HPG Mission System Guide

This is based on a copy of HYPE Online Documentation.

For this Document I used:

- Mission System
- H145 H:Var (from H145 User Guide V 2.1.5)
- H145 Mission System Documentation

If you find something wrong, feel free to inform me or Dave (@davux).
For help please ask the community (creators_chat on HYPE Discord for example) and not me. I'm not a creator, I just worked on this document :).

God luck and have fun
D-VRGL

Content

Introduction & Overview

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

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.

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.

Each of the possible commands are listed in the COMMANDS page. 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}
}
```

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"}
]}}
```

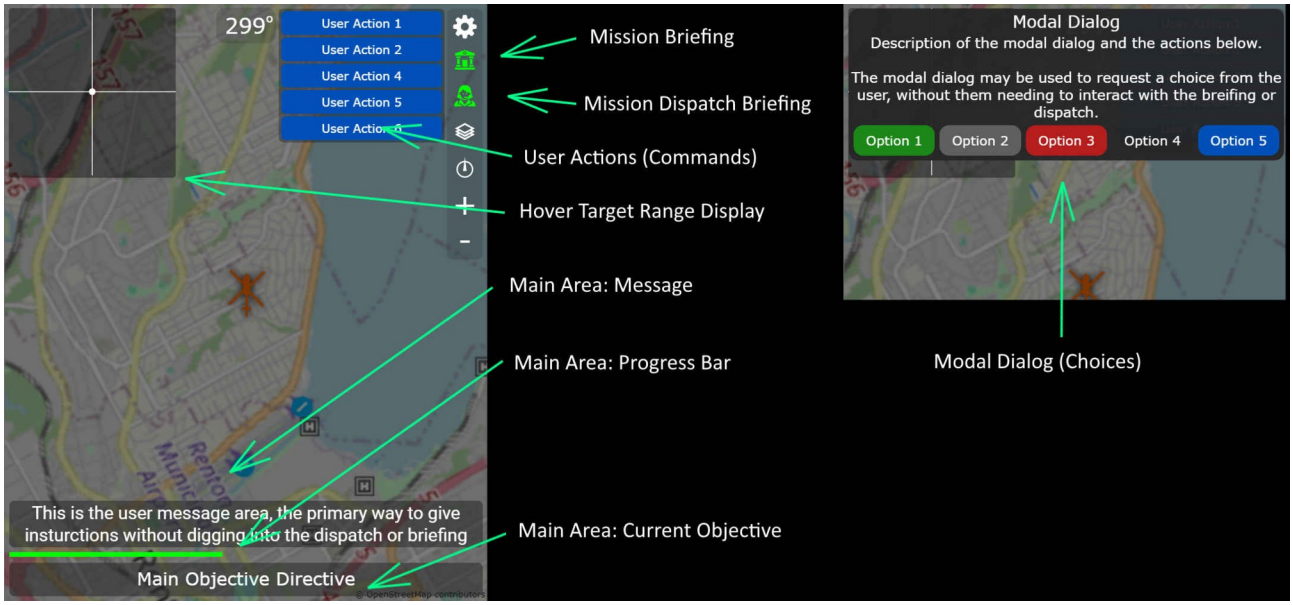
And you'll see the result is unchanged as the table was loaded from disk.
We can use this persistance to save settings, logs of the mission or whatever else we want to. Try to avoid calling `save_table` too rapidly as it is hitting the disk.

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":
"
GQAAAbCSURBVFhH5Vl9bFNVFL+v3x/rttZ2a90G2dhWBqxsYf0g7IvNMI0xRhPm0D9IMHwaE4mJaICYgKiQoJEMEoMhUQOiMZE/JSYOBQywoWMYIpuyBEG7ZRMGW9d1a6/
n3L731te+0q5sqPgLP86955x77n3n3Xc/
OpIOaMvTVmq323fydUIUCoWLL4pQ8BJBkTU17sejVQC2uEHdtLHFQsEqNYACW6QBq9V6ck97CYsf1cSB47i32Ejy8+1HQIieTOkf8JO6x8wERoJVKcCNYnO+mgDZTjleIhKM
ceDEFMFDLx22O9aGNBoSy75ypwOGsIf5ME8eiyht1IRCJJZrsrOrePM0hHTK8V4Plz7MZvMFENRmsx2IapIDeiX0yq1lTPKUIj8//
zMQsQ4SFhQUbAIZzUZOTs4tEJhzWdbV1f0CUpo6yCuNJ29ikDgLuW32+t4XyrGQOKeCxDnYlE0CDh3punh2K/Yf0eCQpyFx1n57h+j/
DJJAIMzqXEgyZKnzebenImizBSEeCynIGcPnW+6GORL2er05vCpz1NbWLoBg4ovc+Op8saxUKic8Ho8ayumhqakJvwsxgMD2Y2UUV5WGVZYEm0qlGq2urpakSsyL07nwxvDw
UK7BYBjTarWTvDptDA0NZefm5gZNpmx/T88l6WJYXl5+SG6FTAf4jba1tTXy1eQTLqRQuOE5W1NxTKW2800kSBq4yJa3GfL0eSzr6+p98bqGqqrE5QeQNLB/
cADXIJERnZJ813l2K8pYfefFLr6FFEkD4+pGYUhhAwQCr/FWs0i0oV5cBWWQNDDEXNdVsbhKOTZFxi1WYvxkiBOIdtRDAcZMvmYN/j8wmUyXYG3+lK/
eP9xu92IQ+IYYYY0Ib9q0+f4WItgwOkCwgLs+KKYGo0rsoKio6D2QMwOsakXxCxEuQMhYnUajCe3YsVMH5dSwWCxfgpAEQLLAEWlggcXFxW+AlEdDQ6MFFiDZhulQp9MFYPT
i+swm+
+rVrdu6u396ua+vr6CsrOwms8wQQttVq1raDh5sP8OrRWDvmULSNvlacZ9gqYgB9sp0kxrNBFQ0WE4G2JMGtYFAvlAFivGSBoZCK8pUAOcv+KIkcDzQKADLIo1Go6QeQwGx5
bnL8YN/ebDtiI9GNdC/4BmmhJuafmqO4j7DILZFJA8M/27CZcPy7K/kwycdxJwHAcHiI1piHb1LQvUmYi/
oRmchRvqBw0YVUQTDsM9ZCFFHXZXXJoi6cwwKHFHA6RG0QgxJ4HigEf87iYHT5IusJZR5KQuJMQyLEjbuXr9BxavI9caVz6NuxGi8yqsESNqmmhXMWWVhB0EGlUYzjhLWayY
fDni9PnVzc7N4QPxXYu/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

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.

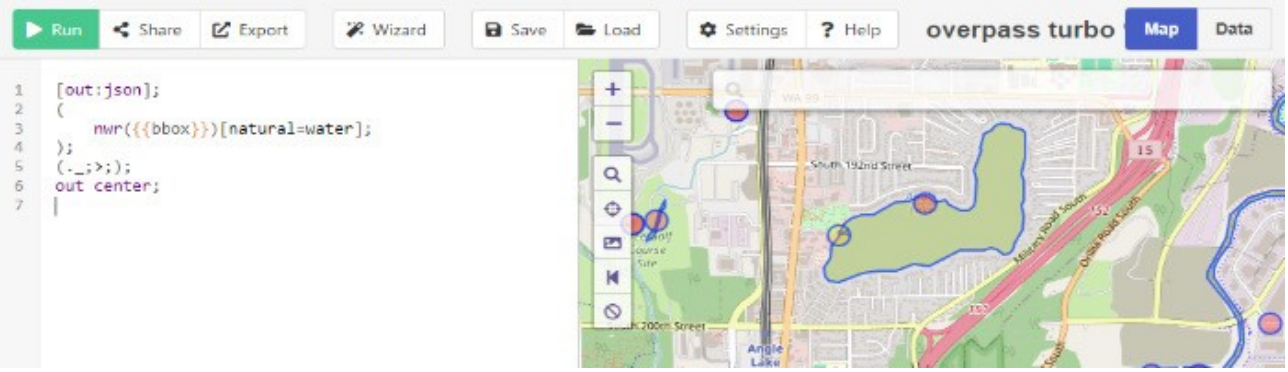
Messages:

- `{"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)

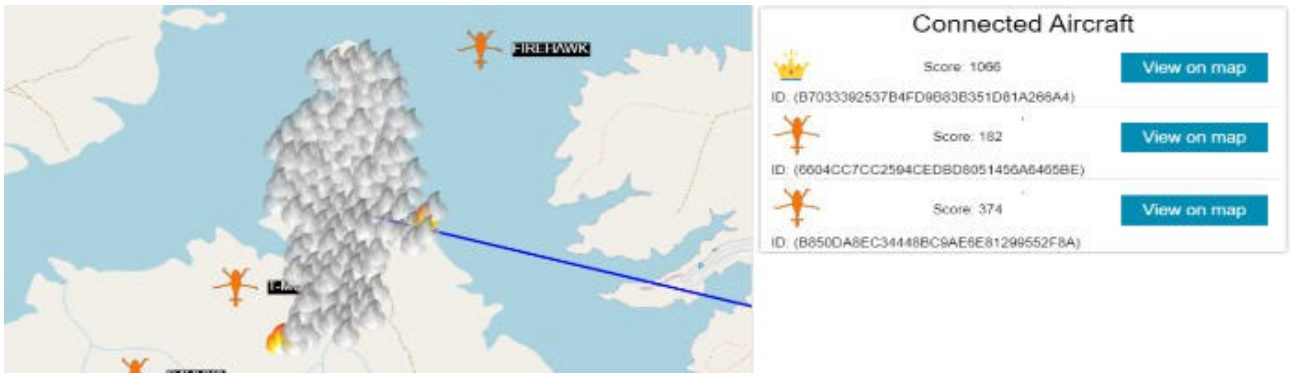
Hoist SDK Events

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.

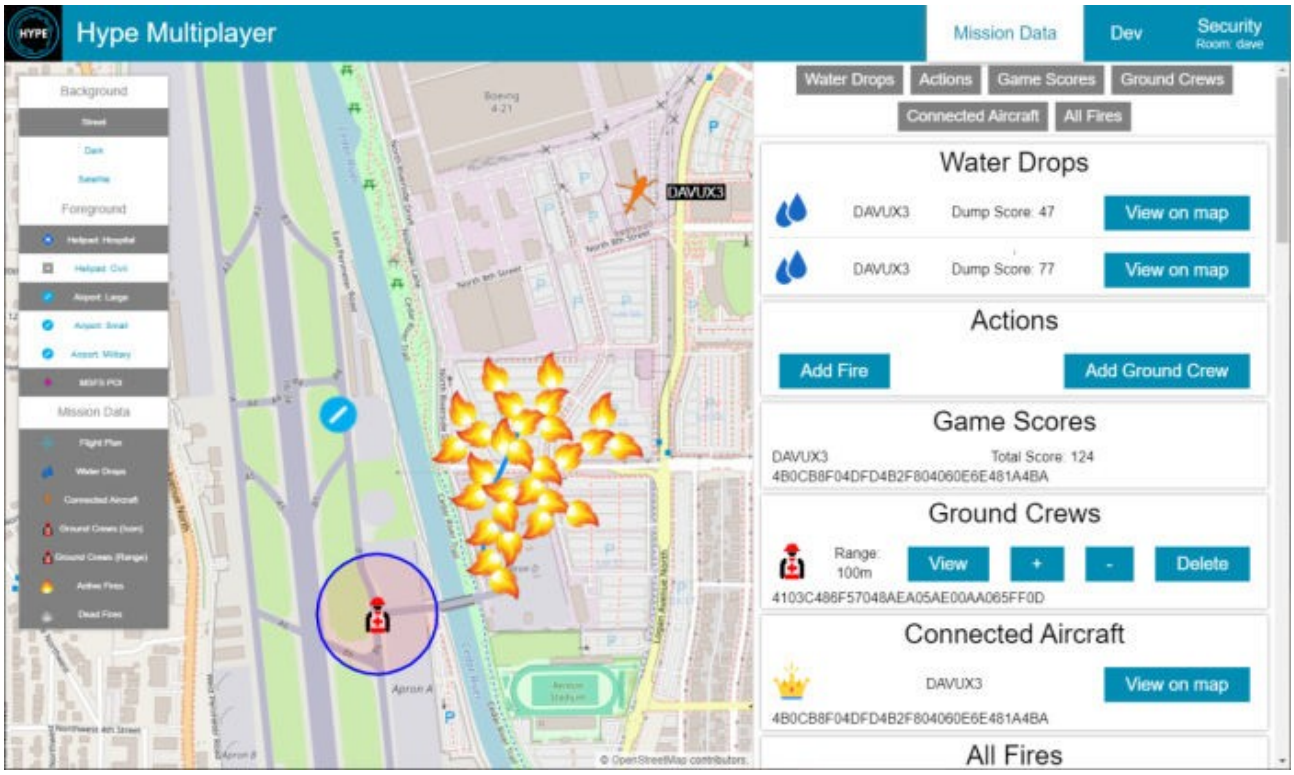
```
{
  "title": "Multiplayer Score Mission Test Program",
  "author":"davux3",
  "api_version": 0.1,
  "aircraft": ["H145"],
  "data":{
    "server_url": "wss://5ed547d.online-server.cloud/mpserver/ws",
    "create_room_url": "https://davux.com/dispatcher/",
    "webConfig": {
      "fligihtPlans": {
        "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":"
sCAEiq+TwxFXJxE2nYpJmQ6TuuCV8PS4OptfE6vvlKZC8SX892zqjF0N2g7Fl4Dbh+SLNChrQxRgIgOodICQT8gVATDCCIFpXenVAxntmLrMnj8t5wCuPR8qHfS4cJm04kmrhZl6M6hFw2JV22S6c/TYCXPAbIUaFUDrTsU/9SD1PqN8XA0eu8sj+wUGWlrX1onJf1Qe6e3YuYJzJkT+d7Lqwi55aTJs9Z+195cvd3ux60KiH/
mA7AbVPwbF5ASTGZ1uxq1DsdBLsdPIdK0hbtL5/AI0KmW8GpN/Dz67MpkU+sqGDRuYmlinHVTe0Nf3VyhRBPKMjy1dugRqVIA/
7qqUinSoUQDhvzSpRj7UsNoQMOH5xWh0Qx2bnj1pb7z0hnPmzAG74xMmAkBRrdkeWxBPnL9Kq64kE/pMmpg07cCB/
XreAYHDyTRNCY56FAhymlQ9CnEYiZoVDZkgV9glj1PViJ2LSp0JcyGdLliSVJLj6uqq8Qi1e3+InWFDubzjHpKejt61N8jUtqa+vfMPwaxOPy7BPFNqby9ZsliLaqfkCf6YO
Kd+uzbU3f0HNpnWF8LicjcTZ3NBo4xMoILjsZnfd3n0voI8g+OLBmjkVExoCKWXo3kpcVug8GI0vuHesqJDUOw7xg0odSWsDa73K0sorQ2UkyYna1f477e79RevDrWRL/
Qe21oIJINMKLn3K/Q96jl841kgcba2F0kkFcZkGr3IQvHiFWNHws/YEJ708vLiwRyagBvlkVx8Wym7pgtLSZkOhJy+NEVZ/0jy4UBLF/
RRoWJQhhyOfri53wycOXMmOHLhyjif+OdReqgANwyPeXBjnMtFzfASBYSBxJHfhDlU8dv2Xc/
w2eLtbQpzNYbsCpbfq8sOH8QeLq0m6nUvDVuvAcxpkVRNrS0PKUmK2Ofn54/CHI2YPXs2cF22ZlS6sfOu/W/pTGst1cNTWYrM7jBbgLlDu9z/
XqGbqq9auXIFa8wn33yebjl0y3EH3ihxORdPVRNNs1qByWCxfFM6/9+M4kT/tcuWZsEctXh4/KjrMuuf824aWG88aclxltUQ91XTRbKawEAbuXRjZvVx/
ap032WL5pfAnFdn7ty5IPzi1ZFEURqJhF3HVzxtddyT5K1badDdvFUpaiHrGyToZsmZznq/ExwcpEsY9P33yxVBbUYz3kDa6WMgCW7g/
LS6IqXscackkZTpArOkjuxWDkT88Q9aFa+SQ5cTFhKGcWtF9BpFukAYhh+2NSIxTOkzlIVbChRTlD9FnHYiBboRHJXpT0bVFKLZxIuG7LsY9e0QC4PHaWkUTjhRCl7/
vknnV0CNhuAGIvmNlBYxZY0k+HQkuTaFiagdeikzUGQ52hOK9ISHoh1TXRRCmQfR3AEIYCr658CDoW+TAgVQ2FGzgZVdGxkTyhwnbiGTaAVetIGxT8KIs5mAT9nkECUeDAk5
QL5cYHkJvwIgAfWlXApLhgJCo9JUc5y6tXg/SGFDoeRJE81sNkARLM0X3oMiJVBi4DkznSHK1YphZY/IBapFixZSO3PWJ/AIhlks6ChWvIjDLt/
8O9T6BEWVGQGLq9ueQoUkw2qoNxT7BGWtbXxd4XYokhyz446BYv/
iuyvAvEerfdCCt2QZZS1ZKzZs3WJ8MCpjMxH3IfnEpGXP8ddMIJIp0iWYTjUf80gpTwqPjJ305ZdfaN1w1P4D/
gKOTJj13fR7pk47jjrofCAu18dTewm+DRrAd8KWwlNjrR1hCrh7967JzBHv1QsL2AjAelmg0m0G1ubtkg0ZVYebcmJ3b/Zcp37C83TU/
bkWfL4UPCaKBXqj+5EuxohoJSCh4vLZKzdGEVNw2kLM0YRcz9jmUCVo67yXoechx1ilKLbsXu0T4v8QxZpmz6c0G1Il+yGTf4zz9Ib35tVrKFpA7cDNzY0x/
mu3WUn69rtO8njOQNIz0PXPPDZWVcT7jxy3YQ6ZtG7He8bu1RVslrp/IDpjpmULWFTdnje9uc437drRiMDAPS/cvrQv/
NQ77dbjNu10MZ8jKOYwgJSYfuuOHLAHNoMl5a2PR1Vm785Luno+KChYQuS8tNIuXLiA8eGseVMLbFy2HHTUmdxaQvxAd4/
jheDCUN1rfdLzmADu4E+g2gG3EdiYiyWbHtYEiAuS93i4r9R8Bf5V2Lhxg+6PUekJZN3pBKtcMYl+6reYMfPmzdW65VKKt/d2BmGM986dHfPPRB+Y09w//
aBacAPDbhbugBqp29QIKJuf6Tu4QQ7VaBvUSH3P9UcHoNYnKBqLW0FxLRhASEFBe4leHaDZMbuIMy0wFzQphjYcvDP2UspUQIkHf35QFwFFcMvSdiUUKYESAxmFtwKgCCL1W
QOhSDPwYj0SGTOTGEsZxT33u74qlAWsBA8HjcSjaAQsqBMXwqQ+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+/
mWhNGuksCKiKSchwNPTU+2gcOHCBWalwfA9+cYWE38bwhzJlzBZEgHuhFai5mgxKDPaAWLQAozMOfLpBU2VH6CSVG5LZRq3qSouNzOlMD+/
oOny5cvw4hejlQNfhXXr1jIdh71jA0ztJon1bCY85OlMjnbiOfLbmExJLf7zMqK2dzz8AlBXdMbFYghUu2BV3yCpLjLs9HUvMVi0AIapGJgaMGSf5jcXjUbFydzmsjsyflFy/
XlZGMT5Y0LI7oZSAvUFWxM5b4i0DKVS/d2X79m0DwEPl7l0Z5hsSMhJmvYGAXJGHr2pHL0Z/BpNVHLz654/
KfE6FUK2T+wPavMq55TWkQBE8GPQ2+eFEZ+5YOayFIlhW2vYAiv0ObRw4rDxruyIWA+DYYN57pNCJ82Zs1brweP7zbVDsd2jjwIcJ1+IRY8UCkqTCABw4eVbVxx08EzkB4ys/
2fwEjFKRGn72uUN7QhW/
vV5co+7mdsXmXYDLYFV9yRZk+tBxtgslv6E5oVLq70lE8tE4182wk6vh2NTQum1Yrc7RpwgTysvQjgK2oYGi+AeLj729xKOy4aUM+SxGvMltBQfL5UFKmCa89kNYW69pGcUUJj/wi5ZAwdxc+rMhWGf/lX4Ru5MyXNxia0W53Aq3Yf6PwgLK5OlSj7fbVgnal7hebdwFeRhto1YQJ5AW3/
ACQknJxDYNTgh+kAuSA9SyRsv3hVEG7JkygUyPA2kph2AJhDRICqbUJ7eylXQ0kWF6ApkJRxdKytiwo0gpaOnBE5WPVW4mScd0+nqYLtHRg4u9n4xBjEdRweI0g8VKYalnzDRrwzf3KMuXo61TR2AKTaQdtY6oJDDR4gbU1b0ZLxR2nqyemJCYmwhw6AcB/AID93zD+9+L6AAAAAElFTkSuQmCC",
    "h160_icon":
"
39m7ut99713719viyb8kpNZ7r0zc2bOzJyZK25kmVACa6QjJ0Mu7T4u/
ZH8Vefa6MALFJOUTBOSOSYkXU0YRp4J04Jp6Y06FdpaiwMm7Hxo6XeyXFpVhhN7B5BIxQZxfRiJ1SYMoaVSAgi66VQUTLsWMGxdLR1tEP2kkwwThmFXKmN9+HtKj6cnyQfmxeAI4+XHGQZC6xoKWQKZpVLRcHalnag9Z31kwr6Me8atxqQx0YROBseqPQeiexMvWi8Hngz/
yNk5tNTXIaW2gbsjc0OUcXTQDjmio4prTBhCtOFoMCGpN2EIzib0gF3WtsGc87JNDoAV3YTgO53qJMN0wqM6qWGeoydZyfT2CB0VAnsy1hBEY9qhxdJs4r7oAZmmo+GEjQ9x
NjHSfHRjXHFPecHEfcFainQ0PvqYMClEXQAMj0Ce0NEQhpv+W6GTGpMX3AviJZtGtnaifGjSQ1mwJUhzzRNjiHcx7oafUS1ABfU1TSK9upgcG7AQLoP/
6FR0fJkRyMI61Ob8yGtlJNYYOimO0sLbTRgTq0LuTwN0VHIhSzk+XFUR5xj14wNw1IROPoeMzM6UtdbYNq5UW3AF5DYI/
aO4GWAVSkF6rs52Jd+8GzcPsYBIPkGqGGJafL9Oesev0dDXmAKx9i3aENPjVCqFWPv5fTrpHb8aWljf2XbOK3ijN8Tv8hc/xjDoz3li6kB5jt/
UrIjPycg0FXbXyU444c2znjpHMc0sBAmdx2aZgi2XgVa5jHmUJSXyMdJqDTbvjWI8UaaawpSrYTl0liCLh9kSE4853l7m4TfY72aiwPOI52WEF8lF4BDeGYE4K3XF68Tfm7l
eKpoqBft9GFt/vyhtCH/TSXd8mfylCgl0yw3/BtrR6fBkmb6WtoJX5BYTDYLKXkLgeRr40pAH6ZysiBp6LsePhnmwCF8NjIRbhXw2/KmJ8r6ZAk1ln8ke/
SgUc4pcDBkMcW2U9ZAb61+QFkgxCqim+Z9vkPaiLeq4dhDCk+cMVBxSaV2zdPR8Vc3RKY2r5JMu2brMfSflxOTtsgZRHjh/
gag7FqvCYShImXUbzqQ5L0sJonzJSQ7eY6OCYPyGIXB6cwUQllFYOV7KNt4q9+I9LhABp/pMu07eBrh/XXM6uw0FsetjTfgFEF6j/
aBSPhltxrSjd56sM3me8TUPDcGuvtgs75qoZ+KpkDcPFp7OE3biqdB+TWW/
ufBEXBWerFeLNYllLAnBzZdzko7vZjPZGWcefR9POKeFG6WwzF0mHgLmbgfmbly95QmbT+OnwYq4WpXV+ZVnj84irgrbedTURL6DTgGzTz+WvsqukCr6nChXXuODkFQdE7mF
8jRRDBkDudrkpYX5nPrfPiDViHs5vrBhoyG8Qr4b4rYY9ji4UBpZPm/wds1Vvshk5qunaWKC7fK/DEInJhL5ECqzju/aBXkzzHO7BRTsmS/nzPNyCK/
F4sb3wuuADZuObWo3XcrT2CMHvqHczT/VU13+2OZKOZAbxbDggC1FsB1yGUJLKIdyVRfgnhZukUKkayFxk6iCFlz/
x2PH2odNJAONfhhpNroRwmPllwgjcqxGAte+qc11R6nsmz9SRuD7Z5DcDKljfiIkS0ELmtM9UGhHBwwsc4PMLBslS96aLYv043Bohzjmljbhg MB/
QVBuHrI+glgeU0IkW0GLvgcWyNEb+ofcp8QEyvGv4H6dSg6pUpCw7KmYn19HuPEIobFVOvKr1N3rcZ2TPFLnQooUtVTKV7GUI+bUN1ankksqFeQK68dCwq4Ok0GyFKTbzZ9O
EyCLzi2XttbVcirHn88xCMLfNgnte0789DBHhPehvKkoaVgpm2BawQ7iwXT/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.

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

Operator	Function
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` 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},
```

`ebug_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`:

```
{"struct": {"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":"
RnQU1BAACxjwv8YQUAAAAJcEhZcwAADsIAAA7CARUoSoAAAA6GSURBVGhD7ZsJWBRHFoCr55IBhvtSTg3gGjWJGrJqvOIaY7Kaj

```
Uk0EjUe6EYNJqioWRUVQxBEFFS8QPF2UYzRKKeigEIUEEVQDpH7ZmYYmnOu3u6emuEadEbapXc//
+9r+r2qpuf1q+t1VTVQR1BE9GIovoQ0DIOSCgY8d2G/
sCAEiq+TwxFXJxE2nYpJmQ6TuuCV8PS4OptfE6vvlKZC8SX892zqjF0N2g7Fl4Dbh+SLNChrQxRgIgOodICQT8gVATDCCIFpXen
VAxntmLrMnj8t5wCuPR8qHfS4cJm04kmrhZl6M6hFw2JV22S6c/TYCXPAbIUaFUDrTsU/
9SD1PqN8XA0eu8sj+wUGWlrX1onJf1Qe6e3YuYJzJkT+d7Lqwi55aTJs9Z+195cvd3ux60KiH/
mA7AbVPwbF5ASTGZ1uxq1DsdBLsdPIdK0hbtL5/AI0KmW8GpN/Dz67MpkU+sqGDRuYmlinHVTe0Nf3VyhRBPKMjy1dugRqVIA/
7qqUinSoUQDhvzSpRj7UsNoQMOH5xWh0Qx2bnj1pb7z0hnPmzAG74xMmAkBRrdkeWxBPnL9Kq64kE/pMmpg07cCB/
XreAYHDyTRNCY56FAhymlQ9CnEYiZoVDZkgV9glj1PViJ2LSp0JcyGdLliSVJLj6uqq8Qi1e3+InWFDubzjHpKejt61N8jUtqa+
vfMPwaxOPy7BPFNqby9ZsliLaqfkCf6YOKd+uzbU3f0HNpnWF8LicjcTZ3NBo4xMoILjsZnfd3n0voI8g+OLBmjkVExoCKWXo3k
pcVug8GI0vuHesqJDUOw7xg0odSWsDa73K0sorQ2UkyYna1f477e79RevDrWRL/Qe21oIJINMKLn3K/
Q96jl841kgcba2F0kkFcZkGr3IQvHiFWNHws/YEJ708vLiwRyagBvlkVx8Wym7pgtLSZkOhJy+NEVZ/0jy4UBLF/
RRoWJQhhyOfri53wycOXMmOHLhyjif+OdReqgANwyPeXBjnMtFzfASBYSBxJHfhDlU8dv2Xc/
w2eLtbQpzNYbsCpbfq8sOH8QeLq0m6nUvDVuvAcxpkVRNrS0PKUmK2Ofn54/CHI2YPXs2cF22ZlS6sfOu/W/
pTGst1cNTWYrM7jBbgLlDu9z/XqGbqq9auXIFa8wn33yebjl0y3EH3ihxORdPVRNNs1qByWCxfFM6/9+M4kT/tcuWZsEctXh4/
KjrMuuf824aWG88aclxltUQ91XTRbKawEAbuXRjZvVx/
ap032WL5pfAnFdn7ty5IPzi1ZFEURqJhF3HVzxtddyT5K1badDdvFUpaiHrGyToZsmZznq/
ExwcpEsY9P33yxVBbUYz3kDa6WMgCW7g/
LS6IqXscackkZTpArOkjuxWDkT88Q9aFa+SQ5cTFhKGcWtF9BpFukAYhh+2NSIxTOkzlIVbChRTlD9FnHYiBboRHJXpT0bVFKLZ
xIuG7LsY9e0QC4PHaWkUTjhRCl7/
vknnV0CNhuAGIvmNlBYxZY0k+HQkuTaFiagdeikzUGQ52hOK9ISHoh1TXRRCmQfR3AEIYCr658CDoW+TAgVQ2FGzgZVdGxkTyhw
nbiGTaAVetIGxT8KIs5mAT9nkECUeDAk5QL5cYHkJvwIgAfWlXApLhgJCo9JUc5y6tXg/
SGFDoeRJE81sNkARLM0X3oMiJVBi4DkznSHK1YphZY/IBapFixZSO3PWJ/AIhlks6ChWvIjDLt/
8O9T6BEWVGQGLq9ueQoUkw2qoNxT7BGWtbXxd4XYokhyz446BYv/
iuyvAvEerfdCCt2QZZS1ZKzZs3WJ8MCpjMxH3IfnEpGXP8ddMIJIp0iWYTjUf80gpTwqPjJ305ZdfaN1w1P4D/
gKOTJj13fR7pk47jjrofCAu18dTewm+DRrAd8KWwlNjrR1hCrh7967JzBHv1QsL2AjAelmg0m0G1ubtkg0ZVYebcmJ3b/
Zcp37C83TU/
bkWfL4UPCaKBXqj+5EuxohoJSCh4vLZKzdGEVNw2kLM0YRcz9jmUCVo67yXoechx1ilKLbsXu0T4v8QxZpmz6c0G1Il+yGTf4zz
9Ib35tVrKFpA7cDNzY0x/mu3WUn69rtO8njOQNIz0PXPPDZWVcT7jxy3YQ6ZtG7He8bu1RVslrp/
IDpjpmULWFTdnje9uc437drRiMDAPS/cvrQv/
NQ77dbjNu10MZ8jKOYwgJSYfuuOHLAHNoMl5a2PR1Vm785Luno+KChYQuS8tNIuXLiA8eGseVMLbFy2HHTUmdxaQvxAd4/
jheDCUN1rfdLzmADu4E+g2gG3EdiYiyWbHtYEiAuS93i4r9R8Bf5V2Lhxg+6PUekJZN3pBKtcMYl+6reYMfPmzdW65VKKt/
d2BmGM986dHfPPRB+Y09w//
aBacAPDbhbugBqp29QIKJuf6Tu4QQ7VaBvUSH3P9UcHoNYnKBqLW0FxLRhASEFBe4leHaDZMbuIMy0wFzQphjYcvDP2UspUQIkH
f35QFwFFcMvSdiUUKYESAxmFtwKgCCL1WQOhSDPwYj0SGTOTGEsZxT33u74qlAWsBA8HjcSjaAQsqBMXwqQ+Q6mBYXbc0cT5b4K
ynWQCBVBnILcBSKoUWwFybp4/TQp0YjAR5xHdC4VdDAFlHtyYXeMHRUqhzMDm3IQghUSjIbgHePFyazu9wFMApa2YCFx/
zq46ApX/b8JiszzJuBLv+IOvP6BP4PG/gpkAhe/+GEZuLqEpFDcRajhxIlyv/jlTZVtzri7yS0CAPVRpBS0dKLb+wANgnd/
uBwDDkZ/+BBVa0b8v1L3wdXpddSRmZglVEnvbRnGJlSH5UkYnaFcD16/3ZETqsro4j6CknsnZtm0rpZsDqIB2DnQe/
9lXoNmIlJnmDYA9UEjKoFUPDBn/xQKFQh9o58D7Vs5eUARLasRPFpW25EAVJBnb/
wxF2kC7PpBdzsckVeTHhiC8OmVeu1TMXmEzhXw7ZFo0Apm9IS37bVqwL/zkMEXwjMd/
eYptNytXrmB2Xq04dObiOPJimkCrJozYTV2vNMlRX0ZuoDt06LDMwRRTLXAILEbTcJsGTbDgN0mVNe1YfA7uTAUhMdk7lek8tIF
WbyW0qYG+vr/q1j5HFGEKqxUUJZ5VrQxgxSmBACgWDtBcHcTHz4+mM7f9yNGYLMVXHfhhVS8i1w47YyhqVO1X23s9U/
EFOg2gTQ2MM7VyhyLYlFlzFIoqvNIbf4ci2PO+/
Qoo9ju0cODatWuQSD2mlUKTgNbs6B7TV4Z1GXiaYq6xrJ7N8fLaTAvb+zWmIhzHM7ewFQ+f7L1z4LjFRJqObT1oszJXaxdSyMcw
gSJG3M5/cgbLvuYtF6GFv/
ziQ+ksrjZQ5sBVq1bqOI4YPcTsLReXKjlnxlNT89FXnVmOogYGQ8ZnASDv5dOlbqxvK44LmKhmewPOoj8r8k8yB2nwWQDuT0YbY
BhLgKGJTP5FASh05tdk2bGkf/CLMtIKsx88Dw7e17Gc3QfUPtGmTf/iDX7nr06Igd3kWibvo8dmJu/+4YzYttRxELkAdwbGwa/
Sxvd45IGPrBxLBJhwJNLZheKcEY2NtzlNZYmYsDg1NTmxLjz8hFa7XBcvXsQa/
8kMc7a+zdhGHYuPc3hGE644cd4WtLKZkjrcNnKzizY2EpVYAhAjMdAxaAO2Ek7L5yVo6lBUGAdERXefP01/+iD1riA29qbicghy
LOLKGDenGelARjhFE/AfYrYBpqkEmOkjspkFzQXvos1Juq1Vt9lNtcl3bkXXhIaG9RhF6YC7+w/sSVM/
G9jAMpzWqmszIUOfMynamecgQAGTbCUyYrZMQ6czpMA/
7cT7yPz5ruBD1zXrE6wGr40dOsASrZAhANXFr9DCoYgYICZSoGcuxT7PR8pGCAVZlrLmZNBcmpz/
MCXff4fP691N1AvEd6POI8cPbdO1/
CjXwHTadTtDl1JOu16baADAGoidUsShaS3F64RuC9C1YAJLqbh9TW6tT3X8SR9t6jjw8PhJ12XKDNsmlunHrVzLsfd47ImxTvo2
wnoJA2vEnS7XgVdqAt5iue2AbYLXZLZM+mWhNGuksCKiKSchwNPTU+2gcOHCBWalwfA9+cYWE38bwhzJlzBZEgHuhFai5mgxKDP
aAWLQAozMOfLpBU2VH6CSVG5LZRq3qSouNzOlMD+/
oOny5cvw4hejlQNfhXXr1jIdh71jA0ztJon1bCY85OlMjnbiOfLbmExJLf7zMqK2dzz8AlBXdMbFYghUu2BV3yCpLjLs9HUvMVi
0AIapGJgaMGSf5jcXjUbFydzmsjsyflFyXlZGMT5Y0LI7oZSAvUFWxM5b4i0DKVS/
d2X79m0DwEPl7l0Z5hsSMhJmvYGAXJGHr2pHL0Z/BpNVHLz654/
KfE6FUK2T+wPavMq55TWkQBE8GPQ2+eFEZ+5YOayFIlhW2vYAiv0ObRw4rDxruyIWA+DYYN57pNCJ82Zs1brweP7zbVDsd2jjwI
cJ1+IRY8UCkqTCABw4eVbVxx08EzkB4ys2fwEjFKRGn72uUN7QhW/
vV5co+7mdsXmXYDLYFV9yRZk+tBxtgslv6E5oVLq70lE8tE4182wk6vh2NTQum1Yrc7RpwgTysvQjgK2oYGi+AeLj729xKOy4aU
M+SxGvMltBQfL5UFKmCa89kNYW69pGcUUJj/wi5ZAwdxc+rMhWGf/
lX4Ru5MyXNxia0W53Aq3Yf6PwgLK5OlSj7fbVgnal7hebdwFeRhto1YQJ5AW3/
ACQknJxDYNTgh+kAuSA9SyRsv3hVEG7JkygUyPA2kph2AJhDRICqbUJ7eylXQ0kWF6ApkJRxdKytiwo0gpaOnBE5WPVW4mScd0+
nqYLtHRg4u9n4xBjEdRweI0g8VKYalnzDRrwzf3KMuXo61TR2AKTaQdtY6oJDDR4gbU1b0ZLxR2nqyemJCYmwhw6AcB/
AID93zD+9+L6AAAAAElFTkSuQmCC",
    "h160_icon":
"
jwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAd+SURBVFhH7ZlpbFVFFMdPdyhlbUG2KDbIYlmUJSCLoEYU06AYIdHIZhUriIU
iglFjTEz8AEhViBLlgzHBoBKNGPWDQdAYRYmgAdGQKsomINAFuvfV/
39m7ut99713371viyb8kpNZ7r0zc2bOzJyZK25kmVACa6QjJ0Mu7T4u/
ZH8Vefa6MALFJOUTBOSOSYkXU0YRp4J04Jp6Y06FdpaiwMm7Hxo6XeyXFpVhhN7B5BIxQZxfRiJ1SYMoaVSAgi66VQUTLsWMGxd
LR1tEP2kkwwThmFXKmN9+HtKj6cnyQfmxeAI4+XHGQZC6xoKWQKZpVLRcHalnag9Z31kwr6Me8atxqQx0YROBseqPQeiexMvWi8
Hngz/
```

```
yNk5tNTXIaW2gbsjc0OUcXTQDjmio4prTBhCtOFoMCGpN2EIzib0gF3WtsGc87JNDoAV3YTgO53qJMN0wqM6qWGeoydZyfT2CB0
VAnsy1hBEY9qhxdJs4r7oAZmmo+GEjQ9xNjHSfHRjXHFPecHEfcFainQ0PvqYMClEXQAMj0Ce0NEQhpv+W6GTGpMX3AviJZtGtn
aifGjSQ1mwJUhzzRNjiHcx7oafUS1ABfU1TSK9upgcG7AQLoP/
6FR0fJkRyMI61Ob8yGtlJNYYOimO0sLbTRgTq0LuTwN0VHIhSzk+XFUR5xj14wNw1IROPoeMzM6UtdbYNq5UW3AF5DYI/
aO4GWAVSkF6rs52Jd+8GzcPsYBIPkGqGGJafL9Oesev0dDXmAKx9i3aENPjVCqFWPv5fTrpHb8aWljf2XbOK3ijN8Tv8hc/
xjDoz3li6kB5jt/
UrIjPycg0FXbXyU444c2znjpHMc0sBAmdx2aZgi2XgVa5jHmUJSXyMdJqDTbvjWI8UaaawpSrYTl0liCLh9kSE4853l7m4TfY72
aiwPOI52WEF8lF4BDeGYE4K3XF68Tfm7leKpoqBft9GFt/vyhtCH/TSXd8mfylCgl0yw3/
BtrR6fBkmb6WtoJX5BYTDYLKXkLgeRr40pAH6ZysiBp6LsePhnmwCF8NjIRbhXw2/KmJ8r6ZAk1ln8ke/
SgUc4pcDBkMcW2U9ZAb61+QFkgxCqim+Z9vkPaiLeq4dhDCk+cMVBxSaV2zdPR8Vc3RKY2r5JMu2brMfSflxOTtsgZRHjh/
gag7FqvCYShImXUbzqQ5L0sJonzJSQ7eY6OCYPyGIXB6cwUQllFYOV7KNt4q9+I9LhABp/pMu07eBrh/
XXM6uw0FsetjTfgFEF6j/aBSPhltxrSjd56sM3me8TUPDcGuvtgs75qoZ+KpkDcPFp7OE3biqdB+TWW/
ufBEXBWerFeLNYllLAnBzZdzko7vZjPZGWcefR9POKeFG6WwzF0mHgLmbgfmbly95QmbT+OnwYq4WpXV+ZVnj84irgrbedTURL6
DTgGzTz+WvsqukCr6nChXXuODkFQdE7mF8jRRDBkDudrkpYX5nPrfPiDViHs5vrBhoyG8Qr4b4rYY9ji4UBpZPm/
wds1Vvshk5qunaWKC7fK/DEInJhL5ECqzju/aBXkzzHO7BRTsmS/nzPNyCK/F4sb3wuuADZuObWo3XcrT2CMHvqHczT/
VU13+2OZKOZAbxbDggC1FsB1yGUJLKIdyVRfgnhZukUKkayFxk6iCFlz/
x2PH2odNJAONfhhpNroRwmPllwgjcqxGAte+qc11R6nsmz9SRuD7Z5DcDKljfiIkS0ELmtM9UGhHBwwsc4PMLBslS96aLYv043B
ohzjmljbhgMB/
QVBuHrI+glgeU0IkW0GLvgcWyNEb+ofcp8QEyvGv4H6dSg6pUpCw7KmYn19HuPEIobFVOvKr1N3rcZ2TPFLnQooUtVTKV7GUI+b
UN1ankksqFeQK68dCwq4Ok0GyFKTbzZ9OEyCLzi2XttbVcirHn88xCMLfNgnte0789DBHhPehvKkoaVgpm2BawQ7iwXT/
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"}
      ]
    }
  ]
}
```

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

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

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

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

H145 Mission System Documentation

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.