show of Parking Brake on tablet status bar. (Useful for VA/mission software that uses this as a cue to begin or end)
Tablet Clock/Time reference	PC Time Sim Time PC Time UTC	PC Time should match your Windows PC clock. Sim Time should match the time you see on the FND or VMS clock, or the MSFS Weather dialog.
Tablet Clock/time format	12-hour 24-hour	Display format AM/PM or 24 hour time.

Crew & Payload

Setting	Options	Notes
Seat Selection - Pilot Seat Selection - Coilot	Hype Asobo	Choose to use Hype pilot models or Asobo. **Note** only the Hype pilots may operate as headless
Seat Selection - 2	Crew Worker	Choose which human model type to use
Seat Selection - 3 Seat Selection - 4	Worker Survivor	Choose which human model type to use
Hoist Selection	Stowed Deployed Crew Crew+Stretcher Worker Crew+Survivor 1 Crew+Survivor 2 Containers Hose	Stowed (Hoist is stowed and off) Deployed (Hoist arm swings out) Choose the hoist objects currently attached. **Note** unless using a mission, the object will not automatically detach when reaching the ground **Note** options available vary per variant

Maps app



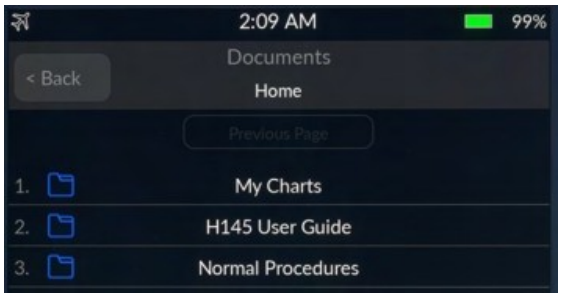
The maps app features zoom controls and the ability to change the map orientation between North-Up and Heading-Up modes. The current magnetic heading is also displayed in a digital/numeric value at the top of the page.

Click **[...]** to reveal the maps selection panel where you may select between a variety of map sources as well as airspace and weather overlays.

Advanced Configuration (Optional)

The maps app uses OpenStreetMap tile data. You may configure the sources and their relevant options through the configuration file located at:
`html_ui\HPGH145-User\Tablet\MapsApp.json`.
Edit this file to change from the default provider of
`https://{a-c}.tile.opentopomap.org/{z}/{x}/{y}.png`.
See here for more providers: http://leaflet-extras.github.io/leaflet-providers/preview/

Documents app



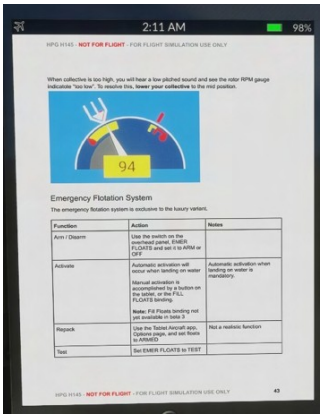
The Documents app is a simple image chart viewer app. It will show documents built into the H145, provided by your livery, or documents of your own choosing.

To show your own “user documents”:

1. Install the UserDocuments package in your Community folder. You will install this new folder alongside the main hpg-airbus-h145 folder, don’t put it inside of the main H145 folder.

2. Place your files in
`hpg-airbus-h145-userdocs\html_ui\HPGH145-User\Documents`

3. Double click the script **Update Docs For Sim** in the UserDocuments. This will overwrite the Index.json which the sim reads the filenames from. It will also update the package layout.json. You must restart your sim after adding or removing files.



Neopad app

Download Neofly and Neopad

The neopad app is present ONLY if neopad is installed within the simulator. You should see Neopad on the sim toolbar. Once installed, the app will be visible on the tablet. The app will simply connect to the NeoFly app, so make sure the toolbar app is working as expected if you are having any problems.

The neopad server url may be changed via:
`Community\hpg-airbus-h145\html_ui\HPGH145-User\Tablet\NeopadApp.json`.

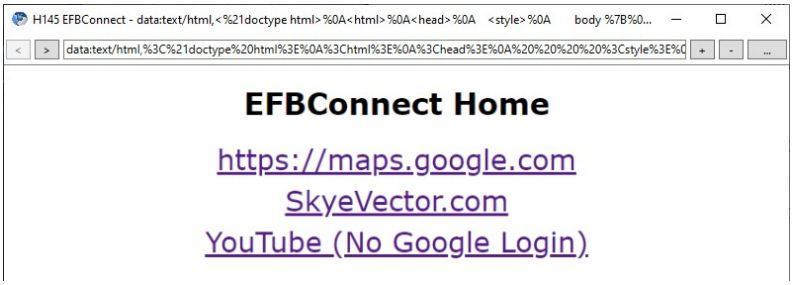
Web Browser

The browser app will load websites which have an access-allow-origin set to allow the game to access it, which is unfortunately rare. Web Browser app will load URLs from
`Community\hpg-airbus-h145\html_ui\HPGH145-User\Tablet\WebBrowserApp.json`.

EFB Connect (Web Browser)

EFBConnect is a companion app which runs on your PC and provides a web browser with more functionality inside the sim, including some ability to stream video.

Download EFBConnect from here:
EFBConnect is here:



https://drive.google.com/file/d/1dRUggLARGRTypUOYagj7junVOX49b15G/view?usp=sharing
(it is currently outdated and refers to H135).

EFBConnect will load to the configurable home page:

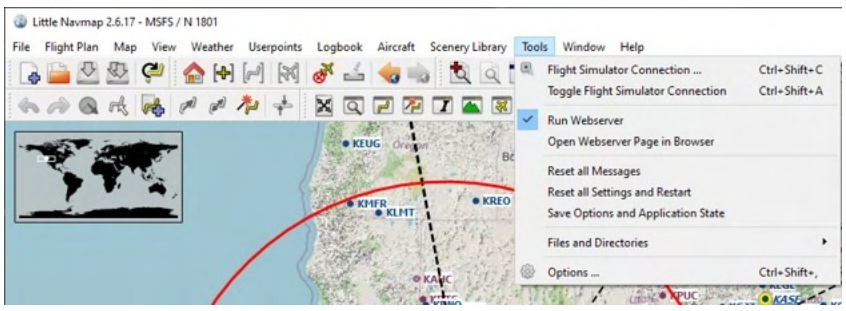
As soon as EFBConnect opens, the tablet EFBConnect app will automatically connect. You are able to send mouse clicks directly on the tablet in the sim, but currently you cannot scroll or zoom or use the back button.

Content that you view in the EFBConnect app will be mirrored in the sim, and light tasks may be accomplished directly through the sim.

LittleNavMap app

Download LittleNavMap

The LittleNavMap app contacts the LittleNavMap webserver. Click Tools then **Run Webserver** to start it. You can test that it is working by viewing the page in your browser.



Advanced Configuration

There is no need to change the below but the options are available if needed, especially changing the Url to another computer.

Configuration file: `html_ui\HPGH145-User\Tablet\LittleNavMapApp.json`.

Parameter	Values	Notes
Url	Default: http://localhost:8965	Server location (ip or host name including protocol)
FrameDelay	Default: 1000	Milliseconds to wait after receiving a farme before loading the next frame
Scale	Default: 1	Requested view size
ZoomScale	Default: 2	Zoom increment
Format	jpg or png (Default: jpg)	
Quality	0-100 (Default: 80)	JPEG compression level

Parameters are passed directly to the LittleNavMap web server.

Events Tester

The Events Tester app is intended as a simple way to check key bindings prior to selecting them in ConfigTool. Events Tester presents a list of categories and within each category are commands which correspond to functions in the H145. The command will execute each time you tap on the entry.

Missions

The missions app enables the user to conduct missions. See the top level Missions section in this document for more information.

Sound Mixer

The Sound Mixer app presents sliders to adjust certain aircraft sounds.

Name	Notes
Rotor Bladeslap	Heavier rotor blade sounds when the air collides with recently disturbed air. Lower values are recommended
Avionics Alerts - Critical	
Avionics Alerts - Advisory	
Crew Audio	Crew guidance audio. **Not yet operative**.

Direction Finder

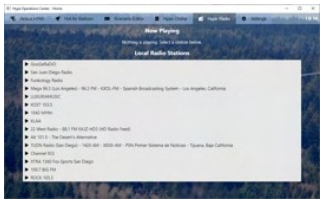
The Direction Finder (DF) app presents an interface to set entries from DFApp json as the active DF source. The DF bearing pointer is available on FND and NAVD under the BRG1 and BRG soft keys.

The file location for entries is
`hpg-airbus-h145\html_ui\HPGH145-User\DFApp\index.json`.

Hype Radio

Listen to internet radio stations based on your flight simulator location. Start the Hype Operations Center app on your PC and then click Hype Radio on the tablet.

Radio stations will be updated every time you launch the app and the radio will remain playing even if you leave the range of the station, or go to the menu to change location.

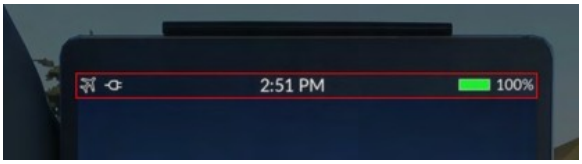


Wallpaper

The tablet wallpaper is located at html_ui\HPGH145-User\Tablet\wallpaper.jpg
The wallpaper may also be set based on the currently active livery. See the **Livery Author Info** section for more info.

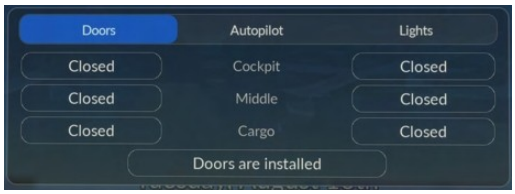
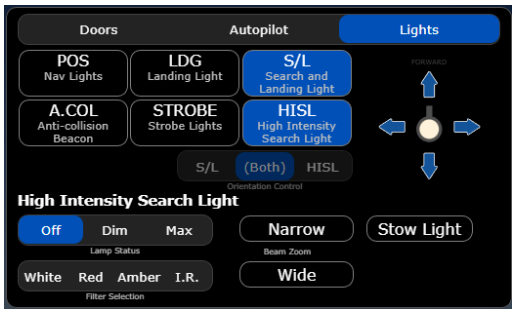
Action center

The Action Center is available from all apps and opens on top of any currently opened app. Click the clock to activate the Action Center.



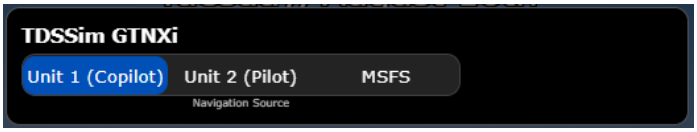
Always-visible sections

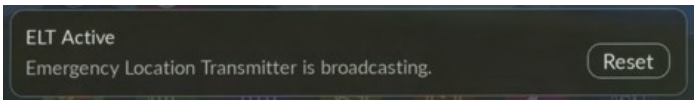
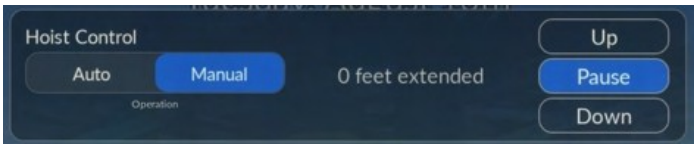
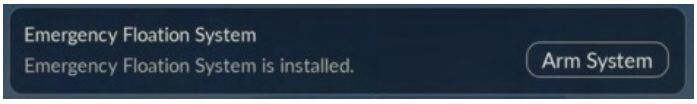
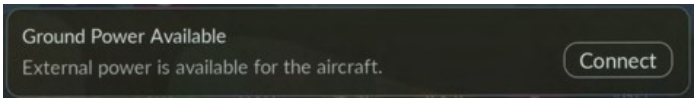
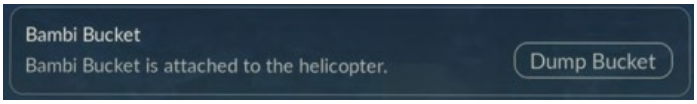
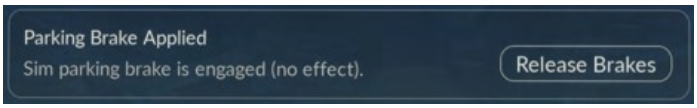
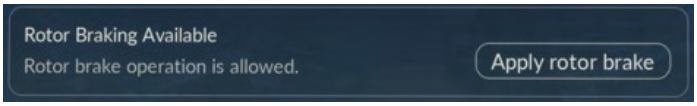
These sections are available at the top of the Action Center under all conditions.

Tablet Brightness control	
Doors	
Autopilot	
Autopilot - extended controls	
Lights	
Lights - with extended HISL controls	

Contextual sections

These sections (or notifications) are available only when certain conditions are met

H145M Weapons	
TDSSim GTNXi Navigation Source	

One Engine Inoperative	
ELT Active	
Hoist Control	
Emergency Flotation System	
Ground Power Available	
Open Door	
Bambi Bucket	
Parking Brake Applied	
Rotor Braking Available	

Status bar

Icons in the status bar are always contextual based on equipment and conditions. Icons will appear and disappear based on events like the GPU being made available, and icons will use a background color to draw additional attention to specific statuses.

GPU status icons

	GPU is connected and on. You must disconnect prior to takeoff.
	GPU is available but not currently connected.

Cargo Hook (with Remote Hook) status icons

	Remote Hook is attached, no load, off the ground.

	Remote Hook is attached, no load, within range of attach/detach
	Remote Hook is attached, with load, off the ground
	Remote Hook is attached, with load, within range of attach/detach

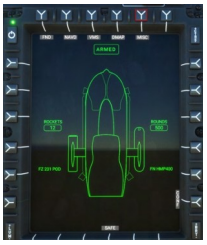
Bambi bucket status icons

	Bambi bucket is attached, empty, and off the ground.
	Bambi bucket is attached, and below the ground level
	Bambi bucket is attached, has water, and above the ground level.
	Bambi bucket is attached and currently dumping

Other status icons

	One or more doors are currently open.
	Rotor braking is available
	ELT is actively broadcasting.
	Parking brake is on and the setting is also on to show it (normally hidden)

H145M Weapons Systems



The H145 military variant is equipped with optional external weapon pods. The weapon pods may be configured for a cannon or a set of 2.75” 70mm **FZ275 LGR** rockets. (currently operating as unguided missiles).

Livery authors can choose to set the weapon pods to be on or off by default. You can use the tablet aircraft app, on the Equipment page to turn the weapon pods on or off at any time during your flight.

The default key binding to fire rockets is covered in the key bindings section of this user guide, and is configurable via ConfigTool. You may also add a custom key binding for **Reload** if you wish.

You may also reload ammunition by clicking on the box behind you in the cockpit. The tablet has an indication of rockets remaining and a button to fire if you haven’t set the key binding (click the top clock).

Weapons status is available on the MISC page on the center and copilot MFDs, as well as on the tablet Action Center.

The **Cleanup** function will immediately remove all rocket holes in the earth and rocket particles. This can be useful after firing many rockets to recover lost fps.

Sensor Pod & Monitor

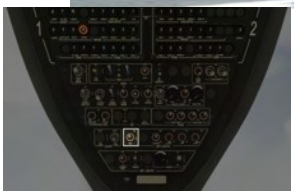
The H145 military variant is equipped with a sensor pod which provides imagery to the internal monitor display. The pod rotates horizontally 180 degrees and vertically 90 degrees (full forward to straight down).

1. Power on/off
2. Pod heading
3. Move Right
4. Move Left
5. Move Forward
6. Move Aft

Key bindings may be configured in ConfigTool for the sensor pod functions.

Action Pack - High Intensity Search Light

The HISL is mounted to the front left skid and controlled with the control panel on the center console as well as a switch on the overhead for the lamp itself. There are also controls on the collective head.

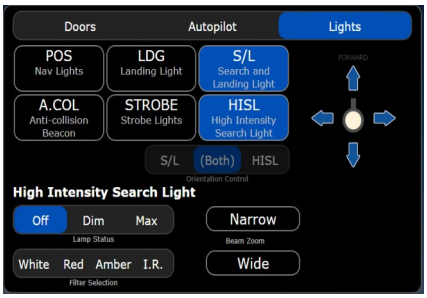


1. System power on/off
2. IR status light (inoperative)
3. Laser control (inoperative)
4. System status display
5. STOW push button with status LED
6. DEPLOY push button with status LED
7. SLAVE push button with status LED
8. LAMP push button with status LED
9. DIM status button with status LED
10. Steering hat (inoperative)
11. Zoom control
12. Filter control
13. Crew select control (inoperative)

Overhead Panel:
HISL Switch: Lamp power

Collective Head:
S/L STEERING: 4-way steering
ORIENT CTRL: Orientation Control, between either S/L or HISL. **Currently inoperative**.

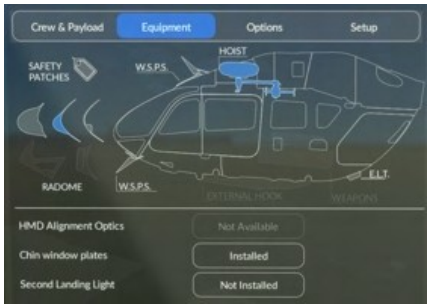
HISL Steering in 4 directions can be accomplished also with the S/L steering events (see the controls setup section of this document



Key features
- Filters
- Zoom
- Lens colors + IR
- 4-way steering

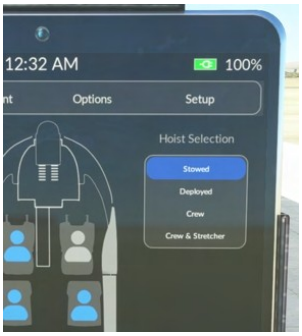
Action Pack - Hoist/Winch

The Hoist is available on the EMS, Firefighter and Offshore variants. It can be used within the context of a mission or it can be managed manually by the user. The Hoist is installed exclusively on the right side of the aircraft



Manual Hoisting

To manage the hoist manually.
1. Ensure the hoist is installed. **Tablet -> Aircraft →Equipment**
2. Deploy the hoist, open the door and select an object. **Tablet -> Aircraft -> Cabin & Payload**. On the top left of the page you may select all of the possible states for the hoist. Don't forget to open the door and select the operator to be positioned.
3. Control the hoist using the key bindings (available in ConfigTool) or by using the tablet without any new bindings. Click the top clock on the tablet, then select **Manual** under **Hoist Control**. You may now move the hoist up, down or pause it in place.



Note: The hoist will still automatically reel up when approaching the ground to avoid the object clipping through the terrain.

Automatic hoisting

1. Start the Rescue + Hospital Transfer (Use Anywhere) mission, or select other hoisting missions such as Road accident rescue.
2. Fly to a hoisting location of your choosing or fly to the target destination of the mission.
3. Select Begin Hoisting.
4. The crew will automatically operate the hoist as you pilot the aircraft.

Action Pack - Cargo Hook & Bambi Bucket

Cargo Hook

Use the `H145M Weapons - Fire (Primary)` key binding to toggle the attachment of the cargo hook to an object, and to detach from the object. To release an object you must be low enough for the hook to be unloaded (the object on the ground and no longer held by the cable). There is an icon at the top of the tablet to assist in the current status.

Enable the cargo hook option in the tablet, Aircraft app, Equipment page. You can set the object on the hook manually on the Crew & Payload page, however the object will not detach when set onto the ground.

Bambi Bucket

Attach the Bambi bucket to the firefighter variant by using the tablet Equipment page and selecting the Bambi and the Cargo Hook.

You may dump the bucket by using the H145M Weapons - Fire (Primary) key binding for activation. There is also a soft button in the tablet action center when applicable.

The DATA soft key on the VMS page will display the cargo hook status. The box will be empty when there is no weight on the hook and otherwise display the detected weight.

The bucket may be filled from any MSFS water source, or from dynamic objects designated as water sources. Two alternative water source objects are provided and may be placed with the Scenario Editor.

The bambi bucket can be controlled on the equipment page in the tablet aircraft app.

Hype Live Map

The Live Map presense may be configure on the tablet.

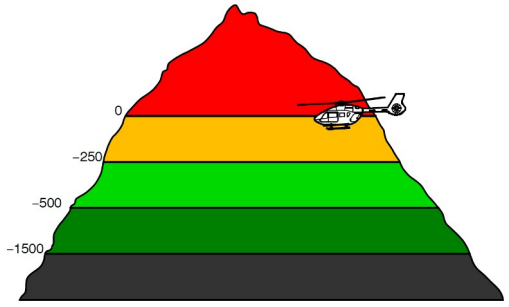
`Tablet` -> `Hype Online` (app). Only `Detailed` information will reveal callsign, which is to be entered at the MSFS World Map page, prior to starting the flight.

Terrain Warning System (HTAWS)

Helicopter Terrain Awareness System (HTAWS). It is based on local stored terrain data (`/Community/hpg-htaws-data/`). This must be installed separate (see pinned msg on Discord)

You may mute the aural warnings by selecting the overhead AUDIO/TAWS switch to STBY or MUTE.

You may also assign a binding in Hype Operations Center, to `HTAWS MUTE-FOR-5-MINS`, which is intended to be clicked shortly before landing.



Starting with .494 the new terrain profile for the HTAWS.

AFCS (Autoflight System)

The H145 autopilot is a comprehensive autoflight system, capable of both basic stabilization and also fully-hands-off upper modes. The system combines redundant Stability Augmentation Systems with redundant aircraft management computers, which take data from aircraft sensors and send commands to the actuators. The systems are monitored by and interacted with through the MFDs, Autopilot control panel (APCP) and controls on the cyclic and collective.

Background

The H145 flight controls are augmented by parallel actuators, called SEMAs (smart electro-mechanical actuator). These parallel actuators are invisible to the pilot (not felt in the controls) and are controlled by aircraft computers directly. The SEMA are quick and powerful but limited in travel. The total SEMA travel will be only 10% of the pitch axis and 20% of the roll and yaw axis. For this reason, the AFCS also needs the ability to re-center the SEMA. The A.TRIM (auto trim) system is able to receive commands from the AFCS computers and then slowly drive the trim motor in the requested direction. As the trims move, the pilot sees and feels their cyclic moving. It is for this reason that the A.TRIM system must be engaged to use UPPER MODES, as otherwise the saturation of the SEMA could not be automatically resolved by the computers.

Community member Josh has created a detailed H145 Autopilot explanation video .

Trim Release

Trim Release (Cyclic Trim Release) is a button on the cyclic control which the pilot holds anytime they are manipulating the cyclic. This is a very important aspect of H145 and doing so both pauses the AFCS (so it doesn't fight you) and also gives you maximum precision. There is also a `Collective Trim Release` , but it isn't as necessary as the cyclic version.

Holding Trim Release is known as **Fly-Through Action**.

Follow-Up Trim

Follow-Up trim will allow you to manipulate the cyclic without using trim release, however this will only work well if you also have a very small H145 cyclic deadzone setting. Even still, Trim Release is recommended for all users. Note that Follow-Up trim is only enabled on ATT mode (and submodes). When follow-up trim is active, attitude setpoints will be updated for any deflection.

Follow-Up trim has 4 settings. The `Only Hover` option is considered the realistic setting, but `Off` is recommended for learning and understanding the systems.

Setting	Mode
Off	Follow-up trim is never active.
Only Hover	When below 30kias, follow-up trim is active.
Only Cruise	When above 30kias, follow-up trim is active.
Both	Follow-up trim is always active.

AFCS OFF and A.TRIM OFF Operation

Pilots are to fly with AFCS engaged during all normal operations. A.TRIM OFF is advised for sloped landings or with a heaving deck. Turning A.TRIM off ensures that the attitude change sensed does not result in flight control input that could injure those working around the aircraft.

AFCS and A.TRIM may be disengaged for training purposes at the pilots discretion.

ATT / Attitude Hold

Attitude hold mode (ATT) is the default autopilot mode and available independently on all 3 axis. Attitude hold mode provides long term attitude stabilization. This mode is also sometimes known as **basic autopilot**. As it is the default mode, it is **not** visualized on the AFCS Status Strip.

Engagement: ATT is engaged automatically when at least one SAS is available and no other mode is selected. It is automatically engaged when disengaging any upper mode.

Reference Management: Use `Cyclic Trim Release` or `Cyclic Beep Trim` to change pitch and roll attitude.

Fly-Through Action: Hold new attitude.

* - ATT mode turn coordination is inoperative.

DSAS / Digital SAS

DSAS mode is actually a degraded sub-mode of ATT. It is engaged automatically on any axis when the A.TRIM function is inoperative. The mode provides short term attitude stabilization, however long term stabilization is not assured due to lack of auto trim.

Engagement: `A.TRIM` Toggle or automatically due to trim failure.

Reference Management:

Use `Cyclic Trim Release` or `Cyclic Beep Trim` to change pitch and roll attitude.
Use `Collective Beep Trim Left/Right` to trim the yaw pedals. Fly-Through Action: Hold new attitude.

HDG and TRK / Heading Hold

Heading hold mode will use yaw in low speed (less than 30kt) and roll and yaw at higher speeds. Track mode will use a GPS ground track. Use the APCP butterfly knob to change to TRK.

Engagement: `HDG` Toggle on the APCP.

Reference Management: `Cyclic Beep Trim Left/Right` will change heading reference.

Fly-Through Action: Return to selected heading.

VS and FPA / Vertical Speed

VS mode will acquire and maintain a vertical speed. FPA (flight path angle) will use a GPS ground reference. Use the APCP butterfly knob to change to FPA.

Engagement: `VS` Toggle on the APCP.

Reference Management: `Collective Beep Trim Up/Down` will change VS reference.

Fly-Through Action: Return to selected VS.

ALT / Altitude Hold

ALT mode will maintain the altitude at engagement.

Engagement: `ALT` Toggle on the APCP.

Reference Management: `Collective Beep Trim Up/Down` will change altitude reference.

Fly-Through Action: Return to selected altitude.

ALT.A / Altitude Acquire

ALT.A mode will use VS mode to acquire and maintain an altitude. Change the ALT.A reference on the APCP, then engage the ALT.A by pushing the knob.

Engagement: `ALT.A` Toggle on the APCP.

Reference Management: `Collective Beep Trim Up/Down` will change altitude reference.

Fly-Through Action: Return to selected altitude.

IAS / Indicated Airspeed Hold

IAS mode will acquire and maintain an selected airspeed.

Engagement: `IAS` Toggle on the APCP.

Reference Management: `Cyclic Beep Trim Up/Down` will change airspeed reference.

Fly-Through Action: Return to selected airspeed.

GA / Go-Around

GA mode will act similarly to IAS and VS. It executes for 15 seconds in cruise and 25 seconds from a hover.

Engagement: `Collective GA` press.

Reference Management:

`Cyclic Beep Trim Up/Down` will change airspeed reference.
`Collective Beep Trim Up/Down` will change the VS reference. Fly-Through Action: Return to selected airspeed and VS.

GTC / Ground Trajectory Command

GTC mode will maintain current ground-referenced speeds on longitudinal and lateral axis, as well as heading hold on the yaw axis.

Engagement: `GTC` press once.

Reference Management:

`Cyclic Beep Trim Up/Down` will change Vy reference.
`Cyclic Beep Trim Left/Right` will change Vx reference. Fly-Through Action: Update Vy and Vx references.

GTC.H / Auto Hover

GTC.H is a sub-mode of GTC and will acquire and maintain a fixed ground-referenced position.

Engagement: `GTC` press twice.

Reference Management:

`Cyclic Beep Trim Up/Down` will change the longitudinal position by approximately 1 meter.
`Cyclic Beep Trim Left/Right` will change the lateral position by approximately 1 meter. Fly-Through Action: Update to new position.

VOR / VOR Navigation

LOC mode will track a localizer in order to conduct an instrument landing.

Engagement: `CPL` key on FND, NAVD or DMAP page.

Reference Management: CRS key may be available (note: MSFS may still restrict changing LOC course), on NAVD or FND pages.

Fly-Through Action: Return to tracking LOC.

LOC / Localizer

LOC mode will track a localizer in order to conduct an instrument landing.

Engagement: `CPL` key on FND, NAVD or DMAP page.

Reference Management: CRS key may be available (note: MSFS may still restrict changing LOC course), on NAVD or FND pages.

Fly-Through Action: Return to tracking LOC.

NAV / FMS Navigation

NAV mode will track a FMS navigation source.

Engagement: `CPL` key on FND, NAVD or DMAP page.

Reference Management: None. Use FMS to change flight plan.

Fly-Through Action: Return to tracking FMS.

GS / Localizer Glideslope

GS mode will track a localizer glideslope in order to conduct an instrument landing.

Engagement: `CPL` key on FND, NAVD or DMAP page.

Reference Management: None.

Fly-Through Action: Return to tracking GS.

APP / Approach

APP mode will track an approach in order to conduct an GPS procedure landing.

Engagement: `CPL` key on FND, NAVD or DMAP page.

Reference Management: None.

Fly-Through Action: Return to tracking APP.

V.APP / Vertical Approach

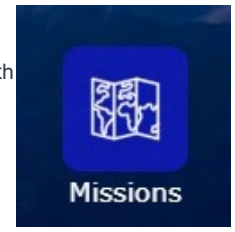
V.APP mode will track a glidepath in order to conduct an GPS procedure landing.

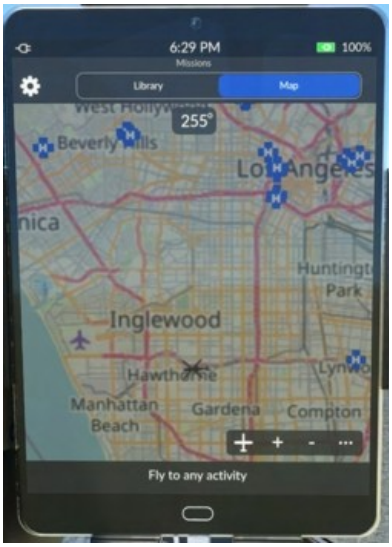
Engagement: `CPL` key on FND, NAVD or DMAP page.

Reference Management: None.

Fly-Through Action: Return to tracking G/P.

Missions (Action Pack)





The H145 includes built-in missions which run in the H145 without the use of any external programs. The Hype mission platform is available for new types of missions to be created. The mission script is a powerful language to interact with H145, data queries powered by OpenStreetMap, and the rest of MSFS. Third party packs are available with objects, missions and mission templates.

There are also multiplayer online missions where multiple users join a session and have a shared mission script which can orchestrate events, games and challenges.

There is also a tool called Scenario Editor where you may build your own simple missions using a graphical editor. This tool is available in Hype Operations Center.

Information for all missions

The HPG mission system is a platform integrated into the aircraft which enables high level orchestration of mission scenarios.
Missions are small text files which are similar to a computer program. These programs have access to the simulator, user aircraft and network, enabling engaging and realistic scenarios which make use of the functions of the aircraft variants.

Component	
Flight Simulator SDK	Access to variables and events within the sim.
HPG Aircraft SDK	Access to variables and events within the HPG aircraft.
AI Object Management	Create and manage AI objects on the ground an in the air.
OpenStreetMap Data Queries	Powerful APIs to query information about the environment.
Pilot Interfaces	Interface with the mission app or touch points within the cockpit like the radios or Rescue Track.
Sound & Text-To-Speech	Play sound files and dynamic text to speech output
Network Communication	Run missions over the network and communicate to enable multiplayer functionality
Debugger & Editor	Test missions using the included debugger for rapid development
Templates	Create mission templates which are finished by the user with a graphical editor

- When you arrive at a location, don't forget to place both engines to IDLE power. This is the cue that is used for the mission to proceed (crew to exit, etc.)
- Completed Missions are optionally logged and visible in Hype Operations Center on the Completed Missions page.

Mission App Settings

These settings are available under the gear icon and persist with each H145 variant.

Mission map update rate	Low (Performance), High This will control how often the map updates and may or may not help performance generally
Mission map resolution	Low (Performance, High The map will have larger but more blurry text and better fps on the Low setting, but will have crisper text and a higher resolution on the High setting.
Mission Flares	Higher Visibility, Realistic (No Smoke) Realistic version will only show smoke for actual marine or mountain rescue. This is configurable from the object by the mission author.
Mission Entry Markers	High Visibility, Map Icon Only Show blue mission entry markers at the site of the location marked on the mission map.
Mission Fire Quality	Low (Performance), High High quality fire has substantially more smoke. May impact fps especially for very large fires. **High quality fire requires Top Gun Maverick DLC to be installed in MSFS.**

Data Query Service	Overpass DE Mail.RU Kumi OSM RU Overpass API choices are driven by `MissionSystem.json`. Previously Mail.RU was very very fast but now we have changed the default to Overpass DE which is reliable but slower.

Authoring mission packs

Missions can be added to any other Community package.

1. Create an `hpgmission` folder within your package, and place a folder hierarchy below with your mission json files.

All contents (folders and json files) below `hpgmission` across all Community packages will be merged into the `Mission Index`.

Mission Index

There are two ways to update the aircraft mission index. You must do this before the aircraft will see new missions or updated missions.

1. `Tools\Update Mission Index.cmd`.
2. Hype Operations Center -> `Refresh Index`.
3. If your aircraft is running, the final step is to `Refresh` in the mission app to pick up the new index.

Note: Using the directly connected `Scenario Developer` will bypass the index and directly load fresh missions through the editor.

Mission Authoring Fundamentals

Mission Format

Missions are JSON files. You should use a JSON validator like jsonlint to confirm the file format is valid. Hype Operations Center also contains the Scenario Developer tool which will check formatting as you make changes.

Development workflow

Mission developers should use `Mission Editor -> Scenario Developer` in Hype Operations Center. Clicking `Connect to Mission Editor on PC` and then `Save` to reload the script in the aircraft. Once the aircraft is connected, `Save` will reload the mission automatically.

The mission may be tested in whole or part in HOC, and then eventually saved to a simple `.json` file for distribution to end users. Using the editor will bypass the mission index.

Adding Mission Packs / Developing Missions

The mission library will show all available missions which do not have a natural start point on the map. Once selected they will show as highlighted to indicate they are active.

REFRESH: Reloads the mission index from disk and then the default mission visualizes all start locations on the map and allows entry into the missions. This will also clear any in progress mission.

Adding mission packs

1. Install one or more mission packs to your Community folder, like any other livery or scenery addon. You install mission packs in their own folder, not into the H145 folders.
2. Run Community\hpg-airbus-h145\Tools\Update Mission Index.cmd. This will scan for all missions within your Community folder, and save them to a file that H145 can read. You can run this script at any time. Alternatively Scenario Editor has a button Update Mission Index which is the same function.
3. If the sim is already running, open the mission catalog and press REFRESH to reload missions from disk.

Updating the mission index

The mission index is stored in the H145 aircraft folders, so it needs to be updated with the content of all the missions that you have installed into your Community folder.

To update the mission index, run `Community\hpg-airbus-h145\Tools\Update Mission Index.cmd`. This will scan for all missions within your Community folder, and save them to a file that H145 can read. Alternatively Scenario Editor has a button **Update Mission Index** which is the same function.

You can run this script at any time, whether the simulator is open or not. If the simulator is open, you'll need to press **Refresh** in the Missions app under the Library tab.

Note: You will also need to update the mission index after each new H145 build is installed.

Developing custom missions

Use the Scenario Editor to begin with mission development. Check the H145 Mission System Documentation included with the Action Pack download. There is a sample object as well as the complete API reference for the mission system commands. Scenario Editor is suitable for all users, editing the exported code is something more advanced that only some will do.

Mission Metadata Sections

Section	•Description
title	Text for the user to identify the mission.
id	id is used to switch to the mission using `load_mission` API. Must be unique.
start_info	Determines start positions on the map.
briefing	Configure information for the user to see when the mission starts.
aircraft	Optional. If present, specifies an array of supported aircraft. `["H145"]`
applicable	Optional. If present, specifies an array of supported variants. `["CIVILCARGO", "MILITARYCARGO"]`
api_version	Not checked with v1 missions. All missions are API version 0.1.

Mission Data Sections

Each of the sections below corresponds to a store for different kinds of data. You can usually define static information up front, or call APIs to create/manipulate/remove data during the mission.

•Section	•Description
locations	Locations (lat/lon)
events	Events (Event handlers)
objects	Objects (live objects)
routes	Routes (lists of locations)
threads	Execution threads
stringTokens	Replace one string with another
userActions	Commands available for the user to interact with
icons	data-uri's representing 44x44 PNG images for use on the map
macros	Functions the mission may use (reusable code)
data	Static data

Mission execution overview

A mission is essentially a computer program. Missions are made up of sets of commands which can work with data within the simulator and on the network.
An extremely simple mission looks like this:

```
•{
  "title": "My simple mission",
  "objectives": [
    {
      "title": "Done",
      "commands": [
        {"sleep": "forever"}
      ]
    }
  ]
•}
```

`title`: The mission title is used by the user to identify it in lists and to select it from the mission library.
`objectives`: Objectives simply contain another `title` (the objective to display at the bottom of the mission app) and `commands` (a `COMMANDLIST`) which will automatically run when the mission starts.
This mission contains only one command, `{"sleep": "forever"}`, which instructs the system to begin waiting and never continue. This `COMMAND` is what prevents the mission from ending.

COMMAND

`COMMAND` is the foundational command in the mission system, executed always in a `COMMANDLIST` form. `QUERY` is used very commonly and is a a component of a command but not a command in of itself.

Commands that take a `QUERY` may use any expression from the QUERY section.

COMMANDLIST

A `COMMANDLIST` is a list of commands which execute sequentially, waiting for each command to finish before continuing.

```
[
  COMMAND1,
  COMMAND2,
  COMMAND3
```

```
]
[
  {"set_message":{"text": "hello world}},
  {"sleep": 1},
  {"#comment": "my hello world program"}
]
```

QUERY

A `QUERY` may be composed of other `QUERY` resulting in an expression that for example fetches a value and adds another value to it.
Each of the below commands are suitable as a `QUERY`, **as well as numbers and strings**.

Examples:

```
1
11.5
{"var":["L:TEST","number"]}
"hello"
{"text": "hello {0}", "params": [ QUERY, ... ]}
```

DATAQUERY

A data query is an OSM Overpass API query. Check your queries on Overpass Turbo. Optionally queries can be post-processed using `logic/groups`.

Examples:

```
“[out:json]; node({{bbox}})[man_made=silo]; out center;”
{
  query:
  "[out:json];(area({{bbox}})[amenity=hospital];area({{bbox}})[aeroway=helipad];); out center;",
  "groups": [
    {amenity: "hospital"},
    {aeroway: "helipad"}
  ],
  logic: {"intersection": 0.2}
}
```

Scenario Editor App

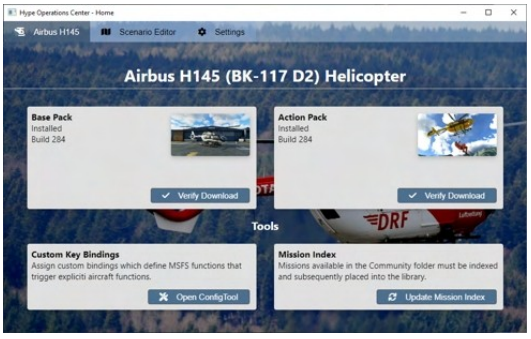
Download the Scenario Editor installer from the H145 download center. Install the program and it will create a shortcut on your Windows Start menu. You may discard the installer after it opens the program.

Settings page

You must set your Community location. MS Store and Steam installations will be automatically detected, but you may need to select a folder manually or pick for a dual installation.

H145 aircraft page

H145 base pack and action pack are detected from your Community folder. If you have an `hpg-airbus-h145` and `hpg-airbus-h145-ap` folder within your selected Community folder they will be shown here with the related tools. Note that you **must not rename the folders** right now or they won't be detected.



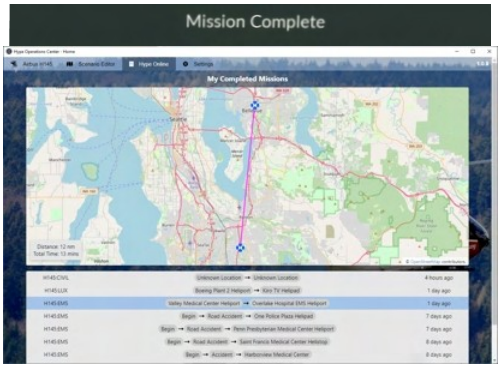
Verify Download will check the integrity of the files within your H145 installation.
Open ConfigTool will launch ConfigTool.
Update Mission Index will update your mission index within H145, enabling mission packs to be read by H145 directly. You should do this each time you

add or remove a mission pack from your Community folder, or when you export one of your own missions. The sim may be running but you must select **Refresh** within the Library in the Mission app on the H145 tablet to actually reload the updated index.

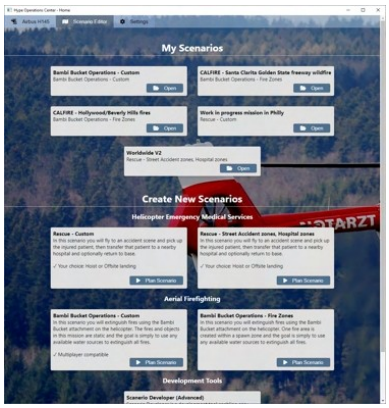
Hype Online page

Hype Online displays your past mission results. Click an item to see that specific mission, or all previously recorded missions will be displayed.

You will be asked at the end of compatible missions if you would like to log the results.



Scenario Editor Dashboard page

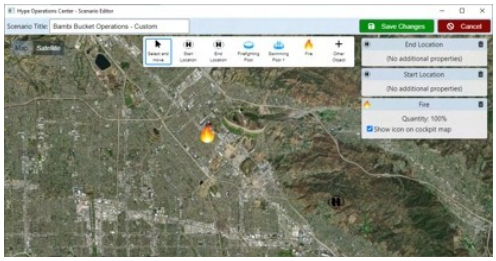


My Scenarios
All the scenarios which you have worked on are presented here for you to open them and begin work again.

Create new Scenarios
All the installed scenario templates are presented here, enabling you to create a new scenario from an existing template. Scenario templates (.scenariometa files) are loaded from the Community folder. Your scenarios are stored at `%appdata%\Hype Aircraft\User Scenarios`

Scenario Developer
At the bottom of the list is the advanced Scenario Developer tool which lets you quickly send mission code to the sim. This is the most complicated way to build scenarios and scenario templates but also the most powerful.

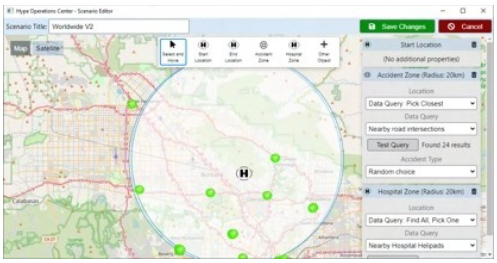
Scenario Editor

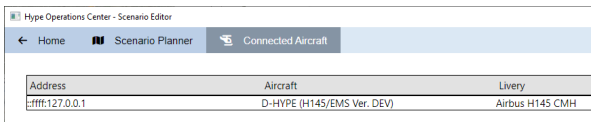


The Scenario Editor is a graphical tool that enables configuring your scenario. The key idea is to use tools from the toolbar to configure objects and zones for the mission.

Mission objects and zones are displayed on the right side when in edit mode. Click the header to center the map on that object or zone. Select the relevant options to configure the objects' appearance, behavior or the behavior of the zone that will be used for the mission.

Once you can see the Scenario Editor map **and have selected Save Changes at least once**, you are ready to connect from the H145. Using the H145 tablet mission app, visit the Library and select **Connect to Scenario Editor on my PC**. Your aircraft will be listed under Connected Aircraft and you will see a helicopter icon somewhere in the world.





Export your missions

When you are finished with a mission and want to test it without needing Scenario Editor or to share with others, select **Export for MSFS**. Choose a location like `Community\my-cool-missions\hpgmission`. (That is to say, make a new folder under Community, then an hpgmission folder within that, and place your mission json within that hpgmission folder). Then, run **Update Mission Index** either by the script in the tools folder or by using the H145 page of Scenario Editor. Once you export a mission, you can use it without using the Scenario Editor app.

Scenario Editor Workflows

Note that Scenario Editor has two modes of operation. The first mode is where you Connect from H145, and are forced into the mission you are developing. The second mode is where you export, Update Mission Index and then are able to load missions without using Scenario Editor. This is what enables end users to avoid installing Scenario Editor at all if they do not prefer it. Scenario Editor is also required for Multiplayer firefighting.

Multiplayer Missions

Only the static fire fighting scenario may be conducted with multiple players.

NOTE: OPENING PORTS TO THE INTERNET POSES A SECURITY RISK. CHANGES TO YOUR FIREWALL SHOULD BE CONDUCTED WITH KNOWLEDGE OF NETWORK SECURITY.

To begin a multiplayer firefighting session:

1. Identify your public IP address. Make a copy of the file in `hpg-airbus-h145\hpgmission\Scenario Editor Server.json`. Edit the file and replace localhost with your public IP address, and to change the title to `My Server`. Name the file `My Server.json`.

2. Share My Server.json to your friends and instruct them to place it into the `hpg-airbus-h145\hpgmission\` folder and then run **Update Mission Index**.

3. Configure your firewall to allow port 40510 to be routed to your PC. This may require both editing the firewall on your router and also the Windows firewall.

4. Start Scenario Editor and load a firefighting mission.

5. Invite other players to load their H145s at the desired location, and to select **My Server** from the mission app on the Library page.

6. Players should be visible on the map and in the aircraft list in Scenario Editor

Scenario Templates

Using Scenario Editor you may build custom scenarios with your own object placements and location choices.

Templates that use “zones” are randomly picking between like zones and data query results in those zones. The point is that you can define an area of POIs or an area that should get random locations, and this way you can create missions for a region which are randomized but still keep you within the defined area and using only the defined locations.

Templates that are “custom” are usually a single mission but everything can be defined, including placing objects at the (e.g.) accident site.

Base Pack - Luxury/Civilian

Passenger Transport

This is a very simple mission where you select a source and destination helipad and fly a leg between them.

Base Pack - Military

Crew Movement

This mission simply sends soldiers running to or from the helicopter.

Action Pack - Medical

Road Accident

Select the range for the accident, then accept the dispatch or cycle to another location.
Fly to the accident site. Either land or use hoisting if needed. After securing the patient, you'll select a nearby hospital.
Fly to the hospital and unload the patient.

Hospital Patient Transfer

Start at a hospital (or at a suitable location to load a patient, like an airport ramp).
The patient will be loaded and you will choose the hospital destination on the map.
Fly to the hospital, land at the designated site or find a suitable nearby landing site.

Aerosoft Offshore Landmarks: North Sea - Medevac

REQUIRED: 'Aerosoft Offshore Landmarks: North Sea'

There are 3 included missions (for the three main types of POIs: Ships, Platforms and Wind farm substations).
Each mission works the same. You may start up to 80nm away, and a random dispatch will be assigned (with the option to cycle the location).
Fly to the platform, land and secure the patient.
Select a nearby hospital from the mission map (it may be very far away)
Fly to the hospital and unload the patient.

HEMS Mission Generator

Check out this awesome third party mission generator .

Action Pack - Firefighter

Nearby Woodland fire

You'll be dispatched to extinguish several small fires nearby. Use your bambi bucket and find a water source.
NOTE: If you land you will be able to spawn (request) a portable water source (firefighting pool) for your use.

Firefighting – Start Fire (Anywhere)

Synopsis

In this mission you and a group of friends may fight one or more constantly evolving fires.
Wildfires are a serious threat to lives, property, and the environment. They can spread rapidly and unpredictably, fueled by dry vegetation, strong winds, and human activity. To combat these fires, aerial firefighting is a vital and effective strategy that involves dropping water, or fire retardants from various fixed and rotary-wing aircraft.
Helicopters are especially well suited for aerial firefighting, as they can maneuver in tight spaces, hover over hot spots, and access remote areas. They can also use a variety of tools, such as the bambi bucket, a collapsible container that can scoop water from lakes or rivers, or the heli-torch, a device that can ignite controlled fires to create firebreaks. Helicopters can also work in coordination with ground crews, providing them with air support, reconnaissance, and transportation.

Mission Key Features:

1. Firefighting with realistic fire spread dynamics based on wind and terrain
2. Single player or multi-player online
3. Bambi Bucket firefighting or Heli-Torch operations to proactively reduce fuel on the ground
4. Ground Crews that inhibit fire spreading in their area of influence

Requirements

Type	Download
Aircraft **Required**	1. H145 Base Pack and 2. H145 Action Pack Minimum Version Required: **H145 Build 435**.

Installation

Install H145 and H145 Action Pack. Follow the instructions here

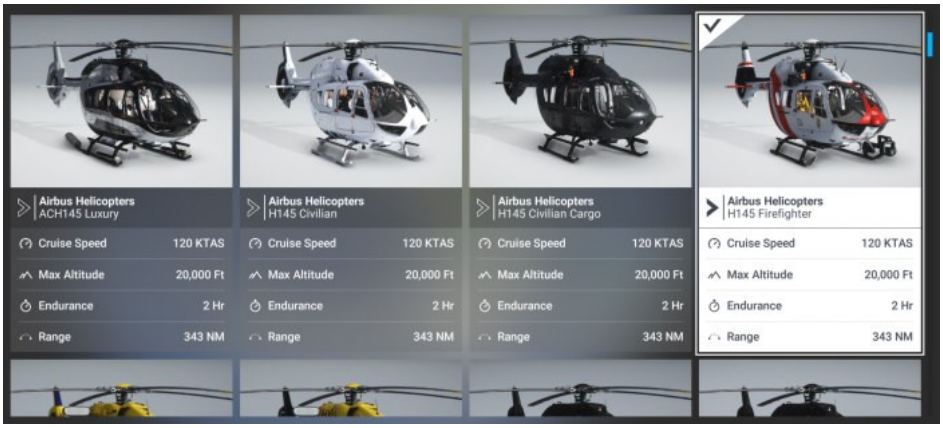
```
Community\
          hpg-airbus-h145
          hpg-airbus-h145-ap
```

Setup

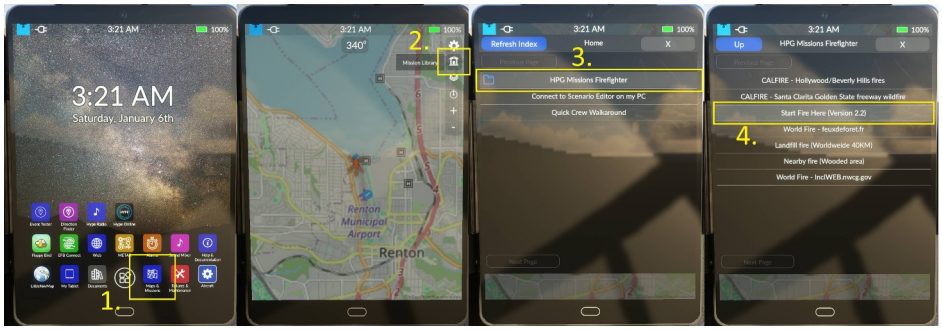
This mission is pre-installed and does not require any mission files to be installed.
You need to prepare a binding on your controller for `TOGGLE YAW DAMPER`. This key will be used in two ways:

1. `Click`: When you are over land and the cargo object is touching the ground, **clicking** will attach or detach the object.
2. `Hold`: When you are in the air or over water, **holding down** the button will open the valve and release the water or diesel.

Start the missions



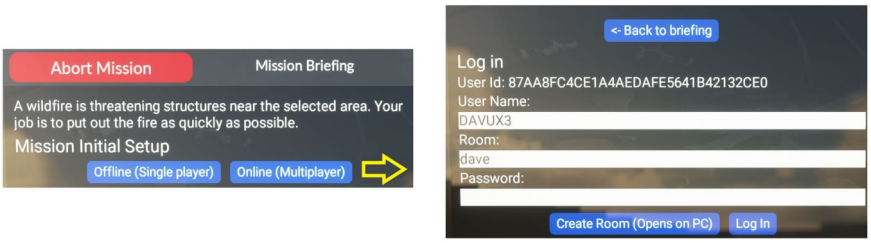
Fly this mission using the **Firefighter** variant of H145.



1. Select the Missions app on the tablet.
2. Select the Library section of the mission app. If you have a mission already in progress, you'll see a red Abort Mission button instead of the library.
3. Aborting the mission will go back to the library. Navigate into the `HPG Missions Firefighter` folder
4. Select the `Start Fire Here v2.x` mission.

You'll now see the mission setup

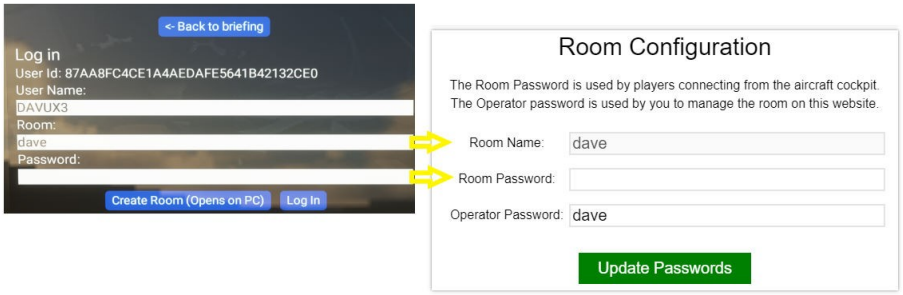
Choose Online or Offline game play



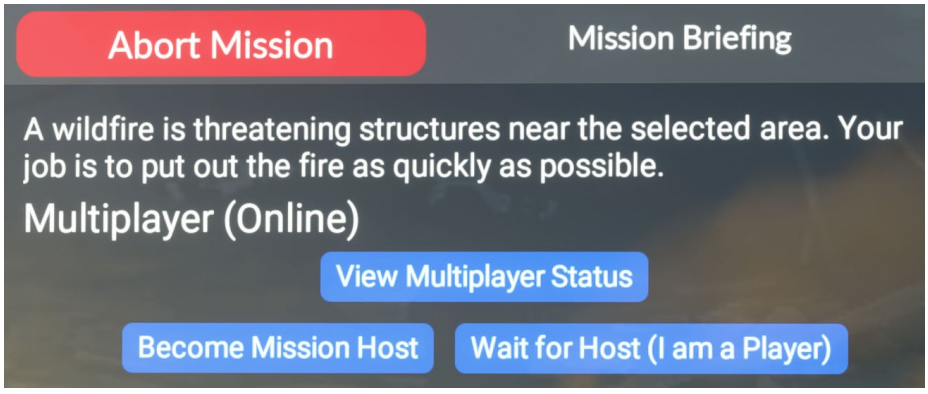
After you launch the mission you will need to decide if you want to play offline or if you will connect to the server and use a room there. Offline game play is single player only.
You can create and manage rooms at any time, by visiting Hype Multiplayer Management

When you choose to play online:

- `UserName`: This is arbitrary and used as a friendly name to identify yourself. Defaults to your MSFS ATC call sign.
- `Room`: This is the room name selected when creating a room.
- `Password`: This is selected in the `Security` area of the management website.



Choose to become the mission host or continue as a player

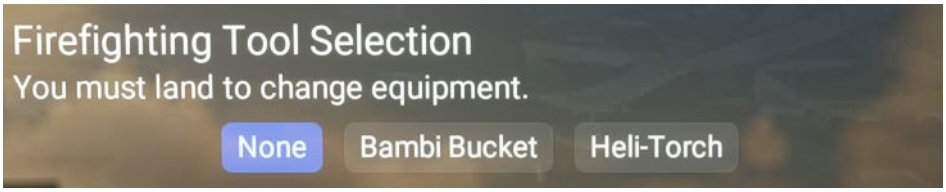


One player must become the host. The host aircraft will conduct fire spreading calculations and manage the fires. **The host must not disconnect or abort the mission**.

- `Become Mission Host`: You will become the host and have access to fire controls
- `Wait for Host (I am a Player)`: You will still need to wait for the host to join

NOTE: When playing offline you will not be asked as you must always be the host when playing alone.

Choose your firefighting attachment



Now that you have set the mission up, you may land your aircraft and choose to attach the bambi bucket or heli-torch.
The object will appear in front of your aircraft after selecting the option in the mission.

Start a fire

The fire isn't started yet, you won't see any fire icons on the tablet or DMAP just yet.
You can start a fire multiple ways:

1. Using the web client, click on the map and then click `Start a fire`.
2. Using the aircraft, click on the map and then click `Start fire`
3. Using the aircraft heli-torch, hold down `TOGGLE YAW DAMPER` to start new fires. It's best to use ground crews to manage these fires instead of allowing them to spread.

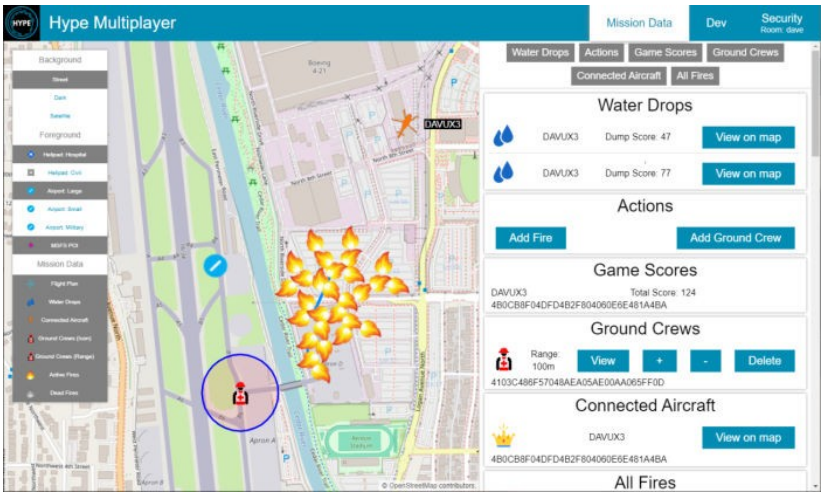
Flight the fires

Once you've attached the bambi bucket, you're ready to hover over a water source and fill up.
Fill up your bucket:

1. Enter a hover and slowly descend to about 20 feet
2. You'll see the bucket start to fill a) on the tablet notification bar and b) on the VMS:DATA display on the MFD.
3. If you are too heavy, you can bleed off water by holding down the `TOGGLE YAW DAMPER` button to bleed off some water. It's recommended to use no more than 30% fuel while doing bambi operations.

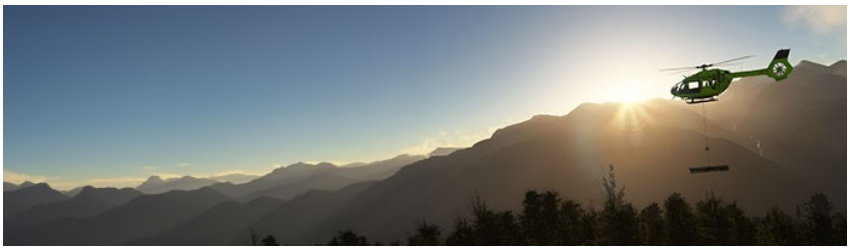
Now you may fly over the fire (don't get too close as it can damage your aircraft) and hold down `TOGGLE YAW DAMPER` for the entire time you want to drop. You can hold down the button just long enough and reserve water to drop again.

Managing the mission



Using the web client you may manage the mission by establishing and removing ground crews, monitoring activity and creating more fires.

Cargo - Heli Logging Revelstoke



Synopsis

In your mission, you will be flying to various locations related to logging activities, including the logging decks and the processing mill, to deliver tools & supplies and move logs down to the mill. This will require precision flying and landing skills, as well as the ability to navigate through challenging terrain.

Helicopters are often used for logging operations because they can access remote areas that are difficult to reach by road or other means of transportation. They can also transport heavy loads of logs and supplies to and from the logging sites with ease.

Revelstoke is a city in southeastern British Columbia, Canada, located 641 kilometers east of Vancouver and 415 kilometers west of Calgary, Alberta. It is situated on the banks of the Columbia River, just south of the Revelstoke Dam and near its confluence with the Illecillewaet River. The area is known for its natural beauty, with the Monashee and Selkirk Mountain Ranges surrounding it.

Mission Key Features:

1. `Tool Delivery`: Deliver supplies & tools to logging deck helipads.
2. `Logging Transport`: Retrieve Logs from the logging sites and return them to the mill.
3. Mission Tasking responsive to scenery detection. If you don't have the Lakeview expansion installed, then you won't be tasked to area.
4. Single player offline & Multiplayer online. Leaderboards and moving map available on the web.

Requirements

Type	Download
Aircraft **Required**	1. H145 Base Pack and 2. H145 Action Pack Minimum Version Required: **H145 Build 435**.
Mission **Required**	Heli Logging Revelstoke by BlueEcko
Scenery **Required**	Heli Logging Revelstoke BC. CA by WingBoss
Scenery Optional (Recommended)	Heli Logging Revelstoke BC. CA - Lake View Addition by WingBoss
Scenery Optional (Recommended)	Glacier Helicopters Base - Revelstoke BC, CA by WingBoss

Installation

1. Install H145 and H145 Action Pack. Follow the instructions here

```
Community\
          hpg-airbus-h145
          hpg-airbus-h145-ap
```

2. Download the scenery (as well as the dependencies) and place them into your Community folder.

```
Community\
            revelstoke-lakeview-scenery
            revelstoke-logging1-scenery
            revelstoke-mill1-scenery
            revelstoke-glacierhp-scenery
            wookie042-bush-and-backcountry-library
            human-library-animated
            esd-modellib-eolib
```

3. Download the mission and place the folder into your Community folder.

```
Community\
          hpg-airbus-h145-z-blueecko-missions
```

4. **You must now update the aircraft mission index** This can be done using one of two methods:

Method 1:

1. Open Hype Operations Center
2. Visit the H145 aircraft page
3. On the **Indexed Missions** section, select `Update`.
4. The mission will now be visible in the list below.
5. If your aircraft is currently running, you must **Refresh** the mission index by visiting `Tablet -> Aircraft -> Missions -> Library` and selecting **Refresh**.

Method 2:

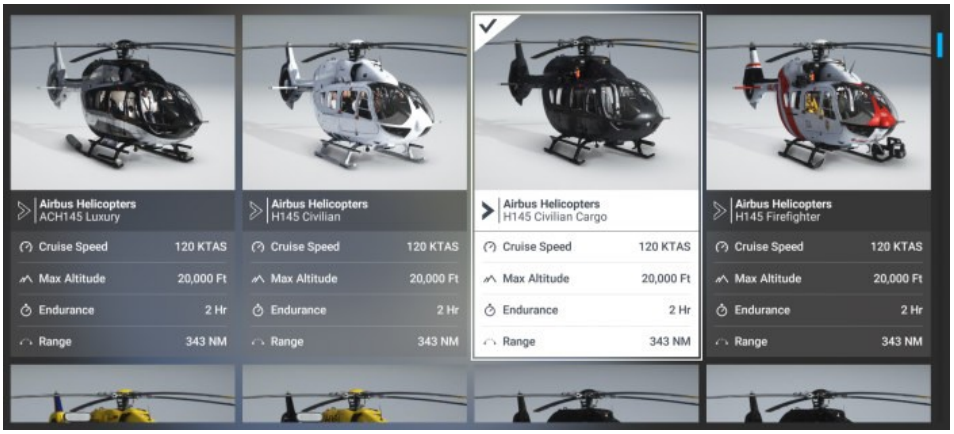
1. Run the script at `Community\hpg-airbus-h145\Tools\Upate Mission Index.cmd`
2. The mission file will be mentioned in the list and you will see a success indication at the bottom of the window. You may close the window.
3. If your aircraft is currently running, you must **Refresh** the mission index by visiting `Tablet -> Aircraft -> Missions -> Library` and selecting **Refresh**

Setup

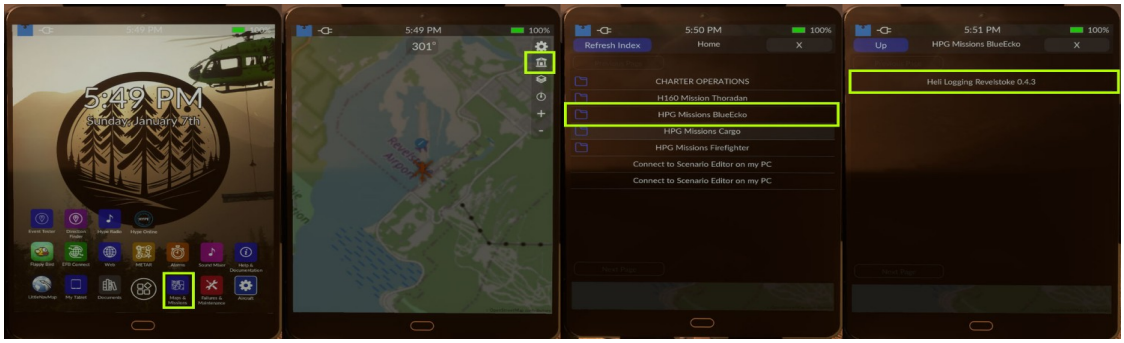
You need to prepare a binding on your controller for `TOGGLE YAW DAMPER`:

- `Click`: Clicking the button will attach or detach the cargo object, when you are in range of the object. Check the tablet status bar for confirmation

Start the mission

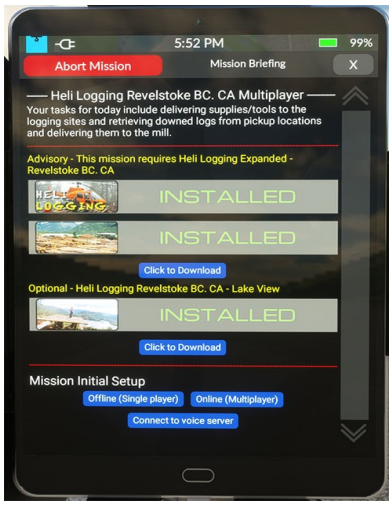


Fly this mission using the **Cargo** variant (**Civil Cargo** or **Military Cargo**) of H145. You may spawn at the mill or perhaps at the nearby Revelstoke Airport (CYRV).



1. Select the `Missions` app on the tablet.
2. Select the Library section of the mission app. If you have a mission already in progress, you'll see a red Abort Mission button instead of the library. Aborting the mission will go back to the library.
3. Navigate into the `HPG Missions BlueEcko` folder
4. Select the `Heli Logging Revelstoke x.x.x` mission.

Mission Setup



After you launch the mission you will see the following screen. The dynamic scenery detection will show `Not Installed` if you don't have the required scenery, and will check for the add-on lakeview scenery and enable additional sites if it's detected.

NOTE: If you follow the link, download and install the scenery your sim will need to be restarted for it to detect and load the scenery properly.

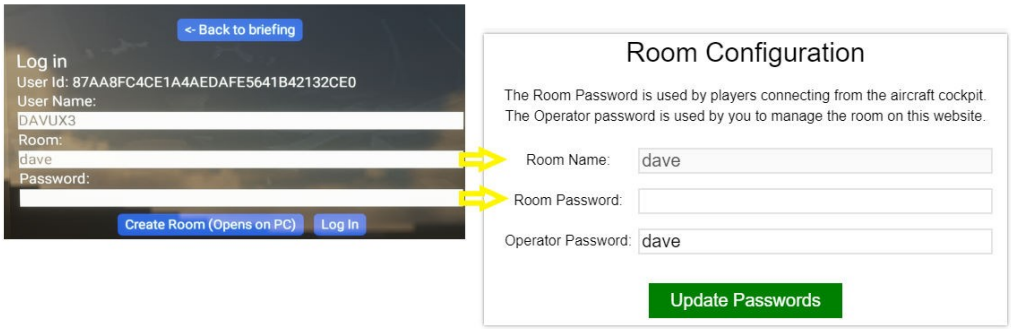
Once Checks are complete, you will need to decide if you want to play offline or if you will connect to the server and use a room there. Offline game play is single player only.

You can create and manage rooms at any time, by visiting Hype Multiplayer Management.

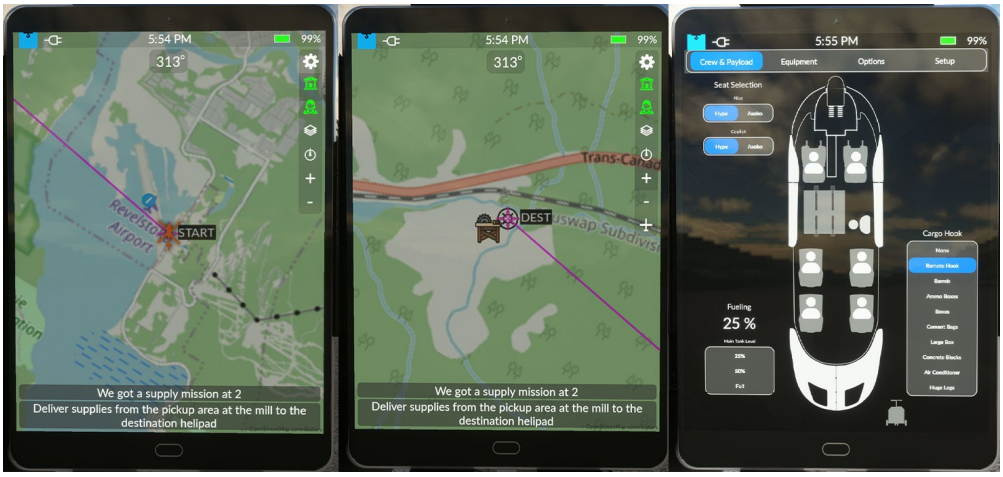
When you choose to play online:

`UserName`: This is arbitrary and used as a friendly name to identify yourself. Defaults to your MSFS ATC call sign.

- `Room`: This is the room name selected when creating a room.
- `Password`: This is selected in the `Security` area of the management website.



Proceeding with the mission



You'll need to select either `Offline (Single Player)` or `Online (Multiplayer)` and then mission tasking will automatically begin.

You should configure your helicopter before flight, you will want to assure "Remote HooK" is enabled and set your fuel accordingly. The Huge Logs Bundle is heavy, and you will not be able to lift with fuel weights too high. I recommend 25%.

Follow your track line to the cargo and complete your first run! After the cargo is released you will have one point added to your score and a new task will be randomly generated.

Now get to work. They're waiting for you.

HEMS – Random Rescue Missions (Anywhere)

Download

HPG Airbus H145 Action pack: HEMS Random and Anywhere mission

Action Pack - Offshore

Nearby water rescue

This mission can be started from the library and will create a random location off the coast (assuming you are close enough to any coastline).
You fly to the location to find either a sinking boat or a crashed airplane with life raft.
Rescue by hoist the survivor(s) (there will be 1 or 2). Deliver them to a nearby airport.

Action Pack – Cargo

Pick up Sling Cargo Here

This mission enables you to pick a source and destination location, and transfer 4 packages using your sling capability.
Start the mission on the mission may, by selecting the pick-up location and then entering the mission.
You'll be immediately asked to set the destination location on the map.
After accepting the destination, the packages will appear at the pick-up location.
Move the packages to the destination. Use the `TOGGLE YAW DAMPER` (Primary Action) to connect and disconnect from the sling cargo.

Multiplayer

Earthquake! 2.0

The Earthquake mission is a combined multiplayer activity. Rounds are 6 hours long. One player joins the mission and clicks "Setup". This loads random locations and saves them for the round.

All players connect, join the same MSFS multiplayer server, and then are able to see the same smoke/fire, EMS incidents, and electrical substations needing hardware.

The mission briefing explains the POI types. POIs will be shown with a green checkmark when completed.

Three variants are supported:

- Firefighter - Your job is to put out the various woodland fires around the city
- Medical - Your job is to rescue people from various accidents around the city
- Civilian Cargo - Your job is to deliver sling cargo to electrical substations, enabling repair of the electrical network in the city

After 6 hours the round will close, and all players must disconnect. Upon reconnecting, Setup may be run by one player and another round may begin immediately.

Working with SimVars and L:Vars

Information from the simulator and the HPG aircraft is available to be accessed at any time.

Aircraft Simulation Variables (A:Vars)

`A:Vars` indicate data from the MSFS simulation, including the environment and default systems.

A few popular examples include:

Variable	Description
`PLANE ALTITUDE, feet`	Altitude above the earth.
`GPS GROUND SPEED, knots`	Speed over the ground
`RADIO HEIGHT, feet`	Radio Altitude (height over the ground)

View The full list of simulation variables in MSFS. Each variable on this page may be prefixed with `A:`, but this is not necessary as it is the default variable prefix.

Examples:

```
{"set_message":{"text":"my variable is: {0}", "params": [
  {"var":["PLANE ALTITUDE", "feet"]}
]}}
```

Note that `A:Vars` are **usually** read-only. Some `A:Vars` however can be written directly, like the transponder:

Examples:

```
{"set":{"var":["TRANSPONDER STATE:1", "enum"]}, "value": 1}
```

Local Variables (L:Vars)

`L:Vars` are much more open than A:Vars.

- Simply writing to any L:Var will create it, if it didn't previously exist.
- Developers are free to use L:Vars for anything they want.

You have access to the HPG Aircraft SDK `L:Vars`, as well as being able to write your own `L:Vars` for your own use. Note that most HPG aircraft SDK L:vars are also read-only.

Examples:

```
{"set_message":{"text":"my variable is: {0}", "params": [
  {"var":["L:H145_SDK_VARIANT_ID", "number"]}
]}}
```

Examples:

```
{"set":{"var":["L:MY_MISSION_VAR", "number"]}, "value": 99}
```

Sending and Receiving Events

You'll send a lot of events, which will trigger various actions within the simulator and the HPG aircraft. You won't receive as many events, receiving events as a special case when the system needs to let you know something happened.

Sending Events

There are two main types of events, `K:` events and `H:` events. `K:` (for keyboard) events are essentially the same control bindings which you may bind in the MSFS preferences. `H:` (for HTML) events are events which are defined by the developer, which means the list is the HPG Aircraft SDK.

Event Prefix	List	Provider
`K:` (Keyboard)	Simulation Events IDs	Microsoft
`H:` (Html)	HPG H145 SDK Events	HPG

Examples:

```
{"trigger": "K:TOGGLE_NAV_LIGHTS"}

{"trigger": "H:H145_SDK_OH_PITOT_1_TOGGLE"}
```

If you need to set the value of a `K:` event, use `set` instead.

Notable received events

This is a list of common events (not exhaustive) which you may respond to:

Event Name	Description
`ON_MISSION_ABORTING`	Called just before unloading the mission. You should do critical work here only.
`H145_SDK_CARGO_COUPLE_FAILED`	Called when the cargo couple button was pressed but no object could be coupled or uncoupled.
`H145_SDK_CARGO_COUPLE_ACTIVATED`	Called when the cargo couple button was pressed and an object was subsequently coupled successfully.
`H145_SDK_CARGO_DECOUPLE_ACTIVATED`	Called when the cargo couple button was pressed and an object was subsequently uncoupled successfully.
`H145_SDK_HELITOCH_IGNITE_ACTIVATED`	Called when the Heli-Torch is requesting to create a new fire immediately. (This will be repeated but at the correct rate for you to create fires)
`H145_SDK_BAMBI_BUCKET_FILL_ACTIVATED`	Called when the Bambi bucket begins filling.
`H145_SDK_BAMBI_BUCKET_DUMP_ACTIVATED`	Called when the Bambi bucket valve opens.
`H145_SDK_BAMBI_BUCKET_VALVE_CLOSED`	Called when the Bambi bucket valve closes.

Receiving events

You can be advised of `H:` events by creating an event handler. You can also define the handler up front in the `events` table. Note that you may not have more than one event handler for a given event, and the last writer will win when setting the second time.

Example:

```
{"create_event_handler": "H145_SDK_BAMBI_BUCKET_DUMP_ACTIVATED", "commands":[
  {"set_message":{"text":"bambi dumped!"}}
]}
```

Note: Working in the cockpit (flipping switches) will actually NOT generate the 'expected' SDK event for most controls.

Working with Data

Within your mission you will need to access and store data.

The lowest form of storage is the `param`. Params are a collection of key/value pairs which are associated with your execution context. When you first start a mission, you are running on the main objective thread and `params` will be empty. If you get some data and need to refer to it, you can set that data to a param like `my_param`. The key `my_param` is used to reference some data which can then be accessed or written to at any time. When you call a macro using `call_macro`, the params will be passed explicitly, which means by default you will not get a copy of any existing params unless you pass each one by name. The point of params is that you have local exclusive space to manage your data.

The second lowest form is `locals`. Locals are available across your mission, but work otherwise exactly like params. `locals` are shared data which is also available in the `briefing` and `dispatch` where params are not available. You will use locals very often and they are also useful for debugging so you may example an otherwise "inaccessible" param value. `locals` are still within the mission platform and do not incur any extra costs or time when reading/writing.

If you need to store large amounts of data between missions, use the `table` API. Tables are key/value pairs loaded and saved to disk, and you may store as much data as you need within a table.

If you have static information that you define up front with your mission, the `static` command will retrieve any path from the `data` section of your mission. This is a great place to define high level configuration options so that somebody can adjust your mission.

Avoid using `global` data, a table may be used instead for any persisted data you need.

You can read and store data in `L:Vars` which you then also view with the MSFS Behaviors window, however `L:Vars` incur a small performance penalty over `params` and `locals`.

String interpretation

`local` and `static` data as well as `params` (no prefix) can be referenced using strings and braces.

In the cases below `my_id` exists in `params`, `locals` or `static` data respectively.

Examples:

```
"object{my_id}"

"object{local:my_id}"
"object{static:my_id}"
```

Data storage options

- `param` - isolated to each macro+child threads and the objective thread+child threads.
- `local` - shared for the mission and between missions when using `reload_mission` and `load_mission`.
- `global` - read a global variable which is persisted. **Avoid** storing data here when you can use a table instead!
- `table` - read table data which is persisted (a group of keys may be stored under a table name)
- `static` - read static data from the mission `data` table, such as configuration settings
- `location` read locations from the mission `locations` table
- `var` - `L:Vars`: global to MSFS, visible in the behaviors window and `A:Vars`: MSFS Aircraft SDK variables.

Data Tables

Data tables are used to store and retrieve information. Data tables are optionally stored to disk so you may have persisted information.

Data tables are identified by their `name`, which will then be used as a filename on disk. Each table is a JSON object with keys.

You can access your tables on disk at `%LocalAppData%\Packages\Microsoft.FlightSimulator_8wekyb3d8bbwe\LocalState\packages\hpg-airbus-h145-ap\work`

Table API

To begin, `open_table` with the table `name` that you will use. This step will either load a table from disk, or create a new empty one for you. If you want to remove everything from a table, you can `clear_table` anytime after it is opened.

After that, you can use `set` commands to set a specific `key` in the table. Each `key` in the table is unique, and you can store any data within that key. Writing to the same key twice will remove the previous data, and it wll be lost if it wasn't previously copied.

You can read data from table by using `table` with the specified `key`

Let's see this in action, we'll open a table named `test1` and set `item1` to `99` and then save it. After that, print `item1` to the screen using `set_message`.

```
{"open_table": "test1"}

{"set": {"table":"test1", "key": "item1"}, "value": 99},

{"save_table": "test1"},

{"set_message":{"text":"The contents of the item1 key are: {0}", "params": [
  {"table": "test1", "key": "item1"}
]}}
```

We will see `The contents of the item1 key are: 99` as expected.
Now, remove the lines which save and modify the table:

```
{"open_table": "test1"}

{"set_message":{"text":"The contents of the item1 key are: {0}", "params": [
  {"table": "test1", "key": "item1"}
]}}
```

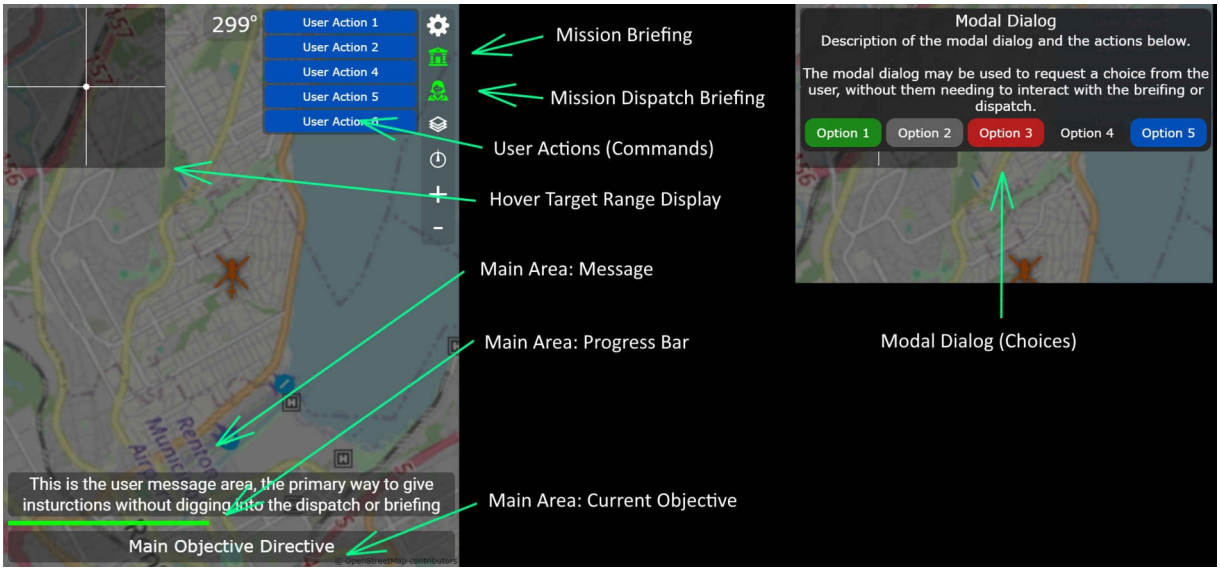
And you'll see the result is unchanged as the table was loaded from disk.
We can use this persistance to save settings, logs of the mission or whatever else we want to. Try to avoid calling `save_table` too rapidly as it is hitting the disk.

Interacting with the user

The mission app is available on the tablet and is used to start missions as well as interact with the mission throughout the duration of its execution.

Main display Widgets

The Main Display widgets are available on top of the mission map. You can configure the elements as needed at anytime.



Element	Description
Mission Briefing	Opens the briefing page (see below)
Mission Dispatch Briefing	Opens the briefing page (see below)
User Actions	Generic commands which the user may also bind to a button on their controller and activate without the tablet open.
Hover Target Display	Provides guidance cross-hair UI to a specific location (target)
Message	Provides text for the user to guide them at this stage of the mission
Progress Bar	Provides status information regarding the current objective or overall mission
Objective Title	Provides information regarding the current main objective
Modal Dialog	Requests the user to make a choice

Each widget is covered in the test program below.

To start you can use `set_message`.

```
{"set_message": "hello world"}
```

And this message will be visible at the bottom of the mission app.

Briefing & Dispatch Widgets

These widgets are available for the `dispatch` and `briefing` pages. The `briefing` is static and defined with the mission, while the `dispatch` is dynamic and may be changed at anytime.

Element	Description
#comment	human-readable description, no effect
title	Display large heading text
text	Display text with various formatting
textbox	Allow the user to type free-form text
buttonbar	Create a row of buttons
buttonbar.button	Create clickable button (with `select_condition`)
link	Create clickable link
image	Draw an image
describe_icon	Draw a small image with text to the right of it
iframe	Display an IFrame
slider	Display an slider which the user may use to pick from a range of values
progressbar	Display a range of values

Each widget has `show_condition` and `disabled_condition`:

- `show_condition`: Optional. QUERY which determines if the element should be visible.
- `disabled_condition`: Optional. QUERY which determines if the element should be non-interactive and visibly disabled.
- `select_condition`: Optional. QUERY which determines if the element should be visually selected. (buttonbar.button only)

Map Widgets

Map widgets enable you to place elements onto the mission map (and also available on DMAP/NAVD on the aircraft MFDs).
You can add points with icon and/or text, and draw lines. You can also draw a precise range circle around a point.

Examples:

```
{"copy_location":{"bearing":330,"dist":500},"to":"P1"},
{"copy_location":{"bearing":30,"dist":500},"to":"P2"},
{"copy_location":{"bearing":120,"dist":500},"to":"P3"},
{"copy_location":{"bearing":240,"dist":500},"to":"P4"},
{"set_map":{"add":{"line":{ "points":["P1","P2","P3","P4", "P1"], "stroke":{"color":"#4287f5", "width":4}}}}},
{"set_map":{"add":{"point":{"location":"P1","text":"waypoint text"}}}},
{"set_map":{"add":{"point":{"location":"P4","icon":"ki_helipad"}}}},
```

You may monitor the user's map selection by looking at the location `$MISSION_SELECTED_POI_LOCATION` and `L:MISSION_SELECTED_POI_TYPE` (zero if not selected).

Widget Test Program

This program has an example of each of the widgets.

```
{

  "title": "Widget Test Program",
  "briefing": [
    {"title": "Briefing Title"},
    {"text": "pagraph text"},
    {"text": "pagraph text with params {0} {1}", "params":[99, 100]},
    {"text": "red text", "color": "red"},
    {"text": "centered text", "align": "center"},
    {"text": "Text input: (value: {0})", "params": [{"local": "my_textbox"}]},
    {"textbox": "my_textbox"},

    {"text": "Button bar (button clicked count={0})", "params":[{"local":"test_local"}]},
    {"buttonbar":[
      {"title":"add", "commands":[
        {"set": {"local":"test_local"}, "value": {"add":[{"local":"test_local"}, 1]}}
      ]},
      {"title":"subtract", "commands":[
        {"set": {"local":"test_local"}, "value": {"add":[{"local":"test_local"}, -1]}}
      ]},
      {
        "title":"my button (with conditions)",
        "commands":[
          {"set": {"local":"test_local"}, "value": {"add":[{"local":"test_local"}, 1]}}
        ],
        "show_condition":{"require":{"local":"test_local"}, "gt":0},
        "select_condition":{"require":{"local":"test_local"}, "eq":2},
        "disabled_condition":{"require":{"local":"test_local"}, "eq":3}
      }
    ]},

    {"link":"my link", "commands": [
      {"set": {"local":"test_local"}, "value": {"add":[{"local":"test_local"}, 1]}}
    ]},

    {"image": "https://www.hypeperformancegroup.com/cdn/shop/files/HPG_Solid_Transparent_180x.png"},
    {"describe_icon": "fire_station", "description": "icon description here"},
    {"iframe": "https://davux.com/docs", "height": 400},
    {"slider":{ "min":0, "max":100, "var":["L:TEST","number"] }},
    {"progressbar":{ "min":0, "max":100, "color":"red", "var":["L:TEST","number"] }},
    {"#comment":"you may add comments as needed"}
  ],
  "objectives": [
    {
      "title": "Done",
      "commands": [
        {"set": {"local":"test_local"}, "value":0},
        {"set":{"var":["L:TEST","number"]},"value":45},
        {"set_modal":{}},
        {"create_user_action": {
          "id": "user_action_1",
          "title": "User Action 1",
          "click_commands": [ {"destroy_user_action":"user_action_1"} ]
        }},

        {"create_user_action": {
          "id": "user_action_2",
          "title": "User Action 2",
          "click_commands": [ {"destroy_user_action":"user_action_2"} ]
        }},
        {"create_user_action": {
          "id": "user_action_3",
          "title": "User Action 3",
          "click_commands": [ {"destroy_user_action":"user_action_3"} ]
        }},
        {"create_user_action": {
          "id": "user_action_3",
          "title": "User Action 4",
          "click_commands": [ {"destroy_user_action":"user_action_4"} ]
        }},
```

```
        {"create_user_action": {
          "id": "user_action_5",
          "title": "User Action 5",
          "click_commands": [ {"destroy_user_action":"user_action_5"} ]
        }},
        {"create_user_action": {
          "id": "user_action_6",
          "title": "User Action 6",
          "click_commands": [ {"destroy_user_action":"user_action_6"} ]
        }},

        {"set_hover_display":{"target":"$USER","range":0.1}},
        {"set_message": "This is the user message area, the primary way to give insturctions without digging into the dispatch or briefing"},
        {"set_objective_title":"Main Objective Directive"},
        {"set_progressbar":{"min":0,"max":100,"var":["L:TEST","number"], "color":"green"}},
        {"set_dispatch":[
          {"text":"Example dispatch"}
        ]},

        {"set_modal": {
          "title": "Modal Dialog",
          "text": "Description of the modal dialog and the actions below.<br /><br /> The modal dialog may be used to request a choice from the
user, without them needing to interact with the breifing or dispatch.",
          "options": [
            {"text": "Option 1", "style": "primary", "commands": [
              {"#comment": "use a sleep 0 here to make sure button with empty list still executes"},
              {"sleep": 0}
            ]},
            {"text": "Option 2", "style": "secondary", "commands": [
              {"#comment": "use a sleep 0 here to make sure button with empty list still executes"},
              {"sleep": 0}
            ]},
            {"text": "Option 3", "style": "danger", "commands": [
              {"#comment": "use a sleep 0 here to make sure button with empty list still executes"},
              {"sleep": 0}
            ]},
            {"text": "Option 4", "style": "subtle", "commands": [
              {"#comment": "use a sleep 0 here to make sure button with empty list still executes"},
              {"sleep": 0}
            ]},
            {"text": "Option 5", "style": "", "commands": [
              {"#comment": "use a sleep 0 here to make sure button with empty list still executes"},
              {"sleep": 0}
            ]}
          ]
        }},

        {"copy_location":{"bearing":330,"dist":500},"to":"P1"},
        {"copy_location":{"bearing":30,"dist":500},"to":"P2"},
        {"copy_location":{"bearing":120,"dist":500},"to":"P3"},
        {"copy_location":{"bearing":240,"dist":500},"to":"P4"},
        {"set_map":{"add":{"line":{ "points":["P1","P2","P3","P4", "P1"], "stroke":{"color":"#4287f5", "width":4}}}}},
        {"set_map":{"add":{"point":{"location":"P1","text":"waypoint text"}}}},
        {"set_map":{"add":{"point":{"location":"P4","icon":"fire_station"}}}},
        {"sleep": "forever"}
      ]
    }
  ],
  "icons": {
    "fire_station":
"
5Vl9bFNVFL+v3x/rttZ2a90G2dhWBqxsYf0g7IvNMI0xRhPm0D9IMHwaE4mJaICYgKiQoJEMEoMhUQOiMZE/JSYOBQywoWMYIpuyBEG7ZRMGW9d1a6/
n3L731te+0q5sqPgLP86955x77n3n3Xc/
OpIOaMvTVmq323fydUIUCoWLL4pQ8BJBkTU17sejVQC2uEHdtLHFQsEqNYACW6QBq9V6ck97CYsf1cSB47i32Ejy8+1HQIieTOkf8JO6x8wERoJVKcCNYnO+mgDZTjleIhKMceDEFMFDLx22O9a
GNBoSy75ypwOGsIf5ME8eiyht1IRCJJZrsrOrePM0hHTK8V4Plz7MZvMFENRmsx2IapIDeiX0yq1lTPKUIj8//
zMQsQ4SFhQUbAIZzUZOTs4tEJhzWdbV1f0CUpo6yCuNJ29ikDgLuW32+t4XyrGQOKeCxDnYlE0CDh3punh2K/Yf0eCQpyFx1n57h+j/
DJJAIMzqXEgyZKnzebenImizBSEeCynIGcPnW+6GORL2er05vCpz1NbWLoBg4ovc+Op8saxUKic8Ho8ayumhqakJvwsxgMD2Y2UUV5WGVZYEm0qlGq2urpakSsyL07nwxvDwUK7BYBjTarWTvDp
tDA0NZefm5gZNpmx/T88l6WJYXl5+SG6FTAf4jba1tTXy1eQTLqRQuOE5W1NxTKW2800kSBq4yJa3GfL0eSzr6+p98bqGqqrE5QeQNLB/
cADXIJERnZJ813l2K8pYfefFLr6FFEkD4+pGYUhhAwQCr/FWs0i0oV5cBWWQNDDEXNdVsbhKOTZFxi1WYvxkiBOIdtRDAcZMvmYN/j8wmUyXYG3+lK/
eP9xu92IQ+IYYYY0Ib9q0+f4WItgwOkCwgLs+KKYGo0rsoKio6D2QMwOsakXxCxEuQMhYnUajCe3YsVMH5dSwWCxfgpAEQLLAEWlggcXFxW+AlEdDQ6MFFiDZhulQp9MFYPTi+swm+
+rVrdu6u396ua+vr6CsrOwms8wQQttVq1raDh5sP8OrRWDvmULSNvlacZ9gqYgB9sp0kxrNBFQ0WE4G2JMGtYFAvlAFivGSBoZCK8pUAOcv+KIkcDzQKADLIo1Go6QeQwGx5bnL8YN/
ebDtiI9GNdC/4BmmhJuafmqO4j7DILZFJA8M/27CZcPy7K/kwycdxJwHAcHiI1piHb1LQvUmYi/
oRmchRvqBw0YVUQTDsM9ZCFFHXZXXJoi6cwwKHFHA6RG0QgxJ4HigEf87iYHT5IusJZR5KQuJMQyLEjbuXr9BxavI9caVz6NuxGi8yqsESNqmmhXMWWVhB0EGlUYzjhLWayYfDni9PnVzc7N4QP
xXYu/efXqHw7EdjtZ4f8JXwgjH7f6qquon0OcfB5w4lKWlpdvgonAXqtODNCppWUXiWghHnj6n01kL5QeHXbt2qwsLCzfo9XqcxAmDqnBl0ethD31z/wL61WkXNT+iTvBBwuB/
htuQF8qzj5qaGg4ysw5OPUNQlR2AwBNnloonJuT6V+bJ+sUSjnFdLpdrCZQzB2RSAaemZwwGY8pBxrJgvo4eO7mInuv30Nd2z4ODaeI9NBnxqAjXxO/
hslsO9dSATJLKysqn4HVfh6ps0AdFHLzZbP5mxYraeVBPDbz3gqCZ3n0zBd6ZQdDYe3Ms5uw8MVdIuktjhnt7ezfj74eRSKSHVzN0PddWMDUyYuCrGUNbWDhY/
dHhEb7KgBmmlG6HDK88fvz4KV6dGsmmRITjjsAkiz9RZMwpjpv+SRkw61PCkZcfhtcydS9mGY1T9XX1++Vs8fQtq4Fxp49M5rASiMesVMTYcvp4wrjTx8P/
0W1saWkduXXbw1dlAQcgssxZ3vtDz+WUG0BxWenhd44eFQ/Bs/7RwYRjH91kiY4O7ithnFhkoKFcNe297KKnTy2k/f0ueu5sBf2xczHt63VR/ydl7AO702Zj/
sPb51EKOyDqgHP70QkIP6omwQ1WxnCxhoR1StI3yhFtsZG0f/wXOX1lgvwR4cjbBwZIIE/P2oSaTdE2L1jYj4KzCrkM/+bxNoZ0+lN8ZmaFt+32dyFuCd/
F7Ga45ML5jerguBOKftjsh6PaKPB3WMAA0B9PeGrJ5gC+UyCYLcfvXwtxfcyQBmY0YBjSGqADeX5JZXNEMd183GYj3VteKhTssfx9ZdMW3o3hrtF4Lc7nGG9KiaQDzsrKuoo/NwJHeZUEyy/
3XBo3mfBPZh2QsY7JLNPW6kMHMXMJoJHIiYDVhn8Y64CH7KB6w+tRSyKgv2vYL1ytbvOq/zII+RuQqIYHzX+HtgAAAABJRU5ErkJggg=="

  }
}
```

Working with AI objects

Create and manage objects external to the aircraft. Aircraft and objects are available for creation.

Creating and destroying objects

Objects must first be registered with MSFS before being created.

Create an object using `create_object`. You'll pick a `name` which is how you reference the object while it exists (to update its properties and then later remove it), and also a `title` which is the name that comes from `aircraft.cfg` or `sim.cfg`.

Remove an object when you are finished by using `destroy_object`. If you don't destroy objects, they will be cleaned up when the mission eventually ends.

Examples:

```
{"create_object":{
  "name": "my_object",
  "title": "HPG Airbus H145 Ambulance",
  "location": "$USER"
}},
{"sleep":60},
{"destroy_object": "my_object"}
```

Object properties

Set properties on objects to configure them.

`VAR 1` often is configured to set the animation state for the object.

Examples:

```
{"set": {"object": "my_object", "var":"MODE"}, "value": 1}
```

Move objects

Use `move_object` to instantly change the position and orientation of an object.

Object waypoint navigation

Use `drive_object` to send an object along a series of waypoints at a specific speed.

Flying objects

TODO

Creating third party object packs

Read the Legacy Documentation
(verry old!)

Sounds & Text to Speech

It's possible to override the main aircraft `sound.xml` however this method isn't recommended as **only one** addon will "win" if there are multiple that attempt to do it. Additionally, addons may become stale as the main aircraft `sound.xml` changes. Where possible the `Voice Server` can be used to manage sounds.

Built-in sounds

`play_audio` can play various built-in sounds.

Voice Server

The Voice Server is a an external program which must be running while the user runs the mission. The voice server will accept commands and subsequently play sounds (outside of MSFS).

To use the voice server, it must first be connected. Call `connect_voice_server` with an `on_connected` `COMMANDLIST`. Note that you should gracefully handle the case where the voice server isn't available.

Example:

```
{"connect_voice_server": {
  "on_connected":[
    {"speak":"Speech activated."}
  ],
  "on_disconnected":[
    {"set_message":{"text":"No voice server available"}}
  ]
}}
```

When running the above, you'll either see `No voice server available` or hear `Speech activated`.
You can check if the voice server is connected at any time by the boolean result of `{"fn":"is_voice_server_connected"}`.

Once you are connected, you can send the `speak` command in several variations:

1. `{"speak":"hello world"}` In this case we simply speak some text. `speak` returns instantly after sending the command.
2. `{"speak":{"text":"hello world: {0}", "params":[ 99 ]}}` Here we use `text/params` to build up a string.
3. `{"speak":"hello world", "interrupt: 1}`, Here we are using `interrupt` to abort any active speech/audio, and to immediately begin playing this new message.
4. `{"speak":"hello.wav", "is_audio_file":1}` Here we are playing the sound `hello.wav` (from the audio folder), and the directive `is_audio_file` tells the server that the text which is normally speech is instead a file name.

Implement a compatible voice server (Advanced)

Server must run on `localhost:5997` and be of type websocket.

Messages:

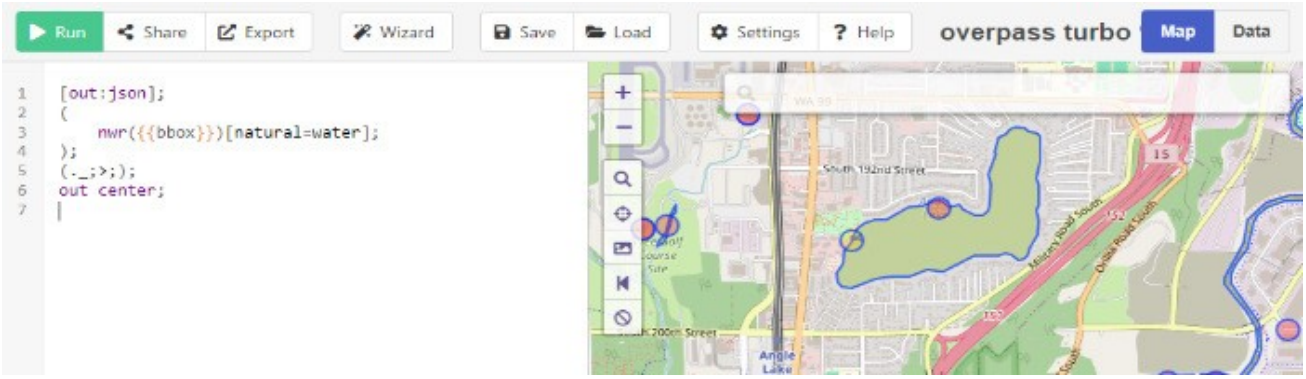
- `{"Text":"","FileName":"test1.wav","Interrupt":false}`
- `{"Text":"hello your name","FileName":"","Interrupt":false}`
- `{"Text":"stop text","FileName":"","Interrupt":true}`

Voice Server Test Program

```
{
 "title": "Voice Server Test Program",
 "briefing": [
 {"title": "Voice Server Test Program"},
 {
 "buttonbar": [ {"title":"Connect to voice server",  "commands":[ {"call_macro":"connect_voice"}]} ],
 "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 0}
 },
 {
 "text":"Not connected to voice server.",
 "color":"red",
 "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 0}
 },
 {
 "text":"Connected to voice server.",
 "color":"green",
 "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 1}
 },
 {"text": "------------------"},
 {"text":".wav file test1", "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 1}},
 {
 "buttonbar": [ {"title":"Play",  "commands":[
 {"speak":"test1.wav", "is_audio_file":1}
 ]}],
 "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 1}
 },
 {"text":"speech recognition test text", "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 1}},
 {
 "buttonbar": [ {"title":"Speak",  "commands":[
 {"speak":"speech recognition test text"}
 ]}],
 "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 1}
 },
 {"text":"INTERRUPT: stop text", "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 1}},
 {
 "buttonbar": [ {"title":"Speak",  "commands":[
 {"speak":"stop text", "interrupt":1}
 ]}],
 "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 1}
    },
```

```
    {"text":{"text":"Custom Message: {0}", "params":[ {"local":"message_textbox"} ]}, "show_condition": {"require":
{"fn":"is_voice_server_connected"}, "eq": 1}},
    {"textbox": "message_textbox"},
    {
      "buttonbar": [ {"title":"Speak",  "commands":[
        {"speak": {"local":"message_textbox"}}
      ]}],
      "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 1}
    },
    {"text":{"text":"Format Message: hello {0}", "params":[ {"local":"message_textbox"} ]}, "show_condition": {"require":
{"fn":"is_voice_server_connected"}, "eq": 1}},
    {
      "buttonbar": [ {"title":"Speak",  "commands":[
        {"speak": {"text":"hello {0}", "params": [ {"local":"message_textbox"} ]}}
      ]}],
      "show_condition": {"require":{"fn":"is_voice_server_connected"}, "eq": 1}
    }
  ],
  "macros": {
    "connect_voice": [
      {"connect_voice_server": {
        "on_connected":[
                {"speak":"Speech activated."}
        ],
        "on_disconnected":[
          {"set_message":"No voice server available"}
        ]
      }}
    ]
  },
  "objectives": [
    {
      "title": "Done",
      "commands": [
        {"set":{"local":"message_textbox"},"value":"your name"},
        {"call_macro":"connect_voice"},
        {"sleep": "forever"}
      ]
    }
  ]
}
```

OpenStreetMap data



OpenStreetMap or OSM data is available worldwide. OSM elements define and tag the world's features.
OSM has three primary types of elements available:

Element	Remarks
node	A node represents a single point. Sometimes featured as tagged as a node if they are small, but often larger features will have their perimeter defined by a way instead. A node may be a point on the road or rail network, or a tree or a hospital building. You'll need to be careful as it is becoming increasingly common to tag features with a way instead of the center point. By adding `out center` you an ask OSM to give you the center point for a way, which can sometimes be helpful also.
way	A way is a container for a list of nodes. Roads and Railways are created from ways which are created from nodes. You'll be able to retrieve the way and its metadata, as well as the complete list of nodes contained by that way. An area is a closed way.
relation	A relation is a way to include nodes an ways into a type of container which may have tags. A relation may be used for an administrative boundary like the city or state.

OSM Developer Workflow

You'll need to:

1. Discover features that you wish to work with. Visit OpenStreetMap and use the `Query Features` tool (right edge) to examine OSM data at a given location.
2. Create the query that will retrieve those features. Visit Overpass Turbo to start working with queries and testing them in different locations on the map.
3. `Choose the way you want to use the data within the mission. You might use create_location and then query_random_result (get all results from the query, pick one randomly) or query_closest_result (start at 100 meters and expand until you find the first result).`

Once you found the data you like and created a working query, you'll integrate that into the mission and test it.

Example OSM queries

All queries below will use the `bbox` bounding box. Don't forget that you must escape quotes " like this: `\"` in JSON.

Get nearby hospitals (`amenity=hospital` tag). We are getting `node` and `way` results, because hospitals are tagged often each way. `out center` is adding a centroid point to each way which enables us to treat them more similarly to nodes.

```
[out:json];
(
  node({{bbox}})[amenity=hospital];
  way({{bbox}})[amenity=hospital];
);
out center;
```

Breakdown of the query:

1. `[out:json];`: Configure for JSON output. Always the same.
2. `(` and `);`: Union that will combine the two groups of results into one list.
3. Get nodes within the view which match the tag `amenity=hospital`.

```
node({{bbox}})[amenity=hospital];
```

4: Get ways within the view which match the tag `amenity=hospital`.

```
way({{bbox}})[amenity=hospital];
```

5: `out center;`: For ways, add a centroid `lat/lon` to each result.

OpenStreetMap APIs

There are powerful APIs for working with data available from OpenStreetMap. You can discover nearby POIs as well as examine polygons and other relationships.

NOTE You'll want to always use `bbox` which will be replaced for you by the location/radius provided. Convert any code using `around` to `bbox`.

API	V1 or V2	Remarks
create_location	V1	Create a location from a list of zones. Still useful.
query_data	V1	Still useful for database query.
query_country	V1	Get the country for a given locatioin. Still useful.
osm_query_data	V2	Query and then process any OSM OverpassAPI data

When using the V2 API, you'll use these accessors:

API	Remarks
osm_get_parent_ways	Get the ways that a given node exists in
osm_get_connected_nodes	Get the nodes before and after this node in the ways it exists in
osm_get_nodes	Get the nodes within this way
osm_get_all_ways	Get all the ways within the data set
osm_get_all_nodes	Get all the nodes within the data set
osm_get_closest_nodes	Get an ordered list of nodes, order is by distance
osm_is_point_within_way	Get a boolean indicating whether the point is within a given closed way
osm_get_area_of_area	Get the area in meters^2 of a given closed way

Show nearby hospital helipads sample

This sample uses `query_data` to find nearby items in the `DB:H_HOSPITAL` (hospital with helipad) database.

```
{
  "title": "Show nearby hospital helipads",
  "author":"davux3",
  "api_version": 0.1,
  "aircraft": ["H145"],
  "objectives": [
    {
      "title": "Done",
      "commands": [


        {"query_data": {
          "query": "DB:H_HOSPITAL",
          "location": "$USER",
          "radius": 5000,
```

```
          "minRadius": 0,
          "bypass_commands": [
            {"#comment":"$ITEMS contains an array of results: type=way, lat, lon, tags. by convention center is copied down into lat/lon"},
            {"for_each":{"param":"$ITEMS"},"do":[
              {"set":{"param":"loc"},"value":{"create_array":[
                {"struct":{"param":"$item"},"path":"lat"},
                {"struct":{"param":"$item"},"path":"lon"}
              ]}},
              {"set_map":{"add":{"point":{"location":{"param":"loc"},"icon":"ki_helipad"}}}}
            ]}
          ]
        }},

        {"sleep": "forever"}
      ]
    }
  ]
}
```

Show nearby power substations sample

This sample uses `query_data` to find nearby OSM items matching `power=substation`.

```
{
  "title": "Show nearby power substations",
  "author":"davux3",
  "api_version": 0.1,
  "aircraft": ["H145"],
  "objectives": [
    {
      "title": "Done",
      "commands": [

        {"query_data": {
          "query": "[out:json]; (node({{bbox}})[power=substation]; area({{bbox}})[power=substation];); out center;",
          "location": "$USER",
          "radius": 2500,
          "minRadius": 0,
          "bypass_commands": [
            {"#comment":"$ITEMS contains an array of results: type=way, lat, lon, tags. by convention center is copied down into lat/lon"},
            {"for_each":{"param":"$ITEMS"},"do":[
              {"set":{"param":"loc"},"value":{"create_array":[
                {"struct":{"param":"$item"},"path":"lat"},
                {"struct":{"param":"$item"},"path":"lon"}
              ]}},
              {"set_map":{"add":{"point":{"location":{"param":"loc"},"icon":"ki_helipad"}}}}
            ]}
          ]
        }},
        {"sleep": "forever"}
      ]
    }
  ]
}
```

Train level crossing objects

This sample accesses the road and rail data to determine a good place to place a crash involving a train and a school bus.

1. Find a nearby level train crossing
2. Get the roads and rails connected to that crossing
3. Place a train on the crossing, and a bus crashing into it from the direction of the road.

```
{
  "title": "Train Level Crossing Test",
  "api_version": 0.1,
  "aircraft": ["H145"],
  "locations":{
    "LOC":"$USER"
  },
  "briefing":[
    {"buttonbar":[ {"title":"teleport to accident","commands":[ {"call_macro":"adv_teleport","params":{ "loc": "accident_location" }} ] }   ]}
  ],
  "macros":{
    "adv_teleport":[
      {"#comment":"param: loc"},
      {"trigger":"K:SLEW_ON"},
      {"sleep":1},
      {"teleport_to":{"param":"loc"}},
      {"set":{"var":["A:PLANE ALTITUDE", "feet"]}, "value":-1000},
      {"set":{"var":["A:PLANE HEADING DEGREES TRUE", "degrees"]}, "value":{"param":"hdg"}},
      {"sleep":2},
      {"trigger":"K:SLEW_OFF"}
    ],
    "create_closest_railway_crossing_accident_scene":[
      {"#comment": "param - location"},
      {"#comment":"first get the closest result and get the node ID"},
      {"create_location": "accident_location", "zones": [
        {"zone": {
          "zone_type": "query_random_result",
          "radius":10000,
          "query": "[out:json];node({{bbox}})[railway=\"level_crossing\"];out center;",
          "location": {"param":"location"},
          "commands":[ {"set":{"param":"railway_crossing_node_id"}, "value":{"param":"$LOCATION:ID"}} ]
        }}
      ]},

      {"#comment":"query nearby highway and railway (very close because it's already the exact position). We get all the nodes in those ways."},
      {"osm_query_data":
```

```
        "[out:json];(way({{bbox}})[railway]; way({{bbox}})[highway];);(._;>;);out;",
        "location":"accident_location",
        "size": 1,
        "result":"my_data"
      },

      {"#comment":"find all the connected nodes and put them into highway and railway lists"},
      {"set":{"param":"highway_nodes"}, "value":{"create_array":[]}},
      {"set":{"param":"railway_nodes"}, "value":{"create_array":[]}},

      {"#comment":"get the nodes connected to our root (intersection) node. these are nodes before/after the root node, on any ways that the node
is apart of"},
      {"osm_get_connected_nodes":{"param":"railway_crossing_node_id"}, "data": {"param":"my_data"}, "result":"my_nodes_connected_to_nearest_node"},
      {"for_each": {"param":"my_nodes_connected_to_nearest_node"}, "do":[
        {"#comment":"draw a line with a different color for each leg of the intersection"},
        {"set":{"param":"first_way_key"}, "value":{"struct":{"object:keys":{"struct":{"param":"$item"}, "path":"_ways"}}, "index": 0}},
        {"set":{"param":"is_road"}, "value":{"struct":{"param":"$item"}, "path":"_ways.{first_way_key}.tags.highway"}},
        {"#comment":{"text":"is_road: {0}","params":[ {"json:stringify":{"param":"is_road"}} ]}},
        {"if":{"param":"is_road"},"ne":null,"then":[
          {"modify_array":{"param":"highway_nodes"}, "append":{"create_array":[ {"struct":{"param":"$item"},"path":"lat"}, {"struct":
"param":"$item"},"path":"lon"} ]}}
        ], "else":[
          {"modify_array":{"param":"railway_nodes"}, "append":{"create_array":[ {"struct":{"param":"$item"},"path":"lat"}, {"struct":
{"param":"$item"},"path":"lon"} ]}}
        ]}
      ]},
      {"return":{"create_struct":{
        "nodeId": {"param":"railway_crossing_node_id"},
        "highway_nodes": {"param":"highway_nodes"},
        "railway_nodes": {"param":"railway_nodes"},
        "data": {"param":"my_data"},
        "location": {"resolve_location": "accident_location"}
      }}}
    ]
  },
  "objectives": [
    {
      "title": "Initializing...",
      "commands": [
        {"#comment":"find the closest railway/road level crossing"},
        {"call_macro":"create_closest_railway_crossing_accident_scene","params":{
          "location": "$USER"
        }, "result":"crossing_ret"},
        {"#comment":"get the extracted data"},
        {"set":{"param":"railway_crossing_node_id"},"value":{"struct":{"param":"crossing_ret"}, "path":"nodeId"}},
        {"set":{"param":"railway_crossing_location"},"value":{"struct":{"param":"crossing_ret"}, "path":"location"}},
        {"set":{"param":"highway_nodes"},"value":{"struct":{"param":"crossing_ret"}, "path":"highway_nodes"}},
        {"set":{"param":"railway_nodes"},"value":{"struct":{"param":"crossing_ret"}, "path":"railway_nodes"}},
        {"#comment":"visualize the results by putting an icon at the crossing and drawing a line to the nearby highway/railway nodes"},
        {"set_map":{"add":{"point":{ "location":{"param":"railway_crossing_location"}, "icon":"ki_waypoint_blue"}}}},
        {"set_map":{"add":{"line":{ "points":{"create_array":[
          {"param":"railway_crossing_location"},
          {"struct":{"param":"highway_nodes"},"index":"0"}
        ]}, "stroke":{"create_struct":{ "color":"red", "width":2}}}}},
        {"set_map":{"add":{"line":{ "points":{"create_array":[
          {"param":"railway_crossing_location"},
          {"struct":{"param":"railway_nodes"},"index":"0"}
        ]}, "stroke":{"create_struct":{ "color":"blue", "width":2}}}}},
        {"#comment":"calculate bearings"},
        {"set":{"param":"train_brg"},"value":{"bearing":{"from":{"param":"railway_crossing_location"},"to":{"struct":
{"param":"railway_nodes"},"index":"0"}}}},
        {"set":{"param":"crash_brg"},"value":{"bearing":{"from":{"param":"railway_crossing_location"},"to":{"struct":
{"param":"highway_nodes"},"index":"0"}}}},
        {"#comment":"place train and crash objects"},
        {"create_object": {"name": "train", "title": "Airbus H145 Train", "location": {"bearing2":{"param":"train_brg"},"dist":20,"object":
{"param":"railway_crossing_location"}}}},
        {"track_object": {"object": "train", "icon": "ki_helipad"}},
        {"create_object": {"name": "crash1", "title": "Airbus H145 School bus", "location": {"bearing2":{"param":"crash_brg"},"dist":-7,"object":
{"param":"railway_crossing_location"}}}},
        {"track_object": {"object": "crash1", "icon": "ki_helipad"}},
        {"sleep": "forever"}
      ]
    }
  ]
}
```

Road network test program

This sample will:

1. Query a second of the road network
2. Draw red lines on all of the roads
3. Find the closest road intersection to `LOC` location.
4. Draw a different color line from each leg of the intersection to the center.

```
{
 "title": "Road Network Test",
 "api_version": 0.1,
 "aircraft": ["H145"],
 "data": {
 "colors": ["hotpink","blue","orange","yellow","green","purple"]
 },
 "locations":{
 "LOC":{"bearing":270,"dist":500}
 },
 "objectives": [
 {
 "title": "Initializing...",
      "commands": [
        {"#comment":"Query a block of road network data and save it into my_data"},
        {"osm_query_data":
          "[out:json];way({{bbox}})[highway~\"^(motorway|trunk|primary|secondary|unclassified|residential|living_street|service|tertiary|(motorway|
```

```
trunk|primary|secondary|tertiary|)_link)$\"];(._;>;);out;",
          "location":"LOC",
          "size": 600,
          "result":"my_data"

        },
        {"#comment":"extract the list of ways into my_ways, and then loop over them and draw them all on the map"},
        {"osm_get_all_ways": {"param":"my_data"}, "result":"my_ways"},
        {"for_each":{"param":"my_ways"},"do":[
          {"#comment":"create a list and then get a list of all the nodes in my_ways. store that list of nodes into my_nodes_on_way"},
{"set":{"param":"node_location_list"}, "value":{"create_array":[]}},
          {"osm_get_nodes":{"struct":{"param":"$item"}, "path":"id"}, "data": {"param":"my_data"}, "result":"my_nodes_on_way"},
          {"#comment":"create a [lat,lon] array from each node and put it into the results list"},
          {"for_each": {"param":"my_nodes_on_way"}, "do":[
            {"modify_array":{"param":"node_location_list"}, "append":{"create_array":[
              {"struct":{"param":"$item"},"path":"lat"},
              {"struct":{"param":"$item"},"path":"lon"}
            ]}},
            {"sleep":0.001}
          ]},
          {"#comment":"draw the road"},
          {"set_map":{"add":{"line":{ "points":{"param":"node_location_list"}, "stroke":{"color":"red", "width":2}}}}}
        ]},

        {"#comment":"get an ordered list of all the nodes by their distance from LOC"},
        {"osm_get_closest_nodes": "LOC", "data": {"param":"my_data"}, "result":"my_closest_nodes"},
        {"#comment":"go through the result nodes and pick only those with at least 2 parents (an intersection)"},
        {"set":{"param":"closest_node"},"value":null},
        {"for_each":{"param":"my_closest_nodes"},"do":[
          {"osm_get_parent_ways":{"struct":{"param":"$item"},"path":"id"}, "data": {"param":"my_data"}, "result":"parents"},
          {"if":{"struct":{"param":"parents"}, "path":"length"},"gt": 1,"then":[
            {"set":{"param":"closest_node"},"value":{"param":"$item"}},
            {"break":1}
          ]},
          {"#comment":"do get parent and check for more than one for an intersection"},
          {"sleep":0.001}
        ]},
        {"set":{"param":"closest_location"}, "value":{"create_array":[
          {"struct": {"param":"closest_node"},"path":"lat"},
          {"struct": {"param":"closest_node"},"path":"lon"}
        ]}},
        {"set_map":{"add":{"point":{ "location":{"param":"closest_location"}, "icon":"ki_waypoint_blue"}}}},
        {"#comment":"get the nodes connected to our root (intersection) node. these are nodes before/after the root node, on any ways that the node
is apart of"},
        {"osm_get_connected_nodes":{"struct":{"param":"closest_node"},"path":"id"}, "data": {"param":"my_data"},
"result":"my_nodes_connected_to_nearest_node"},
        {"for_each": {"param":"my_nodes_connected_to_nearest_node"}, "do":[
          {"#comment":"draw a line with a different color for each leg of the intersection"},
          {"set_map":{"add":{"line":{ "points":{"create_array":[
            {"param":"closest_location"},
            {"create_array":[
              {"struct":{"param":"$item"},"path":"lat"},
              {"struct":{"param":"$item"},"path":"lon"}
            ]}
          ]}, "stroke":{"create_struct":{
            "color":{"struct": {"static": "colors"}, "index": {"param":"$index"}}, "width":2}}}}}
        ]},
        {"#comment":"save some debug stuff"},
        {"set":{"local":"node_location_list2"},"value":{"param":"node_location_list2"}},
        {"set":{"local":"my_closest_nodes"},"value":{"param":"my_closest_nodes"}},
        {"set":{"local":"my_ways"},"value":{"param":"my_ways"}},
        {"set":{"local":"my_data"},"value":{"param":"my_data"}},
        {"sleep": "forever"}
      ]
    }
  ]
}
```

Water polygon test program

You must be essentially on top of a lake for this to work. Change LOC to be in the middle of a lake, or increase the `size`. You should find only one body of water or else the area message will be overwritten.

This sample:

1. Queries nearby water polygons
2. Draws a bunch of points on the map to determine if they are inside or outside of the water

```
{
  "title": "get very nearby water and get area",
  "api_version": 0.1,
  "aircraft": ["H145"],
  "locations":{
    "LOC":"$USER"
  },
  "objectives": [
    {
      "title": "Initializing...",
      "commands": [
        {"#comment":"Query a block of road network data and save it into my_data"},
        {"osm_query_data":
          "[out:json];nwr({{bbox}})[natural=water];(._;>;);out;",
          "location":"LOC",
          "size": 200,
          "result":"my_data"
        },
        {"#comment":"extract the list of ways into my_ways, and then loop over them and draw them all on the map"},
        {"osm_get_all_ways": {"param":"my_data"}, "result":"my_ways"},
        {"for_each":{"param":"my_ways"},"do":[
          {"set":{"param":"way"},"value": {"param":"$item"}},
          {"#comment":"create a list and then get a list of all the nodes in my_ways. store that list of nodes into my_nodes_on_way"},
          {"set":{"param":"node_location_list"}, "value":{"create_array":[]}},
```

```
          {"osm_get_nodes":{"struct":{"param":"$item"}, "path":"id"}, "data": {"param":"my_data"}, "result":"my_nodes_on_way"},
          {"#comment":"create a [lat,lon] array from each node and put it into the results list"},
          {"for_each": {"param":"my_nodes_on_way"}, "do":[
            {"modify_array":{"param":"node_location_list"}, "append":{"create_array":[
              {"struct":{"param":"$item"},"path":"lat"},
              {"struct":{"param":"$item"},"path":"lon"}
            ]}},
            {"sleep":0.001}
          ]},
          {"osm_get_area_of_area":{"struct":{"param":"way"},"path":"id"},  "data": {"param":"my_data"}, "result":"way_area"},
          {"set_message":{"text":"way size {way_area} meters"}},
          {"#comment":"draw the road"},

          {"for_each":{"create_array": 10},"do":[
            {"set":{"param":"dist"},"value":{"multiply":[ {"param":"$index"}, 30 ]}},
            {"for_each":{"create_array": 36},"do":[
              {"set":{"param":"brg"},"value":{"multiply":[ {"param":"$index"}, 10 ]}},
              {"osm_is_point_within_way": {"struct":{"param":"way"}, "path":"id"}, "location":{"bearing":{"param":"brg"},"dist":{"param":"dist"}},
"data":{"param":"my_data"},"result":"is_in"},
              {"if":{"param":"is_in"}, "eq":1,"then":[
                {"set_map":{"add":{"point":{ "location":{"bearing":{"param":"brg"},"dist":{"param":"dist"}}, "icon":"ki_helipad"}}}}
              ],"else":[
                {"set_map":{"add":{"point":{ "location":{"bearing":{"param":"brg"},"dist":{"param":"dist"}}, "icon":"ki_waypoint_blue"}}}}
              ]},
              {"sleep":0.001}           ]},
            {"sleep":0.001}
          ]},
          {"set_map":{"add":{"line":{ "points":{"param":"node_location_list"}, "stroke":{"color":"red", "width":2}}}}}
        ]},
        {"sleep": "forever"}
      ]
    }
  ]
}
```

Buildings test

1. Get the nearby buildings and outline them in red.

```
{
  "title": "Get nearby buildings and outline them",
  "api_version": 0.1,
  "aircraft": ["H145"],
  "locations":{
    "LOC":"$USER"
  },
  "objectives": [
    {
      "title": "Initializing...",
      "commands": [
        {"#comment":"Query a block of road network data and save it into my_data"},
        {"osm_query_data":
          "[out:json];way({{bbox}})[building];(._;>;);out;",
          "location":"LOC",
          "size": 100,
          "result":"my_data"
        },
        {"#comment":"extract the list of ways into my_ways, and then loop over them and draw them all on the
map"},
        {"osm_get_all_ways": {"param":"my_data"}, "result":"my_ways"},
        {"for_each":{"param":"my_ways"},"do":[
          {"#comment":"create a list and then get a list of all the nodes in my_ways. store that list of nodes
into my_nodes_on_way"},
          {"set":{"param":"node_location_list"}, "value":{"create_array":[]}},
          {"osm_get_nodes":{"struct":{"param":"$item"}, "path":"id"}, "data": {"param":"my_data"},
"result":"my_nodes_on_way"},
          {"#comment":"create a [lat,lon] array from each node and put it into the results list"},
          {"for_each": {"param":"my_nodes_on_way"}, "do":[
            {"modify_array":{"param":"node_location_list"}, "append":{"create_array":[
              {"struct":{"param":"$item"},"path":"lat"},
              {"struct":{"param":"$item"},"path":"lon"}
            ]}},
            {"sleep":0.001}
          ]},
          {"#comment":"draw the building"},
          {"set_map":{"add":{"line":{ "points":{"param":"node_location_list"}, "stroke":{"color":"red",
"width":2}}}}}
        ]},
        {"sleep": "forever"}
      ]
    }
  ]
}
```

Operating the hoist

There are V1 and V2 APIs for manipulating the hoist.

Hoist APIs

API	V1 or V2	Remarks
fn.HOIST_SEND_TO_GROUND	V1	Send the hoist to the ground when conditions are met
fn.HOIST_REEL_UP_AND_STOW	V1	Reel up the hoist and stow it
fn.HOIST_REEL_UP	V1	Reel up the hoist only
hoist_control	V2	Provides direct control over reeling in/out.
fn.hoist_get_reel_distance:ft	V2	Distance that the reel is extended, also supports `:m`
fn.hoist_get_distance_from_ground:ft	V2	Distance from the hoist object to the ground, also supports `:m`

Hoist SDK Variables

L:Var	Remarks
L:H145_SDK_HOIST_MODE	0: auto, 1: manual
L:H145_SDK_HOIST_CONTROL	manual control signal
L:H145_SDK_HOIST_CABLE_FT	Distance that the reel is extended
L:H145_SDK_EQUIP_HOIST	0: not installed, 1: installed
L:H145_SDK_OH_HOIST	Overhead switch position (and arm position)

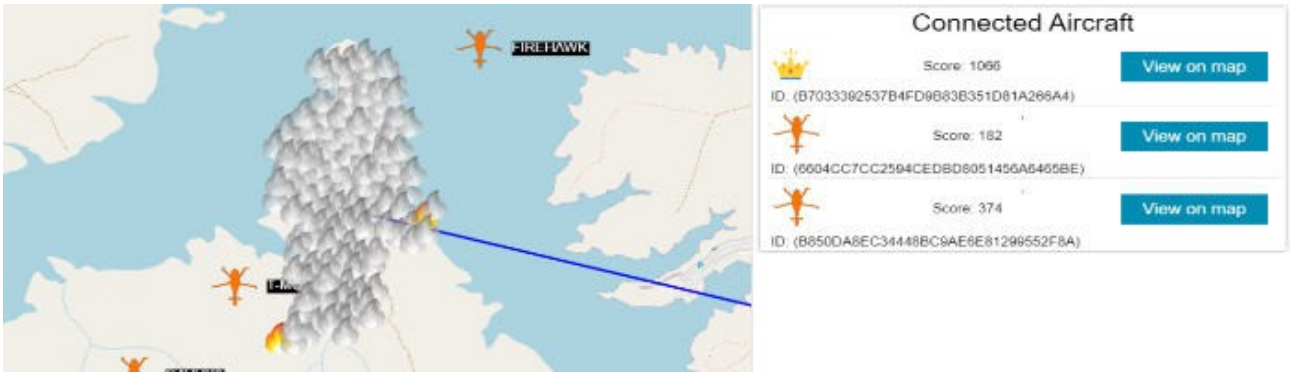
Hoist SDK Events

L:Event	Remarks
H:H145_SDK_EQUIP_HOIST_ON	Equipment Setup - Hoist ON
H:H145_SDK_EQUIP_HOIST_OFF	Equipment Setup - Hoist OFF
H:H145_SDK_EQUIP_HOIST_TOGGLE	Equipment Setup - Hoist TOGGLE
H:H145_SDK_HOIST_CONTROL_MODE_AUTO	Hoist - Hoist Mode AUTO
H:H145_SDK_HOIST_CONTROL_MODE_MANUAL	Hoist - Hoist Mode MANUAL
H:H145_SDK_HOIST_CONTROL_MOTOR_UP	Hoist - Hoist Manual Control UP
H:H145_SDK_HOIST_CONTROL_MOTOR_STOP	Hoist - Hoist Manual Control STOP
H:H145_SDK_HOIST_CONTROL_MOTOR_DOWN	Hoist - Hoist Manual Control DOWN
H:H145_SDK_HOIST_CONTROL_MOTOR_MOMENTARY_UP	Hoist - Hoist Manual Control MOMENTARY_UP
H:H145_SDK_HOIST_CONTROL_MOTOR_MOMENTARY_DOWN	Hoist - Hoist Manual Control MOMENTARY_DOWN
H:H145_SDK_HOIST_ARM_STOW	Hoist - Hoist Arm STOW
H:H145_SDK_HOIST_ARM_DEPLOY	Hoist - Hoist Arm DEPLOY
H:H145_SDK_HOIST_EASY_STOWED	Hoist - Select Stowed
H:H145_SDK_HOIST_EASY_DEPLOYED	Hoist - Select Deployed
H:H145_SDK_HOIST_EASY_CREW	Hoist - Select Crew
H:H145_SDK_HOIST_EASY_WORKER	Hoist - Select Worker
H:H145_SDK_HOIST_EASY_STRETCHER_ANDCREW	Hoist - Select Stetcher_And_Crew
H:H145_SDK_HOIST_EASY_SURVIVOR1_ANDCREW	Hoist - Select Survivor_1_And_Crew
H:H145_SDK_HOIST_EASY_SURVIVOR2_ANDCREW	Hoist - Select Survivor_2_And_Crew
H:H145_SDK_HOIST_EASY_TOOLBAG	Hoist - Select Toolbag
H:H145_SDK_HOIST_EASY_CONTAINERS	Hoist - Select Containers
H:H145_SDK_HOIST_EASY_HOSE	Hoist - Select Hose

Hoist test program

```
{
  "title": "Hoist Test Program",
  "author":"davux3",
  "api_version": 0.1,
  "aircraft": ["H145"],
  "briefing":[
    {"text":"Extended length: {0} ft", "params":[ {"tofixed":{"fn":"hoist_get_reel_distance:ft"}, "digits":2} ]},
    {"text":"Aircraft distance to ground: {0} ft", "params":[ {"tofixed":{"subtract":[{"var":["RADIO HEIGHT", "feet"]}, 5.45]}, "digits":2} ]},
    {"text":"Hoist distance to ground: {0} ft", "params":[ {"tofixed":{"fn":"hoist_get_distance_from_ground:ft"}, "digits":2} ]},
    {"buttonbar":[
      {"title":"Retract", "commands":[ {"hoist_control":"reel_up", "speed":{"scale":[{"local":"speed_slider"}, 0, 100, 0.01, 2]}} ]},
      {"title":"Extend", "commands":[ {"hoist_control":"reel_down", "speed":{"scale":[{"local":"speed_slider"}, 0, 100, 0.01, 2]}} ]}
    ]},
    {"text":"Motor speed factor:"},
    {"slider":{ "min":0, "max":100, "local":"speed_slider" }}
  ],
  "objectives": [
    {
      "title": "Done",
      "commands": [
        {"set":{"local":"speed_slider"},"value":10},
        {"sleep": "forever"}
      ]
    }
  ]
}
```

Multiplayer missions



Multiplayer missions are missions where every player is connected to a common server. That server accepts commands from players and also send notifications to the players. This enables player-to-player communication, as well as adding web operators to interact with the players.

Missions are built on shared data which is mutated and events are generated as a result of those mutations. All clients may subscribe to areas of the data and then be advised of all changes. Consistency is assured by the client and server buffering which enables in-order guarantee on message delivery. Clients disconnecting and reconnecting will have their messages automatically delivered.

Server Termination Commands (`terminationCommands`)

The server will interpret the `terminationCommands` shared data. This section is specifically for clients to register commands which will be run when they are disconnected and then purged from the server after not reconnecting for some time. `terminationCommands` are very important in that they assure server data is not stale even in the face of player disconnects.

Generally, you'll want to add yourself (with an ID) to a `connectedAircraft`. Given that, you'll want to also set up a `terminationCommands.{id}` which goes and removes `connectedAircraft.{id}`. This way, when you disconnect the server is able to automatically clean up for you and clients will be notified of the delete operation, in case they need to respond accordingly.

Shared Data

Shared data is the foundation of the multiplayer platform. The server will store arbitrary values and the clients may subscribe to updates on those values. Each client carefully issues commands to `update` and `delete` data, using a `policy` to avoid conflicts and enable merging of commands.

Both on the aircraft and on the web you may use `set_shared_data` to issue these commands. In the aircraft, `MultiplayerClient` is your gateway to multiplayer data.

MultiplayerClient

`MultiplayerClient` is the type of object returned by `fn.create_multiplayer_connection`. This object represents the interface to the multiplayer server and it has various functions to call and state to access.

Function	Parameters	Remarks
Connect	`url`, `userId`, `roomId`, `roomPassword`	Establish a connection to the server.
Subscribe	`path`, `callback<object>`	Subscribe to a path and retrieve the current data via callback
Get	`path`, `callback<object>`	Get a pat and retrieve the current data via callback
Send	`message`	Send a message object
Close	None.	Disconnect and destroy the connection

Message Types:

Message Type	Parameters	Remarks
read	`path`, `value`	The server has returned an error regarding a recent command you sent.
update	`value`, `value`, `policy`	The server has returned an error regarding a recent command you sent.
delete	`path`	The server is sending you data about shared data changes.

Policy	Remarks
delta	Value is relative.
no_overwrite	Ignore update if path exists.

Property	Remarks
Status	Connection status.

Event Handler	Parameters	Remarks
OnError	`error` (string)	The server has returned an error regarding a recent command you sent.
OnMessage	`data` (object)	The server is sending you data about shared data changes.

Connection status values:

Status	Remarks
Unknown	Default state, this is the status before calling connect.
Disconnected	The connection is disconnected, retry will be automatically attempted.
Connecting	Connecting to the server.
LoggingIn	Server is connected, handshake in progress.
LoginFailed	Fatal. Login was not successful.
Connected	Currently connected.

Multiplayer simple scoring example

This mission creates a score table and each aircraft has a button to set the score for that player.

Additional features:

- Show flight plans on the web client map
- Clients will clean up their `connectedAircraft` and `terminationCommands` entries, but not their score.

```
{
 "title": "Multiplayer Score Mission Test Program",
 "author":"davux3",
 "api_version": 0.1,
 "aircraft": ["H145"],
 "data":{
 "server_url": "wss://5ed547d.online-server.cloud/mpserver/ws",
 "create_room_url": "https://davux.com/dispatcher/",
 "webConfig": {
 "fligihtPlans": {
 "type":"map_line",
 "source":{"static":"flightPlans"},
 "name":"Flight Plan",
 "stroke":{"no_resolve":{"color": "#d303fc", "width":2}},
 "icon":{"static":"icons.wp_blue"}
 },
 "connectedAircraftIcons": {
 "type":"map_point",
 "source":{"static":"connectedAircraft"},
 "name":"Connected Aircraft",
 "text":"{UserName}",
 "icon":{"static":"icons.h160_icon"}
 },
 "scoreList": {
 "type":"list",
 "source":{"static":"gameScores"},
 "title":"Game Scores",
 "emptyText":"No players have connected yet.",
 "rows":{
 "row0":{
 "1": {"text":"{UserName}"},
 "2": {"text":"Total Score: {0}", "params": [ {"round":{"param":"Score"}}  ]},
 "3": {"text":""}
 }
 }
 },
 "connectedAircraftList": {
 "type":"list",
 "source":{"static":"connectedAircraft"},
 "title":"Connected Aircraft",
 "emptyText":"No aircraft are connected right now",
 "rows":{
 "row0":{
 "1": {"icon":{"static":"icons.h160_icon"}},
 "2": {"text":"{UserName}"},
 "4": {"button":"View","commands": [ {"set_map_center": {"param": "location"}, "zoom": 16} ]}
 }
 }
```

```
      }
    }
  },
  "briefing":[
    {"#comment":[
      "MP_MODE ... 0: not set, 1: offline, 2: online"
    ]},
    {"title":"Mission Initial Setup", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
    {"buttonbar":[
      {"title":"Offline (Single player)", "commands": [ {"set":{"local":"MP_MODE"}, "value":1} ]},
      {"title":"Online (Multiplayer)", "commands": [ {"call_macro":"mp_open_login_dialog"}  ]}
    ], "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
    {"title":"Multiplayer (Online)", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 2}},
    {"buttonbar":[
      {"title":"View Multiplayer Status", "commands": [ {"call_macro":"mp_open_login_dialog"}  ]}
    ], "show_condition": {"require":{"local":"MP_MODE"}, "eq": 2}},
    {"title":"Game Score", "show_condition": {"require":{"local":"MP_MODE"},"ne":0}},
    {"text":"My score: {local:MY_SCORE}", "show_condition": {"require":{"local":"MP_MODE"},"ne":0}},
    {"buttonbar":[
      {
        "title":"Increment My Score",
        "commands":[
          {"set":{"local":"MY_SCORE"},"value":{"add":[ {"local":"MY_SCORE"}, 1 ]}},
          {"set_shared_data":"update", "path":"gameScores.{service_auth}.Score", "value": {"local":"MY_SCORE"} }
        ]
      }
    ],
    "show_condition": {"require":{"local":"MP_MODE"},"ne":0}}
  ],
  "events": {
    "ON_MISSION_ABORTING": {
      "commands": [     {"call_macro":"mp_aborting_mission"}    ]
    }
  },
  "macros":{
    "mp_open_login_dialog":[
      {"#comment": "Show the login dialog dispatch (or multiplayer status"},
      {"set_dispatch":[
        {"buttonbar":[ {"title":"<- Back to briefing", "commands": [{"set_briefing_dialog":1} ]} ]},
        {"title":"Log in",  "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"text":"You are playing offline.", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 1}},
        {"text":{"text":"User Id: {0}", "params":[{"local":"service_auth"}]}, "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"text":"User Name:", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"textbox":"mp_userName", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"text":"Room:", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"textbox":"mp_room", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"text":"Password:", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"textbox":"mp_password", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"buttonbar":[
          {"title":"Create Room (Opens on PC)", "commands": [ {"open_url":"{static:create_room_url}?room={local:mp_room}"} ]},
          {"title":"Log In", "commands": [ {"call_macro":"mp_login"} ]}
        ],
        "disabled_condition":{"require":{"struct":{"local":"MP_CONN"}, "path":"Status"},"eq":"Connected"},
        "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"text":{"text":"MP Connection Status: {0}", "params":[
          {"struct":{"local":"MP_CONN"}, "path":"Status"}
        ]}, "show_condition": {"require":{"local":"MP_MODE"}, "ne": 1}},
        {"text":{"text":"MP Server Last Error: {local:MP_LAST_ERROR}"}, "show_condition": {"require":{"local":"MP_MODE"}, "ne": 1}},
        {"title":"Debug Info"},
        {"text":{"text":"Multiplayer Mode: {0}", "params":[
          {"switch":{"local":"MP_MODE"}, "case":{
            "0": "Undecided",
            "1": "Offline, Singleplayer",
            "2": "Multiplayer"
          }}
        ]}},        {"#comment":{"text":"Debug MP Message: {local:MP_MSG}"}, "show_condition": {"require":{"local":"MP_MODE"}, "ne": 1}}
      ]},
      {"set_dispatch_dialog":1}
    ],
    "mp_login":[
      {"#comment":"try to make the actual connection to the server"},
      {"set":{"param":"service_auth"},"value":{"local":"service_auth"}},
      {"set":{"local":"MP_LAST_ERROR"},"value":""},
      {"set":{"local":"MP_CONN"}, "value": {"fn": "create_multiplayer_connection"}},
      {"set":{"local":"MP_CONN", "path":"OnError"}, "value":{"js:create_async_function":[
        {"set":{"local":"MP_LAST_ERROR"},"value":{"struct": {"param":"$args"}, "index": 0}}
      ]}},
      {"set":{"local":"MP_CONN", "path":"OnMessage"}, "value":{"js:create_async_function":[
        {"set":{"param":"arg0"},"value":{"struct": {"param":"$args"}, "index": 0}},
        {"call_macro":"mp_on_message","params":{"msg": {"param":"arg0"}}}
      ]}},
      {"set":{"param":"unused"},"value":{"struct":{"local":"MP_CONN"}, "function":"Connect", "params":[
        {"static":"server_url"}, {"param":"service_auth"}, {"local":"mp_room"}, {"local":"mp_password"}
      ]}},
      {"create_thread":{"commands":[
        {"wait_for":{"struct":{"local":"MP_CONN"}, "path":"Status"},"eq":"Connected"},
        {"#comment":"once we log in once, we're committed to muoltiplayer"},
        {"set":{"local":"MP_MODE"}, "value": 2},
        {"set_briefing_dialog":1},
        {"#comment":"First create terminationCommands with no_overwrite, then add an entry for us, and then populate with commands to clear us from
connectedAircraft and terminationCommands when we become stale on the server"},
        {"set_shared_data":"update",
          "path":"terminationCommands",
          "policy":"no_overwrite",
          "value": {"create_struct":{}}
        },
        {"set_shared_data":"update",
          "path":"terminationCommands.{service_auth}",
          "value": {"create_struct":{
            "removeFromConnectedAircraft":{"create_struct":{
              "type":"delete",
              "path":"connectedAircraft.{service_auth}"
            }},
            "removeFromFlightPlans":{"create_struct":{
              "type":"delete",
              "path":"flightPlans.{service_auth}"
            }},
            "removeFromTerminationCommands":{"create_struct":{
              "type":"delete",
```

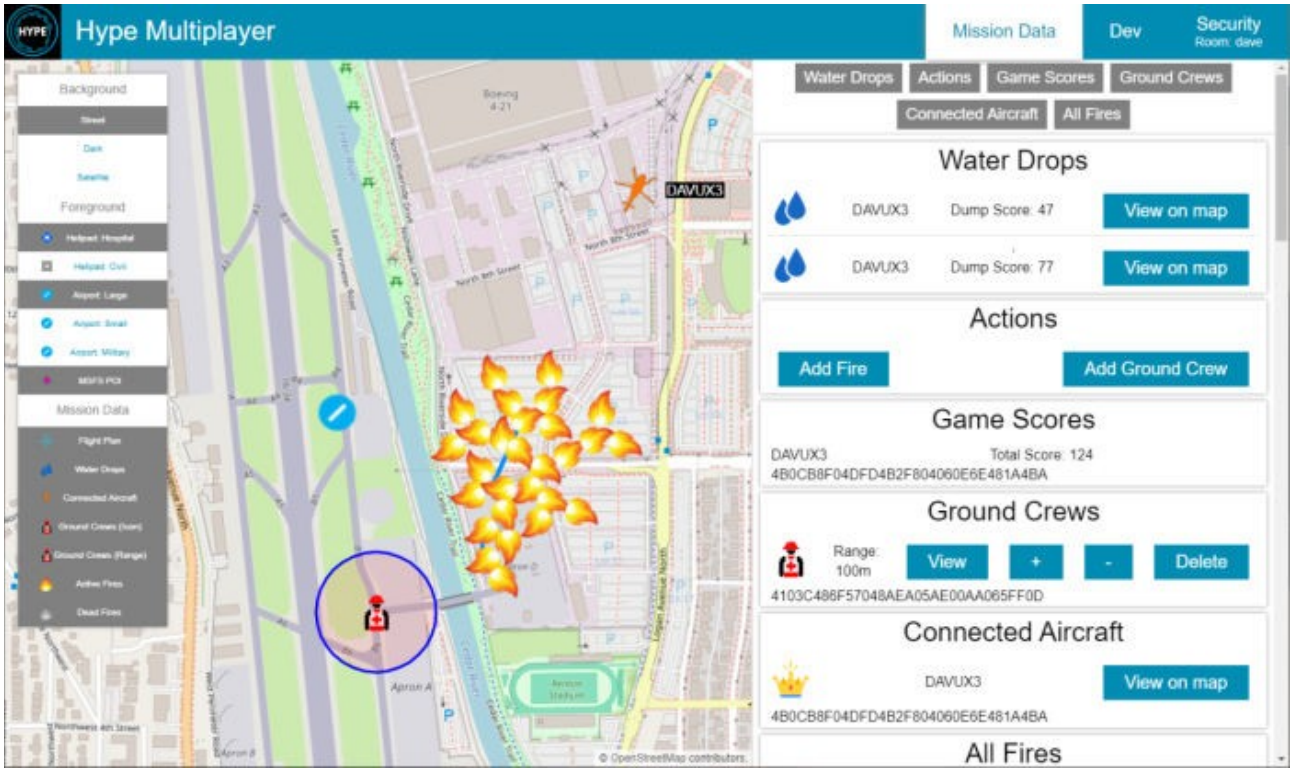
```
              "path":"terminationCommands.{service_auth}"
            }}
        }}},
        {"#comment":"make sure we have connectedAircraft table. all players must use no_overwrite when ensuring the table exists to prevent anybody
from destroying the table."},
        {"set_shared_data":"update", "path":"connectedAircraft", "policy":"no_overwrite", "value": {"create_struct":{}} },
        {"set_shared_data":"update", "path":"icons", "policy":"no_overwrite", "value": {"fn":"get_mission_icons"} },
        {"set_shared_data":"update", "path":"flightPlans", "policy":"no_overwrite", "value": {"create_struct":{}} },
        {"set_shared_data":"update", "path":"webConfig", "policy":"no_overwrite", "value": {"static":"webConfig"} },
        {"set_shared_data":"update", "path":"gameScores", "policy":"no_overwrite", "value": {"create_struct":{}} },
        {"set_shared_data":"update",
          "path":"connectedAircraft.{service_auth}",
          "value": {"create_struct":{
            "location":{"resolve_location":"$USER"},
            "UserName": {"local":"mp_userName"}
        }}},
        {"set_shared_data":"update",
          "path":"gameScores.{service_auth}",
          "value": {"create_struct":{
            "UserName": {"local":"mp_userName"},
            "Score": 0
          }}
        },
        {"#comment":"update our location, score and flightplan (if changed) forever"},
        {"while":1,"eq":1,"do":[
          {"sleep":5},
          {"set_shared_data":"update", "path":"connectedAircraft.{service_auth}.location", "value": {"resolve_location":"$USER"} },
          {"set_shared_data":"update", "path":"gameScores.{service_auth}.Score", "value": {"local":"MY_SCORE"} },
          {"if":{"json:stringify": {"local":"$FLIGHTPLAN"}}, "ne": {"param":"FPL"},"then":[
            {"set":{"param":"FPL"},"value":{"json:stringify": {"local":"$FLIGHTPLAN"}}},
            {"set_shared_data":"update", "path":"flightPlans.{service_auth}", "value": {"create_struct":{
                "points":{"local":"$FLIGHTPLAN"}
            }}}
          ]}
        ]}
      ]}}
    ],
    "mp_initialize":[
      {"#comment":"setup for multiplayer operations later"},
      {"set":{"local":"MP_LAST_ERROR"},"value":""},
      {"set":{"local":"MP_MODE"},"value":0},
      {"#comment":"MP_MODE 0: undecided, 1: offline, 2:online"},
      {"#comment":"these are for debugging only"},
      {"set":{"local":"MP_MSG"},"value":""},
      {"set":{"local":"mp_room"},"value":""},
      {"set":{"local":"mp_password"},"value":""},
      {"set":{"local":"mp_userName"},"value":{"var":["ATC AIRLINE","string"]}},
      {"#comment": "Create or access a unique ID to identify you on the server irrespective of callsign"},
      {"set":{"local":"service_auth"}, "value":{"fn":"create_guid"}},
      {"create_thread":{"commands":[
        {"wait_for":{"local":"MP_MODE"},"ne":0},
        {"call_macro":"mp_begin"}
      ]}}
    ],
    "mp_on_message":[
      {"#comment":"param - msg"},
      {"#comment":"handle READ, UPDATE and DELETE operations below"},
      {"set":{"param":"json"},"value":{"json:stringify":{"param":"msg"}}},
      {"switch":{"struct": {"param":"msg"}, "path": "type"}, "case": {
        "read":[
          {"set":{"local":"MP_MSG"},"value": "we got an read: {json}"}
        ],
        "update": [
          {"set":{"local":"MP_MSG"},"value": "we got an update: {json}"}
        ],
        "delete": [
          {"set":{"local":"MP_MSG"},"value": "we got an delete: {json}"}
        ]
      }}
    ],
    "mp_begin":[
      {"#comment":"called once we decided if we are single or muliplayer. MP_MODE 1:offline, 2:online"},
      {"#comment":"offline case, manually run the logic and complete logic"},
      {"set_objective_title":"Ready to play the game!"}
    ],
    "mp_aborting_mission":[
      {"#comment":"we want to clean up our multiplayer connection if it was created"},
      {"if":{"local":"MP_CONN"},"ne":null, "then":[
        {"set":{"param":"unused"},"value":{"struct":{"local":"MP_CONN"}, "function":"Close", "params":[]}}
      ]}
    ]
  },
  "objectives": [
    {
      "title": "Setup required",
      "commands": [
        {"set":{"local":"MY_SCORE"},"value":0},
        {"call_macro":"mp_initialize"},
        {"sleep": "forever"}
      ]
    }
  ],
  "icons":{
"wp_blue":"
6GSURBVGhD7ZsJWBRHFoCr55IBhvtSTg3gGjWJGrJqvOIaY7KajUk0EjUe6EYNJqioWRUVQxBEFFS8QPF2UYzRKKeigEIUEEVQDpH7ZmYYmnOu3u6emuEadEbapXc//
+9r+r2qpuf1q+t1VTVQR1BE9GIovoQ0DIOSCgY8d2G/
sCAEiq+TwxFXJxE2nYpJmQ6TuuCV8PS4OptfE6vvlKZC8SX892zqjF0N2g7Fl4Dbh+SLNChrQxRgIgOodICQT8gVATDCCIFpXenVAxntmLrMnj8t5wCuPR8qHfS4cJm04kmrhZl6M6hFw2JV22S
6c/TYCXPAbIUaFUDrTsU/9SD1PqN8XA0eu8sj+wUGWlrX1onJf1Qe6e3YuYJzJkT+d7Lqwi55aTJs9Z+195cvd3ux60KiH/mA7AbVPwbF5ASTGZ1uxq1DsdBLsdPIdK0hbtL5/AI0KmW8GpN/
Dz67MpkU+sqGDRuYmlinHVTe0Nf3VyhRBPKMjy1dugRqVIA/7qqUinSoUQDhvzSpRj7UsNoQMOH5xWh0Qx2bnj1pb7z0hnPmzAG74xMmAkBRrdkeWxBPnL9Kq64kE/pMmpg07cCB/
XreAYHDyTRNCY56FAhymlQ9CnEYiZoVDZkgV9glj1PViJ2LSp0JcyGdLliSVJLj6uqq8Qi1e3+InWFDubzjHpKejt61N8jUtqa+vfMPwaxOPy7BPFNqby9ZsliLaqfkCf6YOKd+uzbU3f0HNpnW
F8LicjcTZ3NBo4xMoILjsZnfd3n0voI8g+OLBmjkVExoCKWXo3kpcVug8GI0vuHesqJDUOw7xg0odSWsDa73K0sorQ2UkyYna1f477e79RevDrWRL/Qe21oIJINMKLn3K/
Q96jl841kgcba2F0kkFcZkGr3IQvHiFWNHws/YEJ708vLiwRyagBvlkVx8Wym7pgtLSZkOhJy+NEVZ/0jy4UBLF/
RRoWJQhhyOfri53wycOXMmOHLhyjif+OdReqgANwyPeXBjnMtFzfASBYSBxJHfhDlU8dv2Xc/w2eLtbQpzNYbsCpbfq8sOH8QeLq0m6nUvDVuvAcxpkVRNrS0PKUmK2Ofn54/
CHI2YPXs2cF22ZlS6sfOu/W/pTGst1cNTWYrM7jBbgLlDu9z/XqGbqq9auXIFa8wn33yebjl0y3EH3ihxORdPVRNNs1qByWCxfFM6/9+M4kT/tcuWZsEctXh4/
KjrMuuf824aWG88aclxltUQ91XTRbKawEAbuXRjZvVx/ap032WL5pfAnFdn7ty5IPzi1ZFEURqJhF3HVzxtddyT5K1badDdvFUpaiHrGyToZsmZznq/
ExwcpEsY9P33yxVBbUYz3kDa6WMgCW7g/
LS6IqXscackkZTpArOkjuxWDkT88Q9aFa+SQ5cTFhKGcWtF9BpFukAYhh+2NSIxTOkzlIVbChRTlD9FnHYiBboRHJXpT0bVFKLZxIuG7LsY9e0QC4PHaWkUTjhRCl7/
vknnV0CNhuAGIvmNlBYxZY0k+HQkuTaFiagdeikzUGQ52hOK9ISHoh1TXRRCmQfR3AEIYCr658CDoW+TAgVQ2FGzgZVdGxkTyhwnbiGTaAVetIGxT8KIs5mAT9nkECUeDAk5QL5cYHkJvwIgAfW
lXApLhgJCo9JUc5y6tXg/SGFDoeRJE81sNkARLM0X3oMiJVBi4DkznSHK1YphZY/IBapFixZSO3PWJ/AIhlks6ChWvIjDLt/
```

```
8O9T6BEWVGQGLq9ueQoUkw2qoNxT7BGWtbXxd4XYokhyz446BYv/iuyvAvEerfdCCt2QZZS1ZKzZs3WJ8MCpjMxH3IfnEpGXP8ddMIJIp0iWYTjUf80gpTwqPjJ305ZdfaN1w1P4D/
gKOTJj13fR7pk47jjrofCAu18dTewm+DRrAd8KWwlNjrR1hCrh7967JzBHv1QsL2AjAelmg0m0G1ubtkg0ZVYebcmJ3b/Zcp37C83TU/
bkWfL4UPCaKBXqj+5EuxohoJSCh4vLZKzdGEVNw2kLM0YRcz9jmUCVo67yXoechx1ilKLbsXu0T4v8QxZpmz6c0G1Il+yGTf4zz9Ib35tVrKFpA7cDNzY0x/
mu3WUn69rtO8njOQNIz0PXPPDZWVcT7jxy3YQ6ZtG7He8bu1RVslrp/IDpjpmULWFTdnje9uc437drRiMDAPS/cvrQv/
NQ77dbjNu10MZ8jKOYwgJSYfuuOHLAHNoMl5a2PR1Vm785Luno+KChYQuS8tNIuXLiA8eGseVMLbFy2HHTUmdxaQvxAd4/
jheDCUN1rfdLzmADu4E+g2gG3EdiYiyWbHtYEiAuS93i4r9R8Bf5V2Lhxg+6PUekJZN3pBKtcMYl+6reYMfPmzdW65VKKt/d2BmGM986dHfPPRB+Y09w//
aBacAPDbhbugBqp29QIKJuf6Tu4QQ7VaBvUSH3P9UcHoNYnKBqLW0FxLRhASEFBe4leHaDZMbuIMy0wFzQphjYcvDP2UspUQIkHf35QFwFFcMvSdiUUKYESAxmFtwKgCCL1WQOhSDPwYj0SGTOT
GEsZxT33u74qlAWsBA8HjcSjaAQsqBMXwqQ+Q6mBYXbc0cT5b4KynWQCBVBnILcBSKoUWwFybp4/TQp0YjAR5xHdC4VdDAFlHtyYXeMHRUqhzMDm3IQghUSjIbgHePFyazu9wFMApa2YCFx/
zq46ApX/b8JiszzJuBLv+IOvP6BP4PG/gpkAhe/+GEZuLqEpFDcRajhxIlyv/jlTZVtzri7yS0CAPVRpBS0dKLb+wANgnd/uBwDDkZ/
+BBVa0b8v1L3wdXpddSRmZglVEnvbRnGJlSH5UkYnaFcD16/3ZETqsro4j6CknsnZtm0rpZsDqIB2DnQe/9lXoNmIlJnmDYA9UEjKoFUPDBn/xQKFQh9o58D7Vs5eUARLasRPFpW25EAVJBnb/
wxF2kC7PpBdzsckVeTHhiC8OmVeu1TMXmEzhXw7ZFo0Apm9IS37bVqwL/zkMEXwjMd/
eYptNytXrmB2Xq04dObiOPJimkCrJozYTV2vNMlRX0ZuoDt06LDMwRRTLXAILEbTcJsGTbDgN0mVNe1YfA7uTAUhMdk7lek8tIFWbyW0qYG+vr/
q1j5HFGEKqxUUJZ5VrQxgxSmBACgWDtBcHcTHz4+mM7f9yNGYLMVXHfhhVS8i1w47YyhqVO1X23s9U/EFOg2gTQ2MM7VyhyLYlFlzFIoqvNIbf4ci2PO+/
Qoo9ju0cODatWuQSD2mlUKTgNbs6B7TV4Z1GXiaYq6xrJ7N8fLaTAvb+zWmIhzHM7ewFQ+f7L1z4LjFRJqObT1oszJXaxdSyMcwgSJG3M5/cgbLvuYtF6GFv/
ziQ+ksrjZQ5sBVq1bqOI4YPcTsLReXKjlnxlNT89FXnVmOogYGQ8ZnASDv5dOlbqxvK44LmKhmewPOoj8r8k8yB2nwWQDuT0YbYBhLgKGJTP5FASh05tdk2bGkf/
CLMtIKsx88Dw7e17Gc3QfUPtGmTf/iDX7nr06Igd3kWibvo8dmJu/+4YzYttRxELkAdwbGwa/Sxvd45IGPrBxLBJhwJNLZheKcEY2NtzlNZYmYsDg1NTmxLjz8hFa7XBcvXsQa/
8kMc7a+zdhGHYuPc3hGE644cd4WtLKZkjrcNnKzizY2EpVYAhAjMdAxaAO2Ek7L5yVo6lBUGAdERXefP01/+iD1riA29qbicghyLOLKGDenGelARjhFE/
AfYrYBpqkEmOkjspkFzQXvos1Juq1Vt9lNtcl3bkXXhIaG9RhF6YC7+w/sSVM/G9jAMpzWqmszIUOfMynamecgQAGTbCUyYrZMQ6czpMA/
7cT7yPz5ruBD1zXrE6wGr40dOsASrZAhANXFr9DCoYgYICZSoGcuxT7PR8pGCAVZlrLmZNBcmpz/MCXff4fP691N1AvEd6POI8cPbdO1/
CjXwHTadTtDl1JOu16baADAGoidUsShaS3F64RuC9C1YAJLqbh9TW6tT3X8SR9t6jjw8PhJ12XKDNsmlunHrVzLsfd47ImxTvo2wnoJA2vEnS7XgVdqAt5iue2AbYLXZLZM+mWhNGuksCKiKSch
wNPTU+2gcOHCBWalwfA9+cYWE38bwhzJlzBZEgHuhFai5mgxKDPaAWLQAozMOfLpBU2VH6CSVG5LZRq3qSouNzOlMD+/
oOny5cvw4hejlQNfhXXr1jIdh71jA0ztJon1bCY85OlMjnbiOfLbmExJLf7zMqK2dzz8AlBXdMbFYghUu2BV3yCpLjLs9HUvMVi0AIapGJgaMGSf5jcXjUbFydzmsjsyflFyXlZGMT5Y0LI7oZS
AvUFWxM5b4i0DKVS/d2X79m0DwEPl7l0Z5hsSMhJmvYGAXJGHr2pHL0Z/BpNVHLz654/
KfE6FUK2T+wPavMq55TWkQBE8GPQ2+eFEZ+5YOayFIlhW2vYAiv0ObRw4rDxruyIWA+DYYN57pNCJ82Zs1brweP7zbVDsd2jjwIcJ1+IRY8UCkqTCABw4eVbVxx08EzkB4ys2fwEjFKRGn72uUN
7QhW/vV5co+7mdsXmXYDLYFV9yRZk+tBxtgslv6E5oVLq70lE8tE4182wk6vh2NTQum1Yrc7RpwgTysvQjgK2oYGi+AeLj729xKOy4aUM+SxGvMltBQfL5UFKmCa89kNYW69pGcUUJj/
wi5ZAwdxc+rMhWGf/lX4Ru5MyXNxia0W53Aq3Yf6PwgLK5OlSj7fbVgnal7hebdwFeRhto1YQJ5AW3/
ACQknJxDYNTgh+kAuSA9SyRsv3hVEG7JkygUyPA2kph2AJhDRICqbUJ7eylXQ0kWF6ApkJRxdKytiwo0gpaOnBE5WPVW4mScd0+nqYLtHRg4u9n4xBjEdRweI0g8VKYalnzDRrwzf3KMuXo61TR
2AKTaQdtY6oJDDR4gbU1b0ZLxR2nqyemJCYmwhw6AcB/AID93zD+9+L6AAAAAElFTkSuQmCC",

    "h160_icon":
"
7ZlpbFVFFMdPdyhlbUG2KDbIYlmUJSCLoEYU06AYIdHIZhUriIUiglFjTEz8AEhViBLlgzHBoBKNGPWDQdAYRYmgAdGQKsomINAFuvfV/
39m7ut99713371viyb8kpNZ7r0zc2bOzJyZK25kmVACa6QjJ0Mu7T4u/
ZH8Vefa6MALFJOUTBOSOSYkXU0YRp4J04Jp6Y06FdpaiwMm7Hxo6XeyXFpVhhN7B5BIxQZxfRiJ1SYMoaVSAgi66VQUTLsWMGxdLR1tEP2kkwwThmFXKmN9+HtKj6cnyQfmxeAI4+XHGQZC6xoK
WQKZpVLRcHalnag9Z31kwr6Me8atxqQx0YROBseqPQeiexMvWi8Hngz/
yNk5tNTXIaW2gbsjc0OUcXTQDjmio4prTBhCtOFoMCGpN2EIzib0gF3WtsGc87JNDoAV3YTgO53qJMN0wqM6qWGeoydZyfT2CB0VAnsy1hBEY9qhxdJs4r7oAZmmo+GEjQ9xNjHSfHRjXHFPecH
EfcFainQ0PvqYMClEXQAMj0Ce0NEQhpv+W6GTGpMX3AviJZtGtnaifGjSQ1mwJUhzzRNjiHcx7oafUS1ABfU1TSK9upgcG7AQLoP/
6FR0fJkRyMI61Ob8yGtlJNYYOimO0sLbTRgTq0LuTwN0VHIhSzk+XFUR5xj14wNw1IROPoeMzM6UtdbYNq5UW3AF5DYI/
aO4GWAVSkF6rs52Jd+8GzcPsYBIPkGqGGJafL9Oesev0dDXmAKx9i3aENPjVCqFWPv5fTrpHb8aWljf2XbOK3ijN8Tv8hc/xjDoz3li6kB5jt/
UrIjPycg0FXbXyU444c2znjpHMc0sBAmdx2aZgi2XgVa5jHmUJSXyMdJqDTbvjWI8UaaawpSrYTl0liCLh9kSE4853l7m4TfY72aiwPOI52WEF8lF4BDeGYE4K3XF68Tfm7leKpoqBft9GFt/
vyhtCH/TSXd8mfylCgl0yw3/BtrR6fBkmb6WtoJX5BYTDYLKXkLgeRr40pAH6ZysiBp6LsePhnmwCF8NjIRbhXw2/KmJ8r6ZAk1ln8ke/
SgUc4pcDBkMcW2U9ZAb61+QFkgxCqim+Z9vkPaiLeq4dhDCk+cMVBxSaV2zdPR8Vc3RKY2r5JMu2brMfSflxOTtsgZRHjh/
gag7FqvCYShImXUbzqQ5L0sJonzJSQ7eY6OCYPyGIXB6cwUQllFYOV7KNt4q9+I9LhABp/pMu07eBrh/XXM6uw0FsetjTfgFEF6j/
aBSPhltxrSjd56sM3me8TUPDcGuvtgs75qoZ+KpkDcPFp7OE3biqdB+TWW/
ufBEXBWerFeLNYllLAnBzZdzko7vZjPZGWcefR9POKeFG6WwzF0mHgLmbgfmbly95QmbT+OnwYq4WpXV+ZVnj84irgrbedTURL6DTgGzTz+WvsqukCr6nChXXuODkFQdE7mF8jRRDBkDudrkpYX
5nPrfPiDViHs5vrBhoyG8Qr4b4rYY9ji4UBpZPm/wds1Vvshk5qunaWKC7fK/DEInJhL5ECqzju/aBXkzzHO7BRTsmS/nzPNyCK/F4sb3wuuADZuObWo3XcrT2CMHvqHczT/
VU13+2OZKOZAbxbDggC1FsB1yGUJLKIdyVRfgnhZukUKkayFxk6iCFlz/
x2PH2odNJAONfhhpNroRwmPllwgjcqxGAte+qc11R6nsmz9SRuD7Z5DcDKljfiIkS0ELmtM9UGhHBwwsc4PMLBslS96aLYv043Bohzjmljbhg MB/
QVBuHrI+glgeU0IkW0GLvgcWyNEb+ofcp8QEyvGv4H6dSg6pUpCw7KmYn19HuPEIobFVOvKr1N3rcZ2TPFLnQooUtVTKV7GUI+bUN1ankksqFeQK68dCwq4Ok0GyFKTbzZ9OEyCLzi2XttbVcir
Hn88xCMLfNgnte0789DBHhPehvKkoaVgpm2BawQ7iwXT/KWmatF39MOGlwWnIzVhReY/qSgP89m5V8iw65cVsW5dzJa5rlkCv12QZkrxy4f56AcLtx9Mqa1ewX2ul/
J2VJRnVF6T1um3Kx6wZf5XcuX+hrNKviOBZy9BtytWiEmcgrCwahRjNs0X57paC1ZN/AL/QqSD8ht4Rf+YNgVxfXyGbCnIjl2W2G25HOyF0GhSRRpB/
DzlKXLJpLochvBzhHXDn0cw7HMW9Jh4GOxMdRo+lXufEBT2gOajnvcutEiiokoFIs/N9mWi80LQXovJtOtlJG7orZ6M6KfyhcxKGI87BOQE5ywxX00kSnCs7T9WF3FgpoNxMBMlSjlyC/
AhRypF0KEhqB22VOSauqGlS5v6zTqWOdClIaDZBJrwjzyNgj6eUdCp4prlNLXaK6hrlc6b8YiedCtbuOCLfmzhRq1yqSdUqOgan/Z/
8Fr74U9n09mF1FnTbW32RKgW5D3Fztv+v6dovX5aeWa4cCG7uvOawLzJ093g9T48l5D/P/
4VR1h0OpUuWrEQe98iUks45GFxgyKACddmUctKp4HFs9sFVs7pW3dMk5Vriv0T33fPkGEKeSq6QOCL/Ap3jTWX/GRgjAAAAAElFTkSuQmCC"

  }
}
```

Multiplayer: Web Client



The Web Client is the website used to manage the multiplayer room. The Web Client will read the `webConfig` data from the shared data and create an experience based on the configuration.
You can show lists of data, click buttons and open a dialog with various list/text/slider controls. You can also have icons and text on the map as well as lines and range rings.

list (WebConfig)

The `list` widget will create an entry for every key within the source object. Each entry can have multiple rows, and each row is made up of items which can be of the following type:

```
Text
icon
button
```

map_point (WebConfig)

`map_point` will generate an element for the specified location. The element can be

1. `range` ring. Always used if `range` is specified (even if zero). Not compatible with icon or text (they will be ignored)
2. `icon`. Can be any URI including a data uri.
3. `text` which will be displayed next to the icon (icon itself is not required).

See `stroke` and `fill` to style.

map_line (WebConfig)

`map_line` will generate a line string from a given array of points.

See `stroke` to style.

event (WebConfig)

`event` can be used to execute a command list when keys are added/removed/updated within a given source. You might use this to play a sound, to activate a dialog, or even to execute logic or update locals. Most logic should live within the aircraft but this tool is available.

Dialog widgets

You can use `show_dialog` to display a list of widgets in a modal fashion. Normally triggered as a result of a button click.

```
Text
icon
button
textbox
listbox
textarea
slider
```

WEB COMMANDS

•The web commands are a limited subset of the mission command set.

•Web commands:

```
•set_map_center
show_dialog
close_dialog
•play_sound
```

Available commands from the normal mission command set:

```
•set (param and local only)
sleep
if
while
switch
try
for_each
modify_array
create_thread
throw_error
debug_write
break
continue
set_shared_data
•fetch
```

WEB QUERY

The web query commands are a limited subset of the mission query command set.

```
•create_array
create_struct
struct
string:split
string:join
json:stringify
json:parse
json:copy
object:keys
create_number
param
has_param
static
has_static
local
has_local
rand
add
add360
subtract
multiply
right_shift
left_shift
remainder
xor
exponent
divide
round
tofixed
floor
```

```
Math.* (same)
ceil
abs
clamp
scale
and
or
not
if
switch
require
compare
text
typeof
isNaN
parseInt
parseFloat
convert (weight & length units)
no_resolve
fn.create_guid
fn.create_date
fn.get_time_string
fn.has_selected_poi
fn.selected_poi_info
fn.selected_poi_location
•fn.is_dialog_open
```

Supporting multiple languages

To support multiple languages in your mission, do the following:

1. Define a `data.translation` table for each language you want to support beyond the default language
2. Populate the `data.translation.Language` table with the keys being the default language strings, and the values being the string to use for the specific language.

`local:$MISSION_LANGUAGE` contains the currently selected language name. If it is null or undefined, then the default language will be used without attempting to swap to any other language.

Keys that aren't found in the target language will be rendered in the default language.

Examples:

```
"data":{
  "translation": {
    "French": {
      "hello world": "Bonjour le monde",
      "hello world {0}": "Bonjour le monde ({0})"
    },
    "German": {
      "hello world": "Hallo Welt",
      "hello world {0}": "Hallo Welt ({0})"
    }
  }
},
…
{"set_message": {"text":"hello world"}},
{"set_objective_title": "hello world"},
```

Translation test program

```
{
 "title": "Translation Test Program",
 "data":{
 "translation": {
 "French": {
 "hello world": "Bonjour le monde",
 "hello world {0}": "Bonjour le monde ({0})"
      },
      "German": {
        "hello world": "Hallo Welt",
        "hello world {0}": "Hallo Welt ({0})"
```

```
      }
    }
  },
  "briefing": [
    {"text":"Language selection:"},
    {"buttonbar":[
      {
        "title":"English (default)",
        "commands":[ {"set": {"local":"$MISSION_LANGUAGE"}, "value": null} ],
        "select_condition":{"require":{"local":"$MISSION_LANGUAGE"}, "eq":null}
      },
      {
        "title":"French",
        "commands":[ {"set": {"local":"$MISSION_LANGUAGE"}, "value": "French"} ],
        "select_condition":{"require":{"local":"$MISSION_LANGUAGE"}, "eq":"French"}
      },
      {
        "title":"German",
        "commands":[ {"set": {"local":"$MISSION_LANGUAGE"}, "value": "German"} ],
        "select_condition":{"require":{"local":"$MISSION_LANGUAGE"}, "eq":"German"}
      }
    ]},
    {"text":"hello world"},
    {"text":{"text":"hello world {0}", "params":[ 99 ]}}
  ],
  "objectives": [
    {
      "title": "Done",
      "commands": [
        {"#comment":"adding a loop here only because it won't re-evaluate ever otherwise"},
        {"while":1, "eq":1, "do":[
          {"set_message": {"text":"hello world"}},
          {"set_objective_title": "hello world"},
          {"sleep":1}
        ]},
        {"sleep": "forever"}
      ]
    }
  ]
}
```

Server (Remote) Missions

Missions can instead run on the server and stream commands to the client.

When the user selects the mission your server will be contacted and at that point you will be able to manage the mission system indefinitely until the user selects another mission manually.

```
{
  "title": "Connect to Server",
  "aircraft": ["H145"],
  "api_version": 0.1,
  "url": "localhost:9998"
}
```

A mission server may dynamically generate and apply mission descriptors as well as send other commands and observe status. The server is essentially just a websocket server which listens for the simulator to connect and then speaks a JSON RPC type protocol.
A very simple Mission Server Sample in node.js is included in the Tools folder.

Command sent from aircraft to the server

Message	Description
{"control_msg": “hello”}	After connecting the H145 will alert you that it is ready for you to send a mission
{"control_msg": “canceled_by_user”}	The H145 is alerting you that the user has selected another mission and you are no longer active. The connection will disconnect after this message
{"remote_notify": "my_message_name", "params": [0, 99, 3]}	Use of `remote_notify` command will emit events.

Commands sent from the server to aircraft

Message	Description
{"load_mission": `MISSION_DESCRIPTOR`}	After connecting the H145 will alert you that it is ready for you to send a mission
{"exec_commands": `COMMANDLIST`}	The H145 is alerting you that the user has selected another mission and you are no longer active. The connection will disconnect after this message

`MISSION_DESCRIPTOR` is simply the JSON (already parsed) which would normally be a flat file mission.

`exec_commands` enables you as the mission server to intervene at any time with logic.

Action Pack API Reference – COMMAND

All commands are listed below.

#comment

`#comment` is used to add human-readable information within command lists. It has no effect and runs instantly.

Examples:

```
{"#comment": "This section of code is very delicate"},
{"#comment": [
  "This section of code is very delicate",
  "This section of code is very delicate",
  "This section of code is very delicate"
]},
```

sleep

`sleep` is used to wait or delay execution for some time.

Examples:

```
{"sleep": QUERY},
{"sleep": 0.25},
{"sleep": 1},
{"sleep": {"rand":[0, 60]}},
{"sleep": {"var":["L:MY_SLEEP_TIME", "number]}},
{"sleep": "forever"},
```

wait_for

`wait_for` will not proceed to the next command until the comparison between QUERY_1 and QUERY_2 is satisfied. Format:

```
{"wait_for": QUERY_1, "eq": QUERY_2},
```

Where eq is the operator, and it can be any of

Operator	Function
eq	Equal To
ne	Not Equal To
lt	Less Than
lte	Less Than Or Equal To
gt	Greater Than
gte	Greater Than Or Equal To

Examples:

```
{"wait_for": {"var":["L:MY_TEST_VAR", "number"]}, "eq": 1},
```

if

`if` allows to check a condition (one time) and then proceed down the `then` branch of commands, or optionally the `else` branch of commands. Once the selected branch is executed, processing returns to the next command after `if`.

Format:

```
{"if": QUERY_1, "eq": QUERY_2, "then": COMMANDLIST},
{"if": QUERY_1, "eq": QUERY_2, "then": COMMANDLIST, "else": COMMANDLIST},
```

Where `eq` is an operator and using the same list as `wait_for`.

Examples:

```
{"if": 1, "eq": 1, "then":[
  {"set_message":{"text":"1 is always equal to 1}}
], "else":[
  {"set_message":{"text":"this never executes, since 1 always equals 1"}}
]},
```

while

`while` enables to run a `do` COMMANDLIST until a condition is satisfied.

Format:

```
{"while": QUERY, "do": COMMANDLIST}
```

Examples:

```
{"while": {"var":["L:MY_TEST_VAR","number"]}, "gt": 1, "do":[
  {"set_message":{"text":"this message runs over and over while L:TEST_VAR is greater than 1"}}
]},
{"set_message":{"text":"this message runs once, after L:MY_TEST_VAR becomes less than one"}},
```

for_each

`for_each` is used to call a `do` COMMANDLIST for each element in an array. `$index` and `$item` params will be defined for each iteration.

Format:

```
{"for_each": QUERY, "do": COMMANDLIST},
```

Examples:

```
{"for_each": {"create_array": 4}, "do": [
    {"set_message": {"text": "my array item: idx={0} item={1}", "params": [{"param": "$index"},{"param":
"$item"}]}},
    {"sleep":5}
]},
-------------------------------------------------
{"set":{"param":"my_array"}, "value":[1,2,3,4]},
{"set":{"param":"my_result_array"}, "value":[]},
{"for_each":{"param":"my_array"}, "do":[
  {"if":{"param":"$index"},"eq":1, "then":[
    {"continue":1}
  ]},
  {"if":{"param":"$index"},"eq":3, "then":[
    {"break":1}
  ]},
  {"modify_array":{"param":"my_result_array"}, "append":{"param":"$index"}}
]},
{"set_message":{"text":"ret={0}", "params": [
  {"json:stringify": {"param":"my_result_array"}}
]}},
```

try

`try` and `catch` may be used to trap an error which would otherwise result in a message to the user. `$ERROR` will be defined with the error result.

Examples:

```
{"try":[
  {"set": {"object":""}}
], "catch":[
  {"set_message":"oops! {$ERROR}"}
]},
```

switch

`switch` is used to select from a set of known results (each a `COMMANDLIST`).

Examples:

```
{"switch":2, "case":{
  "0": [ {"set_message":"You selected 0"} ],
  "1": [ {"set_message":"You selected 1"} ],
  "2": [ {"set_message":"You selected 2"} ],
  "default": [ {"set_message":"You selected another number"} ]
}},
```

set

`set` enables setting variables in MSFS and in the mission system and on mission objects. You may prefix the MSFS events with `K:`, the list is available here
Examples:

```
{"set": {"object": "my_object", "var":"MODE"}, "value": QUERY}
{"set": {"var":["L:TEST", "number"]}, "value": QUERY}
{"set": {"table":"my_table", "key":{"text":"blah{0}","params":[99]}}, "value": QUERY}
{"set": {"local":"my_local"}, "value": QUERY}
{"set": {"param":"my_param"}, "value": QUERY}
{"set": {"global":"my_global"}, "value": QUERY}
```

`trigger`

`trigger` is a shorthand which is intended to be used to send `H:` and `K:` events to the sim.
You can send all of the HPG SDK events to the aircraft, and any of the applicable Sim Events.

Examples:

```
{"trigger": "H:MY_EVENT"}
{"trigger": ["H:EVENT_1", "H:EVENT_2"]}
```

`call_macro`

`call_macro` will synchronously call a macro by `name`. Macros can be defined within the mission or some are built into the product as "system macros".

Macros that use the `return` command will have their result available via the `$RET` param after the call is complete. You can change `$RET` to another param name by using `result`.

Examples:

```
{"call_macro": "do_it_now"}
{"call_macro":"my_calc", "params":{"num1":2, "num2":4}}
{"call_macro":"my_calc", "params":{"num1":2, "num2":4}, "result": "my_result"}
```

`return`

`return` is used to set `$RET` on the calling context, when the function returns.

`return` will also stop processing further commands on the macro (except for threads, which keep running).

Examples:

```
{"return": QUERY}
{"return": {"param":"my_ret"}}
{"return": "ERROR"}
```

`break`

`break` is used to escape from a loop (see `for_each`). After `break`, no more iterations of the loop will execute.

`continue`

`continue` is used to escape from a single loop (see `for_each`) iteration but still continue on the next iteration.

`private_macros`

`private_macros` enables you to provide a list of macros which is visible only within that scope.

Examples:

```
{"private_macros":{
  "my_macro_name: [
    {"#comment":"macro commands here"}
  ]
}}
```

create_thread

`create_thread` enables running code (a `COMMANDLIST`) asynchronously.

- `commands`: required.
- `interval`: optional. default to 100ms

Examples:

```
{"create_thread": {"commands":[
  {"sleep": 100},
  {"set_message":{"text":"this runs 100 seconds later!}}
]}}
{"set_message":{"text":"this runs instantly and the next command continues}}
```

create_event_handler

`create_event_handler` enables you to listen for MSFS H:Events.

Examples

```
{"create_event_handler": "BAMBI_BUCKET_DUMPED", "commands":[
  {"set_message":{"text":"bambi dumped!"}}
]}
```

throw_error

`throw_error` enables you to create a custom error.

Examples:

```
{"try":[
  {"throw_error":"my custom error message"}
], "catch":[
  {"set_message":"oops! {$ERROR}"}
]},
```

modify_array

`modify_array` enables some common array operations, like prepending or appending items, or removing an item at an index.

Examples

```
{"modify_array":{"local": "my_array"},"append": QUERY}
{"modify_array":{"local": "my_array"},"prepend": QUERY}
{"modify_array":{"local": "my_array"},"removeIndex": QUERY}
```

reload_mission

`reload_mission` enables resetting the mission without clearing `locals`.

Examples:

```
{"reload_mission": 1}
```

load_mission

`load_mission` enables calling another mission (the current mission will end). The `locals` will not be cleared.

Examples:

```
{"load_mission":"other_mission_id"}
```

create_object

`create_object` instantiates a new AI object in the world. It will be referred to by its `name`. Object names must be unique, and subsequent calls to

`create_object` will fail with the same name. Use `destroy_object` to remove an object when you are finished with it.

- `name`: this is the name that is used to reference the object in subsequent calls like `set`, `drive_object` and `destroy_object`.
- `title`: this is the `title` from `aircraft.cfg/sim.cfg` in MSFS, which uniquely identifies the object.
- `fallback_title`: Should `title` fail to exist, use `fallback_title` automatically.
- `location`: this is the `LOCATIONREF` where the object should be created.
- `is_flight_object`: 1 or 0 depending on whether this is an object which should fly or not.
- `is_ground_object`: 1 or 0 depending on whether this is an aircraft.cfg object or a sim.cfg object.
- `is_static_object`: 1 or 0 depending on whether this is a static type simobject

Examples:

```
{"create_object":{
  "name": "my_object",
  "title": "HPG Airbus H145 Ambulance",
  "location": "$USER"
}}
```

`destroy_object`

`destroy_object` will deallocate an object and wait for it ot be destroyed. It is valid to re-use the object name after this point (`create_object` with the same name).

Examples:

```
{"destroy_object": QUERY}
{"destroy_object": "my_object"}
```

`track_object`

`track_object` will add an icon to the map which follows the specific object. `track_object` uses a thread to so its work, and it returns immediately.

`icon` may be either:

1. data-uri for a 44x44 PNG image
2. a string referring to the `icons` table in the mission, which contains (1)
3. a string referring to a `known icon` (see table below)

Known icons:

Icon	Description
ki_waypoint_blue	Waypoint (blue)
ki_target	Target symbol
ki_helipad	Helipad symbol
ki_medic	Medic symbol

Examples:

```
{"track_obejct": {"object": QUERY, "icon": QUERY}}
{"track_object": {"object": "my_object", "icons": "ki_medic"}}
```

`drive_object`

`drive_object` will send an object along waypoint navigation, and returns only when the object has finished.

- `name`: The name of the object to drive.
- `speed`: Speed to use during the drive, meters per second.
- `to`: ARRAY of `LOCATIONREF` or a `ROUTE`
- `data`: This is `set_drive_data` data.
- `VAR1`: Value to set `VAR1` on the mission object, during the drive.

Examples:

```
{"drive_object": {"name": "soldier_1", "to": ["pax_right_door"], "VAR1": 2, "speed": 10}},
{"drive_object":{
  "name":"tanker1",
  "to":[
    [34.921710973784805, -117.88296989234365, 2200, 100],
    [34.91159609892966, -117.90097049623692, 2500, 100],
    [34.894605381452905, -117.90550330903535, 2600, 100],
    [34.90274380665833, -117.86989409383754, 2700, 100],
```

```
    [34.91631769396497, -117.86277032013513, 2800, 100]
  ],
  "speed":100,
  "data": {
    "use_safety_height": true,
    "safety_height": 100,
    "max_vertical_speed": 50,
    "max_vertical_speed_heightdelta": 100
  }
}},
```

move_object

`move_object` will teleport an object to a new location.

Examples:

```
{"move_object": QUERY, "to": LOCATIONREF}
{"move_object": "my_object", "to": "$USER"}
```

point_object

`point_object` enables orienting an object to point at another object.

Examples:

```
{"point_object": QUERY, "to": LOCATIONREF}
{"point_object": "my_object", "to": "$USER"}
```

set_drive_data

`set_drive_data` lets you configure `drive_object` behaviors after calling `drive_object` (mid-drive).

- `use_safety_height`: Determines whether a flying object is restricted to the safety height (floor).
- `safety_height`: safety height (minimum radio altitude). feet.
- `max_vertical_speed`: Determines the flight objects maximum vertical climb/descend speed
- `max_vertical_speed_heightdelta`: Determines at which altitude delta will result in maximum speed. Values beyond this point will be clamped.

Examples:

```
{set_drive_data: {
  "use_safety_height": true|false,
  "safety_height": 0
  "max_vertical_speed": 0
  "max_vertical_speed_heightdelta": 0
}}
```

set_df

`set_df` can be used to set the active Direction Finder signal location. (DF source on the MFD).

Examples:

```
{"set_df": {"location": LOCATIONREF, "freq": QUERY}}
{"set_df": {"location": "my_boat", "freq": 255.0}}
```

set_carls_radio

`set_carls_radio` will set the displays of the CARLS TACTICAL RADIO in the cockpit.

Examples:

```
{"set_carls_radio": {
  "LSK": ["PG1", "", ""],
```

```
  "RSK": ["", "", "INOP"],
  "Items": [
    ["Group 1", "misc contacts"],
    ["Group 2", "important"],
    ["Group 3", "other"]
  ]
}}
```

```
{"set_carls_radio": {
  "LSK": ["PG1", "", ""],
  "RSK": ["", "", "INOP"],
  "Items": [
    ["Group 1", "misc contacts"],
    {"item": ["Group 2", "important"], "show_condition": …}
    [{"text": {"Group 3 {0}=99, {1}=88"}, "params": [99, 88]}, "other"]
  ]
}}
```

A full sample program is available at Samples.

set_tfm_radio

`set_tfm_radio` works similarly to `set_carls_radio`.

Examples:

```
{"set_tfm_radio":{
      "main": [
        ["DISPATCH", "168.9000"],
        ["BKP DISP", "169.0000"],
      ],
      "guard":[
        ["GUARD 1 NAME", "164.350" ],
        ["GUARD 1 NAME", "168.350" ]
      ]
    }},
    {"set_tfm_radio":{
      "main": [
        ["DISPATCH", "168.9000"],
        ["BKP DISP", "169.0000"]
      ],
      "guard":[
        {"item":["G1 NAME", "165.0000"], "show_condition":{"require":2,"eq":2}},
        [{"text":"G{0} NAME","params":[99]}, "167.0000"],
        ["G 3 NAME", "164.350" ],
        ["G 4 NAME", "168.350" ]
      ]
    }},
```

set_rescuetrack

`set_rescuetrack` configures the DMAP RescueTrack UI.

Examples:

```
{"set_rescuetrack":null},
{"set_rescuetrack":{
    "statusVar": "L:MY_DISPATCH_STATUS",
    "statusMessages": {"static": "statusMessages"},
    "dispatcherMessages": {"local": "Messages"},
    "activate_waypoint_commands":[
      {"#comment":"param - $index - in dispatcherMessages"},
      {"#comment":"param - $command – DIRECTTO"},
      {"set_message":"{$index} {$command}"},
      {"set_route": {"struct": {"struct":{"local": "Messages"}, "index": {"param": "$index"}},
"path":"waypoint"}},
      {"#comment":""}
    ]
}}
```

```
{"set_rescuetrack":{
    "statusVar": "L:MISSION_RESCUETRACK_STATUS",
    "statusMessages": [
      "1. Unavailable for dispatch",
      "2. Ready for dispatch",
      "3. Dispatch accepted, en route to scene"
    ],
    "dispatcherMessages": [
```

```
      {
        "from": "My Dispatcher",
        "time": "00:16:00",
        "text": "the accident site is now clear, proceed to the destination",
        "waypoint": [0, 0]
      },
      {
        "from": "My Dispatcher",
        "time": "00:07:00",
        "text": "the accident site is blocked, enter a hold",
        "waypoint": [0, 0]
      }
    ]
}},
```

open_door

`open_door` will open the aircraft door if it is not already open. if it opens the door, it will also wait for it to finish opening.

Examples:

```
{"open_door": "cockpit_left"}
{"open_door": "pax_left"}
{"open_door": "cargo_left"}
```

close_door

`close_door` will close the aircraft door if it is not already closed. if it closes the door, it will also wait for it to finish closing.

Examples:

```
{"close_door": "cockpit_right"}
{"close_door": "pax_right"}
{"close_door": "cargo_right"}
```

create_fire

`create_fire` will create a set of fires.

- `size`: Count of fires to start
- `title`: Fire object name, such as `Airbus h145 Fire`.
- `showIcon`: Optional. Default `true`. whether or not to show icons for the fires. `L:DEBUG_CREATE_FIRE_DIST_MULT`: (with default) `L:DEBUG_CREATE_FIRE_SIZE_MULT`: (with default)

Examples:

```
{"create_fire": "fire_spawn_area", "size": {"var": ["L:MISSION_FIRE_SIZE", "number"]}, "title": "Airbus H145 Fire"},
```

launch_missile

`launch_missile` will spawn and launch a projectile from one object to another.

Examples:

```
{"launch_missile": {
  "from": "my_ai_fighter_jet",
  "to": "$USER"
}}
```

designate_target

`designate_target` enables setting a target in the targetting computer.

Examples:

```
{"designate_target: "my_target_object"}
{"designate_target: {"location": LOCATIONREF, "alt": QUERY}}
```

```
{"designate_target: {"location": "my_target_ground_location", "alt": 1500}}
```

set_route

`set_route` can be used to set direct-to flight plan on the map

Examples:

```
{"set_route": LOCATIONREF}
{"set_route": "my_location"
```

set_map

`set_map` can be used to:

1. Add, remove or updates points on the map. Points may have an icon and/or text.
2. Add or remove lines on the map

Examples:

```
{"set_map":{"add":{"point":{"location":"$USER", "icon":"ki_helipad", "text":"waypoint text"}}}}
{"copy_location":{"bearing":330,"dist":500},"to":"P1"},
{"copy_location":{"bearing":30,"dist":500},"to":"P2"},
{"copy_location":{"bearing":120,"dist":500},"to":"P3"},
{"copy_location":{"bearing":240,"dist":500},"to":"P4"},
{"set_map":{"add":{"line":{ "points":["P1","P2","P3","P4", "P1"], "stroke":{"color":"#4287f5", "width":4}}}}},
{"set_map":{"add":{"point":{"location":"P1","text":"waypoint text"}}}},
{"set_map":{"add":{"point":{"location":"P4","icon":"ki_helipad"}}}},
```

wait_modal

`wait_modal` can display the (singleton) modal dialog to the user. The user can pick a choice to continue.

Examples:

```
{"wait_modal": {
  "title": "Mission Parameters",
  "text": "Select a sling activity",
  "options": [
    {"text": "Utility", "style": "primary", "commands": [
      {"#command": "use a sleep 0 here to make sure button with empty list still executes"}
      {"sleep": 0}
    ]},
    {"text": "Logging", "style": "", "commands": [
      {"set": {"object":"cargo", "var":"VAR 1"}, "value": 8},
      {"set": {"object":"cargo2", "var":"VAR 1"}, "value": 8},
      {"set": {"object":"cargo3", "var":"VAR 1"}, "value": 8},
      {"set": {"object":"cargo4", "var":"VAR 1"}, "value": 8}
    ]}
  ]
}}
```

set_modal

`set_modal` works exactly the same as `wait_modal`, but does not wait for execution to continue.

set_message

`set_message` displays a message on the bottom of the mission app.

- `align` may be `left`, `center`, or `right`.
- `size` may be `small`, `medium`, `large`, or `extralarge`.
- `color` may be `blue`, `red`, `green`, `orange`, `purple`, `hotpink`, `brown`, `cyan`, or `yellow`.

Examples:

```
{"set_message":{"text":"hello"}}
{"set_message":{"text":"hello {0}", "params":[ "dave" ]}}
```

```
{"set_message":{"text":"hello {0}", "params":[ {"local":"my_local"}]}}
```

set_progressbar

`set_progressbar` will enable display of a progress bar at the bottom of the mission app.

Examples:

```
{"set_progressbar":{"min":0,"max":100,"var":["L:TEST","number"], "color":"green"}}
{"set_progressbar":null}
```

set_dispatch

`set_dispatch` allows setting the dispatch dialog content. This is similar to the briefing but can be changed during the mission. All of the same widgets from the briefing are available.

Examples:

```
{"set_dispatch": [
  {"text":"hello world"}
]}
```

set_briefing_dialog

`set_briefing_dialog` opens or closes the briefing dialog.

Examples:

```
{"set_briefing_dialog": QUERY}
{"set_briefing_dialog": 1}
{"set_briefing_dialog": 0}
```

set_dispatch_dialog

`set_dispatch_dialog` opens or closes the dispatch dialog.

Examples:

```
{"set_dispatch_dialog": QUERY}
{"set_dispatch_dialog": 1}
{"set_dispatch_dialog": 0}
```

scroll_to_briefing_item

`scroll_to_briefing_item` will scroll to the named section on the briefing page.

Examples:

```
{"scroll_to_briefing_item: "header1"}
```

scroll_to_dispatch_item

`scroll_to_dispatch_item` will scroll to the named section on the dispatch page.

Examples:

```
{"scroll_to_dispatch_item: "header1"}
```

set_objective_title

`set_objective_title` enables changing the objective title (text at the bottom of the mission app) at a time other than when the objective list itself

moves to the next objective.

- `color` may be `blue`, `red`, `green`, `orange`, `purple`, `hotpink`, `brown`, `cyan`, or `yellow`.

Examples:

```
{"set_objective_title":QUERY}
{"set_objective_title":"Fly to the target"}
```

set_hover_display

`set_hover_display` enables you to show a target crosshairs on the mission map.

`range`: meters

Examples:

```
{"set_hover_display": {"target":LOCATIONREF,"range":QUERY}},
{"set_hover_display": {"target":"load1_dest","range":0.02}},
```

create_user_action

A User Action is a command available for the user to click, shown at the top of the mission map.

`create_user_action` will create the named user action. `click_commands` is a COMMANDLIST which will be run if the user clicks the button or invokes the hotkey.

Examples:

```
{"create_user_action": {
  "id": "accept_dispatch",
  "title": "Accept Dispatch",
  "click_commands": [
    {"destroy_user_action":"accept_dispatch"}
  ]
}},
{"create_user_action": {"id": "change_accident_location", "title": "Change Location", "click_commands": [
  {"set_message": {"text": ""}},
  {"call_macro": "user_pick_accident_location"},
  {"set_route": "accident_location"},
  {"set_message": {"text": "Accident Location: {0:LOCATION}", "params": [ "accident_location" ]}}
]}},
```

destroy_user_action

`destroy_user_action` will remove an existing user action.

Examples:

```
{"destroy_user_action: "my_action"}
```

trigger_user_action

`trigger_user_action` will manually trigger a user action, as if clicked by the user.

Examples:

```
{"trigger_user_action": "my_action"}
```

set_user_poi

`set_user_poi` will enable clicking on the map on behalf of the user.

Examples:

```
{"clear_user_poi: 1}
```

create_route

`create_route` uses an online service to compute instructions to transit using the road network from one location to another. After calling `create_route`, the `name` will be available to reference with other APIs.

- `type`: Optional. Default to `car`.

Examples:

```
{"create_route": {name: "route-name-here", "query": {
  "location_from": LOCATIONREF,
  "location_to": LOCATIONREF,
  "type": "car|foot|bike"
}}}
{"create_route": {name: "my_route_name", "query": {
  "location_from": "$USER",
  "location_to": {"bearing":0, "dist": 1000},
  "type": "car"
}}}
```

draw_route

`draw_route` will draw lines on the map for the specified route.

`stroke`: Optional. default is `{width: 8, color: '#FF33FF' }`

Examples:

```
{"draw_route": "route_name", "id":"my_route_id"}
```

copy_stringtoken

`copy_stringtoken` copies a string token by name to another name.

Examples:

```
{"copy_stringtoken": "token1", "to": "token2"}
```

open_url

`open_url` opens a web browser window on the user's PC.

Examples:

```
{"open_url": QUERY}
{"open_url":"https://hypeperformancegroup.com/"}
```

copy_location

`copy_location` will take a LOCATIONREF, resolve it right now, and then save it under a new name.

Examples:

```
{"copy_location":LOCATIONREF, "to": "my_new_location_name"}
{"copy_location":"my_location_name", "to": "my_new_location_name"}
```

open_location

`open_location` will open Google Maps to a specific LOCATIONREF

Examples:

```
{"open_location": LOCATIONREF}
{"open_location": "object1"}
```

```
{"open_location": [34.1, -122.9]}
```

create_location

`create_location` will create a location name by selecting from the provided `zones` and creating the location from the zone's information. Zones are picked randomly from the list, you can simply provide one if you like.

Format:

```
{"create_location": "location_name", zones: [ZONE]}
{"create_location": "location_name", zones: [ZONE1, ZONE2, ZONE3, ...]}
{"create_location": "location_name", zones: [ZONE], no_results_commands: COMMANDLIST}
```

`no_results_commands`: Optional. By default a modal dialog will be created when the data query does not succeed.

A `ZONE` is has these properties:

- `location`: `LOCATIONREF` which is the center of the zone.
- `radius`: Radius of the zone in meters.
- `minRadius`: Optional. Defaults to `0`. meters.
- `commands`: `COMMANDLIST` which will be run. `$LOCATION:NAME` param will have the location friendly name.
- `zone_type`: Select from the list below.
- `query`: `DATAQUERY` (only if a data query zone)

Zone Types:

Zone	Description
random_point	Pick a random position inside this location.
query_list_result	Data Query: Execute the query and then present a list of results for the user to choose from.
query_random_result	Data Query: Execute the query and then pick a random result
query_closest_result	Data Query: Execute the query and then pick the closest result. If the query fails, increase the range and try again until there is a result.

If a data query is selected, the following parameters will be populated after call:

- `$LOCATION:NAME` the name tag on the result.
- `$LOCATION:ID` the id on the result.

Examples:

```
{"create_location": "$LOCATION", "zones": [
  {"zone": {
    "zone_type": "query_closest_result",
    "query": "[out:json];way({{bbox}})[highway~\"^(motorway|trunk|primary|secondary|tertiary|(motorway|trunk|
primary|secondary)_link)$\"]→.major;way({{bbox}})[highway~\"^(unclassified|residential|living_street|service)$
\"]→.minor;node(w.major)(w.minor);out;",
    "location": "city_center",
    "radius": 25000,
    "minRadius": 0,
    "commands": []
  }}
]}
```

query_data

`query_data` lets you query for OSM data and get a callback for the results.

Note that this is a legacy API before `for_each`.

- `location`: `LOCATIONREF`
- `query`: `DATAQUERY`
- `radius`: meters.
- `minRadius`: Optional. defaults to `0`
- `commands`: ARRAY OF `COMMANDLIST` to run for each result.

- `no_results_commands`: `COMMANDLIST` to run if there is not enough results (length of)
  - `$LOCATION` (usable temporary location name) will be defined differently in each call back to `commands`.
  - `$LOCATION` (param) will be defined differently in each call back to `commands`.
  - `$LOCATION:NAME` (param) will have the location friendly name.

Use `bypass_commands` and `$ITEMS` to process the full list.

Examples:

```
{"query_data": {
  "query": "[out:json]; ( node({{bbox}})[power=substation]; area({{bbox}})[power=substation]; ); out center;",
  "location": "city_center",
  "radius": 25000,
  "minRadius": 0,
  "commands": [
    [{"set": {"var": ["L:MISSION_LOC_POWER_0", "number"]}, "value": {"location": "$LOCATION"}}, {"set": {"var": ["L:MISSION_SCORE_POWER_0",
"number"]}, "value": 0}],
    [{"set": {"var": ["L:MISSION_LOC_POWER_1", "number"]}, "value": {"location": "$LOCATION"}}, {"set": {"var": ["L:MISSION_SCORE_POWER_1",
"number"]}, "value": 0}],
    [{"set": {"var": ["L:MISSION_LOC_POWER_2", "number"]}, "value": {"location": "$LOCATION"}}, {"set": {"var": ["L:MISSION_SCORE_POWER_2",
"number"]}, "value": 0}],
    [{"set": {"var": ["L:MISSION_LOC_POWER_3", "number"]}, "value": {"location": "$LOCATION"}}, {"set": {"var": ["L:MISSION_SCORE_POWER_3",
"number"]}, "value": 0}],
    [{"set": {"var": ["L:MISSION_LOC_POWER_4", "number"]}, "value": {"location": "$LOCATION"}}, {"set": {"var": ["L:MISSION_SCORE_POWER_4",
"number"]}, "value": 0}],
    [{"set": {"var": ["L:MISSION_LOC_POWER_5", "number"]}, "value": {"location": "$LOCATION"}}, {"set": {"var": ["L:MISSION_SCORE_POWER_5",
"number"]}, "value": 0}],
    [{"set": {"var": ["L:MISSION_LOC_POWER_6", "number"]}, "value": {"location": "$LOCATION"}}, {"set": {"var": ["L:MISSION_SCORE_POWER_6",
"number"]}, "value": 0}],
    [{"set": {"var": ["L:MISSION_LOC_POWER_7", "number"]}, "value": {"location": "$LOCATION"}}, {"set": {"var": ["L:MISSION_SCORE_POWER_7",
"number"]}, "value": 0}],
     [{"set": {"var": ["L:MISSION_LOC_POWER_8", "number"]}, "value": {"location": "$LOCATION"}}, {"set": {"var": ["L:MISSION_SCORE_POWER_8",
"number"]}, "value": 0}],
    [{"set": {"var": ["L:MISSION_LOC_POWER_9", "number"]}, "value": {"location": "$LOCATION"}}, {"set": {"var": ["L:MISSION_SCORE_POWER_9",
"number"]}, "value": 0}]
  ]
}}
```

query_country

`query_country` lets you identify the country name (string) for a location.

- `None` is a special country name that refers to the open ocean.
- `$COUNTRY` (param) will be defined after the call returns.
- `$COUNTRY` (stringToken) will be defined after the call returns.

Examples:

```
{"query_country": {
  "United States of America": [ {"set_message": {"text": "USA country $COUNTRY" }} ],
  "France": [ {"set_message": {"text": "FR country" }} ],
  "Germany": [ {"set_message": {"text": "DE country" }} ],
  "Other": [ {"set_message": {"text": "Other country: $COUNTRY" }} ],
  "None": [ {"set_message": {"text": "You are over open water. ($COUNTRY)" }} ]
}, "location":[65.34528194493097, -12.372530650689942]}
```

osm_query_data

Use `osm_query_data` to query OSM for data within a specific area. Operations on the data after `osm_query_data` will not use the network.

Examples:

```
{"#comment":"Query a block of road network data and save it into my_data"},
{"osm_query_data":
  "[out:json];way({{bbox}})[highway~\"^(motorway|trunk|primary|secondary|unclassified|residential|
living_street|service|tertiary|(motorway|trunk|primary|secondary|tertiary|)_link)$\"];(._;>;);out;",
  "location":"LOC",
  "size": 600,
  "result":"my_data"
},
```

osm_get_parent_ways

Given a `NodeId`, `osm_get_parent_ways` will provide an array of the ways which contain that id. Use this to find out which roads a given node belongs to.

Examples:

```
{"osm_get_parent_ways":{"struct":{"param":"$item"},"path":"id"}, "data": {"param":"my_data"},
"result":"parents"},
```

osm_get_connected_nodes

Use `osm_get_connected_nodes` to discover nodes which are adjacent to the given `nodeId`. This is good for finding the legs of an intersection, or the up/down nodes along a road.

Examples:

```
{"osm_get_connected_nodes":{"struct":{"param":"closest_node"},"path":"id"}, "data": {"param":"my_data"},
"result":"my_nodes_connected_to_nearest_node"},
```

osm_get_nodes

Use `osm_get_nodes` to get an ordered list of all the nodes within a given `wayId`. Use this to get a list of coordinates along a road.

Examples:

```
{"osm_get_nodes":{"struct":{"param":"$item"}, "path":"id"}, "data": {"param":"my_data"},
"result":"my_nodes_on_way"},
```

osm_get_all_ways

Use `osm_get_all_ways` to get a list of all the ways within the data set.

Examples:

```
{"osm_get_all_ways": {"param":"my_data"}, "result":"my_ways"},
```

osm_get_all_nodes

Use `osm_get_all_nodes` to get a list of all the nodes within the data set.

Examples:

```
{"osm_get_all_nodes": {"param":"my_data"}, "result":"my_nodes"},
```

osm_get_closest_nodes

Use `osm_get_closest_nodes` to create an ordered list of nodes, ranked by the distance to the given `LOCATIONREF`.

Examples:

```
{"osm_get_closest_nodes": "LOC", "data": {"param":"my_data"}, "result":"my_closest_nodes"},
```

osm_is_point_within_way

Use `osm_is_point_within_way` to determine if a given `LOCATIONREF` lies within the closed way.

Examples:

```
{"osm_is_point_within_way": {"struct":{"param":"way"}, "path":"id"}, "location":{"bearing":
{"param":"brg"},"dist":{"param":"dist"}}, "data":{"param":"my_data"},"result":"is_in"},
```

osm_get_area_of_area

Use `osm_get_area_of_area` to measure the area of the closed way, in meters squared.

Examples:

```
{"osm_get_area_of_area":{"struct":{"param":"way"},"path":"id"},  "data": {"param":"my_data"},
```

```
"result":"way_area"},
```

open_table

`open_table` opens an existing data table, or creates a new one. Once the table is open, table commands are valid.

Examples:

```
{"open_table": QUERY}
{"open_table": "my_table"}
```

save_table

`save_table` will immediately persist the table to disk. Changes made to tables where `save_table` is not eventually called (before leaving the mission) will be lost.

Examples:

```
{"save_table": QUERY}
{"save_table": "my_table"}
```

clear_table

`clear_table` will remove all keys from a table.

Examples:

```
{"clear_table": QUERY}
{"clear_table": "my_table"}
```

play_audio

`play_audio` enables playback of built-in audio sounds. `play_audio` will not proceed until the sound finishes playing.

Examples:

```
{"play_audio": "hold_position"}
{"play_audio": "4"}
```

Sound list:

```
0 1 2 3 4 5 6 7 8 9 10
we_are_not_in_range
we_are_too_high
we_are_too_low
hold_position
the_cabin_is_secure
forward
backward
left
right
ready_for_you_to_approach_and_hoist
ready_for_you_to_approach_and_land
tablet_alarm1
```

play_guidance_message

`play_guidance_message` can be used to provide audio guidance to a target.

- `target`: The remote target to provide guidance to
- `self`: This is the position on your aircraft that should match the center of the target object, such as `$USER:HOIST` for the hoist fixture position.

Examples:

```
{"play_guidance_message": {"target": LOCATIONREF, "self": LOCATIONREF}},
{"create_thread": {"name": "main_crash_guidance_thread", "commands": [{"while": {"var":
["L:MISSION_GUIDANCE_ENABLED", "number"]}, "eq": 1, "do": [
  {"wait_for": {"location": "main_crash", "var": "distance"}, "lt": 0.03},
  {"play_guidance_message": {"target": "main_crash", "self": "$USER:HOIST"}},
  {"sleep": 2}
]}]}}
```

connect_voice_server

`connect_voice_server` will attempt to connect to the defined voice service. `on_connected` commands will be run on success, `on_disconnected` will run on disconnect, even if much later.

Examples:

```
{"connect_voice_server": {
  "on_connected":[
    {"speak":"Speech activated."}
  ],
  "on_disconnected":[
    {"set_message":{"text":"No voice server available"}}
  ]
}}
```

speak

`speak` will send a command to the voice server to play text-to-speech or an audio file.

`interrupt`: 1 or 0. 1 will cancel the queue and play immediately. `is_audio_file`: 1 or 0. 1 will assume the text is a `filename.wav` in the `audio` directory available to the server.

Examples:

```
{"speak":"hello"}
{"speak":{"text":"hello {0}", "params":["dave"]}}
{"speak":"hello.wav", "is_audio_file":1}
```

Debugger & Remote Commands

cancel_debugger

`cancel_debugger` should be used by non-debug remote missions, this will suppress the extra debugger activity.

- This command applies when using a remote context (a mission connected via websocket) or when using the debugger (a tool based on the same rpc).

remote_notify

`remote_notify` will report data to the remote server, if available.

- This command applies when using a remote context (a mission connected via websocket) or when using the debugger (a tool based on the same rpc).

Examples:

```
{"remote_notify":"my_connected_event"}
{"remote_notify":"hello_event", "params":[
  {"var":["A:PLANE ALTITUDE","feet"]},
  {"var":["A:PLANE BANK DEGREES","bank"]}
]}
```

`teleport_to`

`teleport_to` will set the latitude and longitude of the player aircraft, instantly teleporting them. Note that this needs some work to engage slew mode for the user and adjust the altitude.

`fetch`

`fetch` enables interaction with remote web services using the javascript `fetch` API.

Examples:

```
{"fetch": {
  "url": "http://127.0.0.1:3000/report?key=hello",
  "method": "POST",
  "headers":{
    "Accept": "application/json",
    "Content-Type": "application/json"
  },
  "body": {"param":"msg"}
}},
```

`set_shared_data`

`set_shared_data` will mutate multiplayer shared data state. It implicitly uses the last created multiplayer connection.

Example:

```
{"set_shared_data":"update","path":"connectedAircraft.{service_auth}.isHost", "value": true},
```

`debug_write`

`debug_write` sends a string to `console.log`.

`hoist_control`

`hoist_control` enables reeling in or out the hoist cable. See the hoist topic for more.

Examples:

```
{"hoist_control": "reel_down","speed": 1}
{"hoist_control": "reel_up","speed": 1}
```

API Reference - QUERY

All `QUERY` are listed below. Additionally, `numbers`, `strings`, `null` and `arrays` (pass-through) are valid as a query.

`text`

`text` along with `params` can be used to build up any string. `{N}` is used as a replacement token in the string. `N` starts at 0 and increments, and it matches the elements in `params`.

Examples:

```
{"text":"object_name_{0}", "params":[ 99 ]}
{"text":"object_name_{0}", "params":[ {"var":["L:TEST","number"]} ]}
```

The result is a string which may be sent to another API, with a value like `object_name_99`.

Format specifiers:

```
{0}
{0:TIME}
{0:DMS}
{0:LOCATION}
```

`var`

`var` is the primary way to read an L:Var or A:var from the simulator. A list of variables is Available here.

All `L:Vars` will use the unit `number`.

Examples:

```
{"var": ["L:MY_SIM_VAR_HERE", "number"]}
{"var": ["A:PLANE ALTITUDE", "feet"]}
```

object/var

Read a property on a mission object.

Property	Function
$INDEX	Correlate a mission object name to `L:MISSION OBJECT ...` vars
VAR 1	Generic data slot 1
VAR 2	Generic data slot 2
VAR 3	Generic data slot 3
MODE	Object mode
HEIGHT	Radio Altitude in feet (readonly)
ALT	Altitude in feet
AALT	Actual altitude in feet (readonly)
AHDG	Actual heading in degree (readonly)
CREATED	1: created, 0: not created, -1: failed creation (readonly)
COUPLED	Special obejct mode. See table below
VELOCITY X	Body Velocity X (meters per second)
VELOCITY Y	Body Velocity Y (meters per second)
VELOCITY Z	Body Velocity Z (meters per second)
WP INDEX	Waypoint navigation 0: inactive, >0: represents active waypoint index.
distance	calculate distance to object
distance:ft	calculate distance to object (unit converter)

Generic data slots have a meaning only to the specific object (such as setting an animation state, a common use for `VAR 1`).

These values are applicable to the `MODE` var:

Object Mode	Function
0	Default mode. You may set `VELOCITY Z`
1	Repositioning mode. You may set `LAT`, `LON` and `HDG`
2	3-axis velocity
3	Default MSFS physics
4	Same as 1 but pitch inverted 180 degrees
5	Stop when radio height is under 10ft
6	Flight object repositioning. Set `ALT` and `HDG`

These values are applicable to the `COUPLED` var:

Object Mode	Function
0	Default mode.
1	Coupled to hoist station
2	Coupled to sling station
3	Available to be coupled to sling station
4	Firefighting target VAR 1: quantity of fire
5	Firefighting water source VAR 1: radius in meters, VAR 2: height in feet

`distance:ft` (unit converter) is also supported

Examples:

```
{"object": "my_object", "var": "VELOCITY Z"}
{"object": "my_object", "var": "$INDEX"}
{"object": "my_object", "var": "distance"}
```

location/var

You can get 3 things from a `LOCATIONREF`: distance in nautical miles, latitude and longitude.

`distance:ft` (unit converter) is also supported

Examples:

```
{"location": LOCATIONREF, "var":"distance"}
{"location": LOCATIONREF, "var":"lat"}
{"location": LOCATIONREF, "var":"lon"}
```

bearing

`bearing` calculates the true heading between two `LOCATIONREF`.

Examples:

```
{"bearing": {"to":LOCATIONREF, "from":LOCATIONREF}}
```

has_location

`has_location` will return 1 or 0 based on whether the location name exists already.

Examples:

```
{"has_location": QUERY}
{"has_location": "my_location_name"}
```

resolve_location

`resolve_location` will return `[lat, lon]` from a `LOCATIONREF`.

Examples:

```
{"has_location": QUERY}
{"has_location": "my_location_name"}
```

has_object

`has_object` will return 1 or 0 based on whether the object name exists already.

Examples:

```
{"has_object": QUERY}
{"has_object": "my_object_name"}
```

has_user_action

`has_user_action` returns 1 or 0 depending on whether the user_action is currently active.

Examples:

```
{"has_user_action": QUERY}
{"has_user_action": "my_user_action_name"}
```

has_mission

`has_mission` will return 1 or 0 based on whether the mission id exists in the index.

Examples:

```
{"has_mission": QUERY}
{"has_mission": "my_mission_id_"}
```

has_macro

`has_macro` returns a boolean value indication whether the macro name exists or not.

Examples:

```
{"has_macro": QUERY}
{"has_macro": "my_macro"}
```

no_resolve

`no_resolve` will just return the un-interpreted data.

Examples:

```
{"no_resolve": {"arbitrary_data_here":"my_data"}}
```

resolve_icon

`resolve_icon` will look up an entry in the `icons` table.

Examples:

```
{"resolve_icon": "my_icon_name"}
```

static

`static` enables getting keys under the `data` section of the mission (static data).

Examples:

```
"data": {
  "my_static_key": 99
}
{"static": "my_static_key"}
```

has_static

`has_static` returns a boolean value indicating if the key exists on the mission `data` section.

Examples:

```
{"has_static":"my_static_key"}
```

has_global

`has_mission` will return 1 or 0 based on whether the global name is defined.

Examples:

```
{"has_global": QUERY}
{"has_global": "my_global_name"}
```

global

`global` enables query of a global variable by name.

Examples:

```
{"global": QUERY}
{"global": "my_global_name"}
```

has_route

`has_route` will return a boolean value indicating if the specified route exists.

Examples:

```
{"has_route": QUERY}
{"has_route": "my_route_name"}
```

route

`route` will return the route information for a given route name.

Examples

```
{"route": QUERY}
{"route": "my_route_name"}
```

create_array

`create_array` makes a new array of the specified size. Arrays grow automatically so 0 is fine.

Examples:

```
{"create_array": QUERY}
{"create_array": 10}
```

create_struct

`create_struct` will create a complex object and each key will be evaluated as a QUERY

Examples:

```
{"create_struct": {
  "key1":QUERY,
  "key2":QUERY
}}
```

struct

`struct` is used to access a complex object.

- `path`: access a property.
- `has_path`: 1 or 0 based on whether the property exists.
- `function`: call a function
- `index`: access an array element

Examples:

```
{"struct": ...., "path": "length"}
{"struct": {"js:get":"JSON"}, "function": "stringify", "params": [ {"local": "my_local"}, null, 2]}
{"struct": ...., "index": 0}
```

js:get

`js:get` retrieves an object from the `window`. Examples are `Math` or `JSON`.

Examples:

```
{"js:get": "Math"}
```

js:create_async_function

`js:create_async_function` creates a JS async function which calls a `COMMANDLIST` with `$args` defined as a param.

Examples:

```
{"js:create_async_function": [
  {"set_message":{"text":"js called:{0}", "params": [
    {"struct": {"param": "$args"}, "index": 0}
  ]}}
]}
```

js:function

`js:function` creates a JS function which calls a `QUERY` with `$args` defined as a param. Since it is a `QUERY`, you may return a value synchronously as well.

Examples:

```
{"js:create_callback": [
  {"set_message":{"text":"js called:{0}", "params": [
    {"struct": {"param": "$args"}, "index": 0}
  ]}}
]}
```

`js:new`

`js:new` calls the constructor on an object, providing `params` if defined.

Examples:

```
{"js:new": "my_window_object", "params":[QUERY, QUERY, QUERY]}
```

`json:stringify`

`json:stringify` will transform an object into a JSON string.

Examples:

```
{"json:stringify": {"param":"$RET"}}
```

`json:parse`

`json:parse` will transform JSON string into an object.

Examples:

```
{"json:parse": {"param":"$RET"}}
```

`json:copy`

`json:copy` will create a deep copy of the object. Changes to the new object will not impact the input object.

Examples:

```
{"json:copy": {"param":"$RET"}}
```

`object:keys`

`object:keys` will return an array containing the key names in the target object.

Examples:

```
{"object:keys": {"param":"$RET"}}
```

`string:split`

`string:split` will create an array from the parts of string, specified by the delimiter.

`index`: Optional. This will return only one index in the array, instead of all parts of the array. This is handy if you only want one part of the split string anyway.

Examples:

```
{"string:split": {"struct": {"js:new": "Date"}, "function": "toISOString"}, "delimiter":"T", "index": 1}
```

string:join

`string:join` will create a string by appending each item in the input array, along with a delimiter.

Examples:

```
{"string:join": ["one","two","three"], "delimiter":"_"}
```

create_number

`create_number` will use the js `Number()` to convert a string to a number value.

Examples:

```
{"create_number":QUERY}
{"create_number":"99.5"}
```

has_local

`has_local` will return 1 or 0 based on whether the key exists in the locals.

Examples:

```
{"has_local": "my_local_name"}
```

local

`local` will retreive a local variable by name.

Examples:

```
{"local": "my_local_name"}
{"local": "my_local_name", "path": "key"}
```

gamevar

`gamevar` works like `var` but lets you query `SimVar.GetGameVarValue` in MSFS.

Examples:

```
{"gamevar": ["my_game_var", "my_unit"]}
```

table

`table` lets you read a key from a named table. (Table must be open first)

Examples:

```
{"table": "my_table", "key": "my_key"}
```

param

`param` lets you read a parameter from the params collection. There is a params collection for each macro and one for the main thread. create_thread's take the same params as the calling context.

Examples:

```
{"param": "my_param"}
{"param": "my_param", "path": "my_key"}
```

has_param

`has_param` tells you whether 1 or 0 for whether or not the parameter key exists.

Example:

```
{"has_param": "my_param"}
```

rand

`rand` will create a random decimal value between `QUERY` (minimum) and `QUERY2` (maximum) bounds.

Examples:

```
{"rand":[QUERY1, QUERY2]}
{"rand":[0, 60]}
```

add

`add` will add a list of queries (2 or more).

Examples:

```
{"add":[QUERY1, QUERY2, …]}
{"add":[2,2]}
{"add":[{"var":["L:TEST","number]}, 1]}
```

add360

`add360` is like `add` but the final result is normalized between 0 and 360.

Examples:

```
{"add360":[QUERY1, QUERY2, ...]}
{"add360":[{"var":["L:TEST","number]}, 90]}
```

compare360

`compare360` will provide absolute value between two values 0-360.

Examples:

```
{"compare360":[1, 359]} // -> 2
```

subtract

`subtract` subtracts `QUERY1` - `QUERY2`.

Examples:

```
{"subtract":[QUERY1, QUERY2]}
```

multiply

`multiply` multiplies `QUERY1` * `QUERY2`.

Examples:

```
{"multiply":[QUERY1, QUERY2]}
```

divide

`divide` divides `QUERY1 / QUERY2`. if QUERY2 is zero, the result is `0`.

Examples:

```
{"divide":[QUERY1, QUERY2]}
```

right_shift

`right_shift` is the right bit shift operator >>.

`QUERY` `>>` `QUERY2`

Examples:

```
{"right_shift": [QUERY1, QUERY2]}
{"right_shift": [0xFFFF, 2]}
```

left_shift

`left_shift` is the left bit shift operator <<.

`QUERY` `<<` `QUERY2`

Examples:

```
{"left_shift": [QUERY1, QUERY2]}
{"left_shift": [0xFFFF, 2]}
```

xor

`xor` is the exclusive-or operator.

Examples:

```
{"xor": [QUERY1, QUERY2]}
```

remainder

`remainder` is the mod or remainder operator.

Examples:

```
{"remainder": [QUERY1, QUERY2]}
```

exponent

`exponent` is power operator.

Examples:

```
{"exponent": [QUERY1, QUERY2]}
```

round

`round` will round a number to the nearest integer (whole number) value.

Examples:

```
{"round":QUERY}
{"round": 3.5}
```

`tofixed`

`tofixed` is like round, but lets you configure the number of digits to round to. `digits=2` would result in numbers like `0.00`

Examples:

```
{"tofixed":QUERY, "digits": QUERY}
{"tofixed":3.141592, "digits": 2}
```

`floor`

`floor` returns the closest previous whole number.

Examples:

```
{"floor":QUERY}
{"floor":2.5}
```

`ceil`

`ceil` (ceiling) returns the closest next whole number.

Examples:

```
{"ceil":QUERY}
{"ceil":2.5}
```

`abs`

`abs` returns the absolute value of a number (removing negative sign).

Examples:

```
{"abs":QUERY}
{"abs": -300}
```

`Math. ...` functions

These `Math` functions are also available:

```
Math.sign
Math.log
Math.log2
Math.log10
Math.sin
Math.sinh
Math.asinh
Math.cos
Math.cosh
Math.acosh
Math.atan
Math.atanh
Math.atan2
```

Examples:

```
{"Math.sign": -100}
{"Math.atan2": [QUERY, QUERY]}
```

clamp

`clamp` will return a number which is between the range of `QUERY_MIN` to `QUERY_MAX`. If `QUERY_VAL` is between the min and max, it will be returned directly.

Examples:

```
{"clamp":[QUERY_VAL, QUERY_MIN, QUERY_MAX]}
{"clamp":[5.5, 0, 100]}
```

scale

`scale` is transforms a value in range A to range B.

Examples:

```
{"scale":[QUERY_A_VAL, QUERY_A_MIN, QUERY_A_MAX, QUERY_B_MIN, QUERY_B_MAX]}
{"scale":[0.05, 0, 1, 0, 100]}
```

require

`require` returns 1 or 0 depending on whether the `QUERY1 op QUERY2` condition is true or false.

Examples:

```
{"require": QUERY1, "eq": QUERY2}
{"require": {"var":["L:TEST","number"]}, "eq", 0}
```

and

`and` is the logical AND operator.

`and` returns 1 if each of the queries are 1. If any query isn't 1, the overall result is 0.

Examples:

```
{"and":[
  {"require": QUERY, "eq": QUERY},
  ...
]}
```

or

`or` is the logical OR operator.

`or` will return 1 if any of the queries return 1. It will also short-circuit, in that further queries will not be checked if any subsequent query returns 1.

Examples:

```
{"or":[
  {"require": QUERY, "eq": QUERY},
  ...
]}
```

not

`not` is the logical NOT operator.

`not` will invert (1 to 0 and 0 to 1) any query.

Examples:

```
{"not":QUERY}
```

typeof

`typeof` returns a string which describes the type of input it was given.

Type	Type Name
structs	"object"
null	"object"
arrays	"array"
strings	"string"
numbers	"number"
undefined	"undefined"

isNaN

`isNaN` indicates whether a number is NaN or not.

Examples:

```
{"isNaN": QUERY}
```

parseInt

`parseInt` converts a string to an integer

Examples:

```
{"parseInt": QUERY}
```

parseFloat

`parseFloat` converts a string to an decimal value.

Examples:

```
{"parseFloat": QUERY}
```

if

`if` works as a QUERY or a COMMAND. The syntax is the same, except COMMANDLIST is a QUERY.

Examples:

```
{"if": QUERY, "then": QUERY, "else": QUERY}
{"if": QUERY, "then": QUERY}
```

switch

`switch` works as a QUERY as well as a COMMAND. COMMANDLISTs are instead a QUERY in this form.

Examples:

```
{"switch": QUERY, "case":{
  "0": QUERY,
  "1": QUERY,
  "2": QUERY,
  "3": QUERY,
  "default": QUERY
}}
```

convert

`convert` will do unit conversion.

Examples:

```
{"convert": QUERY, "from":"from_unit", "to":"to_unit"}

{"convert": QUERY, "from":"miles", "to":"meters"}
{"convert": QUERY, "from":"kg", "to":"lb"}
```

You can convert between these units:

Weight Unit	Monikers
Kilogram	kilogram, kg, kilo
Pound	pound, lb

Length Unit	Monikers
Feet	feet, foot, ft
Meter	meter, m
Mile	mile, mi

fn.HOIST_SEND_TO_GROUND

`HOIST_SEND_TO_GROUND` will conditionally deploy and stow the hoist based on proximity to the target.

Examples:

```
{"HOIST_SEND_TO_GROUND", "params": {
  "target": LOCATIONREF,
  "after_deploy_commands": [
    ... commands after deploying the hoist
  ],
  "before_stow_commands": [
    ... commands before stow
  ]
}}
```

fn.HOIST_REEL_UP_AND_STOW

`HOIST_REEL_UP_AND_STOW` will wait for the hoist to be reeled up, and run the commands once when it happens:

`before_stow_commands` will run once when the hoist is stowing.

Examples:

```
"fn":"HOIST_REEL_UP_AND_STOW", "params": {
  "before_stow_commands": [
    .. commands here to run before stow (to swap the objects)
  ]
}}
```

fn.HOIST_REEL_UP

`HOIST_REEL_UP` will return true when the hoist is up and stowed, and false until that happens.

Examples:

```
{"fn":"HOIST_REEL_UP"}
```

fn.hoist_get_reel_distance:ft

Gets the distance which the cable is extended.

Examples:

```
{"fn":"hoist_get_reel_distance:ft"}
{"fn":"hoist_get_reel_distance:m"}
```

fn.hoist_get_distance_from_ground:ft

Gets the distance from the hoist object to the ground.

Examples:

```
{"fn":"hoist_get_distance_from_ground:ft"}
{"fn":"hoist_get_distance_from_ground:m"}
```

fn.score_bambi_dump

`score_bambi_dump` will return a score value, which is calculated from all of the objects which **were created with** `COUPLED=4`. The total score is the summation of `VAR 2` from each of those objects.

Examples:

```
{"fn":"score_bambi_dump"}
```

fn.all_fires_extinguished

`all_fires_extinguished` indicates if there are any objects with `COUPLED=4` that are alive. Returns 1 if any fire is active.

Examples:

```
{"fn":"all_fires_extinguished"}
```

fn.has_remote_notify

Returns 1 or 0 based on whether the mission is running from a server.

Examples:

```
{"fn":"has_remote_notify"}
```

fn.is_voice_server_connected

`fn:is_voice_server_connected` indicates whether or not the voice server is currently connected.

Examples:

```
{"fn": "is_voice_server_connected"}
```

fn.create_guid

`fn:create_guid` will generate a globally unique identifier string.

Examples:

```
{"fn": "create_guid"}
```

fn.create_date

`create_date` will create a JS Date object.

Examples:

```
{"set_message":{"text":"{0}", "params":[ {"fn":"create_date"} ]}},
```

fn.get_time_string

`get_time_string` will get a 24-hour UTC timestamp like `07:05:57`

Examples:

```
{"set_message":{"text":"{0}", "params":[
  {"fn":"get_time_string"}
]}},
```

fn.get_mission_objects

`get_mission_objects` will get the active mission objects (for enumeration).

Examples:

```
{"set_message":{"text":"{0}", "params":[
  {"json:stringify": {"fn":"get_mission_objects"}}
]}},
```

fn.get_aircraft_moniker

`fn.get_aircraft_moniker` will get a string identifier for the aircraft, like `H145` or `H160`.

Examples:

```
{"#comment":"copy either H145 or H160 into a string local named HXX"},
{"set":{"local":"HXX"},"value":{"fn":"get_aircraft_moniker"}},
{"#comment":"fire an event where the name is either H145 or H160 but otherwise is the same..."},
{"trigger":"H:{local:HXX}_SDK_DO_RANDOM_THING"},
{"#comment":"just show a message, but note that you can now use {local:HXX} in any string across the
mission"},
{"set_message":"hello from {local:HXX}"},
```

fn.is_any_sling_object_coupled

`fn.is_any_sling_object_coupled` returns a boolean value indicating whether an object is coupled right now.

fn.get_sling_object_type

`fn.get_sling_object_type` returns a value from 1 to 10 indicating the type of sling object currently coupled.

fn.get_mission_icons

`fn.get_mission_icons` gets the entire `icons` table.

fn.create_multiplayer_connection

`fn.create_multiplayer_connection` creates an `MPClient` multiplayer connection.

API Reference - LOCATION

Locations are a foundational concept within the mission platform.

LOCATIONREF

A `LOCATIONREF` is one of the following:

1. A string referencing an item in the `locations` table, or an object in the `objects` table.
2. An array such as `[34.29, -122.4]` or `[34.29, -122.4, 90]`. The latter 90 being a heading if provided.
3. A special location string like `$USER`.
4. A bearing/dist command
5. a closest command

Examples:

```
"my_location"
"object_name"
"$USER"
[34.29, -122.4]
[34.29, -122.4, 90]
{"bearing": 100, "dist": 100}
{"bearing2": 100, "dist": 100}
{"location_alter": ...}
{"closest": ...}
```

bearing

`bearing` will calculate a bearing based on an input.

- `dist`: meters
- `heading`: optional, defaults to zero

Examples:

```
{
  "bearing": 100,
  "object": "$USER",
  "hearing": 0,
  "dist": 100
}
```

bearing2

`bearing2` will calculate a bearing without considering the user aircraft heading.

- `dist`: meters
- `heading`: optional, defaults to zero

Examples:

```
{
  "bearing2": 100,
  "object": "$USER",
  "hearing": 0,
  "dist": 100
}
```

location_alter

`location_alter` will create a location reference with a modified heading.

Examples:

```
{"location_alter":"$USER", "hdg": 0}
```

closest

`closest` will pick the closest location to `to`. `to` will default to $USER if not provided.

Examples:

```
{"closest": [LOCATIONREF, LOCATIONREF, ...]}
{"closest": [LOCATIONREF, LOCATIONREF, ...], "to": LOCATIONREF}
```

Special Locations

- `$USER`: resolves to `[lat,lon]` of the user aircraft
- `$USER:HOIST`: resolves to `[lat,lon]` of the user aircraft hoist position.
- `$MISSION_START_LOCATION`: resolves to the [lat,lon] which was the start point on the map.
- `$MISSION_SELECTED_POI_LOCATION`: resolves to the [lat,lon] which is currently selected by the user, on the mission map.

Samples

Converted function from JS Example

Given this function written in JS:

```
function polarToCartesian(radius, angleInDegrees) {
    let angleInRadians = (angleInDegrees-90) * Math.PI / 180.0;
    return {
      x: radius * Math.cos(angleInRadians),
      y: radius * Math.sin(angleInRadians)
    };
}
```

The same function, written as a macro, is as follows:

```
"polarToCartesian": [
        {"#comment":[
                "param – radius",
                "param – angleInDegrees"
        ]},
        {"set":{"param":"angleInDegrees"}, "value": {"multiply":[ {"subtract":[ {"param":"angleInDegrees"},
-90 ]}, {"divide": [3.14159 / 180]} ]}},
        {"return": {"create_struct":{
                "x": {"multiply":[ {"param":"radius"}, {"Math.cos": {"param":"angleInDegrees"}} ]},
                "y": {"multiply":[ {"param":"radius"}, {"Math.sin": {"param":"angleInDegrees"}} ]},
        }}}
]
```

Scenery detection Sample

Use `fetch` to query the VFS for content specific to the package that you wish to detect.

```
{
  "title": "Scenery detection Sample",
  "api_version": 0.1,
  "aircraft": ["H145"],
  "briefing":[
    {"image": "/VFS/ContentInfo/revelstoke-logging1-scenery/Thumbnail.jpg", "show_condition":{"require":
{"local":"SCENERY_INSTALLED_1"},"eq":1}},
    {"text":"Installed: {local:SCENERY_INSTALLED_1}"},
    {"image": "/VFS/ContentInfo/revelstoke-mill1-scenery/Thumbnail.jpg", "show_condition":{"require":
{"local":"SCENERY_INSTALLED_2"},"eq":1}},
    {"text":"Installed: {local:SCENERY_INSTALLED_2}"},
    {"image": "/VFS/ContentInfo/revelstoke-lakeview-scenery/Thumbnail.jpg", "show_condition":{"require":
{"local":"SCENERY_INSTALLED_3"},"eq":1}},
    {"text":"Installed: {local:SCENERY_INSTALLED_3}"}
  ],
  "objectives": [
    {
      "title": "Done",
      "commands": [
        {"fetch":{"url":"/VFS/ContentInfo/revelstoke-logging1-scenery/Thumbnail.jpg"}},
        {"set":{"local":"SCENERY_INSTALLED_1"},"value":{"if":
```

```
{"param":"$FETCH_STATUS"},"eq":200,"then":1,"else":0}},
        {"fetch":{"url":"/VFS/ContentInfo/revelstoke-mill1-scenery/Thumbnail.jpg"}},
        {"set":{"local":"SCENERY_INSTALLED_2"},"value":{"if":
{"param":"$FETCH_STATUS"},"eq":200,"then":1,"else":0}},
        {"fetch":{"url":"/VFS/ContentInfo/revelstoke-lakeview-scenery/Thumbnail.jpg"}},
        {"set":{"local":"SCENERY_INSTALLED_3"},"value":{"if":
{"param":"$FETCH_STATUS"},"eq":200,"then":1,"else":0}},
        {"sleep": "forever"}
      ]
    }
  ]
}
```

Get random item from static list

Given this static data:

```
  "data": {
    "cars": [
      "Car Title 1",
      "Car Title 2",
      "Car Title 3",
      "Car Title 4"
    ]
  },
```

Then each call will have a random item from `cars`:

```
{"sruct": {"static": "cars"}, "index": {"floor": {"rand": [0, {"struct":
{"static":"cars"},"path":"length"} ]}}}
```

Example:

```
{
 "title": "test",
 "api_version": 0.1,
 "aircraft": ["H145"],
 "data": {
 "cars": [
 "Car Title 1",
 "Car Title 2",
 "Car Title 3",
 "Car Title 4"
 ]
 },
 "briefing":[
 {"text":"Your selected car: {0}", "params":[
 {"local":"selectedCar"}
 ]},
 {"buttonbar": [
 {"title":"Pick a new car", "commands": [
 {"set": {"local":"selectedCar"}, "value": {"struct": {"static": "cars"}, "index": {"floor": {"rand": [0,
{"struct":{"static":"cars"},"path":"length"} ]}}}}
 ]}
 ]}
 ],
 "objectives": [
 {
 "title": "Done",
 "commands": [
        {"sleep": "forever"}
      ]
    }
  ]
}
```

CARLS Radio Test Program

This program sends information to the radio and also handles the events for clicking the buttons.

```
{
  "title": "Radio test program",
  "api_version": 0.1,
  "aircraft": ["H145"],
  "macros": {
    "render": [
      {"if": {"var": ["L:MY_PAGE", "number"]}, "eq":0, "then": [
        {"set_carls_radio": {
          "LSK": ["PG1", "", ""],
          "RSK": ["", "", "INOP"],
          "Items": [
            ["Group 1", "misc contacts"],
            ["Group 2", "important"],
            ["Group 3", "other"]
          ]
        }}
      ]},
      {"if": {"var": ["L:MY_PAGE", "number"]}, "eq":1, "then": [
        {"set_carls_radio": {
          "LSK": ["PG2", "", ""],          "RSK": ["", "", "INOP"],
          "Items": [
            ["Contact 1", "000-5555-1234"],
            ["Contact 2", ""],
            ["Contact 3", ""]
          ]
        }}
      ]}
    ]
  },
  "objectives": [
    {
      "title": "Initializing...",
      "commands": [
        {"#comment": "select keys"},
        {"create_event_handler": "MISSION_RADIO_CARLS_L1", "commands": [{"set_message": {"text": "LSK1" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_L2", "commands": [{"set_message": {"text": "LSK2" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_L3", "commands": [{"set_message": {"text": "LSK3" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_R1", "commands": [{"set_message": {"text": "RSK1" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_R2", "commands": [{"set_message": {"text": "RSK2" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_R3", "commands": [{"set_message": {"text": "RSK3" }}]},
        {"#comment": "dial pad"},
        {"create_event_handler": "MISSION_RADIO_CARLS_0", "commands": [{"set_message": {"text": "Num 0" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_1", "commands": [{"set_message": {"text": "Num 1" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_2", "commands": [{"set_message": {"text": "Num 2" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_3", "commands": [{"set_message": {"text": "Num 3" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_4", "commands": [{"set_message": {"text": "Num 4" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_5", "commands": [{"set_message": {"text": "Num 5" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_6", "commands": [{"set_message": {"text": "Num 6" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_7", "commands": [{"set_message": {"text": "Num 7" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_8", "commands": [{"set_message": {"text": "Num 8" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_9", "commands": [{"set_message": {"text": "Num 9" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_STAR", "commands": [{"set_message": {"text": "Num
*" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_SHARP", "commands": [{"set_message": {"text": "Num
#" }}]},
        {"#comment": "phone keys"},
        {"create_event_handler": "MISSION_RADIO_CARLS_PICK", "commands": [{"set_message": {"text":
"PICK" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_HANG", "commands": [{"set_message": {"text":
"HANG" }}]},
        {"create_event_handler": "MISSION_RADIO_CARLS_WARNING", "commands": [{"set_message": {"text":
"WARNING" }}]},
        {"#comment": "change page when using <- and -> arrows "},
        {"create_event_handler": "MISSION_RADIO_CARLS_LEFT", "commands": [
          {"set": {"var": ["L:MY_PAGE", "number"]}, "value": 0},
          {"sleep":0.2},
          {"call_macro":"render"}
        ]},
        {"create_event_handler": "MISSION_RADIO_CARLS_RIGHT", "commands": [
          {"set": {"var": ["L:MY_PAGE", "number"]}, "value": 1},
          {"sleep":0.2},
          {"call_macro":"render"}
        ]},
        {"#comment": "Use L:CARLS_LIST_SELECTED_INDEX to get the highlighted list item !!!!"},
        {"call_macro":"render"}
      ]
    },
    {
      "title": "Done",
      "commands": [
        {"sleep": "forever"}
      ]
```

```
    }
  ]
}
```

Remote dispatcher test program

HEMS dispatcher (multiplayer only)

1. Each user connects to the server and uploads a list of 'valid mission choices'
2. Operator can dispatch aircraft and the aircraft can accept the dispatch.

```
{
 "title": "Multiplayer Dispatch Test Program V2",
 "author":"davux3",
 "api_version": 0.1,
 "aircraft": ["H145"],
 "data":{
 "server_url": "wss://5ed547d.online-server.cloud/mpserver/ws",
 "create_room_url": "https://davux.com/dispatcher/",
 "webConfig": {
 "fligihtPlans": {
 "type":"map_line",
 "source":{"static":"flightPlans"},
 "name":"Flight Plan",
 "stroke":{"no_resolve":{"color": "#d303fc", "width":2}},
 "icon":{"static":"icons.wp_blue"}
 },
 "connectedAircraftIcons": {
 "type":"map_point",
 "source":{"static":"connectedAircraft"},
 "name":"Connected Aircraft",
 "text":"{UserName}",
 "icon":{"static":"icons.h160_icon"}
 },
 "aircraftDispatches": {
 "type":"map_point",
 "source":{"static":"messagesToAircraft"},
 "name":"Dispatch Locations",
 "text":"{to}",
 "icon":{"static":"icons.wp_blue"}
 },
 "connectedAircraftList": {
 "type":"list",
 "source":{"static":"connectedAircraft"},
 "title":"Connected Aircraft",
 "emptyText":"No aircraft are connected right now",
 "rows":{
 "row0":{
 "1": {"icon":{"static":"icons.h160_icon"}},
 "2": {"text":"{UserName}"},
 "3": {"button":"Send Dispatch","commands": [
 {"if": {"fn":"has_selected_poi"},"eq":0,"then":[
 {"show_dialog": {
 "title":"Send Message",
 "content": [
 {"text": "POI not selected. Click on a POI on the map and try again"}
 ]
 }}
 ], "else":[
 {"show_dialog": {
 "title":"Send Message",
 "content": [
 {"text": "Select Mission"},
 {"listbox": "lst_mission", "source": {"param":"MissionList"}, "text": "{id}", "value":{"param":"id"}},
 {"text": "Dispatcher Name:"},
 {"textbox": "txt_from"},
 {"text": "Message Text:"},
 {"textarea": "txt_message"},
 {"text": "Patient Life score (0-100%):"},
 {"slider":"slider_value","min":0,"max":100,"value":{"rand":[0,100]}},
 {"button": "Send Message", "commands": [
 {"set_shared_data": "update", "path": "messagesToAircraft.{id}", "value": {"create_struct":{
 "from": {"param":"txt_from"},
 "to": {"param":"UserName"},
                                                "message": {"param":"txt_message"},
                                                "mission": {"param": "lst_mission"},
                                                "location": {"fn": "selected_poi_location"},
                                                "lifeScore": {"param":"slider_value"}
                                        }}},
                                        {"close_dialog":1}
```

```
                                ]}
                  ]
                }}
              ]}
            ]},
            "4": {"button":"View","commands": [ {"set_map_center": {"param": "location"}, "zoom": 16} ]}
          },
          "row1":{
            "1": {"text":"{0}", "params": [ {"struct":{"static":"statusMessages"}, "index":{"param":"Status"}}
]}
          }
        }
      },
      "incomingMessageList": {
        "type":"list",
        "source":{"static":"messagesToDispatcher"},
        "title":"Message Inbox",
        "emptyText":"No messages right now",
        "rows":{
          "row0":{
            "1": {"text":"{UserName}"},
            "2": {"button":"Delete","commands": [ {"set_shared_data":"delete", "path":"messagesToDispatcher.
{id}"} ]}
          },
          "row1":{
            "1": {"text":"{Text}"}
          }
        }
      },
      "outgoingMessageList": {
        "type":"list",
        "source":{"static":"messagesToAircraft"},
        "title":"Recent Dispatches",
        "emptyText":"No messages right now",
        "rows":{
          "row0":{
            "1": {"text":"{from} {to} "},
            "2": {"text":"{mission}"},
            "3": {"button":"Delete","commands": [ {"set_shared_data":"delete", "path":"messagesToAircraft.
{id}"} ]},
            "4": {"button":"View","commands": [ {"set_map_center": {"param": "location"}, "zoom": 16} ]}
          },
          "row1":{
            "1": {"text":"{message}"}
          }
        }
      }
    },
    "statusMessages": [
      "0. Dispatch accepted",
      "1. On the way to the scene",
      "2. At the scene",
      "3. On the way to the hospital",
      "4. At the hospital",
      "5. On the way back to base (Available)",
      "6. At Home base (Available)",
      "7. Unavailable for dispatch"
    ],
    "missionList1": {
      "0. Road Accident": 0,
      "1. Motorcycle Crash": 1,
      "2. Tipped over tractor": 2
    },
    "missionList2": {
      "0. Hospital": 0,
      "1. Meet Ambulance": 1
    }
  },
  "briefing":[
    {"#comment":[
      "MP_MODE ... 0: not set, 1: offline, 2: online"
    ]},
    {"title":"Mission Initial Setup", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
    {"buttonbar":[
      {"title":"Offline (Single player)", "commands": [ {"set":{"local":"MP_MODE"}, "value":1} ],
"disabled_condition":{"require":1,"eq":1}},
      {"title":"Online (Multiplayer)", "commands": [ {"call_macro":"mp_open_login_dialog"}  ]}
    ], "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
    {"title":"Multiplayer (Online)", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 2}},
    {"buttonbar":[
      {"title":"View Multiplayer Status", "commands": [ {"call_macro":"mp_open_login_dialog"}  ]}
    ], "show_condition": {"require":{"local":"MP_MODE"}, "eq": 2}},
```

```
    {"title":"My Status", "show_condition": {"require":{"local":"MP_MODE"},"ne":0}},
    {"text":"My status: {0}", "params": [ {"var":["L:MY_DISPATCH_STATUS","number"]}  ], "show_condition":
{"require":{"local":"MP_MODE"},"ne":0}},
    {"text":"{0}", "params": [
      {"struct": {"static":"statusMessages"}, "index":{"var":["L:MY_DISPATCH_STATUS","number"]}}
    ], "show_condition": {"require":{"local":"MP_MODE"},"ne":0}},
    {"text":"Change my status:", "show_condition": {"require":{"local":"MP_MODE"},"ne":0}},
    {"buttonbar":[
      { "title":"1", "commands":[ {"set":{"var":["L:MY_DISPATCH_STATUS","number"]},"value":1}] },
      { "title":"2", "commands":[ {"set":{"var":["L:MY_DISPATCH_STATUS","number"]},"value":2}] },
      { "title":"3", "commands":[ {"set":{"var":["L:MY_DISPATCH_STATUS","number"]},"value":3}] },
      { "title":"4", "commands":[ {"set":{"var":["L:MY_DISPATCH_STATUS","number"]},"value":4}] },
      { "title":"5", "commands":[ {"set":{"var":["L:MY_DISPATCH_STATUS","number"]},"value":5}] },
      { "title":"6", "commands":[ {"set":{"var":["L:MY_DISPATCH_STATUS","number"]},"value":6}] }
    ],
    "show_condition": {"require":{"local":"MP_MODE"},"ne":0}},
    {"buttonbar":[
      { "title":"Send message to dispatcher", "commands": {"call_macro":"open_dispatcher_msg_dialog"} ]}
    ],
    "show_condition": {"require":{"local":"MP_MODE"},"ne":0}},
    {"text":"Change my avaialble missions for dispatch:", "show_condition":
{"require":"local":"MP_MODE"},"ne":0}},
    {"buttonbar":[
      {
        "title":"Set Mission Set 1 (rescue)",
        "commands":[
          {"set_shared_data":"update",
            "path":"connectedAircraft.{local:service_auth}.MissionList",
            "value": {"static": "missionList1"}
          },
          {"set":{"local":"ACTIVE_MISSION_SET"},"value":1}
        ],
        "select_condition": {"require":{"local":"ACTIVE_MISSION_SET"},"eq":1}
      },
      {
        "title":"Set Mission Set 2 (hospital etc.)",
        "commands":[
          {"set_shared_data":"update",
            "path":"connectedAircraft.{local:service_auth}.MissionList",
            "value": {"static": "missionList2"}
          },
          {"set":{"local":"ACTIVE_MISSION_SET"},"value":2}
        ],
        "select_condition": {"require":{"local":"ACTIVE_MISSION_SET"},"eq":2}
      }
    ],
    "show_condition": {"require":{"local":"MP_MODE"},"ne":0}},
    {"title":"Incoming Dispatch"},
    {"text":"Dispatcher Name: {local:DISPATCH_FROM}"},
    {"text":"Selected Mission: {local:DISPATCH_MISSION}"},
    {"text":"Location: {local:DISPATCH_LOCATION}"},
    {"text":"Text Message: {local:DISPATCH_MESSAGE}"},
    {"text":"Patient Life Score: {local:DISPATCH_LIFESCORE}"}
  ],
  "events": {
    "ON_MISSION_ABORTING": {
      "commands": [     {"call_macro":"mp_aborting_mission"}    ]
    }
  },
  "macros":{
    "open_dispatcher_msg_dialog": [
      {"set_dispatch":[
        {"title":"Send message"},
        {"textbox":"mp_dispatcher_msg"},
        {"buttonbar":[
          {"title":"Send Message to dispatcher", "commands": [
            {"set":{"param":"id"}, "value": {"fn": "create_guid"}},
            {"set_shared_data":"update",
              "path":"messagesToDispatcher.{id}",
              "value": {"create_struct":{
                "Text":{"local":"mp_dispatcher_msg"},
                "UserName": {"local":"mp_userName"}
            }}},
            {"set_briefing_dialog":1}
          ]}
        ]}
      ]},
      {"set_dispatch_dialog":1}
    ],
    "mp_open_login_dialog":[
      {"#comment": "Show the login dialog dispatch (or multiplayer status"},
      {"set_dispatch":[
```

```
        {"buttonbar":[ {"title":"<- Back to briefing", "commands": [{"set_briefing_dialog":1} ]} ]},
        {"title":"Log in",  "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"text":"You are playing offline.", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 1}},
        {"text":{"text":"User Id: {0}", "params":[{"local":"service_auth"}]}, "show_condition": {"require":
{"local":"MP_MODE"}, "eq": 0}},
        {"text":"User Name:", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"textbox":"mp_userName", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"text":"Room:", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"textbox":"mp_room", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"text":"Password:", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"textbox":"mp_password", "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"buttonbar":[
          {"title":"Create Room (Opens on PC)", "commands": [ {"open_url":"{static:create_room_url}?
room={local:mp_room}"} ]},
          {"title":"Log In", "commands": [ {"call_macro":"mp_login"} ]}
        ],
        "disabled_condition":{"require":{"struct":{"local":"MP_CONN"}, "path":"Status"},"eq":"Connected"},
        "show_condition": {"require":{"local":"MP_MODE"}, "eq": 0}},
        {"text":{"text":"MP Connection Status: {0}", "params":[
          {"struct":{"local":"MP_CONN"}, "path":"Status"}
        ]}, "show_condition": {"require":{"local":"MP_MODE"}, "ne": 1}},
        {"text":{"text":"MP Server Last Error: {local:MP_LAST_ERROR}"}, "show_condition": {"require":
{"local":"MP_MODE"}, "ne": 1}},
        {"title":"Debug Info"},
        {"text":{"text":"Multiplayer Mode: {0}", "params":[
          {"switch":{"local":"MP_MODE"}, "case":{
            "0": "Undecided",
            "1": "Offline, Singleplayer",
            "2": "Multiplayer"
          }}
        ]}},
        {"#comment":{"text":"Debug MP Message: {local:MP_MSG}"}, "show_condition": {"require":
{"local":"MP_MODE"}, "ne": 1}}
      ]},
      {"set_dispatch_dialog":1}
    ],
    "mp_login":[
      {"#comment":"try to make the actual connection to the server"},
      {"set":{"param":"service_auth"},"value":{"local":"service_auth"}},
      {"set":{"local":"MP_LAST_ERROR"},"value":""},
      {"set":{"local":"MP_CONN"}, "value": {"fn": "create_multiplayer_connection"}},
      {"set":{"local":"MP_CONN", "path":"OnError"}, "value":{"js:create_async_function":[
        {"set":{"local":"MP_LAST_ERROR"},"value":{"struct": {"param":"$args"}, "index": 0}}
      ]}},
      {"set":{"local":"MP_CONN", "path":"OnMessage"}, "value":{"js:create_async_function":[
        {"set":{"param":"arg0"},"value":{"struct": {"param":"$args"}, "index": 0}},
        {"call_macro":"mp_on_message","params":{"msg": {"param":"arg0"}}}
      ]}},
      {"set":{"param":"unused"},"value":{"struct":{"local":"MP_CONN"}, "function":"Connect", "params":[
        {"static":"server_url"}, {"param":"service_auth"}, {"local":"mp_room"}, {"local":"mp_password"}
      ]}},
      {"create_thread":{"commands":[
        {"wait_for":{"struct":{"local":"MP_CONN"}, "path":"Status"},"eq":"Connected"},
        {"#comment":"once we log in once, we're committed to muoltiplayer"},
        {"set":{"local":"MP_MODE"}, "value": 2},
        {"set_briefing_dialog":1},
        {"#comment":"First create terminationCommands with no_overwrite, then add an entry for us, and then
populate with commands to clear us from connectedAircraft and terminationCommands when we become stale on the
server"},
        {"set_shared_data":"update",
          "path":"terminationCommands",
          "policy":"no_overwrite",
          "value": {"create_struct":{}}
        },
        {"set_shared_data":"update",
          "path":"terminationCommands.{service_auth}",
          "value": {"create_struct":{
            "removeFromConnectedAircraft":{"create_struct":{
              "type":"delete",
              "path":"connectedAircraft.{service_auth}"
            }},
            "removeFromFlightPlans":{"create_struct":{
              "type":"delete",
              "path":"flightPlans.{service_auth}"
            }},
            "removeFromTerminationCommands":{"create_struct":{
              "type":"delete",
              "path":"terminationCommands.{service_auth}"
            }}
        }}}},

        {"#comment":"make sure we have connectedAircraft table. all players must use no_overwrite when
```

```
ensuring the table exists to prevent anybody from destroying the table."},
 {"set_shared_data":"update", "path":"connectedAircraft", "policy":"no_overwrite", "value": {"create_struct":
{}} },
 {"set_shared_data":"update", "path":"messagesToDispatcher", "value": {"create_struct":{}} },
 {"set_shared_data":"update", "path":"messagesToAircraft", "value": {"create_struct":{}} },
 {"set":{"param":"unused"},"value":{"struct":{"local":"MP_CONN"}, "function":"Subscribe", "params":
[ "messagesToAircraft" ]}},
 {"set_shared_data":"update", "path":"icons", "policy":"no_overwrite", "value": {"fn":"get_mission_icons"} },
 {"set_shared_data":"update", "path":"flightPlans", "policy":"no_overwrite", "value": {"create_struct":{}} },
 {"set_shared_data":"update", "path":"webConfig", "policy":"no_overwrite", "value": {"static":"webConfig"} },
 {"set_shared_data":"update", "path":"statusMessages", "policy":"no_overwrite", "value":
{"static":"statusMessages"} },
 {"set_shared_data":"update",
 "path":"connectedAircraft.{service_auth}",
 "value": {"create_struct":{
 "location":{"resolve_location":"$USER"},
 "UserName": {"local":"mp_userName"},
 "Status": 0,
 "MissionList": {"static": "missionList1"}
        }}},
         {"set":{"local":"ACTIVE_MISSION_SET"}, "value":1},
        {"#comment":"update our location, score and flightplan (if changed) forever"},
{"while":1,"eq":1,"do":[
         {"sleep":5},
         {"set_shared_data":"update", "path":"connectedAircraft.{service_auth}.location", "value":
{"resolve_location":"$USER"} },
         {"set_shared_data":"update", "path":"connectedAircraft.{service_auth}.Status", "value": {"var":
["L:MY_DISPATCH_STATUS","number"]} },
         {"if":{"json:stringify": {"local":"$FLIGHTPLAN"}}, "ne": {"param":"FPL"},"then":[
           {"set":{"param":"FPL"},"value":{"json:stringify": {"local":"$FLIGHTPLAN"}}},
           {"set_shared_data":"update", "path":"flightPlans.{service_auth}", "value": {"create_struct":{
               "points":{"local":"$FLIGHTPLAN"}
           }}}
         ]}
       ]}
    ],
    "mp_initialize":[
      {"#comment":"set up for multiplayer operations later"},
      {"set":{"local":"MP_LAST_ERROR"},"value":""},
      {"set":{"local":"MP_MODE"},"value":0},
      {"#comment":"MP_MODE 0: undecided, 1: offline, 2:online"},
      {"#comment":"these are for debugging only"},
      {"set":{"local":"MP_MSG"},"value":""},
      {"set":{"local":"mp_room"},"value":""},
      {"set":{"local":"mp_password"},"value":""},
      {"set":{"local":"mp_userName"},"value":{"var":["ATC AIRLINE","string"]}},
      {"#comment": "Create or access a unique ID to identify you on the server irrespective of callsign"},
      {"set":{"local":"service_auth"}, "value":{"fn":"create_guid"}},
      {"create_thread":{"commands":[
        {"wait_for":{"local":"MP_MODE"},"ne":0},
        {"call_macro":"mp_begin"}
      ]}}
    ],
    "mp_on_message":[
      {"#comment":"param - msg"},
      {"#comment":"handle READ, UPDATE and DELETE operations below"},
      {"set":{"param":"json"},"value":{"json:stringify":{"param":"msg"}}},
      {"switch":{"struct": {"param":"msg"}, "path": "type"}, "case": {
        "read":[
          {"set":{"local":"MP_MSG"},"value": "we got an read: {json}"}
        ],
        "update": [
          {"set":{"local":"MP_MSG"},"value": "we got an update: {json}"},
          {"#comment":"split the path into parts based on ."},
          {"set":{"param":"parts"}, "value":{"string:split":{"struct": {"param":"msg"}, "path": "path"},
"delimiter":"."}},
          {"#comment":"messagesToAircraft NEW MESSAGE"},
          {"if": {"and":[
            {"require": {"struct": {"param":"parts"}, "path":"length"}, "eq": 2},
            {"require": {"struct": {"param":"parts"}, "index":"0"}, "eq": "messagesToAircraft"},
            {"require": {"struct": {"param":"parts"}, "index":"1"}, "eq": "{local:service_auth}"}
          ]}, "eq":1, "then":[
            {"call_macro":"on_got_dispatch","params":{
              "dispatchInfo": {"struct": {"param":"msg"}, "path": "value"}
            }}
          ]}
        ],
        "delete": [
          {"set":{"local":"MP_MSG"},"value": "we got an delete: {json}"}
        ]
      }}
```

```
    ],
    "mp_begin":[
      {"#comment":"called once we decided if we are single or muliplayer. MP_MODE 1:offline, 2:online"},
      {"#comment":"offline case, manually run the logic and complete logic"},
      {"call_macro":"Update_RescueTrack"},
      {"set_objective_title":"Ready to play the game!"}
    ],
    "mp_aborting_mission":[
      {"#comment":"we want to clean up our multiplayer connection if it was created"},
      {"if":{"local":"MP_CONN"},"ne":null, "then":[
        {"set":{"param":"unused"},"value":{"struct":{"local":"MP_CONN"}, "function":"Close", "params":[]}}
      ]}
    ],
    "on_got_dispatch":[
      {"#comment":"param – dispatchInfo"},
      {"#comment":"this macro is called when the dispatch is received from the web"},
      {"set":{"local":"DISPATCH_FROM"}, "value":{"struct": {"param":"dispatchInfo"}, "path": "from"}},
      {"set":{"local":"DISPATCH_MISSION"}, "value":{"struct": {"param":"dispatchInfo"}, "path": "mission"}},
      {"set":{"local":"DISPATCH_LOCATION"}, "value":{"struct": {"param":"dispatchInfo"}, "path": "location"}},
      {"set":{"local":"DISPATCH_MESSAGE"}, "value":{"struct": {"param":"dispatchInfo"}, "path": "message"}},
      {"set":{"local":"DISPATCH_LIFESCORE"}, "value":{"struct": {"param":"dispatchInfo"}, "path":
"lifeScore"}},
      {"modify_array":{"local":"Messages"},"append":{"create_struct":{
        "from": {"local":"DISPATCH_FROM"},
        "time": {"fn":"get_time_string"},
        "text": {"local":"DISPATCH_MESSAGE"},
        "mission": {"local":"DISPATCH_MISSION"},
        "waypoint": {"local":"DISPATCH_LOCATION"}
      }}},
      {"call_macro":"Update_RescueTrack"},
      {"#comment":""}
    ],
    "Update_RescueTrack":[
      {"set_rescuetrack":{
        "statusVar": "L:MY_DISPATCH_STATUS",
        "statusMessages": {"static": "statusMessages"},
        "dispatcherMessages": {"local": "Messages"},
        "activate_waypoint_commands":[
          {"#comment":"param - $index - in dispatcherMessages"},
          {"#comment":"param - $command – DIRECT-TO"},
          {"#comment":"below we set a nav line to the location, and we can select the type of mission scene to
spawn there"},
          {"set_route": {"struct": {"struct":{"local": "Messages"}, "index": {"param": "$index"}},
"path":"waypoint"}},
          {"switch": {"struct": {"struct":{"local": "Messages"}, "index": {"param": "$index"}},
"path":"mission"}, "case":{
            "0. Road Accident": [ {"#comment":"TODO: Set up for 0. Road Accident"} ],
            "1. Motorcycle Crash": [ {"#comment":"TODO: Set up for 1. Motorcycle Crash"} ],
            "2. Tipped over tractor": [ {"#comment":"TODO: Set up for 2. Tipped over tractor"} ],
            "0. Hospital":  [ {"#comment":"TODO: Set up for 0. Hospital"} ],
            "1. Meet Ambulance":  [ {"#comment":"TODO: Set up for 1. Meet Ambulance"} ],
            "default": [ {"#comment":"TODO: Set up for unknown mission"} ]
          }}
        ]
      }}
    ]
  },
  "objectives": [
    {
      "title": "Setup required",
      "commands": [
        {"set":{"local":"Messages"},"value":[]},
        {"call_macro":"mp_initialize"},
        {"sleep": "forever"}
      ]
    }
  ],
  "icons":{
"wp_blue":"
wv8YQUAAAAJcEhZcwAADsIAAA7CARUoSoAAAA6GSURBVGhD7ZsJWBRHFoCr55IBhvtSTg3gGjWJGrJqvOIaY7KajUk0EjUe6EYNJqioWRUVQxB
EFFS8QPF2UYzRKKeigEIUEEVQDpH7ZmYYmnOu3u6emuEadEbapXc//+9r+r2qpuf1q+t1VTVQR1BE9GIovoQ0DIOSCgY8d2G/
sCAEiq+TwxFXJxE2nYpJmQ6TuuCV8PS4OptfE6vvlKZC8SX892zqjF0N2g7Fl4Dbh+SLNChrQxRgIgOodICQT8gVATDCCIFpXenVAxntmLrMnj
8t5wCuPR8qHfS4cJm04kmrhZl6M6hFw2JV22S6c/TYCXPAbIUaFUDrTsU/
9SD1PqN8XA0eu8sj+wUGWlrX1onJf1Qe6e3YuYJzJkT+d7Lqwi55aTJs9Z+195cvd3ux60KiH/
mA7AbVPwbF5ASTGZ1uxq1DsdBLsdPIdK0hbtL5/AI0KmW8GpN/Dz67MpkU+sqGDRuYmlinHVTe0Nf3VyhRBPKMjy1dugRqVIA/
7qqUinSoUQDhvzSpRj7UsNoQMOH5xWh0Qx2bnj1pb7z0hnPmzAG74xMmAkBRrdkeWxBPnL9Kq64kE/pMmpg07cCB/
XreAYHDyTRNCY56FAhymlQ9CnEYiZoVDZkgV9glj1PViJ2LSp0JcyGdLliSVJLj6uqq8Qi1e3+InWFDubzjHpKejt61N8jUtqa+vfMPwaxOPy7
BPFNqby9ZsliLaqfkCf6YOKd+uzbU3f0HNpnWF8LicjcTZ3NBo4xMoILjsZnfd3n0voI8g+OLBmjkVExoCKWXo3kpcVug8GI0vuHesqJDUOw7x
g0odSWsDa73K0sorQ2UkyYna1f477e79RevDrWRL/Qe21oIJINMKLn3K/Q96jl841kgcba2F0kkFcZkGr3IQvHiFWNHws/
YEJ708vLiwRyagBvlkVx8Wym7pgtLSZkOhJy+NEVZ/0jy4UBLF/
RRoWJQhhyOfri53wycOXMmOHLhyjif+OdReqgANwyPeXBjnMtFzfASBYSBxJHfhDlU8dv2Xc/
w2eLtbQpzNYbsCpbfq8sOH8QeLq0m6nUvDVuvAcxpkVRNrS0PKUmK2Ofn54/CHI2YPXs2cF22ZlS6sfOu/W/pTGst1cNTWYrM7jBbgLlDu9z/
```

```
XqGbqq9auXIFa8wn33yebjl0y3EH3ihxORdPVRNNs1qByWCxfFM6/9+M4kT/tcuWZsEctXh4/
KjrMuuf824aWG88aclxltUQ91XTRbKawEAbuXRjZvVx/
ap032WL5pfAnFdn7ty5IPzi1ZFEURqJhF3HVzxtddyT5K1badDdvFUpaiHrGyToZsmZznq/ExwcpEsY9P33yxVBbUYz3kDa6WMgCW7g/
LS6IqXscackkZTpArOkjuxWDkT88Q9aFa+SQ5cTFhKGcWtF9BpFukAYhh+2NSIxTOkzlIVbChRTlD9FnHYiBboRHJXpT0bVFKLZxIuG7LsY9e0
QC4PHaWkUTjhRCl7/
vknnV0CNhuAGIvmNlBYxZY0k+HQkuTaFiagdeikzUGQ52hOK9ISHoh1TXRRCmQfR3AEIYCr658CDoW+TAgVQ2FGzgZVdGxkTyhwnbiGTaAVetI
GxT8KIs5mAT9nkECUeDAk5QL5cYHkJvwIgAfWlXApLhgJCo9JUc5y6tXg/SGFDoeRJE81sNkARLM0X3oMiJVBi4DkznSHK1YphZY/
IBapFixZSO3PWJ/AIhlks6ChWvIjDLt/8O9T6BEWVGQGLq9ueQoUkw2qoNxT7BGWtbXxd4XYokhyz446BYv/
iuyvAvEerfdCCt2QZZS1ZKzZs3WJ8MCpjMxH3IfnEpGXP8ddMIJIp0iWYTjUf80gpTwqPjJ305ZdfaN1w1P4D/
gKOTJj13fR7pk47jjrofCAu18dTewm+DRrAd8KWwlNjrR1hCrh7967JzBHv1QsL2AjAelmg0m0G1ubtkg0ZVYebcmJ3b/Zcp37C83TU/
bkWfL4UPCaKBXqj+5EuxohoJSCh4vLZKzdGEVNw2kLM0YRcz9jmUCVo67yXoechx1ilKLbsXu0T4v8QxZpmz6c0G1Il+yGTf4zz9Ib35tVrKFp
A7cDNzY0x/mu3WUn69rtO8njOQNIz0PXPPDZWVcT7jxy3YQ6ZtG7He8bu1RVslrp/IDpjpmULWFTdnje9uc437drRiMDAPS/cvrQv/
NQ77dbjNu10MZ8jKOYwgJSYfuuOHLAHNoMl5a2PR1Vm785Luno+KChYQuS8tNIuXLiA8eGseVMLbFy2HHTUmdxaQvxAd4/
jheDCUN1rfdLzmADu4E+g2gG3EdiYiyWbHtYEiAuS93i4r9R8Bf5V2Lhxg+6PUekJZN3pBKtcMYl+6reYMfPmzdW65VKKt/
d2BmGM986dHfPPRB+Y09w//
aBacAPDbhbugBqp29QIKJuf6Tu4QQ7VaBvUSH3P9UcHoNYnKBqLW0FxLRhASEFBe4leHaDZMbuIMy0wFzQphjYcvDP2UspUQIkHf35QFwFFcMv
SdiUUKYESAxmFtwKgCCL1WQOhSDPwYj0SGTOTGEsZxT33u74qlAWsBA8HjcSjaAQsqBMXwqQ+Q6mBYXbc0cT5b4KynWQCBVBnILcBSKoUWwFyb
p4/TQp0YjAR5xHdC4VdDAFlHtyYXeMHRUqhzMDm3IQghUSjIbgHePFyazu9wFMApa2YCFx/zq46ApX/b8JiszzJuBLv+IOvP6BP4PG/
gpkAhe/+GEZuLqEpFDcRajhxIlyv/jlTZVtzri7yS0CAPVRpBS0dKLb+wANgnd/uBwDDkZ/
+BBVa0b8v1L3wdXpddSRmZglVEnvbRnGJlSH5UkYnaFcD16/3ZETqsro4j6CknsnZtm0rpZsDqIB2DnQe/9lXoNmIlJnmDYA9UEjKoFUPDBn/
xQKFQh9o58D7Vs5eUARLasRPFpW25EAVJBnb/wxF2kC7PpBdzsckVeTHhiC8OmVeu1TMXmEzhXw7ZFo0Apm9IS37bVqwL/zkMEXwjMd/
eYptNytXrmB2Xq04dObiOPJimkCrJozYTV2vNMlRX0ZuoDt06LDMwRRTLXAILEbTcJsGTbDgN0mVNe1YfA7uTAUhMdk7lek8tIFWbyW0qYG+vr
/q1j5HFGEKqxUUJZ5VrQxgxSmBACgWDtBcHcTHz4+mM7f9yNGYLMVXHfhhVS8i1w47YyhqVO1X23s9U/
EFOg2gTQ2MM7VyhyLYlFlzFIoqvNIbf4ci2PO+/
Qoo9ju0cODatWuQSD2mlUKTgNbs6B7TV4Z1GXiaYq6xrJ7N8fLaTAvb+zWmIhzHM7ewFQ+f7L1z4LjFRJqObT1oszJXaxdSyMcwgSJG3M5/
cgbLvuYtF6GFv/
ziQ+ksrjZQ5sBVq1bqOI4YPcTsLReXKjlnxlNT89FXnVmOogYGQ8ZnASDv5dOlbqxvK44LmKhmewPOoj8r8k8yB2nwWQDuT0YbYBhLgKGJTP5F
ASh05tdk2bGkf/CLMtIKsx88Dw7e17Gc3QfUPtGmTf/iDX7nr06Igd3kWibvo8dmJu/+4YzYttRxELkAdwbGwa/
Sxvd45IGPrBxLBJhwJNLZheKcEY2NtzlNZYmYsDg1NTmxLjz8hFa7XBcvXsQa/
8kMc7a+zdhGHYuPc3hGE644cd4WtLKZkjrcNnKzizY2EpVYAhAjMdAxaAO2Ek7L5yVo6lBUGAdERXefP01/+iD1riA29qbicghyLOLKGDenGel
ARjhFE/AfYrYBpqkEmOkjspkFzQXvos1Juq1Vt9lNtcl3bkXXhIaG9RhF6YC7+w/sSVM/
G9jAMpzWqmszIUOfMynamecgQAGTbCUyYrZMQ6czpMA/
7cT7yPz5ruBD1zXrE6wGr40dOsASrZAhANXFr9DCoYgYICZSoGcuxT7PR8pGCAVZlrLmZNBcmpz/MCXff4fP691N1AvEd6POI8cPbdO1/
CjXwHTadTtDl1JOu16baADAGoidUsShaS3F64RuC9C1YAJLqbh9TW6tT3X8SR9t6jjw8PhJ12XKDNsmlunHrVzLsfd47ImxTvo2wnoJA2vEnS7
XgVdqAt5iue2AbYLXZLZM+mWhNGuksCKiKSchwNPTU+2gcOHCBWalwfA9+cYWE38bwhzJlzBZEgHuhFai5mgxKDPaAWLQAozMOfLpBU2VH6CSV
G5LZRq3qSouNzOlMD+/
oOny5cvw4hejlQNfhXXr1jIdh71jA0ztJon1bCY85OlMjnbiOfLbmExJLf7zMqK2dzz8AlBXdMbFYghUu2BV3yCpLjLs9HUvMVi0AIapGJgaMG
Sf5jcXjUbFydzmsjsyflFyXlZGMT5Y0LI7oZSAvUFWxM5b4i0DKVS/d2X79m0DwEPl7l0Z5hsSMhJmvYGAXJGHr2pHL0Z/BpNVHLz654/
KfE6FUK2T+wPavMq55TWkQBE8GPQ2+eFEZ+5YOayFIlhW2vYAiv0ObRw4rDxruyIWA+DYYN57pNCJ82Zs1brweP7zbVDsd2jjwIcJ1+IRY8UCk
qTCABw4eVbVxx08EzkB4ys2fwEjFKRGn72uUN7QhW/
vV5co+7mdsXmXYDLYFV9yRZk+tBxtgslv6E5oVLq70lE8tE4182wk6vh2NTQum1Yrc7RpwgTysvQjgK2oYGi+AeLj729xKOy4aUM+SxGvMltBQ
fL5UFKmCa89kNYW69pGcUUJj/wi5ZAwdxc+rMhWGf/lX4Ru5MyXNxia0W53Aq3Yf6PwgLK5OlSj7fbVgnal7hebdwFeRhto1YQJ5AW3/
ACQknJxDYNTgh+kAuSA9SyRsv3hVEG7JkygUyPA2kph2AJhDRICqbUJ7eylXQ0kWF6ApkJRxdKytiwo0gpaOnBE5WPVW4mScd0+nqYLtHRg4u9
n4xBjEdRweI0g8VKYalnzDRrwzf3KMuXo61TR2AKTaQdtY6oJDDR4gbU1b0ZLxR2nqyemJCYmwhw6AcB/
AID93zD+9+L6AAAAAElFTkSuQmCC",

    "h160_icon":
"
JcEhZcwAADsMAAA7DAcdvqGQAAAd+SURBVFhH7ZlpbFVFFMdPdyhlbUG2KDbIYlmUJSCLoEYU06AYIdHIZhUriIUiglFjTEz8AEhViBLlgzHBo
BKNGPWDQdAYRYmgAdGQKsomINAFuvfV/39m7ut99713371viyb8kpNZ7r0zc2bOzJyZK25kmVACa6QjJ0Mu7T4u/
ZH8Vefa6MALFJOUTBOSOSYkXU0YRp4J04Jp6Y06FdpaiwMm7Hxo6XeyXFpVhhN7B5BIxQZxfRiJ1SYMoaVSAgi66VQUTLsWMGxdLR1tEP2kkww
ThmFXKmN9+HtKj6cnyQfmxeAI4+XHGQZC6xoKWQKZpVLRcHalnag9Z31kwr6Me8atxqQx0YROBseqPQeiexMvWi8Hngz/
yNk5tNTXIaW2gbsjc0OUcXTQDjmio4prTBhCtOFoMCGpN2EIzib0gF3WtsGc87JNDoAV3YTgO53qJMN0wqM6qWGeoydZyfT2CB0VAnsy1hBEY9
qhxdJs4r7oAZmmo+GEjQ9xNjHSfHRjXHFPecHEfcFainQ0PvqYMClEXQAMj0Ce0NEQhpv+W6GTGpMX3AviJZtGtnaifGjSQ1mwJUhzzRNjiHcx
7oafUS1ABfU1TSK9upgcG7AQLoP/
6FR0fJkRyMI61Ob8yGtlJNYYOimO0sLbTRgTq0LuTwN0VHIhSzk+XFUR5xj14wNw1IROPoeMzM6UtdbYNq5UW3AF5DYI/
aO4GWAVSkF6rs52Jd+8GzcPsYBIPkGqGGJafL9Oesev0dDXmAKx9i3aENPjVCqFWPv5fTrpHb8aWljf2XbOK3ijN8Tv8hc/
xjDoz3li6kB5jt/
UrIjPycg0FXbXyU444c2znjpHMc0sBAmdx2aZgi2XgVa5jHmUJSXyMdJqDTbvjWI8UaaawpSrYTl0liCLh9kSE4853l7m4TfY72aiwPOI52WEF
8lF4BDeGYE4K3XF68Tfm7leKpoqBft9GFt/vyhtCH/TSXd8mfylCgl0yw3/
BtrR6fBkmb6WtoJX5BYTDYLKXkLgeRr40pAH6ZysiBp6LsePhnmwCF8NjIRbhXw2/KmJ8r6ZAk1ln8ke/
SgUc4pcDBkMcW2U9ZAb61+QFkgxCqim+Z9vkPaiLeq4dhDCk+cMVBxSaV2zdPR8Vc3RKY2r5JMu2brMfSflxOTtsgZRHjh/
gag7FqvCYShImXUbzqQ5L0sJonzJSQ7eY6OCYPyGIXB6cwUQllFYOV7KNt4q9+I9LhABp/pMu07eBrh/XXM6uw0FsetjTfgFEF6j/
aBSPhltxrSjd56sM3me8TUPDcGuvtgs75qoZ+KpkDcPFp7OE3biqdB+TWW/
ufBEXBWerFeLNYllLAnBzZdzko7vZjPZGWcefR9POKeFG6WwzF0mHgLmbgfmbly95QmbT+OnwYq4WpXV+ZVnj84irgrbedTURL6DTgGzTz+Wvs
qukCr6nChXXuODkFQdE7mF8jRRDBkDudrkpYX5nPrfPiDViHs5vrBhoyG8Qr4b4rYY9ji4UBpZPm/wds1Vvshk5qunaWKC7fK/
DEInJhL5ECqzju/aBXkzzHO7BRTsmS/nzPNyCK/F4sb3wuuADZuObWo3XcrT2CMHvqHczT/
VU13+2OZKOZAbxbDggC1FsB1yGUJLKIdyVRfgnhZukUKkayFxk6iCFlz/
x2PH2odNJAONfhhpNroRwmPllwgjcqxGAte+qc11R6nsmz9SRuD7Z5DcDKljfiIkS0ELmtM9UGhHBwwsc4PMLBslS96aLYv043BohzjmljbhgM
B/
QVBuHrI+glgeU0IkW0GLvgcWyNEb+ofcp8QEyvGv4H6dSg6pUpCw7KmYn19HuPEIobFVOvKr1N3rcZ2TPFLnQooUtVTKV7GUI+bUN1ankksqFe
QK68dCwq4Ok0GyFKTbzZ9OEyCLzi2XttbVcirHn88xCMLfNgnte0789DBHhPehvKkoaVgpm2BawQ7iwXT/KWmatF39MOGlwWnIzVhReY/
qSgP89m5V8iw65cVsW5dzJa5rlkCv12QZkrxy4f56AcLtx9Mqa1ewX2ul/
J2VJRnVF6T1um3Kx6wZf5XcuX+hrNKviOBZy9BtytWiEmcgrCwahRjNs0X57paC1ZN/AL/
QqSD8ht4Rf+YNgVxfXyGbCnIjl2W2G25HOyF0GhSRRpB/
DzlKXLJpLochvBzhHXDn0cw7HMW9Jh4GOxMdRo+lXufEBT2gOajnvcutEiiokoFIs/
N9mWi80LQXovJtOtlJG7orZ6M6KfyhcxKGI87BOQE5ywxX00kSnCs7T9WF3FgpoNxMBMlSjlyC/
AhRypF0KEhqB22VOSauqGlS5v6zTqWOdClIaDZBJrwjzyNgj6eUdCp4prlNLXaK6hrlc6b8YiedCtbuOCLfmzhRq1yqSdUqOgan/Z/
8Fr74U9n09mF1FnTbW32RKgW5D3Fztv+v6dovX5aeWa4cCG7uvOawLzJ093g9T48l5D/P/
4VR1h0OpUuWrEQe98iUks45GFxgyKACddmUctKp4HFs9sFVs7pW3dMk5Vriv0T33fPkGEKeSq6QOCL/Ap3jTWX/GRgjAAAAAElFTkSuQmCC"

  }
```

```
}
```

RescueTrack Test Program

This program shows basic usage of the RescueTrack UI.

```
{
 "title": "Rescue Track Test Program",
 "data":{
 "statusMessages": [
 "0. Dispatch accepted",
 "1. On the way to the scene",
 "2. At the scene",
 "3. On the way to the hospital",
 "4. At the hospital",
 "5. On the way back to base (Available)",
 "6. At Home base (Available)",
 "7. Unavailable for dispatch"
 ]
 },
 "briefing": [
 {"title":"Rescue Track Test"},
 {"text":"Enable Rescue Track (visible on DMAP STATUS):"},
 {"buttonbar":[
 {
 "title":"Enabled",
 "commands":[
 {"set": {"local":"IS_RESCUE_TRACK_ENABLED"}, "value": 1},
 {"call_macro":"Update_RescueTrack"}
 ],
 "select_condition":{"require":{"local":"IS_RESCUE_TRACK_ENABLED"}, "eq":1}
 },
 {
 "title":"Disabled",
 "commands":[
 {"set": {"local":"IS_RESCUE_TRACK_ENABLED"}, "value": 0},
 {"call_macro":"Update_RescueTrack"}
 ],
 "select_condition":{"require":{"local":"IS_RESCUE_TRACK_ENABLED"}, "eq":0}
 }
 ]},
 {"text":"My Status:"},
    {"buttonbar":[
      {
        "title": {"struct": {"static": "statusMessages"}, "index": 0},
        "commands":[
          {"set": {"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "value": 0},
          {"call_macro":"Update_RescueTrack"}
        ],
        "select_condition":{"require":{"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "eq":0}
      }
    ]},
    {"buttonbar":[
      {
        "title": {"struct": {"static": "statusMessages"}, "index": 1},
        "commands":[
          {"set": {"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "value": 1},
          {"call_macro":"Update_RescueTrack"}
        ],
        "select_condition":{"require":{"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "eq":1}
      }
    ]},
    {"buttonbar":[
      {
        "title": {"struct": {"static": "statusMessages"}, "index": 2},
        "commands":[
          {"set": {"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "value": 2},
          {"call_macro":"Update_RescueTrack"}
        ],
        "select_condition":{"require":{"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "eq":2}
      }
    ]},
    {"buttonbar":[
      {
        "title": "Other status",
        "commands":[
          {"set": {"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "value": 3},
{"call_macro":"Update_RescueTrack"}
```

```
        ],
        "select_condition":{"require":{"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "gt":2}
      }
    ]},
    {"title":"Dispatcher messages"},
    {"text":"Location:"},
    {
      "text":"You must select a location on the map prior to sending a dispatch. Go back to the map and click
a location, then come back to send your message.",
      "color":"red",
      "show_condition": {"require": {"has_location": "$MISSION_SELECTED_POI_LOCATION"}, "eq": 0}
    },
    {
      "text":{"text":"{0:LOCATION}", "params": ["$MISSION_SELECTED_POI_LOCATION"]},
      "show_condition": {"require": {"has_location": "$MISSION_SELECTED_POI_LOCATION"}, "eq": 1}
    },
    {
      "text":"Name: (required field)",
      "color":"red",
      "show_condition": {"require":{"local": "dispatcher_textbox_name"}, "eq":""}
    },
    {
      "text":"Name:",
      "show_condition": {"require":{"local": "dispatcher_textbox_name"}, "ne":""}
    },
    {"textbox":"dispatcher_textbox_name"},
    {
      "text":"Message: (required field)",
      "color":"red",
      "show_condition": {"require":{"local": "dispatcher_textbox_msg"}, "eq":""}
    },
    {
      "text":"Message:",
      "show_condition": {"require":{"local": "dispatcher_textbox_msg"}, "ne":""}
    },
    {"textbox":"dispatcher_textbox_msg"},
    {"buttonbar":[
      {
        "title":"Send Message",
        "commands":[
          {"modify_array": {"local": "Dispatcher_Messages"}, "prepend": {"create_struct": {
            "from": {"local": "dispatcher_textbox_name"},
            "time": {"fn":"get_time_string"},
            "text":{"local": "dispatcher_textbox_msg"},
            "waypoint": {"location": "$MISSION_SELECTED_POI_LOCATION"}
          }}},
          {"set": {"local":"dispatcher_textbox_msg"}, "value": ""},
          {"call_macro":"Update_RescueTrack"}
        ],
        "select_condition":{"and":[
          {"require":{"local": "dispatcher_textbox_msg"}, "ne":0},
          {"require": {"has_location": "$MISSION_SELECTED_POI_LOCATION"}, "eq": 1}
        ]}
      }
    ]},
    {"text":"outgoing messages (Dispatcher_Messages):"},
    {"text": {"json:stringify":{"local":"Dispatcher_Messages"}}},
    {"#comment":"you may add comments as needed"}
  ],
  "macros": {
    "Update_RescueTrack":[
      {"if":{"local":"IS_RESCUE_TRACK_ENABLED"}, "eq":1,"then":[
        {"set_rescuetrack":{
          "statusVar": "L:MISSION_RESCUETRACK_STATUS",
          "statusMessages": {"static": "statusMessages"},
          "dispatcherMessages": {"local": "Dispatcher_Messages"}
        }, "activate_waypoint_commands":[
          {"set":{"param":"ACTIVE_MSG"}, "value": {"struct": {"local": "Dispatcher_Messages"}, "index":
{"param":"$index"}}},
          {"copy_query_to_location": {"struct": {"param":"ACTIVE_MSG"}, "path": "waypoint"}, "to":"temp"},
          {"set_route":"temp"},
          {"set_message":{"text": "Go direct: {0}<br />\n{1:LOCATION}", "params": [
            {"json:stringify":{"struct": {"param":"ACTIVE_MSG"}, "path": "waypoint"}},
            "temp"
          ]}}
        ]}
      ], "else":[
        {"set_rescuetrack": null}
      ]}
    ]
  },
  "objectives": [
```

```
    {
      "title": "Done",
      "commands": [
        {"set":{"local":"IS_RESCUE_TRACK_ENABLED"}, "value":1},
        {"set":{"local": "Dispatcher_Messages"}, "value": []},
        {"set": {"local":"dispatcher_textbox_name"}, "value": ""},
        {"set": {"local":"dispatcher_textbox_msg"}, "value": ""},
        {"call_macro":"Update_RescueTrack"},
        {"sleep": "forever"}
      ]
    }
  ]
}{
 "title": "Rescue Track Test Program",
 "data":{
 "statusMessages": [
 "0. Dispatch accepted",
 "1. On the way to the scene",
 "2. At the scene",
 "3. On the way to the hospital",
 "4. At the hospital",
 "5. On the way back to base (Available)",
 "6. At Home base (Available)",
 "7. Unavailable for dispatch"
 ]
 },
 "briefing": [
 {"title":"Rescue Track Test"},
 {"text":"Enable Rescue Track (visible on DMAP STATUS):"},
 {"buttonbar":[
 {
 "title":"Enabled",
 "commands":[
 {"set": {"local":"IS_RESCUE_TRACK_ENABLED"}, "value": 1},
 {"call_macro":"Update_RescueTrack"}
 ],
 "select_condition":{"require":{"local":"IS_RESCUE_TRACK_ENABLED"}, "eq":1}
 },
 {
 "title":"Disabled",
 "commands":[
 {"set": {"local":"IS_RESCUE_TRACK_ENABLED"}, "value": 0},
 {"call_macro":"Update_RescueTrack"}
 ],
 "select_condition":{"require":{"local":"IS_RESCUE_TRACK_ENABLED"}, "eq":0}
 }
 ]},
 {"text":"My Status:"},
 {"buttonbar":[
 {
 "title": {"struct": {"static": "statusMessages"}, "index": 0},
 "commands":[
 {"set": {"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "value": 0},
 {"call_macro":"Update_RescueTrack"}
 ],
 "select_condition":{"require":{"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "eq":0}
 }
 ]},
 {"buttonbar":[
 {
 "title": {"struct": {"static": "statusMessages"}, "index": 1},
 "commands":[
 {"set": {"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "value": 1},
 {"call_macro":"Update_RescueTrack"}
 ],
 "select_condition":{"require":{"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "eq":1}
 }
 ]},
 {"buttonbar":[
 {
 "title": {"struct": {"static": "statusMessages"}, "index": 2},
 "commands":[
 {"set": {"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "value": 2},
 {"call_macro":"Update_RescueTrack"}
 ],
 "select_condition":{"require":{"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "eq":2}
 }
    ]},
    {"buttonbar":[
      {        "title": "Other status",
        "commands":[
          {"set": {"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "value": 3},
```

```
          {"call_macro":"Update_RescueTrack"}
        ],
        "select_condition":{"require":{"var":["L:MISSION_RESCUETRACK_STATUS", "number"]}, "gt":2}
      }
    ]},
    {"title":"Dispatcher messages"},
    {"text":"Location:"},
    {
      "text":"You must select a location on the map prior to sending a dispatch. Go back to the map and click
a location, then come back to send your message.",
      "color":"red",
      "show_condition": {"require": {"has_location": "$MISSION_SELECTED_POI_LOCATION"}, "eq": 0}
    },
    {
      "text":{"text":"{0:LOCATION}", "params": ["$MISSION_SELECTED_POI_LOCATION"]},
      "show_condition": {"require": {"has_location": "$MISSION_SELECTED_POI_LOCATION"}, "eq": 1}
    },
    {
      "text":"Name: (required field)",
      "color":"red",
      "show_condition": {"require":{"local": "dispatcher_textbox_name"}, "eq":""}
    },
    {
      "text":"Name:",
      "show_condition": {"require":{"local": "dispatcher_textbox_name"}, "ne":""}
    },
    {"textbox":"dispatcher_textbox_name"},
    {
      "text":"Message: (required field)",
      "color":"red",
      "show_condition": {"require":{"local": "dispatcher_textbox_msg"}, "eq":""}
    },
    {
      "text":"Message:",
      "show_condition": {"require":{"local": "dispatcher_textbox_msg"}, "ne":""}
    },
    {"textbox":"dispatcher_textbox_msg"},
    {"buttonbar":[
      {
        "title":"Send Message",
        "commands":[
          {"modify_array": {"local": "Dispatcher_Messages"}, "prepend": {"create_struct": {
            "from": {"local": "dispatcher_textbox_name"},
            "time": {"fn":"get_time_string"},
            "text":{"local": "dispatcher_textbox_msg"},
            "waypoint": {"location": "$MISSION_SELECTED_POI_LOCATION"}
          }}},
          {"set": {"local":"dispatcher_textbox_msg"}, "value": ""},
        {"call_macro":"Update_RescueTrack"}
        ],
        "select_condition":{"and":[
          {"require":{"local": "dispatcher_textbox_msg"}, "ne":0},
          {"require": {"has_location": "$MISSION_SELECTED_POI_LOCATION"}, "eq": 1}
        ]}
      }
    ]},
    {"text":"outgoing messages (Dispatcher_Messages):"},
    {"text": {"json:stringify":{"local":"Dispatcher_Messages"}}},
    {"#comment":"you may add comments as needed"}
  ],
  "macros": {
    "Update_RescueTrack":[
      {"if":{"local":"IS_RESCUE_TRACK_ENABLED"}, "eq":1,"then":[
        {"set_rescuetrack":{
          "statusVar": "L:MISSION_RESCUETRACK_STATUS",
          "statusMessages": {"static": "statusMessages"},
          "dispatcherMessages": {"local": "Dispatcher_Messages"}
        }, "activate_waypoint_commands":[
          {"set":{"param":"ACTIVE_MSG"}, "value": {"struct": {"local": "Dispatcher_Messages"}, "index":
{"param":"$index"}}},
          {"copy_query_to_location": {"struct": {"param":"ACTIVE_MSG"}, "path": "waypoint"}, "to":"temp"},
          {"set_route":"temp"},
          {"set_message":{"text": "Go direct: {0}<br />\n{1:LOCATION}", "params": [
            {"json:stringify":{"struct": {"param":"ACTIVE_MSG"}, "path": "waypoint"}},
            "temp"
          ]}}
        ]}
      ], "else":[
        {"set_rescuetrack": null}
      ]}
    ]
  },
  "objectives": [
```

```
    {
      "title": "Done",
      "commands": [
        {"set":{"local":"IS_RESCUE_TRACK_ENABLED"}, "value":1},
        {"set":{"local": "Dispatcher_Messages"}, "value": []},
        {"set": {"local":"dispatcher_textbox_name"}, "value": ""},
        {"set": {"local":"dispatcher_textbox_msg"}, "value": ""},
        {"call_macro":"Update_RescueTrack"},
        {"sleep": "forever"}
      ]
    }
  ]

}
```

H145 Mission System Documentation (old)

This documentation is early and subject to change.

Last Update: 2022/6/23

Basic mission details

A mission json file is referred to as a Mission Descriptor. It can be loaded into H145 and then operate alone while the user conducts the mission.

title	Title used when displaying your mission in a list
aircraft	Must be H145 (array of supported aircraft)
applicable	Array of variants. If omitted, all variants will apply. Inapplicable missions will be hidden in the mission catalog. EMS FIREFIGHTER
api_version	Must be 0.1

start_info	The start location or start locations can be specified. This will prevent showing the mission in the Library as it has a natural start point on the map.
	If you do not specify a start_info, then you will use the library to begin your mission.
	
	`location`
	Specify [lat, lon] for the fixed starting location.
	`icon_src`
	Specify an HTTPS or data URI. This icon will be shown on the map.
	Suggested size 32x32px.
	`query`
	A data query in the same format as used below in missions

Loading missions from a server

To load missions from a server, do not provide locations/objects/threads/objectives, instead provide a URL which is a websocket server. When the user selects the mission your server will be contacted and at that point you will be able to manage the mission system indefinitely until the user selects another mission manually.

Url	"localhost:40510"

Authoring mission packs

Missions can be added to any other Community package or be authored alone, the only thing to do is create an hpgmission folder within your package, and place a folder hierarchy below with your mission json files. All contents (folders and json files) below hpgmission across all Community packages will be merged into the catalog list. Feel free to create a folder structure for regions or otherwise create organization.

Mission sections

locations	Table of locations referenced throughout the mission file. These are locations like “accident_site” or “hospital_helipad” that mark the coordinates. You can easily copy/paste a location from Bing maps or Google maps by right clicking and selecting the coordinates from the menu.
objects	Table of dynamic objects created when the mission starts. The objects have a title which is what identifies them in MSFS (like an airplane), and they have a default location you may place them at.
threads	Table of background execution threads which occur regardless of the current objective. This allows parallel processing of logic. You may wait for a specific variable to be true, enter some processing, and then quit forever or start the process again. This can be used to design triggers and add other logic to your mission, like enabling a sequence of events only when the user enters an area, regardless of where they are in the mission objective list.
objectives	List of Sequential tasks the user will work through. Every mission has at least one objective and when the list of objectives is complete, the user has finished the mission. Each objective itself is a set of commands which execute sequentially. You can direct the user to an area and then proceed to the next objective only when they have arrived at the area of interest.
userActions	TODO - Not yet documented

OBJECT

Objects are created when the mission starts and manipulated throughout the mission. The VAR 1 variable is commonly used to configure the visual state of the object.

title	string	Title from an aircraft.cfg, registered in MSFS. See the section Creating Dynamic Objects
location	LOCATIONREF	Location to create the object. Optional: objects without a location will be created at Null Island [0, 0] and may be later moved by using move_object.

Special object variables

These variables are interpreted by the system in a special way.

Name	Function
VAR 1 VAR 2	Mapped to simulation vars unique to the object: VAR 1: (A:GENERAL ENG THROTTLE LEVER POSITION:1, percent) VAR 2: (A:SPOILERS LEFT POSITION, percent) These variables are unique for every object and will be available in the model behaviors XML. This allows each object to have independent visual states and behaviors.
COUPLED	Object user coupling mode. When an object is coupled it will be modified automatically based on the coupling state. 0: No coupling 1: Couple to hoist position - Object will be continually snapped to the position below the hoist 2: Couple to external cargo position - Object will be continually snapped to the position below the cargo hook 3: External cargo position auto-couple armed - Object will switch to coupling mode 2 automatically when within range. 4: Firefighting target (fire) - The user may use the Bambi bucket to reduce VAR 1 (quantity of fire) for this target. VAR 2 is set to the most recent quantity reduction by the user bucket dump. 5. Firefighting pool - VAR 1: Radius of pool (METERS). VAR2: Depth of pool (FEET, negative)
MODE	Object mode. The mode is used to control the physics and behavior of the object. 0: Hold position on ground 1: Repositioning mode - Use LAT/LON to configure the next location, and then set mode to 0 to switch back to ground hold. 2: 3-axis Velocity control - Use VELOCITY X, VELOCITY Y and VELOCITY Z to control the object physics over time 3: MSFS default Physics
WP INDEX	Activation navigation index. Set index 1 to activate the waypoint engine and cause the object to rotate on the yaw axis to orient such that velocity z will drive the object to the waypoint. 0: not active 1-5: navigation to waypoint 1-5. The waypoint engine will set the WP INDEX to 0 upon reaching a situation where the next waypoint (WP INDEX + 1) is a waypoint at location 0,0. The waypoint engine will also set VELOCITY Z to 0 at this time.
VELOCITY X VELOCITY Y VELOCITY Z	Object velocities. Only applicable when MODE=2. These velocities will be sent directly to MSFS to instruct the object movement.

THREAD

Threads are background command lists which execute independently of the currently active objective. Threads may be used to schedule activities regardless of where the user is in the objective list.

interval	milliseconds	Update interval (higher is better for performance)
commands	COMMANDLIST	List of commands, execute in order.

OBJECTIVE

Your mission must have at least one objective or it will complete immediately after starting. The objectives each have a list of commands and when one objective is completed the first command in the next objective will be started. When the last command in the last objective finishes, the mission is complete and will end.

title	string	Text to display to the user for this objective
commands	COMMANDLIST	List of commands, execute in order.

Commands

Commands are executed one at a time from a command list, and each command may execute nearly instantly or take some time to finish. See API Reference COMMAND, QUERY and LOCATION

Dynamic Object Library

H145 Crew

The H145 Crew object contains the crew, pilots and stretcher. The visual states below may be configured for the various standing/walking/waving states.

title	$TITLE Crew Airbus H145 ADAC Crew Airbus H145 DRF Crew Airbus H145 CMH Crew Airbus H145 HeliOtago Crew Airbus H145 Norsk Luftambulanse Crew Airbus H145 Bundeswehr Crew Airbus H145 CAL FIRE Crew Airbus H145 San Diego Gas Electric Crew

NOTE: Livery authors should add their title to allow **$TITLE Crew** to work, which is automatically replaced based on the livery name, and with a check for the livery author to have provided a crew title replacement within their livery json file. See the main user guide livery authors section for more information on this.

Visual states

VAR 1	-1: Hidden 0: HEMS standing 1: HEMS standing with (backpack) 2: HEMS walking 3: HEMS walking with (backpack) 4: HEMS crouching on ground 5: HEMS crouching on ground with (backpack) 6: HEMS crouching on ground with (backpack on ground) 7: HEMS waiting 8: stretcher no-patient 9: stretcher patient 10: stretcher walking no-patient 11: stretcher walking with patient 12: stretcher standing1 no-patient 13: stretcher standing1 with patient 14: pilot standing 15: pilot waving 16: pilot walking
VAR 2	**Only applicable to VAR 1 values of 14-17.** 0: Black pilot with headset 1: Black pilot with helmet 2: White pilot with headset 3: White pilot with helmet

H145 Injured Human

The injured human object is a human laying on the ground waiting for medical attention.

title	Airbus H145 Injured Human

Visual states

VAR 1	-1: Hidden 0: Injured human in pain 1: Injured human packed into hoistable stretcher

H145 Waving Civilian

The waving civilian is a human standing waving, attempting to get help for his fallen friend.

title	Airbus H145 Waving Civilian

Visual states

VAR 1	-1: Hidden 0: Civilian waving **NOTE:** Use L:WAVING_CIVILIAN_STOP to 1 to stop waving

H145 Flare

This is a marine flare with orange smoke.

title	Airbus H145 Flare

Visual states

VAR 1	-1: Hidden 0: Smoke auto (ON for high visibility setting, OFF for realism) 1: Smoke on (ON fo both setting positions)

Creating Custom Dynamic Objects

You may create your own dynamic mission objects that H145 can spawn. They can use the same COUPED and MODE flags as the built-in objects.

Unpack the **Mission Object Sample** from **Tools**. Included in the sample is a blender asset which has already been exported for you into the MSFSPackage, which is an SDK project which you load in MSFS to compile the asset and produce a package for redistribution.

The procedure is as follows:

- Prepare an asset. Follow `Blender Asset\Ambulance.blend` as an example.

- Export your asset into `MSFSPackage\PackageSources\SimObjects\Airplanes\ sample-ambulance\model\H145_GenericVehicle`

- In MSFS, enable developer mode and load the project

`MSFSPackage\MSFS_DynamicObjectSample.xml`

- Copy the output package hype-mission-dynamicobjectsample from

`MSFSPackage\Packages to your Community folder.`

Now the object is registered with the simulator and available for creation. Using Scenario Editor, use the More Objects toolbar item and find Sample Ambulance in the list. The object can be placed and used in H145 missions now.

In order to package multiple objects you will need to change the name. To change the name of your object you will need to edit these locations under `MSFSPackage\PackageSources`:

File	Text to change
`ExtraFiles\hpgmission\packageObjects.objmeta`	Airbus H145 Ambulance Sample
`SimObjects\Airplanes\sample-ambulance\aircraft.cfg`	Airbus H145 Ambulance Sample **Tip**: isUserSelectable=1 will allow you to see the object directly, and isUserSelectable=0 will ensure that your distributed package doesn't have extra stuff showing up in the aircraft selector menu for the end user.

To combine multiple assets into one package, use MSFSLayoutGenerator.exe to update thelayout.json after combining all of the output folders.

Mission Server

A mission server may dynamically generate and apply mission descriptors as well as send other commands and observe status. The server is essentially just a websocket server which listens for the simulator to connect and then speaks a JSON RPC type protocol.

A very simple **Mission Server Sample** in node.js is included in the **Tools** folder.

Commands sent from the H145 to the mission server

{control_msg: "hello"}	After connecting the H145 will alert you that it is ready for you to send a mission
{control_msg:"canceled_by_user"}	The H145 is alerting you that the user has selected another mission and you are no longer active. The connection will disconnect after this message
{remote_notify: "tag_name", params:[QUERY1, QUERY2, ...]}	Sent from the active H145 mission. This is data that you would like to be advised about. Remote_notify can be used within objectives or configured in a background thread to provide notifications for specific

	conditions and data.

Commands sent from the server to H145

{load_mission: MISSION_DESCRIPTOR}	Request the H145 to clear the current mission and then load your new mission immediately.
{exec_commands: [COMMAND1, COMMAND2, …]	Request the H145 to execute a free-standing command list. This list executes in parallel with the current objective and all background threads.

Livery Author Info

This section is for those who make aircraft paints (liveries). Many liveries are available already at https://flightsim.to/c/liveries/airbus-h145/. Please do share your liveries with the community.

Paint Kit

Download Official Paint Kit (Version 6)

- You may also find these community resources helpful:
- Livery starter templates: https://flightsim.to/file/24614/h145-livery-templates-for-creators
  Getting started with liveries for MSFS: https://www.youtube.com/watch?v=3atVWEEITQ0

Selecting Variant

Your livery aircraft.cfg base_container should point to

Luxury (Base Pack)	hpg-airbus-h145
Civilian (Base Pack)	hpg-airbus-h145-civ
Military (Base Pack)	hpg-airbus-h145-mil
HEMS (Action Pack)	hpg-airbus-h145-ems
Firefighter (Action Pack)	hpg-airbus-h145-fire
Offshore (Action Pack)	hpg-airbus-h145-offshore
Civilian Cargo (Action Pack)	hpg-airbus-h145-civcargo
Military Cargo (Action Pack)	hpg-airbus-h145-milcargo

Texture.cfg (Luxury)

```
[fltsim]
fallback.1=..\..\hpg-airbus-h145\texture
```

Texture.cfg (Civilian)

```
[fltsim]
fallback.1=..\..\hpg-airbus-h145-civ\texture
fallback.2=..\..\hpg-airbus-h145\texture
```

Texture.cfg (Military)

```
[fltsim]
fallback.1=..\..\hpg-airbus-h145-mil\texture
fallback.2=..\..\hpg-airbus-h145-civ\texture
fallback.3=..\..\hpg-airbus-h145\texture
```

Texture.cfg (HEMS)

```
[fltsim]
fallback.1=..\..\hpg-airbus-h145-ems\texture
fallback.2=..\..\hpg-airbus-h145-mil\texture
fallback.3=..\..\hpg-airbus-h145-civ\texture
fallback.4=..\..\hpg-airbus-h145\texture
```

Texture.cfg (Firefighter)

```
[fltsim]
fallback.1=..\..\hpg-airbus-h145-fire\texture
fallback.2=..\..\hpg-airbus-h145-ems\texture
fallback.3=..\..\hpg-airbus-h145-mil\texture
fallback.4=..\..\hpg-airbus-h145-civ\texture
fallback.5=..\..\hpg-airbus-h145\texture
```

Texture.cfg (Offshore)

```
[fltsim]
fallback.1=..\..\hpg-airbus-h145-ems\texture
fallback.2=..\..\hpg-airbus-h145-civ\texture
fallback.3=..\..\hpg-airbus-h145\texture
fallback.4=..\..\hpg-airbus-h145-mil\texture
fallback.5=..\..\hpg-airbus-h145-fire\texture
fallback.6=..\..\hpg-airbus-h145-civcargo\texture
```

Texture.cfg (Civil Cargo)

```
[fltsim]
fallback.1=..\..\hpg-airbus-h145-civ\texture
fallback.2=..\..\hpg-airbus-h145\texture
fallback.3=..\..\hpg-airbus-h145-offshore\texture
fallback.4=..\..\hpg-airbus-h145-fire\texture
fallback.5=..\..\hpg-airbus-h145-ems\texture
fallback.6=..\..\hpg-airbus-h145-civcargo\texture
```

Texture.cfg (Military Cargo)

```
[fltsim]
fallback.1=..\..\hpg-airbus-h145-mil\texture
fallback.2=..\..\hpg-airbus-h145-civcargo\texture
fallback.3=..\..\hpg-airbus-h145-fire\texture
fallback.4=..\..\hpg-airbus-h145-civ\texture
fallback.5=..\..\hpg-airbus-h145\texture
```

Controlling External Parts

In order to control the parts for your livery, you must include a new file within your livery package named `Livery\`**<Title>**.json. Change only **<Title>** to match your aircraft. The **<Title>** comes from your aircraft.cfg, fltsim.0 title=**<Title>** line. For example the built-in package uses a configuration for the livery matching the title **Airbus H145 Civilian Livery 2**.

Sample Configuration

```
{
"Commands": [
{"Name":"H145_WSPS_Top", "Value":1},
{"Name":"H145_WSPS_Bottom", "Value":1},
{"Name":"H145_WSPS_Skids", "Value":1},
{"Name":"H145_SkidSettlingPreventers", "Value":1},
{"Name":"H145_Radome", "Value":1}
],
"CrewTitle": "Airbus H145 Example Crew"
}
```

NOTE: Invalid JSON will be rejected. Use JSON Validator to check your file

Command	Values	Notes

`H145_WSPS_Top`
`H145_WSPS_Bottom`
`H145_WSPS_Skids`
	0 or 1	Wire Strike Protection system options.
		Bottom WSPS not available on Military variant.

`H145_SkidSettlingPreventers`	0 or 1	Skid-settling-preventers presence. Not compatible with skid snow skis.
`H145_SkidStrutCaps`	0 or 1	Skid struct covers near cockpit door
`H145_SECOND_LANDING_LIGHT`	0 or 1	Add a second fixed LDG light
`H145_ROTOR_BLUR_STRIPES`	0 or 1	Turn on rotor visuals with a split top and bottom texture, better for rotor designs where the top has stripes but the bottom is black.
`H145_Radome`	0: none 1: small 2: large	Weather Radar and Radome presence. **Large radome on HEMS, Firefighter, Offshore, Civil Cargo variants only.**
`H145_RocketPods`	0 or 1	External Weapon Pods installed **Military variant only.**
`H145_WeaponSight`	0 or 1	Cockpit Alignment optics installed **Military variant only.**

H145_CHIN_WINDOW_PLATES	0 or 1	Chin window plates installed instead ofglass. **Civil, HEMS, Firefighter, Offshore, Civil Cargo variants only.**
H145_HOIST	0 or 1	Hoist installed **HEMS, Firefighter, Offshore variants only.**
H145_ELT	0 or 1	ADELT installed **HEMS, Firefighter, Offshore, Civil Cargo variants only.**
H145_SNOWSKI	0 or 1	Snow Skis installed. Not compatible with floats Not compatible with skid settling preventers **HEMS, Firefighter, Civil Cargo variants only.**
H145_HISL	0 or 1	High Intensity Search Light installed **HEMS, Firefighter, Offshore, Civil Cargo variants only.**
H145_FLOATS	0 or 1	Emergency Floats installed **HEMS, Firefighter, Civil Cargo variant only.**
H145_LONG_SKID	0 or 1	Long Skids installed (implicit with floats). **HEMS, Firefighter, Civil Cargo variant only.**
H145_CARGO_HOOK	0 or 1	Cargo Hook installed. **Firefighter, Offshore, Civil Cargo, Military Cargo variants only.**

Configuring external crew

To set the crew models which appear detached from the helicopter, you can use the CrewTitle token to select any of the default crews that come with H145. **Action Pack only.**

Title	Description
Airbus H145 ADAC Crew	ADAC style
Airbus H145 DRF Crew	DRF style
Airbus H145 CMH Crew	Generic red style
Airbus H145 Norsk Luftambulanse Crew Airbus H145 HeliOtago Crew Airbus H145 Bundeswehr Crew	Generic orange style
Airbus H145 CAL FIRE Crew	CAL FIRE style

Custom external crew

If you paint the crew models you must include a second aircraft.cfg which points to the crew base container. The second aircraft.cfg is based on hpg-airbus-h145-crew and should have a texture.cfg which points to your aircraft main livery texture folder. There is no need to duplicate any textures, you will simply add one more aircraft.cfg and texture.cfg to your livery package. The CrewTitle should then be set to your crew livery name, which is recommended to be in the form of `YourLiveryTitle Crew`

Setting Tablet Wallpaper

Include a **wallpaper.jpg** file in your package: `html_ui\Livery\<Title>\wallpaper.jpg`

Adding documents to the Documents app

Your livery may add documents to the Documents app. They will be merged in with the documents provided by the system and the user.

1. See the Documents app section for how to generate user documents.
2. Move everything (including Index.json) in: `hpg-airbus-h145-userdocs\html_ui\HPGH145-User`**\Documents**
   To: `<your livery>\html_ui\HPGH145-User`**\LiveryDocuments\<livery title>**

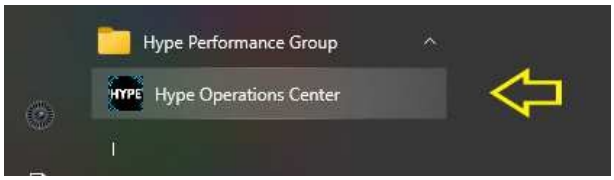
Download and Install H145

If you have installed MSFS 2024, please read the chapter Copy and Install below.
If you are performing a manual installation, please continue reading the chapter "Manual Installation."

Download & Install

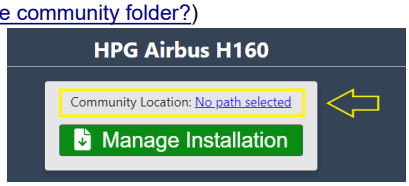
Install either H145 or H160 using this guide.

1. Download and install Hype Operations Center.

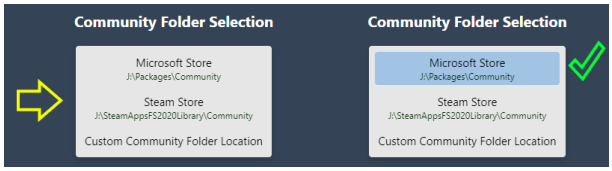


2. Open Hype Operations Center from your Start menu.
3. Visit the product page (either H145 or H160) for the aircraft you'd like to install.

4. Click to select the Community path. (see How do I find the community folder?)

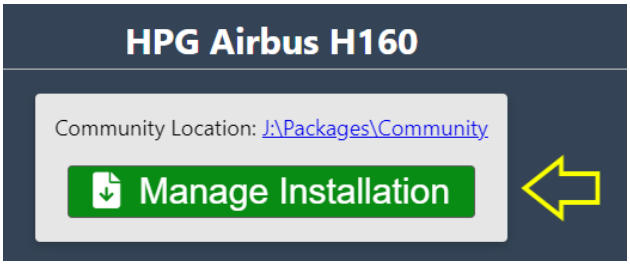


5. Select your Community location. The item must show as highlighted in blue. You may select a custom path for use with Addon Linker, or if the automatic detection is not correct.



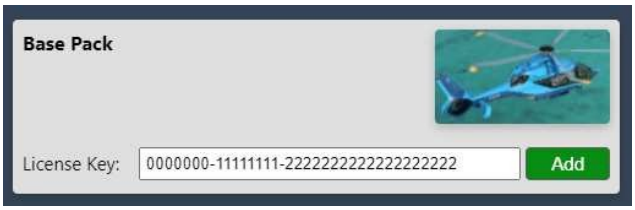
6. Select your aircraft from the side menu again.

7. Confirm the Community location is correct, and then click Manage Installation.

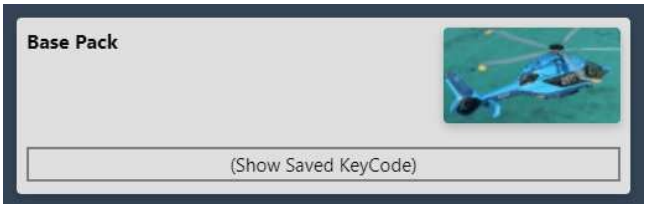


8. Enter your Base Pack license key. You will have been emailed this from Hype Performance Group Downloads. Help me find my details!

9. Copy & Paste your license key into the box. The green ADD button will be available only when the correct length key is entered. Check for extra spaces before or after the key, if you have trouble.

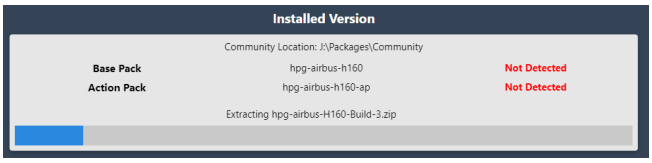


10. The key has been entered successfully.

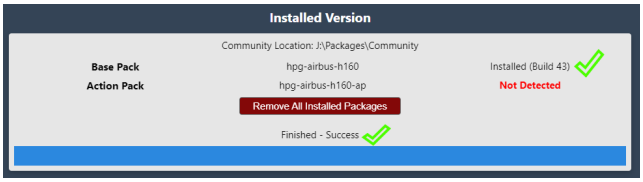


11. If you have the Action Pack too, enter the License Key for the Action Pack before you go to the next step
12. Select a version for installation. Usually the top-most version is best.

13. Wait for download and installation.



14. Check for installation success. You are ready to fly.



15. Activation for MSFS 2024. Workaround for SU2

- Close MSFS 2024
- Download MSFSLayoutgenerator.exe from GitHub: https://github.com/HughesMDflyer4/MSFSLayoutGenerator/releases/download/v1.1.1/MSFSLayoutGenerator.exe
- Locate your Community folder
- Drag the `layout.json` file from the `hpg-airbus-h160` directory onto the downloaded `.exe` file

The 4 most common mistakes we see here are:

1 - Duplicate Community folders and paths not correct
2 - Opening or running MSFSLayoutGenerator (DON'T!)
3 - Trying to modify .json files while FS24 is running. Close first, then restart the sim after mods.
4 - Copying or moving the layout.json files before dragging them over LoG. They must be grabbed from their home location in the HPG folder structure so LoG can do its voodoo magic. If LoG senses the .json files come from another location, it will do it's comparative analysis from that location and will fail in its function

Copy & Install

If you have installed the H160 on MSFS 2020 you can copy the H160 folder to MSFS 2024. Copy the following folders (as applicable):

- hpg-airbus-h160
- hpg-airbus-h160-usersetup
- hpg-hatws-data (download here)
- pms50-instrument-gtn750 (download here)

Manual installation with files from Discord

First, try the normal installation. If that does not work, even though it was performed correctly, you can perform the manual installation. Please close FlightSimulator first. To install the H145 and, if applicable, the Action Pack, follow these steps:

- Download the necessary file(s) from Google Drive (H145 BasePack, H145 ActionPack)
- Unzip the BasePack or both ZIP files into your Community folder

Enter your license keys in the license files (please delete the number entered there).

- Community\hpg-airbus-h145\html_ui\HPGH145-System\product.json (**BasePack**)
- Community\hpg-airbus-h145-ap\html_ui\HPGH145-System\product-ap.json (**ActionPack**)

Start FlightSimulator.

How to Install test build

To install a test version join the H160/H145-discussion our Discord server, go to the pinned messages and download the last Build. Manually unpack the zip-file and copy the files to Community. If you do it correctly then you will be asked to overwrite the files and you must answer yes. HOC will report the new build once installed. If you have the Action Pack, you must install the latest build and the Action Pack for that build!

How to Install HTAWS Database

To install the local HTAWS Database join the H160/H145-discussion our Discord server, go to the pinned messges and download the hpg-htaws-data.zip or use this link https://drive.google.com/file/d/1eMd6cjDGPVaIm4CBajPVXmtnBG-L4k1z/view?usp=sharing Copy the folder included in the .zip file to your Community folder. You need to do this once for H160 and H1145.

How to Install Action Pack

To install Action Pack, follow these steps:

1. Add the Action Pack license key (use steps 8-10 from the procedure above)
2. After the Action Pack key is saved, select **Remove all installed packages**
3. Select **Install** on a version from the available versions list

Now Action Pack will be downloaded and installed along with a matched version of the Base Pack

After Installation

- Upgrading to a new version does not require uninstall, however rolling back to an earlier version does.
- To use an older version, click `Remove all packages` and then `Install`. You may need to reveal all versions by selecting `Show Hidden Versions`

Download Cache

ZIP files are cached to %appdata%\Hype Aircraft\Downloads. You may clear this directory as needed and the large files will be re-downloaded, however this may be slow. You may also junction this location to another drive if needed.

Products are composed of a main package (large) and an update package (much smaller). Keeping the main package will allow you to avoid any extra download time when changing builds (specifically rolling back or reinstalling).

Use of Addon Linker

Use of the third party program Addon Linker is supported: you should place all of your HPG addons into one folder somewhere, and then point Hype Operations Center to that folder as if it was your Community location. Then link in your content as normally in addon linker.

Note that Hype Operations Center assumes it can find content like mission packs by navigating to other folders in your selected Community Location. For this reason you should place HPG addons, mission packs and object packs into one combined location so that Hype Operations Center sees it as a 'normal Community folder'.

Activation trouble

Activation is on Microsoft Azure and highly reliable. If you have trouble activating in the cockpit, check these items (the most likely listed first):

- Check that **Online Functionality** is enabled in MSFS DATA Settings
- Ensure that your PC Date, Time and Timezone are accurate. Go into PC Settings to update your PC time. **Set time automatically** is highly recommended.

- Check your firewall. You must be able to access This URL(opens new window). You should see **Not Authorized**.
- Reinstall the aircraft to ensure that you don't have a corrupt copy installed.
- Remove all other addons from your Community folder
- Remove addons from exe.xml

Troubleshooting

Microsoft Teams or other app won't install

The Squirrel installer has a bug where it can confuse itself with stale data. You may see Microsoft Teams by Hype Performance Group.

Resolution:

1. Open `%LocalAppData%\SquirrelTemp` in the File Explorer address bar.
2. Delete SquirrleTemp folder.
3. Attempt installation again (of Hype Operations Center or another Squirrel installer).

Known Issues

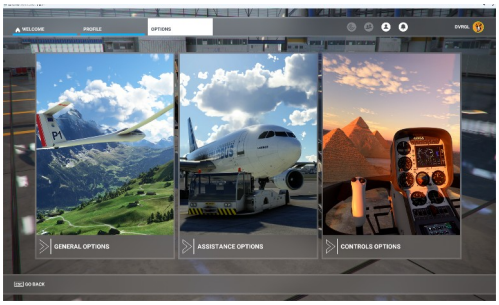
Issue	Workaround
Error: path is not absolute	Community Location was not properly set, re-select the Community Location and try again
Error: end of central directory index	Visit Settings -> Download Cache Location. Delete the small zip file (13kb or so) and try again

We are working to remove known issues and above will all be fixed in a subsequent update.

How are helicopter buttons configured?

•

With the MSFS you can easily assign buttons to functions. As with other vehicles, this is done in the Flight Simulator options. However, this can be somewhat “difficult” if you have to search for the correct functions or if no or unknown functions are used for individual buttons. The HYPE helicopters have a function for each button that can be assigned as required, even if a corresponding function is missing in the MSFS. In general, this is done by using unused functions of the MSFS using the Hype Operations Center. The assignments of pre-assigned functions can also be found here. The procedure is identical for the H145 and the H160.
In general, you should think about the functions you want to have on the sticks and their buttons on the controllers beforehand. Which functions do I need? How easy is it to reach the buttons/switches/...? It is also helpful to record the assignment on a picture and place it in the user docs. You can have a quick look here during the flight.



Normal assignment

Known functions are assigned to individual buttons in the MSFS options. You probably know the procedure well by now, but here it is again for the sake of completeness.

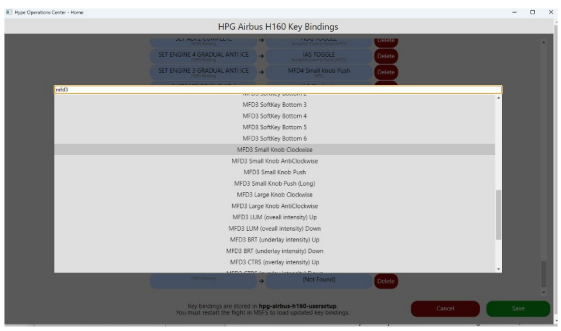
The whole thing is made even easier by the fact that you only have to press the button when selecting.
So select the desired function in the MSFS, click in the assignment field and press the desired button when the “Select input” function is selected.

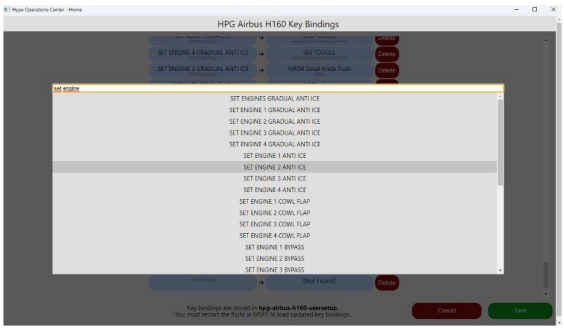
Assignment with HOC

•If the desired function of the helicopter is not included in the functions of the MSFS, the Hype Operations Center is required. Here, the required function is assigned to an unused function of the MSFS and can then be assigned to a button as shown above.

In the helicopter, you will find the Keybindings item. All standard assignments are already defined in the upper area. For your own function assignments, click on “Add Binding”.

The MSFS functions can be found on the left and the helicopter functions on the right.

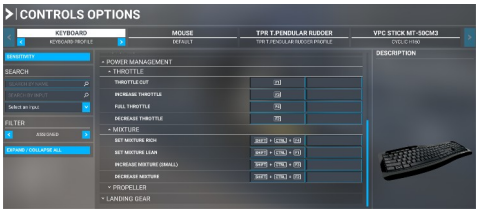


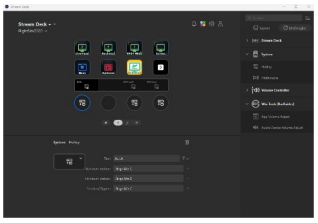


Simply select the function, save it and the function can be assigned to a button in the MSFS. Note, however, that the flight must be restarted for a new assignment!

How do I assign a stream deck?

In general, buttons or controls (Stream Deck +) are assigned in exactly the same way as normal buttons. However, there is an additional step with the Stream Deck SW.





Here again, the desired helicopter function is first assigned to an MSFS function. Then assign the function to a free key combination in the MSFS and finally assign it to the desired key or, as in the example here, to a knob in the Stream Deck software.

SDK H:Events

Home Cockpit SDK

See `hpg-airbus-h145\html_ui\HPGH145-System\H145_Keys.txt` for a full listing of events for your build.
H:Events or Html Events may be used with tools like FSUIPC and SPAD.NEXT.

Overhead Panel

Name	Event
Bus Tie 1 ON	H:H145_SDK_OH_BUSTIE_1_ON
Bus Tie 1 OFF	H:H145_SDK_OH_BUSTIE_1_OFF
Bus Tie 2 ON	H:H145_SDK_OH_BUSTIE_2_ON
Bus Tie 2 OFF	H:H145_SDK_OH_BUSTIE_2_OFF
Generator 1 ON	H:H145_SDK_OH_GEN_1_ON
Generator 1 OFF	H:H145_SDK_OH_GEN_1_OFF
Generator 2 ON	H:H145_SDK_OH_GEN_2_ON
Generator 2 OFF	H:H145_SDK_OH_GEN_2_OFF Emergency Shed Bus
Generator 2 ON	H:H145_SDK_OH_EMER_SHED_BUS_ON
Emergency Shed Bus OFF	H:H145_SDK_OH_EMER_SHED_BUS_OFF
Master Battery OFF	H:H145_SDK_OH_BAT_MASTER_OFF
Master Battery ON	H:H145_SDK_OH_BAT_MASTER_ON
Master Battery ENGAGE	H:H145_SDK_OH_BAT_MASTER_ENGAGE
Master Battery UP	H:H145_SDK_OH_BAT_MASTER_UP
Master Battery DOWN	H:H145_SDK_OH_BAT_MASTER_DOWN
HYD System 1 TEST	H:H145_SDK_OH_HYD_TEST_SYS1
HYD Test OFF	H:H145_SDK_OH_HYD_TEST_OFF
HYD System 2 TEST	H:H145_SDK_OH_HYD_TEST_SYS2
Engine 1 Fire Test OFF	H:H145_SDK_OH_FIRE_ENG1_TEST_OFF
Engine 1 Fire Test EXT	H:H145_SDK_OH_FIRE_ENG1_TEST_EXT Engine 1 Fire Test
EXT+WARN	H:H145_SDK_OH_FIRE_ENG1_TEST_EXT_WARN
Engine 2 Fire Test OFF	H:H145_SDK_OH_FIRE_ENG2_TEST_OFF
Engine 2 Fire Test EXT	H:H145_SDK_OH_FIRE_ENG2_TEST_EXT Engine 2 Fire Test
EXT+WARN	H:H145_SDK_OH_FIRE_ENG2_TEST_EXT_WARN
TEST PREFLIGHT	H:H145_SDK_OH_LAMP_TEST_PREFLIGHT
TEST OFF	H:H145_SDK_OH_LAMP_TEST_OFF
TEST LAMP	H:H145_SDK_OH_LAMP_TEST_LAMP
DC Receptacles OFF	H:H145_SDK_OH_DC_RECEPT_OFF
DC Receptacles ON	H:H145_SDK_OH_DC_RECEPT_ON
Standby Battery ON	H:H145_SDK_OH_STANDBY_BATTERY_ON
Standby Battery OFF	H:H145_SDK_OH_STANDBY_BATTERY_OFF Standby Battery
TOGGLE	H:H145_SDK_OH_STANDBY_BATTERY_TOGGLE
Avionics Bus 1 ON	H:H145_SDK_OH_AVIONICS_1_ON
Avionics Bus 1 OFF	H:H145_SDK_OH_AVIONICS_1_OFF
Avionics Bus 1 TOGGLE	H:H145_SDK_OH_AVIONICS_1_TOGGLE
Avionics Bus 2 ON	H:H145_SDK_OH_AVIONICS_2_ON
Avionics Bus 2 OFF	H:H145_SDK_OH_AVIONICS_2_OFF
Avionics Bus 2 TOGGLE	H:H145_SDK_OH_AVIONICS_2_TOGGLE
Emergency Floats OFF	H:H145_SDK_OH_EMER_FLOATS_OFF
Emergency Floats ARM	H:H145_SDK_OH_EMER_FLOATS_ARM
Emergency Floats TEST	H:H145_SDK_OH_EMER_FLOATS_TEST
Fuzz Burner OFF	H:H145_SDK_OH_FUZZ_CHIP_BURNER_OFF
Fuzz Burner ON	H:H145_SDK_OH_FUZZ_CHIP_BURNER_ON
LAVCS OFF	H:H145_SDK_OH_LAVCS_OFF
LAVCS PIL	H:H145_SDK_OH_LAVCS_PIL
LAVCS PAX	H:H145_SDK_OH_LAVCS_PAX
Windshield Wiper OFF	H:H145_SDK_OH_WINDSHIELD_WIPER_OFF
Windshield Wiper SLOW	H:H145_SDK_OH_WINDSHIELD_WIPER_SLOW
Windshield Wiper FAST	H:H145_SDK_OH_WINDSHIELD_WIPER_FAST
Air Conditioning OFF	H:H145_SDK_OH_AIR_CONDITIONING_OFF
Air Conditioning ON	H:H145_SDK_OH_AIR_CONDITIONING_ON
Cockpit Vent OFF	H:H145_SDK_OH_COCKPIT_VENT_OFF
Cockpit Vent ON	H:H145_SDK_OH_COCKPIT_VENT_ON
Pitot Heater 1 ON	H:H145_SDK_OH_PITOT_1_ON
Pitot Heater 1 OFF	H:H145_SDK_OH_PITOT_1_OFF
Pitot Heater 1 TOGGLE	H:H145_SDK_OH_PITOT_1_TOGGLE
Pitot Heater 2 ON	H:H145_SDK_OH_PITOT_2_ON
Pitot Heater 2 OFF	H:H145_SDK_OH_PITOT_2_OFF
Pitot Heater 2 TOGGLE	H:H145_SDK_OH_PITOT_2_TOGGLE
IBF 1 CLOSED	H:H145_SDK_OH_IBF_1_CLOSED
IBF 1 OPEN	H:H145_SDK_OH_IBF_1_OPEN
IBF 1 NORMAL	H:H145_SDK_OH_IBF_1_NORMAL

IBF 2 CLOSED	H:H145_SDK_OH_IBF_2_CLOSED
IBF 2 OPEN	H:H145_SDK_OH_IBF_2_OPEN
IBF 2 NORMAL	H:H145_SDK_OH_IBF_2_NORMAL
IBF RECALL OFF	H:H145_SDK_OH_IBF_RECALL_OFF
IBF RECALL ON	H:H145_SDK_OH_IBF_RECALL_ON
Fuel Engine 1 Prime OFF	H:H145_SDK_OH_FUEL_ENG1_PRIME_OFF
Fuel Engine 1 Prime ON	H:H145_SDK_OH_FUEL_ENG1_PRIME_ON
Fuel Engine 2 Prime OFF	H:H145_SDK_OH_FUEL_ENG2_PRIME_OFF
Fuel Engine 2 Prime ON	H:H145_SDK_OH_FUEL_ENG2_PRIME_ON
Fuel Transfer Forward OFF	H:H145_SDK_OH_FUEL_TRANSFER_FWD_OFF
Fuel Transfer Forward ON	H:H145_SDK_OH_FUEL_TRANSFER_FWD_ON
Fuel Transfer Aft OFF	H:H145_SDK_OH_FUEL_TRANSFER_AFT_OFF
Fuel Transfer Aft ON	H:H145_SDK_OH_FUEL_TRANSFER_AFT_ON
ACAS MUTE	H:H145_SDK_OH_AUDIO_ACAS_MUTE
ACAS NORMAL	H:H145_SDK_OH_AUDIO_ACAS_NORMAL
ACAS TEST	H:H145_SDK_OH_AUDIO_ACAS_TEST
HTAWS MUTE	H:H145_SDK_OH_AUDIO_HTAWS_MUTE
HTAWS MUTE-FOR-5-MINS	H:H145_SDK_OH_AUDIO_HTAWS_MUTE_5MIN
HTAWS NORMAL	H:H145_SDK_OH_AUDIO_HTAWS_NORMAL
HTAWS STANDBY	H:H145_SDK_OH_AUDIO_HTAWS_STANDBY
Int Lights Cargo/Pax OFF	H:H145_SDK_OH_INT_LIGHT_CARGO_PAX_OFF
Int Lights Cargo/Pax PAX	H:H145_SDK_OH_INT_LIGHT_CARGO_PAX_PAX
Int Lights Cargo/Pax BOTH	H:H145_SDK_OH_INT_LIGHT_CARGO_PAX_ON
Int Emergency Exits OFF	H:H145_SDK_OH_INT_LIGHT_EMERGENCY_EXITS_OFF
Int Emergency Exits ARM	H:H145_SDK_OH_INT_LIGHT_EMERGENCY_EXITS_ARM
Int Emergency Exits ON	H:H145_SDK_OH_INT_LIGHT_EMERGENCY_EXITS_ON
Int Panel Lights DAY	H:H145_SDK_OH_INT_LIGHT_INSTRUMENT_PANEL_DAY
Int Panel Lights NIGHT	H:H145_SDK_OH_INT_LIGHT_INSTRUMENT_PANEL_NIGHT
Int Panel Lights NVG	H:H145_SDK_OH_INT_LIGHT_INSTRUMENT_PANEL_NVG
Ext Lights HISL ON	H:H145_SDK_OH_EXT_LIGHT_HISL_ON
Ext Lights HISL OFF	H:H145_SDK_OH_EXT_LIGHT_HISL_OFF
Ext Lights HISL TOGGLE	H:H145_SDK_OH_EXT_LIGHT_HISL_TOGGLE
Cockpit Vent INCREASE	H:H145_SDK_OH_COCKPIT_VENT_POT_INC
Cockpit Vent DECREASE	H:H145_SDK_OH_COCKPIT_VENT_POT_DEC
Bleed Heading INCREASE	H:H145_SDK_OH_BLEED_HEATING_POT_INC
Bleed Heading DECREASE	H:H145_SDK_OH_BLEED_HEATING_POT_DEC
Panel Lights INCREASE	H:H145_SDK_OH_INT_LIGHT_INSTRUMENT_PANEL_KNOB_INC
Panel Lights DECREASE	H:H145_SDK_OH_INT_LIGHT_INSTRUMENT_PANEL_KNOB_DEC
Front Light TOGGLE	H:H145_SDK_OH_COCKPIT_LIGHT_FRONT_TOGGLE
Front Light ON	H:H145_SDK_OH_COCKPIT_LIGHT_FRONT_ON
Front Light OFF	H:H145_SDK_OH_COCKPIT_LIGHT_FRONT_OFF
Rear Light TOGGLE	H:H145_SDK_OH_COCKPIT_LIGHT_REAR_TOGGLE
Rear Light ON	H:H145_SDK_OH_COCKPIT_LIGHT_REAR_ON
Rear Light OFF	H:H145_SDK_OH_COCKPIT_LIGHT_REAR_OFF

Engine Control Panel (ECP)

Name	Event
Toggle both engines FLIGHT/IDLE	H:H145_SDK_ECP_FADEC_DUAL_TOGGLE
Main 1 FLIGHT	H:H145_SDK_ECP_MAIN_1_FLIGHT
Main 1 IDLE	H:H145_SDK_ECP_MAIN_1_IDLE
Main 1 OFF	H:H145_SDK_ECP_MAIN_1_OFF
Main 1 UP	H:H145_SDK_ECP_MAIN_1_UP
Main 1 DOWN	H:H145_SDK_ECP_MAIN_1_DOWN
Main 2 FLIGHT	H:H145_SDK_ECP_MAIN_2_FLIGHT
Main 2 IDLE	H:H145_SDK_ECP_MAIN_2_IDLE
Main 2 OFF	H:H145_SDK_ECP_MAIN_2_OFF
Main 2 DOWN	H:H145_SDK_ECP_MAIN_2_DOWN
Main 2 UP	H:H145_SDK_ECP_MAIN_2_UP
Main 1 Latch OFF	H:H145_SDK_ECP_MAIN_LATCH_1_OFF
Main 1 Latch ON	H:H145_SDK_ECP_MAIN_LATCH_1_ON
Main 2 Latch OFF	H:H145_SDK_ECP_MAIN_LATCH_2_OFF
Main 2 Latch ON	H:H145_SDK_ECP_MAIN_LATCH_2_ON
FADEC EMER 1 OFF	H:H145_SDK_ECP_FADEC_EMER_1_OFF
FADEC EMER 1 ON	H:H145_SDK_ECP_FADEC_EMER_1_ON
FADEC EMER 1 Latch CLOSE	H:H145_SDK_ECP_FADEC_EMER_1_CAP_CLOSE
FADEC EMER 1 Latch OPEN	H:H145_SDK_ECP_FADEC_EMER_1_CAP_OPEN
FADEC EMER 2 OFF	H:H145_SDK_ECP_FADEC_EMER_2_OFF
FADEC EMER 2 ON	H:H145_SDK_ECP_FADEC_EMER_2_ON

FADEC EMER 2 Latch CLOSE	H:H145_SDK_ECP_FADEC_EMER_2_CAP_CLOSE
FADEC EMER 2 Latch OPEN	H:H145_SDK_ECP_FADEC_EMER_2_CAP_OPEN

Autopilot Control Panel (APCP)

Name	Event
A.TRIM TOGGLE	H:H145_SDK_APCP_ATRIM_TOGGLE
A.TRIM ON	H:H145_SDK_APCP_ATRIM_ON
A.TRIM OFF	H:H145_SDK_APCP_ATRIM_OFF
AP1 TOGGLE	H:H145_SDK_APCP_AP1_TOGGLE
AP1 ON	H:H145_SDK_APCP_AP1_ON
AP1 OFF	H:H145_SDK_APCP_AP1_OFF
AP2 TOGGLE	H:H145_SDK_APCP_AP2_TOGGLE
AP2 ON	H:H145_SDK_APCP_AP2_ON
AP2 OFF	H:H145_SDK_APCP_AP2_OFF
BKUP TOGGLE	H:H145_SDK_APCP_BKUP_TOGGLE
BKUP ON	H:H145_SDK_APCP_BKUP_ON
BKUP OFF	H:H145_SDK_APCP_BKUP_OFF
ALT TOGGLE	H:H145_SDK_APCP_ALT_TOGGLE
ALT ON	H:H145_SDK_APCP_ALT_ON
ALT OFF	H:H145_SDK_APCP_ALT_OFF
(VS/FPA HDG/TRK) TOGGLE	H:H145_SDK_APCP_GPSMODE_TOGGLE
(VS/FPA HDG/TRK) VS/HDG	H:H145_SDK_APCP_GPSMODE_TRAD
(VS/FPA HDG/TRK) TRK/FPA	H:H145_SDK_APCP_GPSMODE_GPS
ALT.A TOGGLE	H:H145_SDK_APCP_ALTA_TOGGLE
ALT.A ON	H:H145_SDK_APCP_ALTA_ON
ALT.A OFF	H:H145_SDK_APCP_ALTA_OFF
ALT.A Clockwise	H:H145_SDK_APCP_ALTA_Clockwise
ALT.A AntiClockwise	H:H145_SDK_APCP_ALTA_AntiClockwise
CR.HT TOGGLE	H:H145_SDK_APCP_CRHT_TOGGLE
CR.HT ON	H:H145_SDK_APCP_CRHT_ON
CR.HT OFF	H:H145_SDK_APCP_CRHT_OFF
CR.HT Clockwise	H:H145_SDK_APCP_CRHT_Clockwise
CR.HT AntiClockwise	H:H145_SDK_APCP_CRHT_AntiClockwise
VS TOGGLE	H:H145_SDK_APCP_VS_TOGGLE
VS ON	H:H145_SDK_APCP_VS_ON
VS OFF	H:H145_SDK_APCP_VS_OFF
VS Clockwise	H:H145_SDK_APCP_VS_Clockwise
VS AntiClockwise	H:H145_SDK_APCP_VS_AntiClockwise
HDG TOGGLE	H:H145_SDK_APCP_HDG_TOGGLE
HDG ON	H:H145_SDK_APCP_HDG_ON
HDG OFF	H:H145_SDK_APCP_HDG_OFF
HDG Clockwise	H:H145_SDK_APCP_HDG_Clockwise
HDG AntiClockwise	H:H145_SDK_APCP_HDG_AntiClockwise
IAS TOGGLE	H:H145_SDK_APCP_IAS_TOGGLE
IAS ON	H:H145_SDK_APCP_IAS_ON
IAS OFF	H:H145_SDK_APCP_IAS_OFF
IAS Clockwise	H:H145_SDK_APCP_IAS_Clockwise
IAS AntiClockwise	H:H145_SDK_APCP_IAS_AntiClockwise
Easy AFCS Toggle	H:H145_SDK_AP_AFCS_EASY_TOGGLE
Easy AFCS On	H:H145_SDK_AP_AFCS_EASY_ON
Easy AFCS Off	H:H145_SDK_AP_AFCS_EASY_OFF
Warning Unit (WU) - Engine 1 Fire Shutoff PUSH	H:H145_SDK_WU_ENG1_FIRE_SHUTOFF
Warning Unit (WU) - Engine 2 Fire Shutoff PUSH	H:H145_SDK_WU_ENG2_FIRE_SHUTOFF
Warning Unit (WU) - Engine 1 Fire Extinguish PUSH	H:H145_SDK_WU_ENG1_FIRE_EXTINGUISH
Warning Unit (WU) - Engine 2 Fire Extinguish PUSH	H:H145_SDK_WU_ENG2_FIRE_EXTINGUISH

Cyclic Control

Name	Event
AP/BKUP ON	H:H145_SDK_AP_APBKUPON_UP
AP/BKUP ON (AP1 Only)	H:H145_SDK_AP_APBKUPON_LEFT
AP/BKUP ON (AP2 Only)	H:H145_SDK_AP_APBKUPON_RIGHT
AP/BKUP CUT	H:H145_SDK_AP_APBKUPCUT
AP/UM OFF	H:H145_SDK_AP_UM_OFF
AP/GTC	H:H145_SDK_AP_GTCGTCH
AP/GTC (Direct to GTC.H)(Advanced)	H:H145_SDK_AP_GTCGTCH_HOVER
Cyclic Beep Trim RIGHT	H:H145_SDK_CYCLIC_BEEP_RIGHT
Cyclic Beep Trim LEFT	H:H145_SDK_CYCLIC_BEEP_LEFT
Cyclic Beep Trim UP	H:H145_SDK_CYCLIC_BEEP_UP

Cyclic Beep Trim DOWN	H:H145_SDK_CYCLIC_BEEP_DOWN
Cyclic Beep Trim RESET/Zero(Uncommon)	H:H145_SDK_CYCLIC_BEEP_RESET
Set New Cyclic Center	H:H145_SDK_CYCLIC_FORCE_TRIM_SET_NEW_CENTER
Displace Cyclic Center (Force Trim)	H:H145_SDK_CYCLIC_FORCE_TRIM_DISPLACE_CENTER
Trim Release (HOLD)	H:H145_SDK_CYCLIC_TRIM_RELEASE_HOLD
Trim Release (Latch: Open)	H:H145_SDK_CYCLIC_TRIM_RELEASE_LATCH_PUSH
Trim Release (Latch: Closed)	H:H145_SDK_CYCLIC_TRIM_RELEASE_LATCH_RELEASE
Trim Release (Latch: Toggle)	H:H145_SDK_CYCLIC_TRIM_RELEASE_LATCH_TOGGLE
Message List RESET	H:H145_SDK_MESSAGELIST_RESET

Collective Control

Name	Event
Collective Beep Trim RIGHT	H:H145_SDK_COLLECTIVE_BEEP_RIGHT
Collective Beep Trim LEFT	H:H145_SDK_COLLECTIVE_BEEP_LEFT
Collective Beep Trim UP	H:H145_SDK_COLLECTIVE_BEEP_UP
Collective Beep Trim DOWN	H:H145_SDK_COLLECTIVE_BEEP_DOWN
Collective Beep Trim ATT YAW AUTORESET	H:H145_SDK_COLLECTIVE_YAW_TRIM_AUTO_RESET
Collective Trim Release (HOLD)	H:H145_SDK_COLLECTIVE_TRIM_RELEASE_HOLD
Collective Trim Release (Latch: Open)	H:H145_SDK_COLLECTIVE_TRIM_RELEASE_LATCH_PUSH
Collective Trim Release (Latch: Closed)	H:H145_SDK_COLLECTIVE_TRIM_RELEASE_LATCH_RELEASE
Collective Trim Release (Latch: Toggle)	H:H145_SDK_COLLECTIVE_TRIM_RELEASE_LATCH_TOGGLE
OEI HI/LO (Low)	H:H145_SDK_COLLECTIVE_OEI_HILO_LO
OEI HI/LO (High)	H:H145_SDK_COLLECTIVE_OEI_HILO_HI
OEI HI/LO (Toggle)	H:H145_SDK_COLLECTIVE_OEI_HILO_TOGGLE
Fill Floats	H:H145_SDK_FILL_FLOATS
GA (Go Around)	H:H145_SDK_COLLECTIVE_GA

H145M Weapons

Name	Event
Fire (Primary)	H:H145_SDK_PRIMARY_ACTION_COMMAND
Fire (Secondary)	H:H145_SDK_SECONDARY_ACTION_COMMAND
Installed (Toggle)	H:H145_SDK_EQUIP_WEAPONS_TOGGLE
Installed (On)	H:H145_SDK_EQUIP_WEAPONS_ON
Installed (Off)	H:H145_SDK_EQUIP_WEAPONS_OFF
Pod Left Type (Toggle)	H:H145_SDK_EQUIP_WEAPONS_POD_LEFT_TOGGLE
Pod Left Type (Gun)	H:H145_SDK_EQUIP_WEAPONS_POD_LEFT_GUN
Pod Left Type (Rockets)	H:H145_SDK_EQUIP_WEAPONS_POD_LEFT_ROCKETS
Pod Right Type (Toggle)	H:H145_SDK_EQUIP_WEAPONS_POD_RIGHT_TOGGLE
Pod Right Type (Gun)	H:H145_SDK_EQUIP_WEAPONS_POD_RIGHT_GUN
Pod Right Type (Rockets)	H:H145_SDK_EQUIP_WEAPONS_POD_RIGHT_ROCKETS
Reload Rockets	H:H145_SDK_WEAPON_RELOAD
Cleanup All Rockets	H:H145_SDK_WEAPON_CLEANUP
Master Arm TOGGLE	H:H145_SDK_EQUIP_WEAPONS_MASTER_ARM_TOGGLE
Master Arm OFF (SAFE)	H:H145_SDK_EQUIP_WEAPONS_MASTER_ARM_OFF
Master Arm ON (ARMED)	H:H145_SDK_EQUIP_WEAPONS_MASTER_ARM_ON

Cabin

Name	Event
Cockpit Door Left TOGGLE	H:H145_SDK_DOOR_COCKPIT_L_TOGGLE
Cockpit Door Left OPEN	H:H145_SDK_DOOR_COCKPIT_L_OPEN
Cockpit Door Left CLOSE	H:H145_SDK_DOOR_COCKPIT_L_CLOSE
Cockpit Door Right TOGGLE	H:H145_SDK_DOOR_COCKPIT_R_TOGGLE
Cockpit Door Right OPEN	H:H145_SDK_DOOR_COCKPIT_R_OPEN
Cockpit Door Right CLOSE	H:H145_SDK_DOOR_COCKPIT_R_CLOSE
Pax Door Left TOGGLE	H:H145_SDK_DOOR_PAX_L_TOGGLE
Pax Door Left OPEN	H:H145_SDK_DOOR_PAX_L_OPEN
Pax Door Left CLOSE	H:H145_SDK_DOOR_PAX_L_CLOSE
Pax Door Right TOGGLE	H:H145_SDK_DOOR_PAX_R_TOGGLE
Pax Door Right OPEN	H:H145_SDK_DOOR_PAX_R_OPEN
Pax Door Right CLOSE	H:H145_SDK_DOOR_PAX_R_CLOSE
Cargo Door Left TOGGLE	H:H145_SDK_DOOR_CARGO_L_TOGGLE
Cargo Door Left OPEN	H:H145_SDK_DOOR_CARGO_L_OPEN
Cargo Door Left CLOSE	H:H145_SDK_DOOR_CARGO_L_CLOSE

Cargo Door Right TOGGLE	H:H145_SDK_DOOR_CARGO_R_TOGGLE
Cargo Door Right OPEN	H:H145_SDK_DOOR_CARGO_R_OPEN
Cargo Door Right CLOSE	H:H145_SDK_DOOR_CARGO_R_CLOSE
Cockpit And Pax Doors TOGGLE	H:H145_SDK_DOORS_TOGGLE
Cockpit And Pax Doors INSTALL ALL	H:H145_SDK_DOORS_INSTALLED
Cockpit And Pax Doors REMOVE ALL	H:H145_SDK_DOORS_REMOVED
Pilot TOGGLE	H:H145_SDK_PILOT_CAPT_TOGGLE
Pilot ON	H:H145_SDK_PILOT_CAPT_ON
Pilot OFF	H:H145_SDK_PILOT_CAPT_OFF
Copilot TOGGLE	H:H145_SDK_PILOT_FO_TOGGLE
Copilot ON	H:H145_SDK_PILOT_FO_ON
Copilot OFF	H:H145_SDK_PILOT_FO_OFF
HEMS Stretcher Toggle	H:H145_SDK_HEMS_STRETCHER_TOGGLE
HEMS Stretcher Eject	H:H145_SDK_HEMS_STRETCHER_EJECT
HEMS Stretcher Retract	H:H145_SDK_HEMS_STRETCHER_RETRACT
HEMS Stretcher Removed	H:H145_SDK_HEMS_STRETCHER_REMOVED
HEMS Stretcher Present without patient	H:H145_SDK_HEMS_STRETCHER_NOPATIENT
HEMS Stretcher Present with patient	H:H145_SDK_HEMS_STRETCHER_PATIENT
Pax 1 Toggle	H:H145_SDK_PAX_1_TOGGLE
Pax 1 On	H:H145_SDK_PAX_1_ON
Pax 1 Off	H:H145_SDK_PAX_1_OFF
Pax 2 Toggle	H:H145_SDK_PAX_2_TOGGLE
Pax 2 On	H:H145_SDK_PAX_2_ON
Pax 2 Off	H:H145_SDK_PAX_2_OFF
Pax 3 Toggle	H:H145_SDK_PAX_3_TOGGLE
Pax 3 On	H:H145_SDK_PAX_3_ON
Pax 3 Off	H:H145_SDK_PAX_3_OFF
Pax 4 Toggle	H:H145_SDK_PAX_4_TOGGLE
Pax 4 On	H:H145_SDK_PAX_4_ON
Pax 4 Off	H:H145_SDK_PAX_4_OFF
Pax 5 Toggle	H:H145_SDK_PAX_5_TOGGLE
Pax 5 On	H:H145_SDK_PAX_5_ON
Pax 5 Off	H:H145_SDK_PAX_5_OFF
Pax 6 Toggle	H:H145_SDK_PAX_6_TOGGLE
Pax 6 On	H:H145_SDK_PAX_6_ON
Pax 6 Off	H:H145_SDK_PAX_6_OFF
Pax 7 Toggle	H:H145_SDK_PAX_7_TOGGLE
Pax 7 On	H:H145_SDK_PAX_7_ON
Pax 7 Off	H:H145_SDK_PAX_7_OFF
Pax 8 Toggle	H:H145_SDK_PAX_8_TOGGLE
Pax 8 On	H:H145_SDK_PAX_8_ON
Pax 8 Off	H:H145_SDK_PAX_8_OFF

•Misc

Name	Event
State Load READY FOR TAKEOFF	H:H145_SDK_MISC_CMD_READYFORTAKEOFF
State Load COLD AND DARK	H:H145_SDK_MISC_CMD_COLDANDDARK
Rotor Brake TOGGLE	H:H145_SDK_ROTOR_BRAKE_TOGGLE
Rotor Brake ON	H:H145_SDK_ROTOR_BRAKE_ON
Rotor Brake OFF	H:H145_SDK_ROTOR_BRAKE_OFF
FMS1 Source TOGGLE	H:H145_SDK_MISC_FMS1_TOGGLE
FMS1 Source ON	H:H145_SDK_MISC_FMS1_ON
FMS1 Source OFF	H:H145_SDK_MISC_FMS1_OFF
FMS2 Source TOGGLE	H:H145_SDK_MISC_FMS2_TOGGLE
FMS2 Source ON	H:H145_SDK_MISC_FMS2_ON
FMS2 Source OFF	H:H145_SDK_MISC_FMS2_OFF
Master Brightness Increase	H:H145_SDK_MASTERBRIGHTNESS_INC
Master Brightness Decrease	H:H145_SDK_MASTERBRIGHTNESS_DEC
Luxury Divider Wall TOGGLE	H:H145_SDK_LUX_DIVIDER_TOGGLE
Luxury Divider Wall UP	H:H145_SDK_LUX_DIVIDER_UP
Luxury Divider Wall DOWN	H:H145_SDK_LUX_DIVIDER_DOWN
TDSSim GTNXi Nav Source UNIT1	H:H145_SDK_MISC_GTN750_TDSSIM_NAVSOURCE_UNIT_1
TDSSim GTNXi Nav Source UNIT2	H:H145_SDK_MISC_GTN750_TDSSIM_NAVSOURCE_UNIT_2
TDSSim GTNXi Nav Source MSFS	H:H145_SDK_MISC_GTN750_TDSSIM_NAVSOURCE_MSFS
TDSSim GTNXi Nav Source NEXT	H:H145_SDK_MISC_GTN750_TDSSIM_NAVSOURCE_NEXT
Marker Beacon Sensitivity HIGH	H:H145_SDK_MISC_MKR_HIGH
Marker Beacon Sensitivity LOW	H:H145_SDK_MISC_MKR_LOW

Hoist

Name	Event
Hoist Mode AUTO	H:H145_SDK_HOIST_CONTROL_MODE_AUTO
Hoist Mode MANUAL	H:H145_SDK_HOIST_CONTROL_MODE_MANUAL
Hoist Manual Control UP	H:H145_SDK_HOIST_CONTROL_MOTOR_UP
Hoist Manual Control STOP	H:H145_SDK_HOIST_CONTROL_MOTOR_STOP
Hoist Manual Control DOWN	H:H145_SDK_HOIST_CONTROL_MOTOR_DOWN
Hoist Manual Control MOMENTARY_UP	H:H145_SDK_HOIST_CONTROL_MOTOR_MOMENTARY_UP
Hoist Manual Control MOMENTARY_DOWN	H:H145_SDK_HOIST_CONTROL_MOTOR_MOMENTARY_DOWN
Hoist Arm STOW	H:H145_SDK_HOIST_ARM_STOW
Hoist Arm DEPLOY	H:H145_SDK_HOIST_ARM_DEPLOY

Center Console WXRCP

Name	Event
Weather Radar Power OFF	H:H145_SDK_WXR_OFF
Weather Radar Power STANDBY	H:H145_SDK_WXR_STBY
Weather Radar Power TEST	H:H145_SDK_WXR_TEST
Weather Radar Power ON	H:H145_SDK_WXR_ON
Weather Radar Power Knob UP	H:H145_SDK_WXR_UP
Weather Radar Power Knob NEXT	H:H145_SDK_WXR_UP_LOOP
Weather Radar Power Knob DOWN	H:H145_SDK_WXR_DOWN
Weather Radar Tilt Knob UP	H:H145_SDK_WXR_TILT_UP
Weather Radar Tilt Knob DOWN	H:H145_SDK_WXR_TILT_DOWN

Search Light

Name	Event
Light TOGGLE	H:H145_SDK_SL_LIGHT_TOGGLE
Light OFF	H:H145_SDK_SL_LIGHT_OFF
Light ON	H:H145_SDK_SL_LIGHT_ON
Steering UP	H:H145_SDK_SL_STEER_UP
Steering DOWN	H:H145_SDK_SL_STEER_DOWN
Steering LEFT	H:H145_SDK_SL_STEER_LEFT
Steering RIGHT	H:H145_SDK_SL_STEER_RIGHT
Steering HOME	H:H145_SDK_SL_STEER_HOME

Landing Light

Name	Event
Light TOGGLE	H:H145_SDK_LDG_LIGHT_TOGGLE
Light OFF	H:H145_SDK_LDG_LIGHT_OFF
Light ON	H:H145_SDK_LDG_LIGHT_ON

Center Console HISLCP

Name	Event
HISL Deploy or Stow TOGGLE	H:H145_SDK_HISL_STOW_TOGGLE
HISL STOW	H:H145_SDK_HISL_STOW
HISL DEPLOY	H:H145_SDK_HISL_DEPLOY
HISL Dim TOGGLE	H:H145_SDK_HISL_DIM_TOGGLE
HISL Dim ON	H:H145_SDK_HISL_DIM_ON
HISL Dim OFF	H:H145_SDK_HISL_DIM_OFF
HISL Lamp TOGGLE	H:H145_SDK_HISL_LAMP_TOGGLE
HISL Lamp ON	H:H145_SDK_HISL_LAMP_ON
HISL Lamp OFF	H:H145_SDK_HISL_LAMP_OFF
Easy HISL TOGGLE	H:H145_SDK_HISL_EASYTOGGLE
Easy HISL OFF	H:H145_SDK_HISL_EASY_OFF
Easy HISL ON	H:H145_SDK_HISL_EASY_ON
Beam Zoom (Wide)	H:H145_SDK_HISL_ZOOM_WIDE
Beam Zoom (Narrow)	H:H145_SDK_HISL_ZOOM_NARROW
Filter ENTER	H:H145_SDK_HISL_FILTER_ENTER
Filter SELECT	H:H145_SDK_HISL_FILTER_SELECT
Directly Select Filter 1	H:H145_SDK_HISL_FILTER_EASYSELECT_1
Directly Select Filter 2	H:H145_SDK_HISL_FILTER_EASYSELECT_2

Directly Select Filter 3	H:H145_SDK_HISL_FILTER_EASYSELECT_3
Directly Select Filter 4	H:H145_SDK_HISL_FILTER_EASYSELECT_4

Tablet

Name	Event
Hinge Open/Close	H:H145_SDK_TABLET_OPENCLOSE
Home (Push)	H:H145_SDK_TABLET_HOME_PUSH
Home (Push Long)	H:H145_SDK_TABLET_HOME_PUSH_LONG
Open Action Center	H:H145_SDK_TABLET_OPEN_ACTIONCENTER
Launch Maps	H:H145_SDK_TABLET_OPENAPP_MAPS
Launch Missions	H:H145_SDK_TABLET_OPENAPP_MISSIONS
Launch Setup	H:H145_SDK_TABLET_OPENAPP_SETUP
Launch Documents	H:H145_SDK_TABLET_OPENAPP_DOCUMENTS
Launch EFBConnect	H:H145_SDK_TABLET_OPENAPP_WEB_EFBCONNECT
Launch Web Browser	H:H145_SDK_TABLET_OPENAPP_WEB
Launch METAR	H:H145_SDK_TABLET_OPENAPP_METAR
Launch LittleNavMap	H:H145_SDK_TABLET_OPENAPP_LITTLENAVMAP
Launch Navigraph Charts	H:H145_SDK_TABLET_OPENAPP_NAVIGRAPH
Launch Flappy Bird	H:H145_SDK_TABLET_OPENAPP_FLAPPYBIRD
Launch Alarms & Clock	H:H145_SDK_TABLET_OPENAPP_CLOCK
Launch Activity Log	H:H145_SDK_TABLET_OPENAPP_ACTIVITYLOG
Launch Direction Finder	H:H145_SDK_TABLET_OPENAPP_DF
Launch Hype Radio	H:H145_SDK_TABLET_OPENAPP_HYPERADIO
Launch Neopad	H:H145_SDK_TABLET_OPENAPP_NEOPAD
Map ZOOM IN	H:H145_SDK_TABLET_MAPSAPP_ZOOM_IN
Map ZOOM OUT	H:H145_SDK_TABLET_MAPSAPP_ZOOM_OUT
Map ZOOM Level 3	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_3
Map ZOOM Level 4	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_4
Map ZOOM Level 5	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_5
Map ZOOM Level 6	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_6
Map ZOOM Level 7	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_7
Map ZOOM Level 8	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_8
Map ZOOM Level 9	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_9
Map ZOOM Level 10	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_10
Map ZOOM Level 11	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_11
Map ZOOM Level 12	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_12
Map ZOOM Level 13	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_13
Map ZOOM Level 14	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_14
Map ZOOM Level 15	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_15
Map ZOOM Level 16	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_16
Map ZOOM Level 17	H:H145_SDK_TABLET_MAPSAPP_ZOOM_SET_17
Map FollowMe TOGGLE	H:H145_SDK_TABLET_MAPSAPP_FOLLOWME_TOGGLE
Map FollowMe ON	H:H145_SDK_TABLET_MAPSAPP_FOLLOWME_ON
Map FollowMe OFF	H:H145_SDK_TABLET_MAPSAPP_FOLLOWME_OFF
Map Orientation TOGGLE	H:H145_SDK_TABLET_MAPSAPP_ORIENTATION_TOGGLE
Map Orientation NorthUP	H:H145_SDK_TABLET_MAPSAPP_ORIENTATION_NORTHUP
Map Orientation HeadingUP	H:H145_SDK_TABLET_MAPSAPP_ORIENTATION_HEADINGUP
Map DB Layer Hospital Helipad ON	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_ON_H_HOSPITAL
Map DB Layer Civil Helipad ON	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_ON_H_CIVIL
Map DB Layer Airport Primary ON	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_ON_AIRPORT
Map DB Layer Airport Secondary ON	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_ON_AIRPORT-NOCODE
Map DB Layer Hospital Helipad OFF	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_OFF_H_HOSPITAL
Map DB Layer Civil Helipad OFF	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_OFF_H_CIVIL
Map DB Layer Airport Primary OFF	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_OFF_AIRPORT
Map DB Layer Airport Secondary OFF	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_OFF_AIRPORT-NOCODE
Map DB Layer Hospital Helipad TOGGLE	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_TOGGLE_H_HOSPITAL
Map DB Layer Civil Helipad TOGGLE	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_TOGGLE_H_CIVIL
Map DB Layer Airport Primary TOGGLE	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_TOGGLE_AIRPORT
Map DB Layer Airport Secondary TOGGLE	H:H145_SDK_TABLET_MAPSAPP_SET_DB_LAYER_TOGGLE_AIRPORT-NOCODE
Mission Command 1 PRESS	H:H145_SDK_MISSION_ACTION_COMMAND_1
Mission Command 2 PRESS	H:H145_SDK_MISSION_ACTION_COMMAND_2
Mission Command 3 PRESS	H:H145_SDK_MISSION_ACTION_COMMAND_3
Mission Command 4 PRESS	H:H145_SDK_MISSION_ACTION_COMMAND_4
Mission Command 5 PRESS	H:H145_SDK_MISSION_ACTION_COMMAND_5
Mission Command 6 PRESS	H:H145_SDK_MISSION_ACTION_COMMAND_6

Hype Radio App

Name	Event
Connect_Reconnect_SyncLocation	H:H145_SDK_HYPERADIO_CONNECT

Volume Down	H:H145_SDK_HYPERADIO_VOLUME_DOWN
Volume Up	H:H145_SDK_HYPERADIO_VOLUME_UP
Stop	H:H145_SDK_HYPERADIO_STOP
Select Previous Station	H:H145_SDK_HYPERADIO_STATION_PREV
Select Next Station	H:H145_SDK_HYPERADIO_STATION_NEXT
Select Station 1	H:H145_SDK_HYPERADIO_STATION_1
Select Station 2	H:H145_SDK_HYPERADIO_STATION_2
Select Station 3	H:H145_SDK_HYPERADIO_STATION_3
Select Station 4	H:H145_SDK_HYPERADIO_STATION_4
Select Station 5	H:H145_SDK_HYPERADIO_STATION_5
Select Station 6	H:H145_SDK_HYPERADIO_STATION_6
Select Station 7	H:H145_SDK_HYPERADIO_STATION_7
Select Station 8	H:H145_SDK_HYPERADIO_STATION_8
Select Station 9	H:H145_SDK_HYPERADIO_STATION_9
Select Station 10	H:H145_SDK_HYPERADIO_STATION_10
Select Station 11	H:H145_SDK_HYPERADIO_STATION_11
Select Station 12	H:H145_SDK_HYPERADIO_STATION_12
Select Station 13	H:H145_SDK_HYPERADIO_STATION_13
Select Station 14	H:H145_SDK_HYPERADIO_STATION_14
Select Station 15	H:H145_SDK_HYPERADIO_STATION_15

Equipment Setup

Name	Event
Radome TOGGLE	H:H145_SDK_EQUIP_RADOME_TOGGLE
Radome 1 ON	H:H145_SDK_EQUIP_RADOME_ON
Radome 2 ON	H:H145_SDK_EQUIP_RADOME_2_ON
Radome OFF	H:H145_SDK_EQUIP_RADOME_OFF
WSPS Top TOGGLE	H:H145_SDK_EQUIP_WSPS_TOP_TOGGLE
WSPS Top ON	H:H145_SDK_EQUIP_WSPS_TOP_ON
WSPS Top OFF	H:H145_SDK_EQUIP_WSPS_TOP_OFF
WSPS Bottom TOGGLE	H:H145_SDK_EQUIP_WSPS_BOTTOM_TOGGLE
WSPS Bottom ON	H:H145_SDK_EQUIP_WSPS_BOTTOM_ON
WSPS Bottom OFF	H:H145_SDK_EQUIP_WSPS_BOTTOM_OFF
WSPS Skid TOGGLE	H:H145_SDK_EQUIP_WSPS_SKID_TOGGLE
WSPS Skid ON	H:H145_SDK_EQUIP_WSPS_SKID_ON
WSPS Skid OFF	H:H145_SDK_EQUIP_WSPS_SKID_OFF
Skid Settling Preventers TOGGLE	H:H145_SDK_EQUIP_SKID_SETTLING_PREVENTERS_TOGGLE
Skid Settling Preventers ON	H:H145_SDK_EQUIP_SKID_SETTLING_PREVENTERS_ON
Skid Settling Preventers OFF	H:H145_SDK_EQUIP_SKID_SETTLING_PREVENTERS_OFF
Air Conditioning TOGGLE	H:H145_SDK_EQUIP_AIRCONDITIONING_TOGGLE
Air Conditioning ON	H:H145_SDK_EQUIP_AIRCONDITIONING_ON
Air Conditioning OFF	H:H145_SDK_EQUIP_AIRCONDITIONING_OFF
Fuel Flow Sensor TOGGLE	H:H145_SDK_EQUIP_FUELFLOWSENSOR_TOGGLE
Fuel Flow Sensor ON	H:H145_SDK_EQUIP_FUELFLOWSENSOR_ON
Fuel Flow Sensor OFF	H:H145_SDK_EQUIP_FUELFLOWSENSOR_OFF
ACAS (Traffic) TOGGLE	H:H145_SDK_EQUIP_ACAS_TOGGLE
ACAS (Traffic) ON	H:H145_SDK_EQUIP_ACAS_ON
ACAS (Traffic) OFF	H:H145_SDK_EQUIP_ACAS_OFF
HTAWS (Terrain) TOGGLE	H:H145_SDK_EQUIP_HTAWS_TOGGLE
HTAWS (Terrain) ON	H:H145_SDK_EQUIP_HTAWS_ON
HTAWS (Terrain) OFF	H:H145_SDK_EQUIP_HTAWS_OFF
IBF (Filter) TOGGLE	H:H145_SDK_EQUIP_IBF_TOGGLE
IBF (Filter) ON	H:H145_SDK_EQUIP_IBF_ON
IBF (Filter) OFF	H:H145_SDK_EQUIP_IBF_OFF
Cockpit Weapon Sights TOGGLE	H:H145_SDK_EQUIP_WEAPONS_SIGHT_TOGGLE
Cockpit Weapon Sights OFF	H:H145_SDK_EQUIP_WEAPONS_SIGHT_OFF
Cockpit Weapon Sights ON	H:H145_SDK_EQUIP_WEAPONS_SIGHT_ON
Helmet Display ON	H:H145_SDK_EQUIP_HMD_ON
Helmet Display OFF	H:H145_SDK_EQUIP_HMD_OFF
Helmet Display TOGGLE	H:H145_SDK_EQUIP_HMD_TOGGLE
Bambi Bucket ON	H:H145_SDK_EQUIP_BAMBI_ON
Bambi Bucket OFF	H:H145_SDK_EQUIP_BAMBI_OFF
Bambi Bucket TOGGLE	H:H145_SDK_EQUIP_BAMBI_TOGGLE
Cargo Hook ON	H:H145_SDK_EQUIP_HOOK_ON
Cargo Hook OFF	H:H145_SDK_EQUIP_HOOK_OFF
Cargo Hook TOGGLE	H:H145_SDK_EQUIP_HOOK_TOGGLE
Fabric Glareshields ON	H:H145_SDK_EQUIP_FABRIC_FLARESHIELDS_ON
Fabric Glareshields OFF	H:H145_SDK_EQUIP_FABRIC_FLARESHIELDS_OFF
Fabric Glareshields TOGGLE	H:H145_SDK_EQUIP_FABRIC_FLARESHIELDS_TOGGLE

Sun Visors ON	H:H145_SDK_EQUIP_SUN_VISORS_ON
Sun Visors OFF	H:H145_SDK_EQUIP_SUN_VISORS_OFF
Sun Visors TOGGLE	H:H145_SDK_EQUIP_SUN_VISORS_TOGGLE
Safety Patches ON	H:H145_SDK_EQUIP_SAFETY_PATCHES_ON
Safety Patches OFF	H:H145_SDK_EQUIP_SAFETY_PATCHES_OFF
Safety Patches TOGGLE	H:H145_SDK_EQUIP_SAFETY_PATCHES_TOGGLE
ELT (ADELT) ON	H:H145_SDK_EQUIP_ADELT_ON
ELT (ADELT) OFF	H:H145_SDK_EQUIP_ADELT_OFF
ELT (ADELT) TOGGLE	H:H145_SDK_EQUIP_ADELT_TOGGLE
Hoist ON	H:H145_SDK_EQUIP_HOIST_ON
Hoist OFF	H:H145_SDK_EQUIP_HOIST_OFF
Hoist TOGGLE	H:H145_SDK_EQUIP_HOIST_TOGGLE
HISL ON	H:H145_SDK_EQUIP_HISL_ON
HISL OFF	H:H145_SDK_EQUIP_HISL_OFF
HISL TOGGLE	H:H145_SDK_EQUIP_HISL_TOGGLE
Snow Skis ON	H:H145_SDK_EQUIP_SKID_SKI_ON
Snow Skis OFF	H:H145_SDK_EQUIP_SKID_SKI_OFF
Snow Skis TOGGLE	H:H145_SDK_EQUIP_SKID_SKI_TOGGLE
Emergency Floats ON	H:H145_SDK_EQUIP_SKID_FLOATS_ON
Emergency Floats OFF	H:H145_SDK_EQUIP_SKID_FLOATS_OFF
Emergency Floats TOGGLE	H:H145_SDK_EQUIP_SKID_FLOATS_TOGGLE
Long Skids ON	H:H145_SDK_EQUIP_SKID_LONG_ON
Long Skids OFF	H:H145_SDK_EQUIP_SKID_LONG_OFF
Long Skids TOGGLE	H:H145_SDK_EQUIP_SKID_LONG_TOGGLE
Second Landing Light ON	H:H145_SDK_EQUIP_SECOND_LANDING_LIGHT_ON
Second Landing Light OFF	H:H145_SDK_EQUIP_SECOND_LANDING_LIGHT_OFF
Second Landing Light TOGGLE	H:H145_SDK_EQUIP_SECOND_LANDING_LIGHT_TOGGLE
Chin Window Plates ON	H:H145_SDK_EQUIP_CHIN_WINDOW_PLATES_ON
Chin Window Plates OFF	H:H145_SDK_EQUIP_CHIN_WINDOW_PLATES_OFF
Chin Window Plates TOGGLE	H:H145_SDK_EQUIP_CHIN_WINDOW_PLATES_TOGGLE

MFDs

Name	Event
MFD1 SoftKey Top 1	H:MFD1_SoftKey_T1
MFD1 SoftKey Top 2	H:MFD1_SoftKey_T2
MFD1 SoftKey Top 3	H:MFD1_SoftKey_T3
MFD1 SoftKey Top 4	H:MFD1_SoftKey_T4
MFD1 SoftKey Top 5	H:MFD1_SoftKey_T5
MFD1 SoftKey Top 6	H:MFD1_SoftKey_T6
MFD1 SoftKey Left 1	H:MFD1_SoftKey_L1
MFD1 SoftKey Left 2	H:MFD1_SoftKey_L2
MFD1 SoftKey Left 3	H:MFD1_SoftKey_L3
MFD1 SoftKey Left 4	H:MFD1_SoftKey_L4
MFD1 SoftKey Left 5	H:MFD1_SoftKey_L5
MFD1 SoftKey Left 6	H:MFD1_SoftKey_L6
MFD1 SoftKey Right 1	H:MFD1_SoftKey_R1
MFD1 SoftKey Right 2	H:MFD1_SoftKey_R2
MFD1 SoftKey Right 3	H:MFD1_SoftKey_R3
MFD1 SoftKey Right 4	H:MFD1_SoftKey_R4
MFD1 SoftKey Right 5	H:MFD1_SoftKey_R5
MFD1 SoftKey Right 6	H:MFD1_SoftKey_R6
MFD1 SoftKey Bottom 1	H:MFD1_SoftKey_B1
MFD1 SoftKey Bottom 2	H:MFD1_SoftKey_B2
MFD1 SoftKey Bottom 3	H:MFD1_SoftKey_B3
MFD1 SoftKey Bottom 4	H:MFD1_SoftKey_B4
MFD1 SoftKey Bottom 5	H:MFD1_SoftKey_B5
MFD1 SoftKey Bottom 6	H:MFD1_SoftKey_B6
MFD1 Small Knob Clockwise	H:MFD1_SoftKey_KnobInnerClockwise
MFD1 Small Knob AntiClockwise	H:MFD1_SoftKey_KnobInnerAntiClockwise
MFD1 Small Knob Push	H:MFD1_SoftKey_KnobInnerPush
MFD1 Small Knob Push (Long)	H:MFD1_SoftKey_KnobInnerPushLong
MFD1 Large Knob Clockwise	H:MFD1_SoftKey_KnobOuterClockwise
MFD1 Large Knob AntiClockwise	H:MFD1_SoftKey_KnobOuterAntiClockwise
MFD1 LUM (oveall intensity) Up	H:MFD1_SoftKey_LUM_UP
MFD1 LUM (oveall intensity) Down	H:MFD1_SoftKey_LUM_DOWN
MFD1 BRT (underlay intensity) Up	H:MFD1_SoftKey_BRT_UP
MFD1 BRT (underlay intensity) Down	H:MFD1_SoftKey_BRT_DOWN
MFD1 CTRS (overlay intensity) Up	H:MFD1_SoftKey_CTRS_UP

MFD1 CTRS (overlay intensity) Down	H:MFD1_SoftKey_CTRSW_DOWN
MFD1 Power	H:MFD1_SoftKey_POWER
MFD2 SoftKey Top 1	H:MFD2_SoftKey_T1
MFD2 SoftKey Top 2	H:MFD2_SoftKey_T2
MFD2 SoftKey Top 3	H:MFD2_SoftKey_T3
MFD2 SoftKey Top 4	H:MFD2_SoftKey_T4
MFD2 SoftKey Top 5	H:MFD2_SoftKey_T5
MFD2 SoftKey Top 6	H:MFD2_SoftKey_T6
MFD2 SoftKey Left 1	H:MFD2_SoftKey_L1
MFD2 SoftKey Left 2	H:MFD2_SoftKey_L2
MFD2 SoftKey Left 3	H:MFD2_SoftKey_L3
MFD2 SoftKey Left 4	H:MFD2_SoftKey_L4
MFD2 SoftKey Left 5	H:MFD2_SoftKey_L5
MFD2 SoftKey Left 6	H:MFD2_SoftKey_L6
MFD2 SoftKey Right 1	H:MFD2_SoftKey_R1
MFD2 SoftKey Right 2	H:MFD2_SoftKey_R2
MFD2 SoftKey Right 3	H:MFD2_SoftKey_R3
MFD2 SoftKey Right 4	H:MFD2_SoftKey_R4
MFD2 SoftKey Right 5	H:MFD2_SoftKey_R5
MFD2 SoftKey Right 6	H:MFD2_SoftKey_R6
MFD2 SoftKey Bottom 1	H:MFD2_SoftKey_B1
MFD2 SoftKey Bottom 2	H:MFD2_SoftKey_B2
MFD2 SoftKey Bottom 3	H:MFD2_SoftKey_B3
MFD2 SoftKey Bottom 4	H:MFD2_SoftKey_B4
MFD2 SoftKey Bottom 5	H:MFD2_SoftKey_B5
MFD2 SoftKey Bottom 6	H:MFD2_SoftKey_B6
MFD2 Small Knob Clockwise	H:MFD2_SoftKey_KnobInnerClockwise
MFD2 Small Knob AntiClockwise	H:MFD2_SoftKey_KnobInnerAntiClockwise
MFD2 Small Knob Push	H:MFD2_SoftKey_KnobInnerPush
MFD2 Small Knob Push (Long)	H:MFD2_SoftKey_KnobInnerPushLong
MFD2 Large Knob Clockwise	H:MFD2_SoftKey_KnobOuterClockwise
MFD2 Large Knob AntiClockwise	H:MFD2_SoftKey_KnobOuterAntiClockwise
MFD2 LUM (oveall intensity) Up	H:MFD2_SoftKey_LUM_UP
MFD2 LUM (oveall intensity) Down	H:MFD2_SoftKey_LUM_DOWN
MFD2 BRT (underlay intensity) Up	H:MFD2_SoftKey_BRT_UP
MFD2 BRT (underlay intensity) Down	H:MFD2_SoftKey_BRT_DOWN
MFD2 CTRS (overlay intensity) Up	H:MFD2_SoftKey_CTRS_UP
MFD2 CTRS (overlay intensity) Down	H:MFD2_SoftKey_CTRSW_DOWN
MFD2 Power	H:MFD2_SoftKey_POWER
MFD4 SoftKey Top 1	H:MFD4_SoftKey_T1
MFD4 SoftKey Top 2	H:MFD4_SoftKey_T2
MFD4 SoftKey Top 3	H:MFD4_SoftKey_T3
MFD4 SoftKey Top 4	H:MFD4_SoftKey_T4
MFD4 SoftKey Top 5	H:MFD4_SoftKey_T5
MFD4 SoftKey Top 6	H:MFD4_SoftKey_T6
MFD4 SoftKey Left 1	H:MFD4_SoftKey_L1
MFD4 SoftKey Left 2	H:MFD4_SoftKey_L2
MFD4 SoftKey Left 3	H:MFD4_SoftKey_L3
MFD4 SoftKey Left 4	H:MFD4_SoftKey_L4
MFD4 SoftKey Left 5	H:MFD4_SoftKey_L5
MFD4 SoftKey Left 6	H:MFD4_SoftKey_L6
MFD4 SoftKey Right 1	H:MFD4_SoftKey_R1
MFD4 SoftKey Right 2	H:MFD4_SoftKey_R2
MFD4 SoftKey Right 3	H:MFD4_SoftKey_R3
MFD4 SoftKey Right 4	H:MFD4_SoftKey_R4
MFD4 SoftKey Right 5	H:MFD4_SoftKey_R5
MFD4 SoftKey Right 6	H:MFD4_SoftKey_R6
MFD4 SoftKey Bottom 1	H:MFD4_SoftKey_B1
MFD4 SoftKey Bottom 2	H:MFD4_SoftKey_B2
MFD4 SoftKey Bottom 3	H:MFD4_SoftKey_B3
MFD4 SoftKey Bottom 4	H:MFD4_SoftKey_B4
MFD4 SoftKey Bottom 5	H:MFD4_SoftKey_B5
MFD4 SoftKey Bottom 6	H:MFD4_SoftKey_B6
MFD4 Small Knob Clockwise	H:MFD4_SoftKey_KnobInnerClockwise
MFD4 Small Knob AntiClockwise	H:MFD4_SoftKey_KnobInnerAntiClockwise
MFD4 Small Knob Push	H:MFD4_SoftKey_KnobInnerPush
MFD4 Small Knob Push (Long)	H:MFD4_SoftKey_KnobInnerPushLong
MFD4 Large Knob Clockwise	H:MFD4_SoftKey_KnobOuterClockwise
MFD4 Large Knob AntiClockwise	H:MFD4_SoftKey_KnobOuterAntiClockwise
MFD4 LUM (oveall intensity) Up	H:MFD4_SoftKey_LUM_UP
MFD4 LUM (oveall intensity) Down	H:MFD4_SoftKey_LUM_DOWN

MFD4 BRT (underlay intensity) Up	H:MFD4_SoftKey_BRT_UP
MFD4 BRT (underlay intensity) Down	H:MFD4_SoftKey_BRT_DOWN
MFD4 CTRS (overlay intensity) Up	H:MFD4_SoftKey_CTRS_UP
MFD4 CTRS (overlay intensity) Down	H:MFD4_SoftKey_CTRSW_DOWN
MFD4 Power	H:MFD4_SoftKey_POWER

IESI

Name	Event
Baro Knob Clockwise	H:H145_SDK_IESI_BARO_CLOCKWISE
Baro Knob AntiClockwise	H:H145_SDK_IESI_BARO_ANTICLOCKWISE
Baro STD	H:H145_SDK_IESI_BARO_STD
Cage	H:H145_SDK_IESI_CAGE
Brightness Up	H:H145_SDK_IESI_BRT_UP
Brightness Down	H:H145_SDK_IESI_BRT_DOWN

Center Console Other

Name	Event
ELTCP ELT ON	H:H145_SDK_ELT_SWITCH_ON
ELTCP ELT ARM	H:H145_SDK_ELT_SWITCH_ARM
ELTCP ELT RESET	H:H145_SDK_ELT_SWITCH_RESET
GPUCP Ground Power (LIGHTS) ON	H:H145_SDK_GPCP_PWR_ON
GPUCP Ground Power (LIGHTS) OFF	H:H145_SDK_GPCP_PWR_OFF
AIRCP DEFOG ON	H:H145_SDK_AIRCP_DEFOG_ON
AIRCP DEFOG OFF	H:H145_SDK_AIRCP_DEFOG_OFF
AIRCP AIR MIX ON	H:H145_SDK_AIRCP_AIRMIX_ON
AIRCP AIR MIX OFF	H:H145_SDK_AIRCP_AIRMIX_OFF

Sensor Pod

Name	Event
Power TOGGLE	H:H145_SDK_SENSORPOD_MONITOR_POWER_TOGGLE
Power ON	H:H145_SDK_SENSORPOD_MONITOR_POWER_ON
Power OFF	H:H145_SDK_SENSORPOD_MONITOR_POWER_OFF
Move RIGHT	H:H145_SDK_SENSORPOD_MOVE_RIGHT
Move LEFT	H:H145_SDK_SENSORPOD_MOVE_LEFT
Move FORWARD	H:H145_SDK_SENSORPOD_MOVE_FWD
Move AFT	H:H145_SDK_SENSORPOD_MOVE_AFT

System Failures

Note that more failures are directly writable to their L:Var.

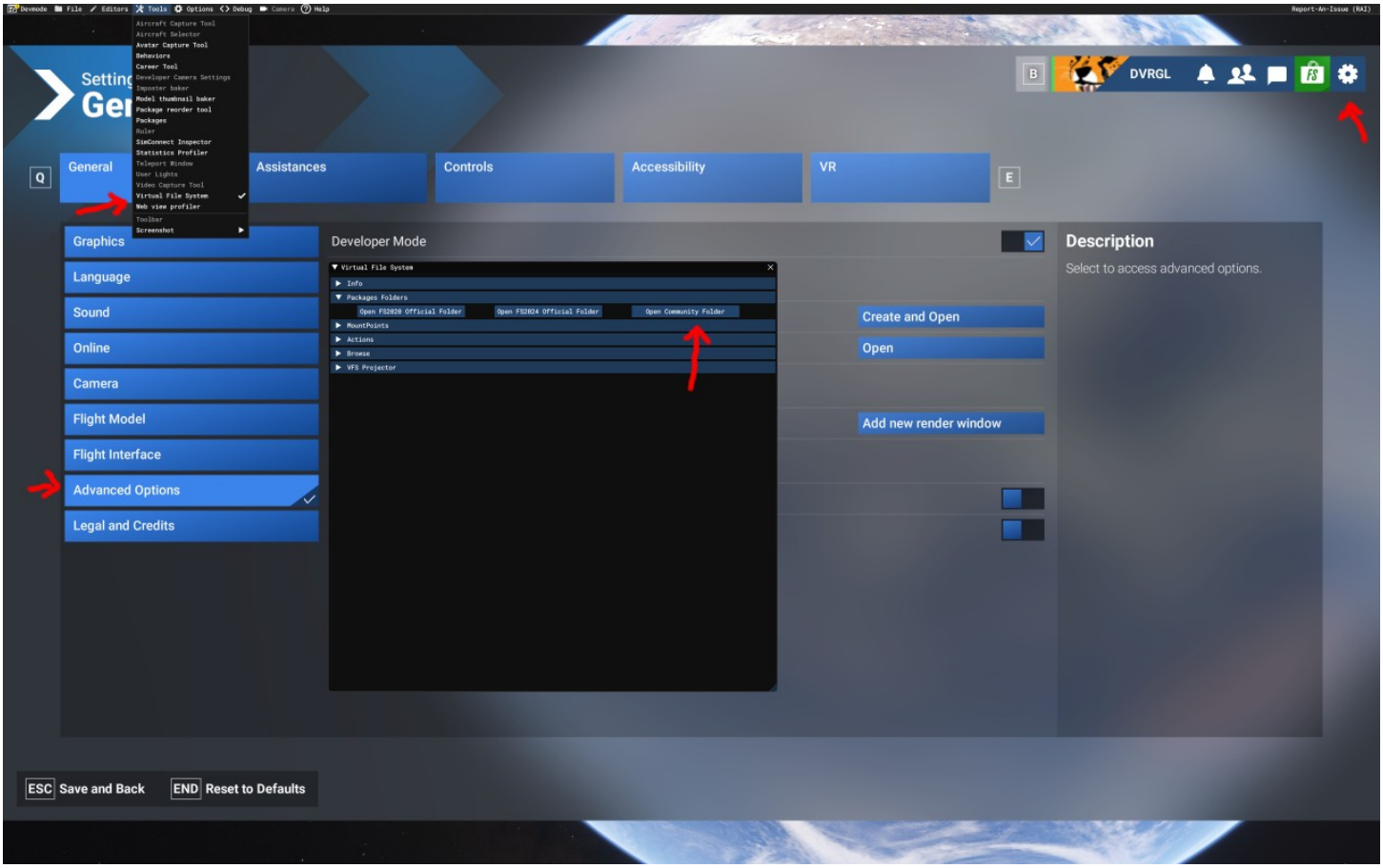
Name	Event
Engine 1 Failure ON	H:H145_SDK_FAILURE_ENG1_FAIL_ON
Engine 1 Failure OFF	H:H145_SDK_FAILURE_ENG1_FAIL_OFF
Engine 1 Failure TOGGLE	H:H145_SDK_FAILURE_ENG1_FAIL_TOGGLE
Engine 2 Failure ON	H:H145_SDK_FAILURE_ENG2_FAIL_ON
Engine 2 Failure OFF	H:H145_SDK_FAILURE_ENG2_FAIL_OFF
Engine 2 Failure TOGGLE	H:H145_SDK_FAILURE_ENG2_FAIL_TOGGLE
Engine 1 FIRE ON	H:H145_SDK_FAILURE_ENG1_FIRE_ON
Engine 1 FIRE OFF	H:H145_SDK_FAILURE_ENG1_FIRE_OFF
Engine 1 FIRE TOGGLE	H:H145_SDK_FAILURE_ENG1_FIRE_TOGGLE
Engine 2 FIRE ON	H:H145_SDK_FAILURE_ENG2_FIRE_ON
Engine 2 FIRE OFF	H:H145_SDK_FAILURE_ENG2_FIRE_OFF
Engine 2 FIRE TOGGLE	H:H145_SDK_FAILURE_ENG2_FIRE_TOGGLE
Mast Moment Exceed OFF	H:H145_SDK_MASTMOMENT_EXCEED_OFF
Mast Moment Exceed ON	H:H145_SDK_MASTMOMENT_EXCEED_ON
Fire Bottle 1 EMPTY	H:H145_SDK_FIREBOTTLE1_EMPTY
Fire Bottle 1 CHARGED	H:H145_SDK_FIREBOTTLE1_FULL
Fire Bottle 2 EMPTY	H:H145_SDK_FIREBOTTLE2_EMPTY
Fire Bottle 2 CHARGED	H:H145_SDK_FIREBOTTLE2_FULL

Tipps and tricks

How do I find the community folder?

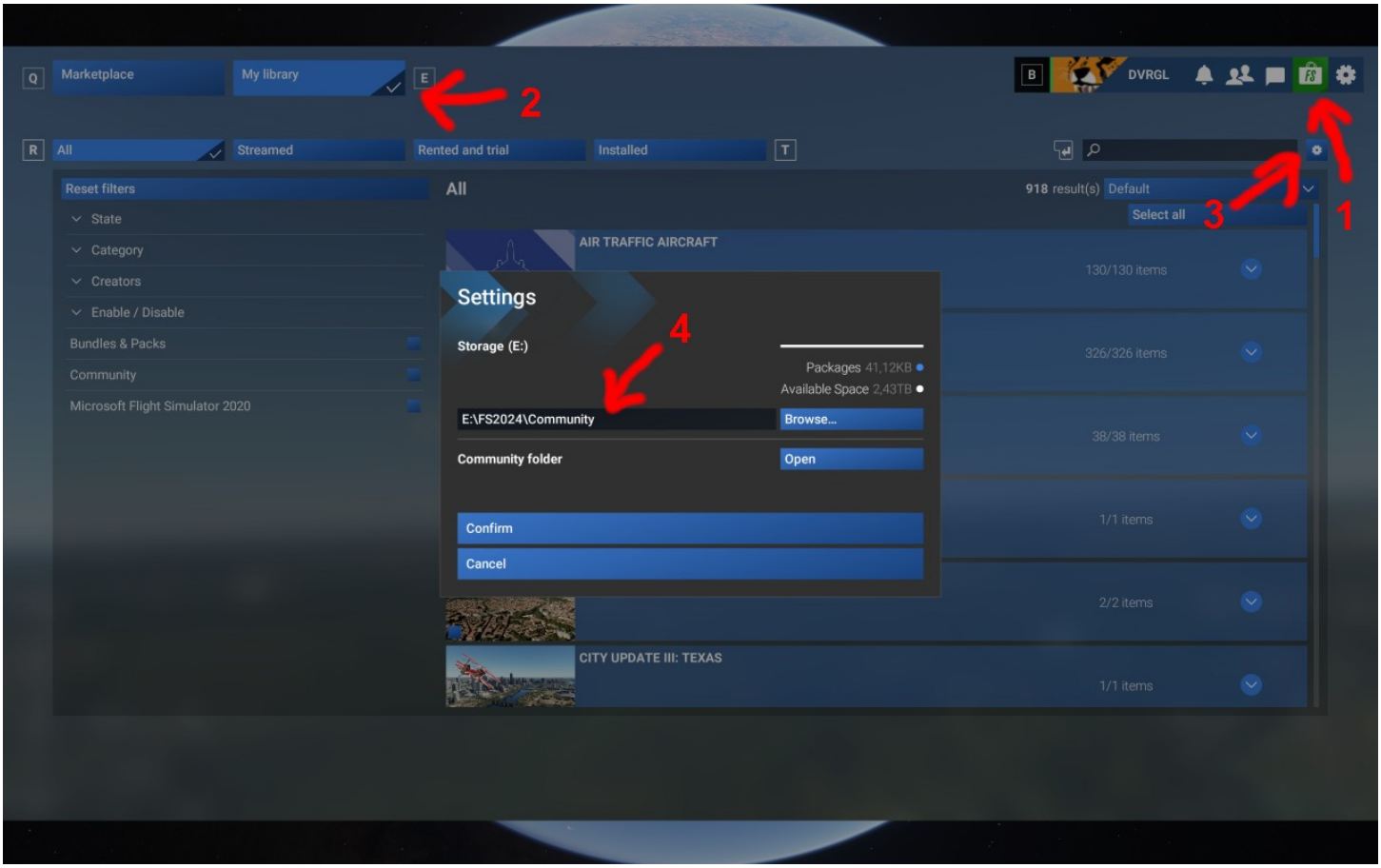
Here are two methods to find the community folder:

In Developer Mode



- Launch Microsoft Flight Simulator
- Go to Options → General Options
- Enable Developer Mode by clicking on the Developers tab and setting Developer Mode to On
- In the new menu bar at the top of the screen, click Tools → Virtual File System
- Expand "Packages Folders"
- Click "Open Community Folder"
- This method works regardless of how you installed MSFS and is the most reliable way to locate the correct folder

In the settings



- Start Microsoft Flight Simulator
- Click on the Marketplace at the top right
- Click on "My Libary" at the top left
- Click on the cogwheel on the right
- In the settings you will find or can set the folder

GTC and how it works

GTC at low speed (like hover):
cyclic beep: up/down: forward/backward longitudinal speed (ground speed)
cyclic beep: left/right: left/right lateral speed (ground speed)
collective beep left/right: heading reference

GTC at high speed:
cyclic beep: up/down: forward/backward longitudinal speed (ground speed)
cyclic beep: left/right: heading reference
collective beep left/right: no function

HANDS ON detection setting

None:
This means that your stick is always active but you should use Trim Release to really take control otherwise the AFCS will be fighting you the entire time. Note that in this setting, L:FFB_HANDS_ON_CYCLIC or L:FFB_HANDS_ON_CYCLICY/L:FFB_HANDS_ON_CYCLICX can be used to provide CSAS signal.
Motion Rate
This uses motion rate detection (with a slider to adjust sensitivity). When you move the stick it goes into CSAS modes automatically, which is similar to using trim release and more like the 1.0 flight model
Deadzone
This is the traditional deadzone that has been an option for a long time.
Ignore
This is the old AFCS Override: AP Only setting, where your stick is ignored unless you press Trim Release. This is good for people without friction locks or people who absolutely demand it works like the Asobo autopilot.

Behind pedal settings

There are a few types of pedals:

- Force-feedback (FFB) with trim motors (very rare ..)
- no-spring helicopter pedals
- traditional spring-center pedals for fixed wing flight sim

For no-spring pedals you want to use it like this:
Pedal trim system: Hardware
FEET ON detection strategy: Motion

That will turn the vtrim off and the blue line won't move to trim under you. Your pedals are directly connected but the SEMA are still able to fine-tune and provide stabilization. That means you want to do a normal helicopter takeoff with displaced pedals, and you'll generally move your pedals for the correct position in flight. Not much needs to be done here (pedal to the left in hover, slightly to the right in cruise).

You can use the Motion FEET ON detection strategy because the Deadzone style is only around the center and thus is not workable.

For centering spring pedals or xbox controller you want
Pedal trim system: Software (default setting)
FEET ON detection strategy: Deadzone (default setting)

This will provide a virtual trim that enables your spring pedals to always be centered. You'll push left and right to go left and right, and you don't need to worry about the 20% left displacement in hover. Your pedals are still directly connected all the time and you'll still want to be FEET ON. You generally will want to be ahead of the machine and still provide the correct pedal input, but the SEMA will stabilize, but at takeoff you will start moving the vtrim and this could lead to wobbling if you don't quickly get into the air and back off the pedal.

The above is how I intend for people to use the settings to get an authentic experience for their controller. There is an alternate way to fly which is based only on people not liking the stabilized yaw in builds 412+.

The ATT Yaw Mode setting has two positions. Normal is the authentic level of damping. Legacy gives you more traditional helicopter feeling, but the real machine doesn't react like that. I would remove this if possible but people like it and it's been a good mitigation for people who found the yaw stab logic to be annoying.

Behind cyclic settings

For a spring joystick you'll want to use it like this:
Cyclic trim system: software (default setting)
HANDS ON detection strategy: deadzone (default setting)
Follow-up trim: HOVER or BOTH (default setting)

In this configuration you'll have the virtual trim (blue cross) constantly moving to relax your stick position. It's slow so it shouldn't bother you, but it means no matter if you are in cruise or in hover, the stick will slowly relax so you are at zero forces.

Follow-up trim should either be in HOVER or BOTH modes. When you are in ATT mode (blank on the MFD) you can bump the stick and have either "fly back to previous setpoints" or it can "sync to current" to where the aircraft ends up. Follow-up trim depends on HANDS ON detection strategy. The real aircraft will use HOVER only. (Helionix V10 adds TAC mode which is a follow-up trim for cruise).

IMPORTANT: The deadzone strategy for detecting HANDS ON means we expect your stick to reliably re-center itself and you should set a deadzone which is large enough such that you don't have spurious HANDS ON events when you actually are not manipulating the stick.

You have the option (not recommended) of using the DCS-style center-displacement binding. You'll need to switch the cyclic trim system is on hardware as it would compete with the displacement binding.

Add a control binding for MAGNETO 2 START (default binding)
When your control is in a desirable position, click the binding
Your cyclic is now disocnnected. Within the Center displace reset time setting time, you must return your stick to a center position.
Your controls will now be reconnected, but you moved them into the neutral position, so there is no change in output since step 2.
H145 then:

1. disconnects your control entirely
2. sets the trim to your last position (since it was what you wanted)
3. now you have 250ms (controlled by the setting on the tablet) to re-center your controller
4. now your controller is reconnected but at zero, and the trim is where you were holding your controller a second ago

Of course you keep the same current ROTOR TRIM RESET(on press, repeat) binding, the MAGNETO 2 START(on release) works in tandem

This strategy of updating the center position is common on other sims like DCS or the xplane rotorsim EC135, although we believe the Trim Release strategy works better without interruption to the controls.

Trim release

The understanding and correct use of trim release always is a problem. FireHawk wrote a short explanation in the Discord H160_discussion. This is for the H145 correct too.

When your Automated Flight Control System (AFCS) is turned on (all “OFF” indicators not showing on the AFCS panel) this is what’s called Attitude Mode (ATT). In ATT, the aircraft will always want to hold pitch and roll axis (nose up/down and roll right/left) where you last left them when, basically, you take you hands off the stick. So, let’s say you’re hands off, flying 120kts, straight-n-level. You see some birds ahead, so you grab the cyclic and roll right to avoid them. When you release the stick, it will roll left back to level on its own. Same for pitch.

Now, doing the same exercise, S-n-L, 120kts, except this time press and hold Trim Release (TR) while you roll right. When you’re banked over, release TR and the cyclic and now the aircraft will hold that bank angle (that Attitude). So wherever your nose is pointed or tilted, if you move the stick, the aircraft will ‘rubber-band’ right back to where it was. Press TR, move the stick, release TR and now you’ve centered the rubber bands (as it were).
Also, Force Trim Release is NOT the same as Beep Trim. Beep Trim lets you move those ATT set points small ‘beeps’ at a time
Beep Trim functions kinda similar to a fixed-wing elevator or aileron trim system in feel (but is a different function mechanically).

Difference between 'trim release' and 'spring override'

The indication OVERRIDE is used for two different notifications.
1- When using Trim Release in ATT Mode (TR), the override is telling you that you are overriding the AFCS system and the ‘trims’ are disengaged but following your stick movement. When you release TR, the servos are now positioned to that new set point for hands off (or hands light) control of the cyclic.
2- When in ATT or Upper Modes, and you don’t use TR when moving the cyclic, OVERRIDE is telling you that you are ‘pushing against springs’. The servos remain at their set points, and when you release the cyclic, the aircraft will want to return to those set points or UM settings.
Let’s say you are on final approach (ATT Mode) to your LZ and you encounter a flock of birds. You roll right to avoid the birds, roll left back on course and when you let go (or go light touch), the ATT set points are still where they were before.

Correct setting of Trim Release

One source of problems is the correct setting and application of Trim Release button. Dave described this as follows in the H145 discussion forum:
The types of cyclic controls are:
Spring joystick (centering spring)

- no-spring helicopter controls (friction lock available). may or may not have light spring forces.
- no-spring helicopter controls (friction lock NOT available)
- Keyboard or controller buttons. This is difficult and rare.
- Full FFB cyclic. This is pretty rare.

In every case below you'll want to use zero deadzone in MSFS settings.

spring joystick:

Cyclic trim system: software (default setting)
HANDS ON detection strategy: deadzone (default setting)
Follow-up trim: HOVER or BOTH (default setting)
In this configuration you'll have the virtual trim (blue cross) constantly moving to relax your stick position. It's slow so it shouldn't bother you, but it means no matter if you are in cruise or in hover, the stick will slowly relax so you are at zero forces.

Follow-up trim should either be in HOVER or BOTH modes. When you are in ATT mode (blank on the MFD) you can bump the stick and have either "fly back to previous setpoints" or it can "sync to current" to where the aircraft ends up. Follow-up trim depends on HANDS ON detection strategy. The real aircraft will use HOVER only. (Helionix V10 adds TAC mode which is a follow-up trim for cruise).
IMPORTANT: The deadzone strategy for detecting HANDS ON means we expect your stick to reliably re-center itself and you should set a deadzone which is large enough such that you don't have spurious HANDS ON events when you actually are not manipulating the stick.

You have the option (not recommended) of using the DCS-style center-displacement binding. You'll need to switch the cyclic trim system is on hardware as it would compete with the displacement binding.

Add a control binding for MAGNETO 2 START (default binding)
When your control is in a desirable position. click the binding
Your cyclic is now disocnnected. Within the Center displace reset time setting time, you must return your stick to a center position.
Your controls will now be reconnected, but you moved them into the neutral position, so there is no change in output since step 2.

This strategy of updating the center position is common on other sims like DCS or the xplane rotorsim EC135, although we believe the Trim Release strategy works better without interruption to the controls.

no-spring or helicopter cyclic:

Cyclic trim system: Hardware
Follow-up trim: BOTH
Friction Lock available:
HANDS ON detection strategy: Motion
Friction Lock not available:
HANDS ON detection strategy: Ignore

The cyclic virtual trim will not be available, which means your stick is always directly connected to the helicopter output without an offset provided by the trim. As a consequence you'll need to deflect your stick forward in cruise as with any other helicopter. You can enable the software trim system and make use of the virtual trim if you prefer.
Motion HANDS ON detection strategy is necessary for HANDS ON detection because the deadzone is unlikely to be usable due to lack of a centering force. If you can return your stick reliably to the center, you may like to use the centering-spring settings entirely. If the Motion strategy doesn't work well for you, then NONE could be used, along with heavy Trim Release use.
If your control cannot be parked in a position, you'll not be able to use the AFCS unless you use the Ignore strategy for HANDS on detection. Using the Ignore mode means that your stick position will be completely ignored except when you use Trim Release. By this method you may use beep trim and UPPER MODES, while your cyclic is deflected to one corner or in any position.
Follow-up trim is recommended to be on the BOTH setting to avoid the AFCS fighting you. Since your stick is always directly connected, your hand holding it would prevent fly-back behavior and as such it's probably better to just avoid the AFCS trying to fly back regardless.

Force Feedback cyclic:

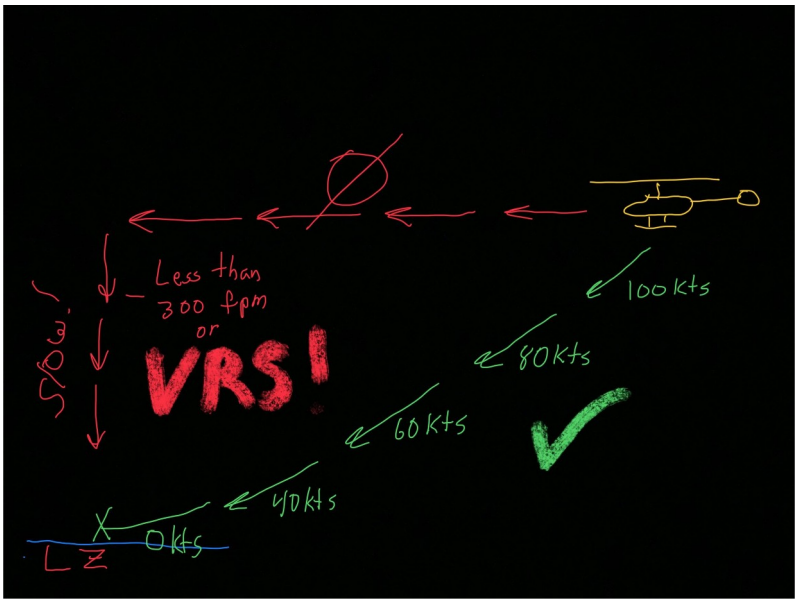
Cyclic trim system: Hardware
Follow-up trim: BOTH
HANDS ON detection strategy: None

FFB controllers will monitor SEMA L:Vars and drive trim motors to de-saturate them, so virtual trim must be disabled.
Follow-up trim should be on BOTH for the reasons mentioned above around avoiding AFCS fly-back.

HANDS ON detection strategy should be None to disable built-in CSAS modes. CSAS detection signal must be provided by L:Var. If you cannot provided CSAS signal then Trim Release should be used.

How to make an approach:

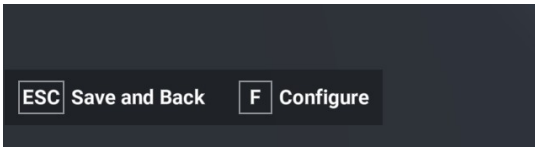
Many pilots think in terms of vertical flight, but a 3-ton aircraft of any wing type (rotor or fixed) generally requires a bit more horizontal thought process. As you approach your landing zone (starting about 3 to 4 miles out, depending on altitude) pitch the nose up to start scrubbing speed and reduce collective to start descending. Look for around 50 to 60 knots and use the collective to keep your LZ steady in your sight picture... not climbing in your windscreen, or falling. As you get closer, the idea is to keep above 30 kts forward speed so the rotor disc continues to act as a total airfoil. The goal then is to scrub speed and altitude so as to end up at zero speed, zero descent, and about 10 feet in hover (that puts you hovering in ground effect.. HIGE) all at the same time. Now, stabilize hover and slowly work the collective until skids/wheels are on the ground. Practice, practice, practice until you find the muscle memory, then it will all come together for you.



(Thx. To FireHawk)

I have installed 2024 or switched to it and cannot find the standard HPG livery (anymore).

2024 organisiert die Livery anders als 2020. Die Livery finden sich in der Flugzeugauswahl (Modell wählen) hinter der Funktion “F Configure” oder einfach durch drücken der Taste “F”.

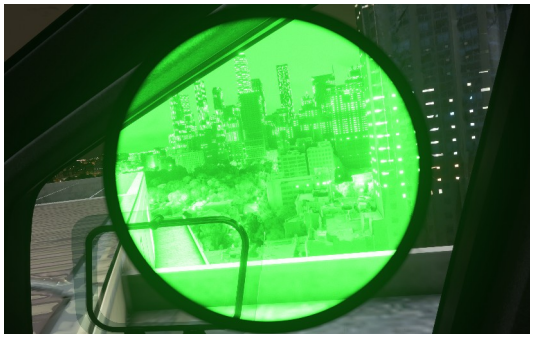


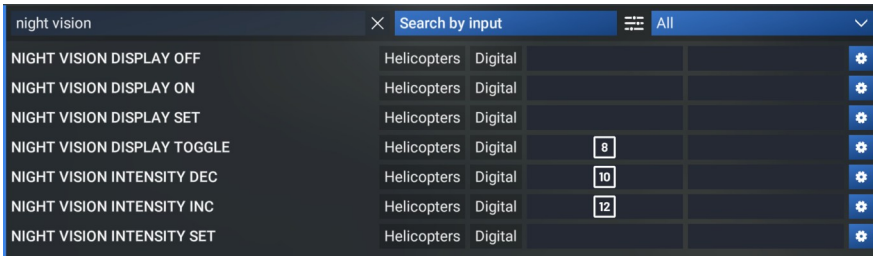
How can I integrate night vision?

Since FS 2024 SU4, night vision has been built in as standard. It only needs to be activated in the aircraft and the switches set up.
To set it up, open the directory /Community\hpg-airbus-h160\Simobjects\Airplanes\hpg-airbus-h160-civ and locate the file system.cfg. Open it with an editor and scroll all the way down. Insert the following lines here:

[NIGHT_VISION]
Available = true
Intensity = 300

Save the file, start the simulator, and enter the necessary key assignments.





What can I do if OSM stops providing map data for the DMAP?

You can register for free at https://cloud.maptiler.com/ and will then receive your private API key, which allows you to access all types of raster maps (including OSM) using the free quota—that should be more than enough.
Next, go to the directory ...*Community\hpg-airbus-h145\html_ui\HPGH145-User\MFD*\ and insert the following lines into the DMAP.json file:

```
{
 "title": "VECTOR",
 "url": "https://api.maptiler.com/maps/openstreetmap/{z}/{x}/{y}.png?key=**************",
 "tileSize": 512,
 "zoomOffset": -1,
 "attributions": "&copy; OpenStreetMap contributors"
 }
```

Thanks for the tip to @Daikan (Discord).A donation to OpenStreetMap wouldn't hurt either.

Alternatively, you can use Google Maps:

```
{
 "title": "STREET",
 "url": "https://mt.google.com/vt/lyrs=m&x={x}&y={y}&z={z}&hl=en.jpeg",
 "tileSize": 256,
 "attributions": "&copy; Google Inc"
 },
 {
 "title": "TERRAIN",
 "url": "https://mt.google.com/vt/lyrs=p&x={x}&y={y}&z={z}&hl=en.jpeg",
 "tileSize": 256,
 "attributions": "&copy; Google Inc"
 },
```

Thanks for the tip to @WhisperDark (Discord)

If you want the new entries to also be available in the NAVD view, you must also make the change in the NAVD.json file, located in the directory mentioned above.
To verify that the entries have the correct syntax, you can use a JSON online validator (e.g., https://jsonlint.com/).

Troubleshooting

I can't find my license key

You can recover your information in these ways:

- You may request your details be re-sent to your email
- You also should find your info in your account after logging in at Hype Performance Group Store

MFD screens are black or the tablet won't open

This almost always means the installation is corrupt or there is an addon conflict.

Resolution:

1. In Hype Operations Center, select `Removes All Packages`.
2. Select `Install` on the latest version of the product.
3. If the aircraft is still not working, **remove all other addons from Community folder**.

GTN750 screens are black

If the GTN750 screens are black then use the tablet to ensure the setting is as desired. Make sure to remove old H145 packages which enabled GTN750 options, as this setting is now built-in.

Resolution:

1. `H145 Tablet` -> `Aircraft` (app) -> `Options` (page). Change `GTN750 Software` to either `pms50` or `TDSSim`.
2. If the GTN750 remains blank, reinstall the vendor software module and try again.
3. If the GTN750 remains black, **remove all other addons from Community folder**.

It feels like I am fighting the aircraft when I fly.

You need to use `Cyclic Trim Release`, this removes the deadzone and also pauses the AFCS so it will not compete with you.

The aircraft is not reliable when flying with autopilot upper modes (HDG, IAS, ALT)

Ensure that your `Cyclic Deadzone` is large enough. You must not have accidental HANDS ON input as this will compromise the autopilot.

Click-spots in the virtual cockpit are offset

Lens Correct will distort the view and break clicking on controls.

Resolution:

1. Turn off the MSFS `Lens Correction` setting.

The camera moves in an erratic way, mostly when taking off and landing.

Something is wrong with the `Camera Shake` setting, it needs to be cycled to fix it.

Resolution:

1. Turn On MSFS `Camera Shake`. Apply.
2. Turn Off MSFS `Camera Shake`. Apply.

I have bound a hotkey but it always sends multiple commands

MSFS Bindings have the option for "On Press" and "On Release". By default, keys will repeat after some short delay.

Resolution:

1. Go into MSFS Control settings and change the binding to **On Release**.

Aircraft has pink textures

A. A common problem is to install only the `Action Pack`. You must also install the `H145 Base Pack`.
B. Very old civil variant liveries need a `texture.cfg` update since they were not authored with one initially:

```
[fltsim]
fallback.1=..\..\hpg-airbus-h145-civ\texture
fallback.2=..\..\hpg-airbus-h145\texture
```

C. If you are a livery author, check the user guide for `texture.cfg` configuration for the variant you are painting.

MGB Over limit / Aircraft won't takeoff

These warnings indicate aircraft damage. This our new detailed damage model in action.

1. On the tablet, use the `Failures & Maintenance app` , and clear the damage by clicking `Do All Maintenance` .
2. (optional) turn `Aircraft Damage` to OFF in the `Aircraft` app on the `Setup` page.

Unable to start at MSFS Helipads

Liveries authored before Sim Update 11 (Nov 2022) will need to be updated in order to avoid preventing H145 from starting at helipads.

Resolution:

1. Locate the `aircraft.cfg` file within the livery package (if you find two, update both)
2. Change `ui_typerole="Helicopter"` to `ui_typerole="Rotorcraft"` .

The built-in H145 liveries are up to date, as of build 360. If you are still using Version 1.0, you would also need to make this change to the base liveries.

How can I find the community folder?

Launch Microsoft Flight Simulator
Go to Options → General Options
Enable Developer Mode by clicking on the Developers tab and setting Developer Mode to On
In the new menu bar at the top of the screen, click Tools → Virtual File System
Expand "Packages Folders"
Click "Open Community Folder"

This method works regardless of how you installed MSFS and is the most reliable way to locate the correct folder.

Problems with WU/SU update

Each time there is a WU or SU update (yes, that is redundant…) and there are H145 loading and/or startup glitches, it's almost always an add-on conflict, an initialization error during load-in, or a msfs binding hiccup. Especially when your H145 has been flying just fine prior. If you experience this, please take a breath before jumping headlong into uninstall, reinstall, reinstall the sim, reinstall Windows, or throw your computer out a Window. 9 times out of 10, a simple restart of the flight brings everything to life, and barring that, a restart of the sim usually solves the easy stuff. After that, suspect add-on conflict, FS20 assets like sceneries and 4th-party liveries, and known culprits like GSX, Flow, and other add-ons that can compete for your bits-and-bytes.

(thx to @FireHawk)

Known Issues

WTT Mode is inoperative with helicopters in MSFS

Due to this bug, we are unable to provide a WTT package for H145.

FSRealistic Low-altitude turbulence is not compatible with helicopters.

FSRealistic software has a mode `Low-altitude turbulence` that will cause problems with the flight model. Turn it off.

Tablet Time option is off by 1 hour

Reading the time is often wrong due to the sim DST database being out of date .

Helicopters don't spawn correctly on some helipads

The position of the helicopter may be forward or aft of the correct position. Asobo knows about the problem .

Helicopter ground effect transition is abrupt when approaching elevated helipads

Asobo has confirmed the bug and indicated a fix will come in a future version of MSFS.

FS 2024: TDS GTNXi is currently blocked

Asobo has confirmed that the developer has what they need to resolve this soon.

Acronym List

Acronym	Meaning
ACAS	Airborne Collision Avoidance System
ACOL	Anti-collision light
ADC	Air data computer
ADELT	Automatic deployable Emergency Locator Transmitter
ADF	Automatic direction finder
AEO	All engines operating
AFCS	Automatic Flight Control System
AGL	Above ground level
AHRS	Attitude Heading Reference System
ALT	Altitude or Altitude hold
ALT.A	Altitude acquire
AMC	Aircraft Management Computer
APCP	Autopilot control panel
A.TRIM	Automatic trim system
ATT	Attitude or Long term attitude hold
BAT	Battery
BKUP	Backup SAS
BOT	Bottle
CRHT, CR.HT	Cruise height
DA	Decision Altitude
DEG, DEGR	Degraded
DG	Directional gyro
DH	Decision height
DISCH	Discharge
DISCON	Disconnected
DMAP	Digital Map System
DME	Distance measuring equipment
DSAS	Digital SAS
DST	Distance
DTD	Data Transfer Device
DTK	Desired Track
EFB	Electronic Flight Bag
ELT	Emergency Locator Transmitter
EMER	Emergency
EMS	Emergency Medical Services
EPU	External power unit
FADEC	Full Authority Digital Engine Control
FDS	Flight Display System
FLI	First limit indicator
FMS	Flight management system
FND	Flight and Navigation Display
FPA	Flight Path Angle
GA	Go Around
GEN	Generator
GPS	Global positioning system
GS	Ground speed
GTC	Ground trajectory control
GTC.H	Ground trajectory control with hover mode
GTN	GARMIN GTN 750
HAT	Height Above Terrain
HDG	Heading
HEMS	Helicopter Emergency Medical Services
HIGE	Hover in ground effect
HISL	High Intensity Search Light
HMD	Helmet Mounted Display
HOGE	Hover out of ground effect
HTAWS	Helicopter Terrain Awareness and Warning System
IAS	Indicated airspeed
IBF	Inlet Barrier Filter
IESI	Integrated Electronic Standby Instrument/Indicator
IMA	Integrated Modular Avionics
LAVCS	Light Helicopter Active Vibration Control System
LDG	Landing (Landing Light)
LNAV	Lateral Navigation approach (nonprecision)
LNAV+V	Non–precision LNAV approach with vertical guidance
LNAV/VNAV	Lateral Navigation and Vertical Navigation approach
LOC	Localizer
LOW ALT	Low altitude
LP	Localizer Performance without vertical guidance

LP+V	Localizer Performance with advisory vertical guidance
LPV	Localizer Performance with vertical guidance
L/VNAV	Lateral Navigation and Vertical Navigation approach
LSK	Line select key
MFD	Multifunction display
MGB	Main gearbox
MISC	Miscellaneous
MSG	Message
MSTR	Master
MTOW	Maximum Takeoff weight
N1	Gas generator speed
N2	Power turbine speed
NAVD	Navigation display
OAT	Outside air temperature
OBS	Omni Bearing Selector
OEI	One engine inoperative
OGE	Out of ground effect
OVHT	Overheat
PAX	Passenger
pb	Push Button
QTY	Quantity
RA	Radar altitude
RNAV	Area Navigation
SAS	Stability augmentation system
SBAS	Satellite Based Augmentation System
SEMA	Smart electro-mechanical actuator
SL, S/L	Search Light
SK	Select Key or Soft Key
STBY	Standby
SUSP	Suspended
SVS	Synthetic Vision System
SYS, SYST	System
TAS	True airspeed
TGB	Tail gearbox
TOP	Takeoff power
TOT	Turbine outlet temperature
TRQ	Torque
TRK	Track
V.APP	Vertical approach
VENT	Ventilation
VMS	Vehicle Management System
VNE	Never-exceed speed
VNE power off	Maximum speed in autorotation
VOR	VHF omnidirectional radio ranging
VRS	Vortex Ring State
VS	Vertical speed
VTOSS	Takeoff safety speed
VY	Best rate-of-climb speed
XFER	Fuel transfer pump

MSFS/H145 Default-Function

MSFS-Function	H145 – System	H145 – Function
TOGGLE YAW DAMPER	H145M Weapons	Fire (Primary)
YAW DAMPER ON	H145M Weapons	Fire (Secondary)
ROTOR TRIM RESET	Cyclic Control	Trim Release (HOLD)
MAGNETO 3 LEFT	Collective Control	Fill Floats
TOGGLE AUTO HOVER	Cyclic Control	AP/GTC
AUTO HOVER ON	Cyclic Control	AP/GTC (Direct to GTC.H) (Advanced)
AUTOPILOT ON	Cyclic Control	AP/BKUP ON
AUTOPILOT OFF	Cyclic Control	AP/UM OFF
TOGGLE DISENGAGE AUTOPILOT	Cyclic Control	AP/BKUP CUT
AUTOTHROTTLE DISCONNECT	Collective Control	Collective Trim Release (HOLD)
ARM AUTO THROTTLE	Collective Control	OEI HI/LO (Toggle)
AUTO THROTTLE TO GA	Collective Control	GA (Go Around)
ANNUNCIATOR SWITCH OFF	Cyclic Control	Message List RESET
AILERON TRIM RIGHT	Cyclic Control	Cyclic Beep Trim RIGHT
AILERON TRIM LEFT	Cyclic Control	Cyclic Beep Trim LEFT
undefined	Cyclic Control	Cyclic Beep Trim UP
undefined	Cyclic Control	Cyclic Beep Trim DOWN
RUDDER TRIM RIGHT	Collective Control	Collective Beep Trim RIGHT
RUDDER TRIM LEFT	Collective Control	Collective Beep Trim LEFT
RESET RUDDER TRIM	Collective Control	Collective Beep Trim ATT YAW AUTORESET
INCREASE AUTOPILOT N1 REFERENCE	Collective Control	Collective Beep Trim UP
DECREASE AUTOPILOT N1 REFERENCE	Collective Control	Collective Beep Trim DOWN
LANDING LIGHTS UP	Search Light	Steering UP
LANDING LIGHTS DOWN	Search Light	Steering DOWN
LANDING LIGHTS LEFT	Search Light	Steering LEFT
LANDING LIGHTS RIGHT	Search Light	Steering RIGHT
LANDING LIGHTS HOME	Search Light	Steering HOME
TOGGLE WING LIGHTS	Search Light	Light TOGGLE
WING LIGHTS OFF	Search Light	Light OFF
WING LIGHTS ON	Search Light	Light ON
SET CONDITION LEVER	Engine Control Panel (ECP)	Toggle both engines FLIGHT/IDLE
CONDITION LEVER 1 CUT OFF	Engine Control Panel (ECP)	Main 1 OFF
CONDITION LEVER 1 LOW IDLE	Engine Control Panel (ECP)	Main 1 IDLE
CONDITION LEVER 1 HIGH IDLE	Engine Control Panel (ECP)	Main 1 Latch ON
DECREASE CONDITION LEVER 1	Engine Control Panel (ECP)	Main 1 DOWN
INCREASE CONDITION LEVER 1	Engine Control Panel (ECP)	Main 1 UP
CONDITION LEVER 2 CUT OFF	Engine Control Panel (ECP)	Main 2 OFF
CONDITION LEVER 2 LOW IDLE	Engine Control Panel (ECP)	Main 2 IDLE
CONDITION LEVER 2 HIGH IDLE	Engine Control Panel (ECP)	Main 2 Latch ON
DECREASE CONDITION LEVER 2	Engine Control Panel (ECP)	Main 2 DOWN
INCREASE CONDITION LEVER 2	Engine Control Panel (ECP)	Main 2 UP
INCREASE AUTOPILOT REFERENCE VS	Autopilot Control Panel (APCP)	VS Clockwise
DECREASE AUTOPILOT REFERENCE VS	Autopilot Control Panel (APCP)	VS AntiClockwise
INCREASE AUTOPILOT REFERENCE AIRSPEED	Autopilot Control Panel (APCP)	IAS Clockwise
DECREASE AUTOPILOT REFERENCE AIRSPEED	Autopilot Control Panel (APCP)	IAS AntiClockwise
TOGGLE AUTOPILOT RADIO ALTITUDE MODE	Autopilot Control Panel (APCP)	CR.HT TOGGLE
AUTOPILOT RADIO ALTITUDE MODE ON	Autopilot Control Panel (APCP)	CR.HT ON
AUTOPILOT RADIO ALTITUDE MODE OFF	Autopilot Control Panel (APCP)	CR.HT OFF
AUTOPILOT AIRSPEED HOLD	Autopilot Control Panel (APCP)	IAS TOGGLE
AUTOPILOT AIRSPEED HOLD ON	Autopilot Control Panel (APCP)	IAS ON
AUTOPILOT AIRSPEED HOLD OFF	Autopilot Control Panel (APCP)	IAS OFF
TOGGLE AUTOPILOT ALTITUDE HOLD	Autopilot Control Panel (APCP)	ALT TOGGLE
AUTOPILOT ALTITUDE HOLD ON	Autopilot Control Panel (APCP)	ALT ON
AUTOPILOT ALTITUDE HOLD OFF	Autopilot Control Panel (APCP)	ALT OFF

TOGGLE AUTOPILOT HEADING HOLD	Autopilot Control Panel (APCP)	HDG TOGGLE
AUTOPILOT HEADING HOLD ON	Autopilot Control Panel (APCP)	HDG ON
AUTOPILOT HEADING HOLD OFF	Autopilot Control Panel (APCP)	HDG OFF
TOGGLE AUTOPILOT VS HOLD	Autopilot Control Panel (APCP)	VS TOGGLE
AUTOPILOT VS HOLD ON	Autopilot Control Panel (APCP)	VS ON
AUTOPILOT VS HOLD OFF	Autopilot Control Panel (APCP)	VS OFF
SET FUEL TRANSFER AUTO	Overhead Panel	Fuel Transfer Forward ON
SET FUEL TRANSFER OFF	Overhead Panel	Fuel Transfer Forward OFF
SET FUEL TRANSFER FORWARD	Overhead Panel	Fuel Transfer Aft ON
SET FUEL TRANSFER AFT	Overhead Panel	Fuel Transfer Aft OFF
TOGGLE PRIMER 1	Overhead Panel	Fuel Engine 1 Prime ON
TOGGLE PRIMER 2	Overhead Panel	Fuel Engine 1 Prime OFF
TOGGLE PRIMER 3	Overhead Panel	Fuel Engine 2 Prime ON
TOGGLE PRIMER 4	Overhead Panel	Fuel Engine 2 Prime OFF
INCREASE ALTITUDE PRESSURE	Overhead Panel	Master Battery UP
DECREASE ALTITUDE PRESSURE	Overhead Panel	Master Battery DOWN
MAGNETO 3 BOTH	Tablet	Hinge Open/Close
INCREASE MAGNETO 3	Cabin	Cockpit Door Right TOGGLE
DECREASE MAGNETO 3	Cabin	Cockpit Door Left TOGGLE
MAGNETO 3 START	Misc	Master Brightness Increase
SET MAGNETO 3	Misc	Master Brightness Decrease
MAGNETO 2 BOTH	Cyclic Control	Set New Cyclic Center
MAGNETO 2 START	Cyclic Control	Displace Cyclic Center (Force Trim)
MAGNETO 4 START	Autopilot Control Panel (APCP)	A.TRIM TOGGLE
AUTOPILOT NAV1 HOLD	MFDs	MFD2 SoftKey Bottom 1
AUTOPILOT NAV1 HOLD ON	MFDs	MFD2 SoftKey Bottom 1
INCREASE MIXTURE 4	Search Light	Steering UP
DECREASE MIXTURE 4	Search Light	Steering DOWN
INCREASE MIXTURE 3	Search Light	Steering LEFT
DECREASE MIXTURE 3	Search Light	Steering RIGHT
TOGGLE VARIOMETER SWITCH	Engine Control Panel (ECP)	Toggle both engines FLIGHT/IDLE
TOGGLE ENGINE MASTER 1	Engine Control Panel (ECP)	Main 1 UP
TOGGLE ENGINE MASTER 2	Engine Control Panel (ECP)	Main 1 DOWN
TOGGLE ENGINE MASTER 3	Engine Control Panel (ECP)	Main 2 UP
TOGGLE ENGINE MASTER 4	Engine Control Panel (ECP)	Main 2 DOWN
DECREASE EGT 3	Engine Control Panel (ECP)	Main 1 Latch OFF
DECREASE EGT 3	Engine Control Panel (ECP)	Main 1 OFF
INCREASE EGT 3	Engine Control Panel (ECP)	Main 1 Latch OFF
INCREASE EGT 3	Engine Control Panel (ECP)	Main 1 IDLE
SET EGT 3	Engine Control Panel (ECP)	Main 1 Latch ON
SET EGT 3	Engine Control Panel (ECP)	Main 1 FLIGHT
DECREASE EGT 4	Engine Control Panel (ECP)	Main 2 Latch OFF
DECREASE EGT 4	Engine Control Panel (ECP)	Main 2 OFF
INCREASE EGT 4	Engine Control Panel (ECP)	Main 2 Latch OFF
INCREASE EGT 4	Engine Control Panel (ECP)	Main 2 IDLE
SET EGT 4	Engine Control Panel (ECP)	Main 2 Latch ON
SET EGT 4	Engine Control Panel (ECP)	Main 2 FLIGHT
MAGNETO 4 BOTH	Collective Control	GA (Go Around)
TOGGLE ENGINE 3 ANTI ICE	Cyclic Control	AP/GTC
TOGGLE ENGINE 4 ANTI ICE	Cyclic Control	Trim Release (HOLD)
MAGNETO 4 OFF	Cyclic Control	AP/BKUP ON
SET MAGNETO 4	Cyclic Control	AP/BKUP CUT
MAGNETO 4 LEFT	Collective Control	Collective Trim Release (HOLD)
MAGNETO 4 RIGHT	Collective Control	OEI HI/LO (Toggle)
MAGNETO 3 OFF	Cyclic Control	Message List RESET
INCREASE MAGNETO 4	Autopilot Control Panel (APCP)	CR.HT TOGGLE
INCREASE PROPELLER 4 PITCH (SMALL)	Collective Control	Collective Beep Trim UP
DECREASE PROPELLER 4 PITCH (SMALL)	Collective Control	Collective Beep Trim DOWN
MAGNETO 3 RIGHT	Cyclic Control	AP/UM OFF

INCREASE PROPELLER 3 PITCH	Cyclic Control	Cyclic Beep Trim RIGHT
DECREASE PROPELLER 3 PITCH	Cyclic Control	Cyclic Beep Trim LEFT
INCREASE PROPELLER 4 PITCH	Cyclic Control	Cyclic Beep Trim UP
DECREASE PROPELLER 4 PITCH	Cyclic Control	Cyclic Beep Trim DOWN
INCREASE PROPELLER 3 PITCH (SMALL)	Collective Control	Collective Beep Trim RIGHT
DECREASE PROPELLER 3 PITCH (SMALL)	Collective Control	Collective Beep Trim LEFT

Changelog H145

Here are the last changes of the H145.

.500

- FM: Revision that feels about the same but enables autorotation and more NR instability during power changes and large cyclic inputs,
- FM: new SAS strategy for hardware (yes new again),
- Settings: Arcade Mode is deprecated (probably not needed anymore),
- Settings: Pedal ATT mode is implicitly "Legacy" for hardware pedal trim system, and "Normal" for software.,
- H145M gets a Nearby Tank Invasion mission. Destroy the tanks using rockets or guns.,
- H145M fix rockets visible while firing guns,
- H145M fix rockets were guided, they are unguided again (correct),
- H145M guns make an impact VFX again,
- H145M Gun/Rocket imparts are reported to missions via WEAPON_IMPACT_ROCKET/WEAPON_IMPACT_GUN (lat/lon)

.499

This update focuses on firefighting and cargo fixes. Note that multiplayer server is still unavailable.

- FM: Reduce the descent fix from 497 (it was negatively impacting the FLI and this fix is still more descent rate than 496)
- Fix IAS mode not clearing the stale bug
- Fix IAS mode current-trend engagement bug limited to 130kt
- Update POI db
- FM: Roll SAS is enabled again
- Firefighting mission now has temporary permanence for the water drops, preventing new fires (if <60kt and <400ft then droplets are visible on the map for 15mins). blue water drops are visible on the map and these will prevent fire spreading.
- Firefighting smoke VFX adjusted/fixed in wind, fires are placed closer together, adjusted default settings
- Firefighting mission has the portable filling pool option again
- Tweak MFD ground color, IESI and Game HUD update slightly faster
- You can now bleed off water when dipping the bambi bucket (if you are high fuel for example)
- Cargo emergency release binding is available and disconnects the cargo
- Cargo loads are detached automatically when dragging on the ground at high speed
- bambi bucket detaches when dragging on the water at high speed
- fire mission has bambi bucket icon in case you disconnect the object so you can find it again
- Game HUD has status for cargo load hooking/load-on-ground
- Game HUD added to cargo variants
- Fix a bug causing bad flight feeling when picking up cargo objects
- Fix a bug causing bad calculations to hook distance for cargo hooking
- Fix a bug causing low sling load air resistance calculations
- Fix a bug where cargo objects jump around when interacting with the ground

HTAWS DATA

- You must download and install hpg-htaws-data into Community for HTAWS to work
- HTAWS no longer uses the network, so you need to install the terrain database into your Community folder.
- Install the terrain data only once, it does not change with each new build.
- HTAWS Terrain Data Download: https://drive.google.com/file/d/1eMd6cjDGPVaIm4CBajPVXmtnBG-L4k1z/view?usp=sharing (Build 1)

.497

- New flight controls/SAS strategy. More stable, easier to hover.
- H:Events DME toggle events, MKR Toggle event, Training, (L:H145_SDK_IS_TRAINING_MODE)
- Faster cyclic vtrim
- FM: less floating
- Mission: VAR1 negative values fix
- Sling load CG effects are reduced

.496

- Adjustments to the yaw axis
- Avoid acceleration during pedal turns
- Less nose wagging and easier to fly
- Identical to 495.1 patch
- New sensitivity/curves recommendation for x52

.495

- GTC lateral is available up to ~30kt before switching to coordinated flight
- Pilot visibility sped up 4x faster
- easier to pull collective to hover (default attitude setpoints)
- slower vtrims (more predictable hover behaviors)
- rotor speed is slower when in high speed visual state
- Plus changes from 494

Version 3.0 (Build 491)

- Compatible with Sim Update 15
- Performance optimization
- Major flight model update
- Sing load flight model and flexible cable visuals
- Mission platform upgrade, new APIs for multiplayer, road network processing, and much more
- Many bugfixes and enhancements

Changelog User Guide

The changelog is constantly updated with the H145 version history on .https://davux.com/docs/h145/

V2.13 May 2026
V2.12.1 April 2026 add Google Maps at OSM tip
V2.12 April 2026
V2.11.2 March 2026: OSM-tip added
V2.11 February 2026
V2.10.2 February 2026: “manual Installation” added
V2.10.1 January 2026: “night vision” added to tips and tricks
V2.10 December 2025
V2.9.3 December 2025: Setup for Xbox-Controller: not existend setup deleted
Reorganized the menustructure and deleted the second Lvar-list
Add to Troubleshooting
V2.9.2 November 2025: Tips and tricks added
V2.9.1 October 2025: Add recommend for non spring cyclic in tablet setup (V2) (“ignore” at “Hands on”)
V2.9 July 2025
V2.8.3 June 2025 Add Historical
V2.8.2 June 2025 Add Specifications
V2.8.1 June 2025 Add “common mistakes with MSFSLayoutGenerator” to the installation chapter and “how to find community folder”
V2.8 Mai 2025
V2.7.1 Add SU2 workaround to Installation; add “copy&install” to Installation; add “TDS GTNXi” to “known issues”; add “how to make an approach” to “Tipps and Tricks”
V2.7 April 2025
01.04.25 Add “Trim release” to Tipps and Tricks
22.03.25 Add MSFS 2024 Settings and add Installation and Known Issues for MSFS 2024
06.01.25 Correct older settings
30.10.24 Error correction page 65 “Collective… change VS (not heading) reference”
V2.6 October 2024
29.10.24 New VMS Main Page and add CTRL Page
09.10.24 add “Correct setting of Trim Release” to Tipps&Tricks
07.10.24 add “How are helicopter buttons configured?”-Chapter
29.09.24 add “How to find the community folder” in troubleshooting
02.09.24 add changelog
V 2.5 August 2024
27.08.24 change tablet picture and info
21.08.24 add Action Pack Dokumentation; change AP-Installation text
09.08.24 add additional information for installation of test build
08.08.24 add User Guide version and download link at first page
V 2.4 August 2024
07.08.24 add pedal and cyclic settings in Tipps&Tricks
06.08.24 add test- and HTAWS local installation
02.08.24 remove WTT-Mode entry (Flightmanagement)
18.07.24 Add more H-Event
V 2.3 July 2024
18.07.24 Add new H-Event; add graphic for new HTAWS
03.07.24 Add Link to EFB Connect
02.07.24 Start for “Tipps & Tricks”
V 2.2 Mai 2024
13.05.24 VMS Subformat changed
18.05.24 Add “PMS-50” “Enter transponder code”
28.04.24 Add “Override” to “Controls & Bindings”, “Trim Release”
V 2.1 April 2024
06.04.24 Add “Recommended MSFS Settings”, “MSFS Cockpit Interaction System” and “H145 in Multiplayer” from the old PDF-User Guide
03.04.24 Add Default Function-Table
V 2.0 March 2024
18.03.24 Added Missions (Cargo and HEMS) from Mission-Website
17.03.24 Added Mission (Firefighing) from Mission Website;
Added Pedal-Settings for Xbox controller (page 11; Source: pinned msg)