Operations File Schema

This document describes the schema for specifying the commands.json’ file to the satellite. First a ‘operations.json’ file is produced from the input command file. The operations file filters feasible commands which can be executed.

In EO-Sim, when animating the satellite operations, the ‘operations.json’ file is turned into CZML format which can be read and animated in Cesium JS.

Examples of the commands.json file are available in:

* ` \examples\mission2\commands.json`
* ` \examples\mission3\RUNxx\` folders

User I/P

(commands.json)

Check Feasibility &

Filter

operations.json

Process to CZML format

CZML file

Animate with Cesium JS

(Some of the schema is identical to the the CZML schema defined in <https://github.com/AnalyticalGraphicsInc/czml-writer/wiki>

An example command.json file is present in the examples/mission2/ folder.

# Overview

* **Structure:** Each mission has the operations defined as a list of JSON objects (packets).
* Each packet corresponds to a spacecraft operation.
* Allowed operations:
  + **TakeImage** - Take an image of a set of ground-points (GP). The corresponding GP lights up and remains lit until the end of mission.
  + **TransmitData** - Transmit data from spacecraft to another spacecraft or ground-station. A ‘line’ is animated corresponding to the time interval indicated.
* Example of TakeImage json-object

{

"@id": “ANI-10000231”,

"@type": "TakeImage",

"spacecraftId": "557",

"startTime": "2018-07-17T15:06:21Z",

"endTime": "2018-07-17T15:08:19Z",

"observedPosition":

{"@type": "cartographicDegrees",

"cartographicDegrees": [[30.2, 30.01, 0],

[30.3, 30.01, 0]]

},

"color": {"rgba": [255,0,0,255]}

}

* Example of TransmitData json-object

{

"@id": “ANI-10000271”,

"@type": "TransmitData",

"txEntityId": "557",

"rxEntityId": "41",

"startTime": “2018-07-17T15:08:36Z”,

"endTime": "2018-07-17T15:08:40Z"

}

# Packet Schema

This section describes the schema of a single packet. Below are the basic fields required for each packet. Depending on the operation-type additional required fields manifest which are described in the sub-sections.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| @id | string | An alphanumeric unique identifier. It shall consist of 3 uppercase characters ‘ANI’ and 8 digit number, separated by a dash ‘-‘(example: ANI-10000008). |
| @type | [**operationType**](#_bq99xswg8899) | Operation type |

## **operationType**

|  |  |  |
| --- | --- | --- |
| **Type** | **Context** | **Description** |
| string | enumeration | Limited to [‘TakeImage’](#_usldx2tvpmd4) and [‘TransmitData’](#_gxa70suxfkq5). |

### TakeImage

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| spacecraftId | string | Unique spacecraft identifier which is making the observation. |
| startTime | [**time**](#_dbnsxgmzisd5) | Start time of observation. |
| endTime | [**time**](#_dbnsxgmzisd5) | Stop time of observation. |
| observedPosition | **list,** [**position**](#_cpjp27wquwv8) | List of observed positions |
| color | [**color**](#_aal4quwbjtyc) | Color of the observed ground-position. |

### 

### 

### TransmitData

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| txEntityId | string | Unique identifier of the transmitter. |
| txEntityType | entityType | Transmitter entity type (Spacecraft or GroundStation)\* |
| rxEntityId | string | Unique identity of the receiver. |
| rxEntityType | entityType | Receiver entity type (Spacecraft or GroundStation)\* |
| startTime | [**time**](#_dbnsxgmzisd5) | Start time of data transmission. |
| endTime | [**time**](#_dbnsxgmzisd5) | Stop time of data transmission. |

\* The transmit data is allowed to be specified only between spacecraft to spacecraft or spacecraft to/from ground-station. (Not from ground-station to ground-station.)

## Other JSON fields

### time

| **Type** | **Units** | **Description** |
| --- | --- | --- |
| string | UTC | Time value shall be in YYYY-MM-DDThh:mm:ssZ format. Reference ISO-8601 standard.  Example: "2021-10-15T10:00:00Z" |

### position

position can be expressed with cartographic-degrees by specifying the “@type”:”cartographicDegrees”

#### cartographicDegrees

| **Type** | **Context** | **Units** | **Description** |
| --- | --- | --- | --- |
| List, float | [Longitude, Latitude, Height] | degrees, meters | Specify the longitude, latitude (in degrees) and height (in meters).  Example: [70.45, -30.32, 0] |

### color

A color specified as an array of color components [Red, Green, Blue, Alpha] where each component is in the range 0-255.

| **Type** | **Context** | **Description** |
| --- | --- | --- |
| List, integer  (0-255) | [Red, Green, Blue, Apha] | Specify the RGBA values between 0-255.  Example: [0,255,0,255] |