

# Elite:Dangerous Player Journal

## 1 Introduction

Elite:Dangerous writes a network log file primarily to help when investigating problems.

Third-party tools developers have been reading some of the entries in the network log file, mainly in order to track the player's location.

There is a clear demand from players for third-party tools, and from tools developers for more information from the game and/or server api.

The new Player Journal provides a stream of information about gameplay events which can be used by tools developers to provide richer, more detailed tools to enhance the player experience. The data records written to this journal are much more high-level then that written to the network log.

A short example of a player journal file (*out of date, some events may have additional data*):

```
{ "timestamp":"2016-06-10T14:31:00Z", "event":"FileHeader", "part":1, "gameversion":"2.2", "build":"r113684 " },
{ "timestamp":"2016-06-10T14:32:03Z", "event":"LoadGame", "Commander":"HRC1", "Ship":"SideWinder", "ShipID":1,
"GameMode":"Open", "Credits":600120, "Loan":0 }
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Rank", "Combat":0, "Trade":0, "Explore":1, "Empire":0, "Federation":0, "CQC":0 }
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Progress", "Combat":0, "Trade":0, "Explore":73, "Empire":0, "Federation":0, "CQC":0
}
{ "timestamp":"2016-06-10T14:32:15Z", "event":"Location", "StarSystem":"Asellus Primus", "StarPos":[-23.938,40.875,-1.344] }
{ "timestamp":"2016-06-10T14:32:16Z", "event":"Docked", "StationName":"Beagle 2 Landing", "StationType":"Coriolis" }
{ "timestamp":"2016-06-10T14:32:38Z", "event":"RefuelAll", "Cost":12, "Amount":0.234493 }
{ "timestamp":"2016-06-10T14:34:25Z", "event":"Undocked", "StationName":"Beagle 2 Landing", "StationType":"Coriolis" }
{ "timestamp":"2016-06-10T14:35:00Z", "event":"FSDJump", "StarSystem":"HIP 78085", "StarPos":[120.250,40.219,268.594],
"JumpDist":36.034 }
{ "timestamp":"2016-06-10T14:35:22Z", "event":"Scan", "BodyName":"HIP 78085 A", "StarType":"G" }
{ "timestamp":"2016-06-10T14:36:10Z", "event":"FSDJump", "StarSystem":"Praea Euq NW-W b1-3",
"StarPos":[120.719,34.188,271.750], "JumpDist":6.823 }
{ "timestamp":"2016-06-10T14:36:42Z", "event":"Scan", "BodyName":"Praea Euq NW-W b1-3", "StarType":"M" }
{ "timestamp":"2016-06-10T14:38:50Z", "event":"Scan", "BodyName":"Praea Euq NW-W b1-3 3", "Description":"Icy body with neon
rich atmosphere and major water geysers volcanism" }
{ "timestamp":"2016-06-10T14:39:08Z", "event":"Scan", "BodyName":"Praea Euq NW-W b1-3 3 a", "Description":"Tidally locked Icy
body" }
{ "timestamp":"2016-06-10T14:41:03Z", "event":"FSDJump", "StarSystem":"Asellus Primus", "StarPos":[-23.938,40.875,-1.344],
"JumpDist":39.112 }
{ "timestamp":"2016-06-10T14:41:26Z", "event":"SupercruiseExit", "StarSystem":"Asellus Primus", "Body":"Beagle 2 Landing" }
{ "timestamp":"2016-06-10T14:41:29Z", "event":"Docked", "StationName":"Beagle 2 Landing", "StationType":"Coriolis" }
{ "timestamp":"2016-06-10T14:41:58Z", "event":"SellExplorationData", "Systems":["HIP 78085", "Praea Euq NW-W b1-3" ],
"Discovered":["HIP 78085 A", "Praea Euq NW-W b1-3", "Praea Euq NW-W b1-3 3 a", "Praea Euq NW-W b1-3 3" ], "BaseValue":10822,
"Bonus":3959 }
```

## 1.1 ChangeLog

### Version 23 – v3.3.02 (Jan 2019)

- SquadronStartup event (§10.11)
- ProspectedAsteroid event (§11.33)
- ReservoirReplenished event (§11.38)
- Include station info in Location event, if starting when docked
- Status.json: more detail on fuel level (§12)

### Bugs Fixed

- Status.json: fix GuiFocus value when viewing Codex (§12)
- CodexEntry: fix SystemAddress (§6.1)

### Documentation Errata

- CrimeVictim event (was in v3.3, but not previously documented) (§11.11)
- Fix in EngineerCraft: (§8.13) note property name is BlueprintName not Blueprint (since v3.0)
- MissionAccepted: (§8.21) note property LocalisedName (does not match automatically-localised strings)

### Version 22 – in v3.3 (beta 4)

- Fixed duplicate scan events
- Fix some wing mission cargo reported as stolen
- FSSSignalDiscovered: add USSType info (§6.6)
- Add new FSSAllBodiesFound event (§6.4)
- Scan events generated automatically when entering system now logged as “ScanType”: “AutoScan” (§6.3)
- Add new event MultiSellExplorationData (§6.10)
- Faction info in Location/FSDJump: if it's the player's squadron faction, add SquadronFaction:true, and flags HappiestSystem:true or HomeSystem:true if relevant (§4.8)

### in v3.3 (beta3)

- FSSSignalDiscovered – if signal is a USS, add a ThreatLevel value; if signal is a station, add IsStation:true; add SystemAddress (§6.6)
- Cargo – fix spurious extra events; add a flag to indicate vessel=Ship or Vessel=SRV (§3.1)
- CodexEntry: add SystemAddress (§6.1)
- Status.json, Flags: add NightVision flag (bit28/0x10000000) (§12)
- SendText: include a journal entry for text to squadron or system chat (§11.42)
- ReceiveText: include a ‘Channel’ parameter to show if message was from squadron or system chat (§11.36)
- SAAScanComplete: remove lists of names of discoverers and mappers (§6.13)

### Version 21 – for v3.3 (beta 2)

- Include info on Faction Happiness– in Location and FSDJump events (§4.8, §4.12)
- Status flags indicate if in Analysis mode, and GUI focus shows if in Orrery, FSS, SAA or Codex view (§12)
- Fixed bug in MissionAccepted and MissionCompleted for donation missions, where the “Donated” key was written twice, one with a string, once with an int (§8.21, §8.22)

### Errata

- Note Faction Active states do not have a Trend value
- Note FID value in Commander (§3.3), LoadGame (§3.8), ClearSavedGame (§3.2), NewCommander (§3.7) events with unique player id number

- Note Status.json (§12) includes fuel and cargo info
- Fix capitalisation for BodyName in SAAScanComplete (§6.13)

**Version 20 – for v3.3 (beta 1) (released 30<sup>th</sup> Oct 2018)**

- Multiple faction activestates – in Location and FSDJump events (§4.8, §4.12)
- The first “Cargo” event written to the journal contains full inventory (§3.1)
- Added “AsteroidCracked” event (§7.1)
- Added “SAAScanComplete” (Surface Area Analysis) event (§6.13)
- Added “CodexDiscovery” event (§6.1)
- Added “FSDDiscoveryScan” and “FSSSignalScan” events (§6.4, §6.6)
- Added several events for Squadrons (§10)
- 

**Version 19 – for v3.3 (preview released 20<sup>th</sup> Sept 2018)**

- Simplify the “Category” in MaterialTrade
- Clarify meaning of bit 14 in status file: was called “under ship” but actually indicates when turret is retracted
- ApproachSettlement now includes Latitude and Longitude
- Note Bounty event is different for Skimmer bounty
- Update description of StoredShips event with InTransit flag
- Add ActiveFine info to Docked event
- EngineerProgress event at startup with summary for all engineers currently known
- Note a new ShipTargetted event is generated after using KillWarrantScanner, with updated bounty for target
- Note the MissionRedirected mission name now has any trailing “\_name” removed
- Added MyReputation in faction list in FsdJump and Location events
- Added “FSDTarget” event when selecting a starsystem to jump to
- Added MissionID to cargo to indicate if it is mission-related: in Cargo, CollectCargo, EjectCargo events
- In ship loadout, indicate if it is ‘hot’
- Cargo summary is now written to a separate file, and updated when data changes
- Add “HullHealth” stat in the “Loadout” event
- MissionCompleted now indicates correct destination after redirection

**Version 18 – for v3.0.4 (27<sup>th</sup> March 2018)**

**Version 17 – for v3.0.3 (19<sup>th</sup> March 2018)**

**Version 16 – for v3.0.2 (5<sup>th</sup> March 2018)**

**Version 15 – for v3.0 – beta3 (6<sup>th</sup> Feb 2018)**

**Version 14 – for v3.0 – beta1 (25/Jan/2018)**

**Version 13 - In 2.4 Open beta (24<sup>th</sup> Aug 2017)**

**Version 12 - In 2.4 beta1 (17<sup>th</sup> Aug 2017)**

**Version 11 published 26/Jun/2017**

**Version 10 published 29/Mar/2017 (for v2.3 beta 5)**

**Version 9 published 20/Feb/2017 (for v2.3 beta)**

**Version 8 published 10/Jan/2017 (for v2.2.03)**

**Version 7 published 15/Nov/2016 (for release 2.2.02)**

**Version 6 published 26/Oct/2016 (for 2.2 public release)**

**Version 1 was published 20/July/2016**

## 2 File Format

The Player Journal is written in line-delimited JSON format (see [json.org](http://json.org) and [jsonlines.org](http://jsonlines.org)), to provide a standard format for ease of machine parsing, while still being intelligible to the human reader.

Each Journal file is a series of lines each containing one Json object.

### 2.1 File Location

The journal files are written into the user's Saved Games folder, eg, for Windows:

C:\Users\User Name\Saved Games\Frontier Developments\Elite Dangerous\

The filename is of the form **Journal**.<timestamp>.<part>.**log**, similar to network log files

In addition to the incremental player journal file, the following files are written into the same folder:

**Market.json** – contains list of commodities available at a station, with price info (written when opening commodity interaction screen). See section §8.17

**Outfitting.json** – contains list of modules and prices at station (written when opening outfitting interaction). See section §8.31

**Shipyard.json** – contains list of ships available in shipyard, with prices (written when opening shipyard screen). See section §8.46

**Status.json** – contains frequently-changing info as displayed in the cockpit gui. This is updated when the data changes. See section §12

### 2.2 Heading entry

The Heading record has a Json object with the following values:

- timestamp: the time in GMT, ISO 8601
- part: the file part number
- language: the language code
- gameversion: which version of the game produced the log (will indicate if beta)
- build: game build number

Example:

```
{ "timestamp": "2016-07-22T10:20:01Z", "event": "fileheader", "part": 1, "language": "French/FR",  
  "gameversion": "2.2 Beta 1", "build": "r114123 " }
```

(If the play session goes on a long time, and the journal gets very large, the file will be closed and a new file started with an increased part number: the heading entry is added at the beginning of every file. See also the "Continued" event)

## 2.3 Event Records

Each event record is a json object.

The object has a "timestamp" value with the time in ISO 8601 format, an "event":"*eventname*" key-value pair identifying the type of event, followed by other key-value pairs providing additional information.

The rest of this document describes each type of event that might be written into the journal, and the data values for each event.

## 2.4 Localisation

Some values written into the log use internal symbol IDs, as used by the game to lookup localised text strings. These have the form "\$symbolname;"

When such values are written into the log, the localised version of the string will also be written (*UTF8 encoded*), as a separate key-value pair, with "\_Localised" appended to the key name.

***Examples throughout this document have not been updated with this extra localised format***

"Government":"\$government\_PrisonColony;", "Government\_Localised":"Colonie pénitentiaire"

In addition, for v3.0, all commodity names and material names will also be localised, eg if we had "Material ":"hyperspacetrajectories", we will get the result "Material\_Localised":"Eccentric Hyperspace Trajectories"

However the localised value will be omitted if it is exactly the same as the original, ie avoid:

```
{ "Name":"iron", "Name_Localised":"Iron", "Count":2 }
```

## 3 Startup

### 3.1 Cargo

When written: at startup, *note this is now written slightly later in startup, after we have initialised missions, so we can detect if any cargo came from an abandoned delivery mission*

The first “Cargo” event in the file will contain the full inventory, others just indicate a separate file has been written

The full data is now written to a separate file Cargo.json

A simple event (with no parameters) is written to the main journal file when the cargo file is updated

Parameters:

- Vessel: “Ship” or “SRV”
- Inventory: array of cargo,
  - Name
  - Count
  - Stolen
  - MissionID (if relevant)

Example:

```
{ "timestamp": "2018-06-26T08:21:17Z", "event": "Cargo", "Vessel": "Ship", "Inventory": [
  { "Name": "gold", "Count": 2, "Stolen": 0 },
  { "Name": "gold", "MissionID": 65397935, "Count": 14, "Stolen": 0 },
  { "Name": "iondistributor", "Name_Localised": "Ion Distributor", "Count": 2, "Stolen": 0 }
]}
```

### 3.2 ClearSavedGame

When written: If you should ever reset your game

Parameters:

- Name: commander name
- FID: player id

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "ClearSavedGame", "Name": "HRC1",
  "FID": "F44396" }
```

### 3.3 Commander

When written: at the start of the LoadGame process

This is written before the inventory, loadout, etc

Parameters:

- Name: commander name

- FID: player id

### 3.4 Loadout

When written: at startup, when loading from main menu, or when switching ships, or after changing the ship in Outfitting

Parameters:

- Ship: current ship type
- ShipID: ship id number (indicates which of your ships you are in)
- ShipName: user-defined ship name
- ShipIdent: user-defined ship ID string
- HullValue – may not always be present
- ModulesValue – may not always be present
- HullHealth
- Rebuy
- Hot: (if wanted at startup – may not always be present)
- Modules: array of installed items, each with:
  - Slot: slot name
  - Item: module name
  - On: bool, indicates on or off
  - Priority: power priority
  - Health
  - Value
  - AmmoInClip: (if relevant)
  - AmmoInHopper: (if relevant)
  - Engineering: (if engineered)
    - EngineerID
    - Engineer: name
    - BlueprintID
    - BlueprintName: blueprint name
    - Level
    - Quality
    - ExperimentalEffect: (name, if applied)
    - Modifications: Json array of objects
      - Label – (see §13.11 below)
      - Value – may not always be present
      - OriginalValue
      - LessIsGood: bool

*(For a passenger cabin, AmmoInClip holds the number of places in the cabin)*

### 3.5 Materials

When written: at startup, when loading from main menu into game

Parameters:

- Raw: array of raw materials (each with name and count)
- Manufactured: array of manufactured items
- Encoded: array of scanned data

Example:

```
{ "timestamp":"2017-02-10T14:25:51Z", "event":"Materials", "Raw":[ { "Name":"chromium",
"Count":28 }, { "Name":"zinc", "Count":18 }, { "Name":"iron", "Count":23 }, { "Name":"sulphur",
"Count":19 } ], "Manufactured":[ { "Name":"refinedfocuscystals", "Count":10 }, {
"Name":"highdensitycomposites", "Count":3 }, { "Name":"mechanicalcomponents", "Count":3 } ],
"Encoded":[ { "Name":"emissiondata", "Count":32 }, { "Name":"shielddensityreports", "Count":23 } }
]
```

### 3.6 Missions

When written: at startup

Parameters:

- Active: (array of objects)
- Failed: (array of objects)
- Complete: (array of objects)

Each object contains:

- MissionID
- Name
- PassengerMission: bool
- Expires: time left in seconds

```
{ "timestamp":"2017-10-02T10:37:58Z", "event":"Missions", "Active":[ { "MissionID":65380900,
"Name":"Mission_Courier_name", "PassengerMission":false, Expires":82751 } ], "Failed":[ ],
"Complete":[ ] }
```

### 3.7 NewCommander

When written: Creating a new commander

Parameters:

- Name: (new) commander name
- FID: player id
- Package: selected starter package

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"NewCommander", "Name":"HRC1",
"FID":"F44396", "Package":"ImperialBountyHunter" }
```

### 3.8 LoadGame

When written: at startup, when loading from main menu into game



Parameters:

- Commander: commander name
- FID: player id
- Horizons: bool
- Ship: current ship type
- ShipID: ship id number (indicates which of your ships you are in)
- StartLanded: true (only present if landed)
- StartDead:true (only present if starting dead: see "Resurrect")
- GameMode: Open, Solo or Group
- Group: name of group (if in a group)
- Credits: current credit balance
- Loan: current loan
- ShipName: user-defined ship name
- ShipIdent: user-defined ship ID string
- FuelLevel: current fuel
- FuelCapacity: size of main tank

Example:

```
{ "timestamp":"2017-02-10T14:25:51Z", "event":"LoadGame", "Commander":"HRC-2",  
"FID":"F44396", "Horizons":true, "Ship":"FerDeLance", "ShipID":19, "ShipName":"jewel of parhoon",  
"ShipIdent":"hr-17f", "FuelLevel":3.964024, "FuelCapacity":8.000000, "GameMode":"Open",  
"Credits":2890718739, "Loan":0 }
```

### 3.9 Passengers

When written: at startup, when loading the saved game file

Parameters:

- Manifest: array of passenger records, each containing:
  - MissionID
  - Type
  - VIP (bool)
  - Wanted (bool)
  - Count

### 3.10 Powerplay

When written: at startup, if player has pledged to a power

Parameters:

- Power: name
- Rank
- Merits
- Votes
- TimePledged (time in seconds)

Example:

```
{ "timestamp":"2018-01-31T10:53:04Z", "event":"Powerplay", "Power":"Edmund Mahon", "Rank":0, "Merits":10, "Votes":0, "TimePledged":433024 }
```

### 3.11 Progress

When written: at startup

Parameters:

- Combat: percent progress to next rank
- Trade: "
- Explore: "
- Empire: "
- Federation: "
- CQC: "

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Progress", "Combat":77, "Trade":9, "Explore":93, "Empire":0, "Federation":0, "CQC":0 }
```

### 3.12 Rank

When written: at startup

Parameters:

- Combat: rank on scale 0-8
- Trade: rank on scale 0-8
- Explore: rank on scale 0-8
- Empire: military rank
- Federation: military rank
- CQC: rank on scale 0-8

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Rank", "Combat":2, "Trade":2, "Explore":5, "Empire":1, "Federation":3, "CQC":0 }
```

### 3.13 Reputation

When written: at startup (after Rank and Progress)

This gives the player's reputation (on a scale of -100..+100) with the superpowers

Parameters:

- Empire
- Federation
- Independent
- Alliance

Note thresholds:

- 100.. -90: hostile
- 90.. -35: unfriendly
- 35..+ 4: neutral
- +4..+35: cordial
- +35..+90: friendly
- +90..+100: allied

### 3.14 Statistics

When written: at startup

This line contains the information displayed in the statistics panel on the right side of the cockpit

Parameters:

- Bank\_Account
  - Current\_Wealth Spent\_On\_Ships
  - Spent\_On\_Outfitting Spent\_On\_Repairs
  - Spent\_On\_Fuel
  - Spent\_On\_Ammo\_Consumables
  - Insurance\_Claims
  - Spent\_On\_Insurance
- Combat
  - Bounties\_Claimed
  - Bounty\_Hunting\_Profit
  - Combat\_Bonds
  - Combat\_Bond\_Profits
  - Assassinations
  - Assassination\_Profits
  - Highest\_Single\_Reward
  - Skimmers\_Killed
- Crime
  - Fines
  - Total\_Fines
  - Bounties\_Received
  - Total\_Bounties
  - Highest\_Bounty
- Smuggling
  - Black\_Markets\_Traded\_With
  - Black\_Markets\_Profits
  - Resources\_Smuggled
  - Average\_Profit
  - Highest\_Single\_Transaction
- Trading
  - Markets\_Traded\_With
  - Market\_Profits
  - Resources\_Traded
  - Average\_Profit
  - Highest\_Single\_Transaction
- Mining
  - Mining\_Profits
  - Quantity\_Mined
  - Materials\_Collected
- Exploration
  - Systems\_Visited

- Fuel\_Scooped
- Fuel\_Purchased
- Exploration\_Profits
- Planets\_Scanned\_To\_Level\_2
- Planets\_Scanned\_To\_Level\_3
- Highest\_Payout
- Total\_Hyperspace\_Distance
- Total\_Hyperspace\_Jumps
- Greatest\_Distance\_From\_Start
- Time\_Played
- Passengers
  - Passengers\_Missions\_Bulk
  - Passengers\_Missions\_VIP
  - Passengers\_Missions\_Delivered
  - Passengers\_Missions\_Ejected
- Search\_And\_Rescue
  - SearchRescue\_Traded
  - SearchRescue\_Profit
  - SearchRescue\_Count
- Crafting
  - Spent\_On\_Crafting
  - Count\_Of\_Used\_Engineers
  - Recipes\_Generated
  - Recipes\_Generated\_Rank\_1
  - Recipes\_Generated\_Rank\_2
  - Recipes\_Generated\_Rank\_3
  - Recipes\_Generated\_Rank\_4
  - Recipes\_Generated\_Rank\_5
  - Recipes\_Applied
  - Recipes\_Applied\_Rank\_1
  - Recipes\_Applied\_Rank\_2
  - Recipes\_Applied\_Rank\_3
  - Recipes\_Applied\_Rank\_4
  - Recipes\_Applied\_Rank\_5
  - Recipes\_Applied\_On\_Previously\_Modified\_Modules
- Crew
  - NpcCrew\_TotalWages
  - NpcCrew\_Hired
  - NpcCrew\_Fired
  - NpcCrew\_Died
- Multicrew
  - Multicrew\_Time\_Total
  - Multicrew\_Gunner\_Time\_Total
  - Multicrew\_Fighter\_Time\_Total
  - Multicrew\_Credits\_Total
  - Multicrew\_Fines\_Total

Note times are in seconds

Example:

```
{ "timestamp":"2017-09-25T15:18:31Z", "event":"Statistics", "Bank_Account":{ "Current_Wealth":148827050,
"Spent_On_Ships":14499177, "Spent_On_Outfitting":30785093, "Spent_On_Repairs":17716, "Spent_On_Fuel":1647,
"Spent_On_Ammo_Consumables":0, "Insurance_Claims":4, "Spent_On_Insurance":1397620 }, "Combat":{
"Bounties_Claimed":0, "Bounty_Hunting_Profit":0, "Combat_Bonds":0, "Combat_Bond_Profits":0, "Assassinations":0,
"Assassination_Profits":0, "Highest_Single_Reward":0, "Skimmers_Killed":0 }, "Crime":{ "Fines":0, "Total_Fines":0,
"Bounties_Received":0, "Total_Bounties":0, "Highest_Bounty":0 }, "Smuggling":{ "Black_Markets_Traded_With":0,
"Black_Markets_Profits":0, "Resources_Smuggled":0, "Average_Profit":0, "Highest_Single_Transaction":0 }, "Trading":{
```

"Markets\_Traded\_With":3, "Market\_Profits":40700, "Resources\_Traded":23, "Average\_Profit":4070,  
 "Highest\_Single\_Transaction":17961 }, "Mining":{ "Mining\_Profits":0, "Quantity\_Mined":0, "Materials\_Collected":100 },  
 "Exploration":{ "Systems\_Visited":228, "Fuel\_Scooped":111, "Fuel\_Purchased":0, "Exploration\_Profits":304469,  
 "Planets\_Scanned\_To\_Level\_2":39, "Planets\_Scanned\_To\_Level\_3":15, "Highest\_Payout":52503,  
 "Total\_Hyperspace\_Distance":844927, "Total\_Hyperspace\_Jumps":295, "Greatest\_Distance\_From\_Start":65222.47204614,  
 "Time\_Played":651060 }, "Passengers":{ "Passengers\_Missions\_Bulk":0, "Passengers\_Missions\_VIP":0,  
 "Passengers\_Missions\_Delivered":0, "Passengers\_Missions\_Ejected":0 }, "Search\_And\_Rescue":{  
 "SearchRescue\_Traded":12, "SearchRescue\_Profit":19467, "SearchRescue\_Count":8 }, "Crafting":{ "Spent\_On\_Crafting":0,  
 "Count\_Of\_Used\_Engineers":2, "Recipes\_Generated":28, "Recipes\_Generated\_Rank\_1":9,  
 "Recipes\_Generated\_Rank\_2":6, "Recipes\_Generated\_Rank\_3":9, "Recipes\_Generated\_Rank\_4":4,  
 "Recipes\_Generated\_Rank\_5":0, "Recipes\_Applied":21, "Recipes\_Applied\_Rank\_1":8, "Recipes\_Applied\_Rank\_2":5,  
 "Recipes\_Applied\_Rank\_3":7, "Recipes\_Applied\_Rank\_4":1, "Recipes\_Applied\_Rank\_5":0,  
 "Recipes\_Applied\_On\_Previously\_Modified\_Modules":0 }, "Crew":{ "NpcCrew\_TotalWages":0, "NpcCrew\_Hired":0,  
 "NpcCrew\_Fired":0, "NpcCrew\_Died":0 }, "Multicrew":{ "Multicrew\_Time\_Total":23327,  
 "Multicrew\_Gunner\_Time\_Total":14241, "Multicrew\_Fighter\_Time\_Total":6070, "Multicrew\_Credits\_Total":0,  
 "Multicrew\_Fines\_Total":0 } }

## 4 Travel

### 4.1 ApproachBody

When written: when in Supercruise, and distance from planet drops to within the 'Orbital Cruise' zone

Parameters:

- StarSystem
- SystemAddress
- Body
- BodyID

```
{ "timestamp": "2017-09-27T15:21:05Z", "event": "ApproachBody", "StarSystem": "Eranin",  
  "Body": "Eranin 2" }
```

### 4.2 Docked

When written: when landing at landing pad in a space station, outpost, or surface settlement

Parameters:

- StationName: name of station
- MarketID
- SystemAddress
- StationType: type of station
- StarSystem: name of system
- CockpitBreach: true (only if landing with breached cockpit)
- StationFaction: station's controlling faction
- FactionState
- StationAllegiance
- StationEconomy : (station's primary economy)
- StationEconomies: (array of name and proportion values)
- StationGovernment
- DistFromStarLS
- StationServices: (Array of strings)
- Wanted: (only if docking when wanted locally)
- ActiveFine: true (if any fine is active)

The 'anonymous docking' protocol comes into effect if you're either Wanted (ie have a local bounty) or have an ActiveFine

Example:

```
{ "timestamp": "2018-03-07T12:22:25Z", "event": "Docked", "StationName": "Jenner Orbital", "StationType": "Outpost",  
  "StarSystem": "Luhman 16", "SystemAddress": "22960358574928", "MarketID": "3228883456", "StationFaction": "Union of  
  Luhman 16 Values Party", "FactionState": "CivilWar", "StationGovernment": "$government_Democracy",  
  "StationGovernment_Localised": "Democracy", "StationAllegiance": "Federation", "StationServices": [ "Dock", "Autodock",  
  "BlackMarket", "Commodities", "Contacts", "Exploration", "Missions", "Outfitting", "CrewLounge", "Rearm", "Refuel",  
  "Workshop", "MissionsGenerated", "FlightController", "StationOperations", "Powerplay", "SearchAndRescue" ],  
  "StationEconomy": "$economy_Refinery", "StationEconomy_Localised": "Refinery", "StationEconomies": [ {
```

```
"Name": "$economy_Refinery", "Name_Localised": "Refinery", "Proportion": 0.760000 }, { "Name": "$economy_Extraction",  
"Name_Localised": "Extraction", "Proportion": 0.240000 } ], "DistFromStarLS": 10.061876 }
```

StationServices can include:

Dock, Autodock, BlackMarket, Commodities, Contacts, Exploration, Initiatives, Missions, Outfitting, CrewLounge, Rearm, Refuel, Repair, Shipyard, Tuning, Workshop, MissionsGenerated, Facilitator, Research, FlightController, StationOperations, OnDockMission, Powerplay, SearchAndRescue,

### 4.3 DockingCancelled

When written: when the player cancels a docking request

Parameters:

- StationName: name of station
- StationType
- MarketID

### 4.4 DockingDenied

When written: when the station denies a docking request

Parameters:

- StationName: name of station
- StationType
- MarketID
- Reason: reason for denial

Reasons include: NoSpace, TooLarge, Hostile, Offences, Distance, ActiveFighter, NoReason

### 4.5 DockingGranted

When written: when a docking request is granted

Parameters:

- StationName: name of station
- StationType
- MarketID
- LandingPad: pad number
- 

### 4.6 DockingRequested

When written: when the player requests docking at a station

Parameters:

- StationName: name of station
- StationType
- MarketID

## 4.7 DockingTimeout

When written: when a docking request has timed out

Parameters:

- StationName: name of station
- StationType
- MarketID

## 4.8 FSDJump

When written: when jumping from one star system to another

Parameters:

- StarSystem: name of destination starsystem
- SystemAddress
- StarPos: star position, as a Json array [x, y, z], in light years
- Body: star's body name
- JumpDist: distance jumped
- FuelUsed
- FuelLevel
- BoostUsed: whether FSD boost was used
- SystemFaction: system controlling faction
- FactionState
- SystemAllegiance
- SystemEconomy
- SystemSecondEconomy
- SystemGovernment
- SystemSecurity
- Population
- Wanted
- Factions: an array of info for the local minor factions
  - Name
  - FactionState
  - Government
  - Influence
  - Happiness
  - MyReputation
  - PendingStates: array (if any) with State name and Trend value
  - RecovingStates: array (if any)with State name and Trend value
  - ActiveStates: array with State names (Note active states do not have a Trend value)
  - SquadronFaction:true (if player is in squadron aligned to this faction)
  - HappiestSystem:true (if player squadron faction, and this is happiest system)
  - HomeSystem:true(if player squadron faction, and this is home system)

If the player is pledged to a Power in Powerplay, and the star system is involved in powerplay,

- Powers: a json array with the names of any powers contesting the system, or the name of the controlling power
- PowerplayState: the system state – one of ("InPrepareRadius", "Prepared", "Exploited", "Contested", "Controlled", "Turmoil", "HomeSystem")
-



Example:

Happiness values are: (Elated, Happy, Discontented, Unhappy, Despondent)

Example:

```
{ "timestamp": "2018-10-29T10:05:21Z", "event": "FSDJump", "StarSystem": "Eranin", "SystemAddress": "2832631632594",
  "StarPos": [-22.84375, 36.53125, -1.18750], "SystemAllegiance": "Independent", "SystemEconomy": "$economy_Agri",
  "SystemEconomy_Localised": "Agriculture", "SystemSecondEconomy": "$economy_Refinery",
  "SystemSecondEconomy_Localised": "Refinery", "SystemGovernment": "$government_Anarchy",
  "SystemGovernment_Localised": "Anarchy", "SystemSecurity": "$GALAXY_MAP_INFO_state_anarchy",
  "SystemSecurity_Localised": "Anarchy", "Population": 450000, "JumpDist": 13.334, "FuelUsed": 0.000000,
  "FuelLevel": 25.630281, "Factions": [ { "Name": "Eranin Expeditionary Institute", "FactionState": "None",
    "Government": "Cooperative", "Influence": 0.170000, "Allegiance": "Independent",
    "Happiness": "$Faction_HappinessBand2", "Happiness_Localised": "Happy", "MyReputation": 0.000000 }, { "Name": "Eranin Peoples Party", "FactionState": "CivilWar", "Government": "Communism", "Influence": 0.226000,
    "Allegiance": "Independent", "Happiness": "$Faction_HappinessBand2", "Happiness_Localised": "Happy",
    "MyReputation": 29.974300, "ActiveStates": [ { "State": "CivilWar" } ] }, { "Name": "Pilots Federation Local Branch",
    "FactionState": "None", "Government": "Democracy", "Influence": 0.000000, "Allegiance": "PilotsFederation",
    "Happiness": "$Faction_HappinessBand2", "Happiness_Localised": "Happy", "MyReputation": 82.918297 }, {
    "Name": "Eranin Industry", "FactionState": "Outbreak", "Government": "Corporate", "Influence": 0.209000,
    "Allegiance": "Independent", "Happiness": "$Faction_HappinessBand3", "Happiness_Localised": "Discontented",
    "MyReputation": 0.000000, "ActiveStates": [ { "State": "Famine" }, { "State": "Lockdown" }, { "State": "Outbreak" } ] }, {
    "Name": "Eranin Federal Bridge", "FactionState": "CivilWar", "Government": "Dictatorship", "Influence": 0.226000,
    "Allegiance": "Independent", "Happiness": "$Faction_HappinessBand2", "Happiness_Localised": "Happy",
    "MyReputation": 0.000000, "ActiveStates": [ { "State": "CivilWar" } ] }, { "Name": "Mob of Eranin",
    "FactionState": "CivilLiberty", "Government": "Anarchy", "Influence": 0.134000, "Allegiance": "Independent",
    "Happiness": "$Faction_HappinessBand1", "Happiness_Localised": "Elated", "MyReputation": 0.000000, "ActiveStates": [ {
      "State": "Boom" }, { "State": "CivilLiberty" } ] }, { "Name": "Terran Colonial Forces", "FactionState": "CivilUnrest",
    "Government": "Confederacy", "Influence": 0.035000, "Allegiance": "Alliance", "Happiness": "$Faction_HappinessBand2",
    "Happiness_Localised": "Happy", "MyReputation": 0.000000, "ActiveStates": [ { "State": "Boom" }, { "State": "CivilUnrest" } ] }
  ], "SystemFaction": "Mob of Eranin", "FactionState": "CivilLiberty" }
```

## 4.9 FSDTarget

When written: when selecting a star system to jump to

Note, when following a multi-jump route, this will typically appear for the *next* star, during a jump, ie after “StartJump” but before the “FSDJump”

Parameters:

- Starsystem
- Name

## 4.10 LeaveBody

When written: when flying away from a planet, and distance increases above the 'Orbital Cruise' altitude

Parameters:

- StarSystem
- SystemAddress
- Body
- BodyID

## 4.11 Liftoff

When written: when taking off from planet surface

Parameters:

- Latitude (only if player flying in ship)
- Longitude (only if player flying in ship)
- PlayerControlled: (bool) false if ship dismissed when player is in SRV, true if player is taking off

Example:

```
{ "timestamp":"2016-07-22T10:53:19Z", "event":"Liftoff", "Latitude":63.468872, "Longitude":157.599380, "PlayerControlled":true }
```

## 4.12 Location

When written: at startup, or when being resurrected at a station

Parameters:

- StarSystem: name of destination starsystem
- SystemAddress
- StarPos: star position, as a Json array [x, y, z], in light years
- Body: star or planet's body name
- BodyID
- BodyType
- Docked: (bool)
- Latitude (if landed)
- Longitude (if landed)
- StationName: station name, (if docked)
- StationType: (if docked)
- MarketID: (if docked)
- SystemFaction: star system controlling faction
- FactionState
- SystemAllegiance
- SystemEconomy
- SystemSecondEconomy
- SystemGovernment
- SystemSecurity
- Wanted
- Factions: an array with info on local minor factions (similar to FSDJump)
- 

If the player is pledged to a Power in Powerplay, and the star system is involved in powerplay,

- Powers: a json array with the names of any powers contesting the system, or the name of the controlling power
- PowerplayState: the system state – one of ("InPrepareRadius", "Prepared", "Exploited", "Contested", "Controlled", "Turmoil", "HomeSystem")

The faction data includes happiness info, and can include multiple active states

If starting docked in a station, also include:

- StationFaction
- FactionState
- StationGovernment
- StationAllegiance
- StationServices
- StationEconomies (Array of (Name,Proportion) pairs )

### 4.13 StartJump

When written: at the start of a Hyperspace or Supercruise jump (start of countdown)

Parameters:

- JumpType: "Hyperspace" or "Supercruise"
- StarSystem: name of destination system (for a hyperspace jump)
- SystemAddress
- StarClass: star type (only for a hyperspace jump)

### 4.14 SupercruiseEntry

When written: entering supercruise from normal space

Parameters:

- Starsystem

Example:

```
{"timestamp":"2016-06-10T14:32:03Z", "event":"SupercruiseEntry", "StarSystem":"Yuetu" }
```

### 4.15 SupercruiseExit

When written: leaving supercruise for normal space

Parameters:

- Starsystem
- Body
- BodyID
- BodyType

Example:

```
{"timestamp":"2016-06-10T14:32:03Z", "event":"SupercruiseExit", "StarSystem":"Yuetu",  
"Body":"Yuetu B" }
```

## 4.16 Touchdown

When written: landing on a planet surface

Parameters:

- Latitude (only if player is landing)
- Longitude (only if player is landing)
- PlayerControlled: (bool) false if ship was recalled from SRV, true if player is landing

Example:

```
{ "timestamp":"2016-07-22T10:38:46Z", "event":"Touchdown", "Latitude":63.468872,  
  "Longitude":157.599380, "PlayerControlled":true }
```

## 4.17 Undocked

When written: liftoff from a landing pad in a station, outpost or settlement

Parameters:

- StationName: name of station
- MarketID

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Undocked", "StationName":"Long Sight Base" }
```

## 5 Combat

### 5.1 Bounty

When written: player is awarded a bounty for a kill

Parameters:

- Rewards: an array of Faction names and the Reward values, as the target can have multiple bounties payable by different factions
- Target: type of ship
- VictimFaction: the victim's faction
- TotalReward
- SharedWithOthers: if credit for the kill is shared with other players, this has the number of other players involved

When the bounty is for a skimmer, it's slightly different:

- Faction: faction paying bounty
- Target: type of target
- Reward: amount
- VictimFaction

Examples:

```
{ "timestamp": "2018-04-17T11:11:02Z", "event": "Bounty", "Rewards": [ { "Faction": "Nehet Patron's Principles", "Reward": 5620 } ], "Target": "empire_eagle", "TotalReward": 5620, "VictimFaction": "Nehet Progressive Party" }
{ "timestamp": "2018-05-20T21:19:58Z", "event": "Bounty", "Faction": "HIP 18828 Empire Consulate", "Target": "Skimmer", "Reward": 1000, "VictimFaction": "HIP 18828 Empire Consulate" }
```

### 5.2 CapShipBond

When written: The player has been rewarded for a capital ship combat

Parameters:

- Reward: value of award
- AwardingFaction
- VictimFaction

### 5.3 Died

When written: player was killed

Parameters:

- KillerName
- KillerShip
- KillerRank

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "Died", "KillerName": "$ShipName_Police_Independent;", "KillerShip": "viper", "KillerRank": "Deadly" }
```

## 5.4 Died

When written: player was killed by a wing

Parameters:

- Killers: a JSON array of objects containing player name, ship, and rank

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "Died", "Killers": [ { "Name": "Cmdr HRC1",  
"Ship": "Vulture", "Rank": "Competent" }, { "Name": "Cmdr HRC2", "Ship": "Python", "Rank": "Master" }  
] }
```

## 5.5 EscapeInterdiction

When written: Player has escaped interdiction

Parameters:

- Interdictor: interdicting pilot name
- IsPlayer: whether player or npc

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "EscapeInterdiction", "Interdictor": "Hrc1",  
"IsPlayer": true }
```

## 5.6 FactionKillBond

When written: Player rewarded for taking part in a combat zone

Parameters:

- Reward
- AwardingFaction
- VictimFaction

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "FactionKillBond", "Reward": 500,  
"AwardingFaction": "Jarildekald Public Industry", "VictimFaction": "Lencali Freedom Party" }
```

## 5.7 FighterDestroyed

When written: when a ship-launched fighter is destroyed

Parameters: none

## 5.8 HeatDamage

When written: when taking damage due to overheating

Parameters: none

## 5.9 HeatWarning

When written: when heat exceeds 100%

Parameters: none

## 5.10 HullDamage

When written: when hull health drops below a threshold (20% steps)

Parameters:

- Health
- PlayerPilot: bool – true if player is piloting the ship/fighter taking damage
- Fighter: bool – true for ship-launched fighter

## 5.11 Interdicted

When written: player was interdicted by player or npc

Parameters:

- Submitted: true or false
- Interdictor: interdicting pilot name
- IsPlayer: whether player or npc
- CombatRank: if player
- Faction: if npc
- Power: if npc working for a power

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"interdicted", "Submitted":false,
  "Interdictor":"Dread Pirate Roberts", "IsPlayer":false, "Faction": "Timocani Purple Posse" }
```

## 5.12 Interdiction

When written: player has (attempted to) interdict another player or npc

Parameters:

- Success : true or false
- Interdicted: victim pilot name
- IsPlayer: whether player or npc
- CombatRank: if a player
- Faction: if an npc
- Power: if npc working for power

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"interdiction", "Success":true, "Interdicted":"Fred
  Flintstone", "IsPlayer":true, "CombatRank":5 }
```

### 5.13 PVPKill

When written: when this player has killed another player

Parameters:

- Victim: name of victim
- CombatRank: victim's rank in range 0..8

### 5.14 ShieldState

When written: when shields are disabled in combat, or recharged

Parameters:

- ShieldsUp 0 when disabled, 1 when restored

Examples:

```
{ "timestamp":"2016-07-25T14:45:48Z", "event":"ShieldState", "ShieldsUp":false }  
{ "timestamp":"2016-07-25T14:46:36Z", "event":"ShieldState", "ShieldsUp":true }
```

### 5.15 ShipTargetted

When written: when the current player selects a new target

The amount of data written depends on the extent to which the target ship has been scanned

Parameters:

- TargetLocked: bool (ie false when losing target)

If target locked:

- Ship: name
- ScanStage: number

If Scan stage >= 1

- PilotName: name
- PilotRank: rank name

If scan stage >= 2

- ShieldHealth
- HullHealth

If scan stage >= 3

- Faction
- LegalStatus
- Bounty
- SubSystem
- SubSystemHealth

### 5.16 SRVDestroyed

When written: when the player's SRV is destroyed

Parameters: none



## 5.17 UnderAttack

When written: when under fire (same time as the Under Attack voice message)

Parameters:

- Target: (Fighter/Mothership/You)

## 6 Exploration

### 6.1 CodexEntry

When written: when a new discovery is added to the Codex

Parameters:

- EntryID: an ID number
- Name: string (+localisation)
- SubCategory: string (+localisation)
- Category: string (+localisation)
- Region: string
- System: string
- SystemAddress
- IsNewEntry: bool
- NewTraitsDiscovered: bool
- Traits: [ array of strings ]

The IsNewEntry and NewTraitsDiscovered fields are optional depending on the results of the scan, and the Traits field is only available for entries that have unlocked traits.

Example:

```
{ "timestamp": "2018-10-09T07:10:25Z", "event": "CodexEntry", "EntryID": 2401001, "Name": "$Codex_Ent_L_Phnl_Part_Cld_001_Name;", "Name_Localised": "L_Phnl_Part_Cld_001", "SubCategory": "$Codex_SubCategory_Micro_Organisms;", "SubCategory_Localised": "Micro Organisms", "Category": "$Codex_Category_Biology;", "Category_Localised": "Biological and Geological", "Region": "Region 18", "System": "Shinrarta Dezhra", "SystemAddress": 3932277478106, "IsNewEntry": true, "NewTraitsDiscovered": true, "Traits": ["Trait1", "Trait2", "Trait3"] }
```

### 6.2 DiscoveryScan

When written: when using the discovery scanner, and new body discoveries are displayed in the cockpit info window. Note you can get two or three of these in a row, where some bodies are discovered by the automatic passive scan, before the active scan is complete.

Parameters:

- SystemAddress
- Bodies: number of new bodies discovered

### 6.3 Scan

When Written: basic or detailed discovery scan of a star, planet or moon

This is also generated when scanning a navigation beacon in a populated system, to record info about all the bodies in the system

Parameters(star)

- ScanType

- Bodyname: name of body
- BodyID
- DistanceFromArrivalLS
- StarType: Stellar classification (for a star) – see §13.2
- StellarMass: mass as multiple of Sol's mass
- Radius
- AbsoluteMagnitude
- RotationPeriod (seconds)
- SurfaceTemperature
- Luminosity – see §13.9
- Age\_MY: age in millions of years
- \* Rings: [ array ] – if present

#### Parameters(Planet/Moon)

- ScanType
- Bodyname: name of body
- BodyID
- Parents: Array of BodyType:BodyID pairs
- DistanceFromArrivalLS
- \* TidalLock: 1 if tidally locked
- \* TerraformState: Terraformable, Terraforming, Terraformed, or null
- PlanetClass – see §13.3
- \* Atmosphere – see §13.4
- \* AtmosphereType
- \* AtmosphereComposition: [ array of info ]
- \* Volcanism – see §13.5
- SurfaceGravity
- \* SurfaceTemperature
- \* SurfacePressure
- \* Landable: true (if landable)
- \* Materials: JSON array with objects with material names and percentage occurrence
- \*Composition: structure containing info on solid composition
  - Ice
  - Rock
  - Metal
- \* Rings: [ array of info ] – if rings present
- \* ReserveLevel: (Pristine/Major/Common/Low/Depleted) – if rings present

#### If rotating:

- RotationPeriod (seconds)
- Axial tilt

#### Orbital Parameters for any Star/Planet/Moon (except main star of single-star system)

- SemiMajorAxis
- Eccentricity
- OrbitalInclination
- Periapsis
- OrbitalPeriod

Rings properties\*

- Name
- RingClass
- MassMT – ie in megatons
- InnerRad
- OuterRad

Note that a basic scan (ie without having a Detailed Surface Scanner installed) will now save a reduced amount of information.

A basic scan on a planet will **include** body name, planet class, orbital data, rotation period, mass, radius, surface gravity; but will **exclude** tidal lock, terraform state, atmosphere, volcanism, surface pressure and temperature, available materials, and details of rings. The info for a star will be largely the same whether a basic scanner or detailed scanner is used.

The "Parents" property provides the body's hierarchical position within the system: in the example below, "Procyon B 3 a" is a moon of a planet (body 11), which is orbiting a star (body 2), which is has a parent body that's a Barycentre

*Entries in the list above marked with an asterisk are only included for a detailed scan*

**ScanType:** one of Basic, Detailed, NavBeacon, NavBeaconDetail, AutoScan

Example:

```
{ "timestamp": "2018-02-02T10:43:05Z", "event": "Scan", "ScanType": "NavBeaconDetail",
  "BodyName": "Procyon B 3 a", "BodyID": 12, "Parents": [ { "Planet": 11, { "Star": 2, { "Null": 0 } },
  "DistanceFromArrivalLS": 10048.152344, "TidalLock": true, "TerraformState": "", "PlanetClass": "Rocky body",
  "Atmosphere": "", "AtmosphereType": "None", "Volcanism": "", "MassEM": 0.025342, "Radius": 2011975.250000,
  "SurfaceGravity": 2.495225, "SurfaceTemperature": 318.448792, "SurfacePressure": 0.000000, "Landable": true,
  "Materials": [ { "Name": "iron", "Percent": 19.315084 }, { "Name": "sulphur", "Percent": 17.321133 }, {
  "Name": "nickel", "Percent": 14.609120 }, { "Name": "carbon", "Percent": 14.565277 }, { "Name": "phosphorus",
  "Percent": 9.324941 }, { "Name": "chromium", "Percent": 8.686635 }, { "Name": "manganese",
  "Percent": 7.976933 }, { "Name": "zinc", "Percent": 5.249117 }, { "Name": "tin", "Percent": 1.200338 }, {
  "Name": "tungsten", "Percent": 1.060592 }, { "Name": "technetium", "Percent": 0.690823 } ], "Composition": {
  "Ice": 0.000000, "Rock": 0.861282, "Metal": 0.138718 }, "SemiMajorAxis": 89655728.000000,
  "Eccentricity": 0.000000, "OrbitalInclination": 4.566576, "Periapsis": 309.656799,
  "OrbitalPeriod": 344902.937500, "RotationPeriod": 352425.468750, "AxialTilt": 0.479157 }
```

## 6.4 FSSAllBodiesFound

When written: after having identified all bodies in the system

Parameters:

- SystemName
- SystemAddress
- Count

## 6.5 FSSDiscoveryScan

When written: when performing a full system scan ("Honk")

Parameters:

- Progress: (a value in range 0-1 showing how completely the system has been scanned)
- BodyCount: number of stellar bodies in system
- NonBodyCount: Number of non-body signals found

## 6.6 FSSignalDiscovered

When written: when zooming in on a signal using the FSS scanner

Parameters:

- SignalName
- SpawningState: the BGS state that triggered this event, if relevant
- SpawningFaction: the minor faction, if relevant
- TimeRemaining: remaining lifetime in seconds, if relevant
- SystemAddress
- ThreatLevel (if a USS)
- USSType: (if a USS) – same types as in USSDrop event
- IsStation: true (if it is a station)

## 6.7 MaterialCollected

When Written: whenever materials are collected

Parameters:

- Category: type of material (Raw/Encoded/Manufactured)
- Name: name of material
- Count: number of units collected

Examples:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "MaterialCollected", "Category": "Raw",
  "Name": "sulphur", "Count": 2 }
{ "timestamp": "2016-06-10T14:32:03Z", "event": "MaterialCollected", "Category": "Encoded",
  "Name": "disruptedwakeechoes", "Count": 1 }
```

## 6.8 MaterialDiscarded

When Written: if materials are discarded

Parameters:

- Category
- Name
- Count

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "MaterialDiscarded", "Category": "Raw",
  "Name": "sulphur", "Count": 5 }
```

## 6.9 MaterialDiscovered

When Written: when a new material is discovered

Parameters:

- Category
- Name
- DiscoveryNumber

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"MaterialDiscovered", "Category":"Manufactured",  
  "Name":"focuscrystals", "DiscoveryNumber":3 }
```

## 6.10 MultiSellExplorationData

When written: when selling exploration data in Cartographics, a page at a time

Parameters:

- Discovered: (array of records, as follows)
  - SystemName
  - NumBodies
- BaseValue
- Bonus
- TotalEarnings

Example

```
{ "timestamp":"2018-11-14T10:35:35Z", "event":"MultiSellExplorationData", "Discovered":[ {  
  "SystemName":"HIP 84742", "NumBodies":23 }, { "SystemName":"Col 359 Sector NY-S b20-1",  
  "NumBodies":9 } ], "BaseValue":2938186, "Bonus":291000, "TotalEarnings":3229186 }
```

## 6.11 NavBeaconScan

When written: when scanning a navigation beacon, before the scan data for all the bodies in the system is written into the journal

Parameters:

- NumBodies
- SystemAddress

## 6.12 BuyExplorationData

When Written: when buying system data via the galaxy map

Parameters:

- System
- Cost

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"BuyExplorationData", "System":"Styx", "Cost":352 }
```

### 6.13 SAAScanComplete

When written: after using the "Surface Area Analysis" scanner

Parameters:

- BodyName
- BodyID
- ~~Discoverers: (array of names)~~
- ~~Mappers: (array of names)~~
- ProbesUsed: (int)
- EfficiencyTarget: (int)

Example:

```
{ "timestamp":"2018-10-05T15:06:12Z", "event":"SAAScanComplete", "BodyName":"Eranin 5",  
"BodyID":5, "ProbesUsed":6, "EfficiencyTarget":9 }
```

### 6.14 SellExplorationData

When Written: when selling exploration data in Cartographics

Parameters:

- Systems: JSON array of system names
- Discovered: JSON array of discovered bodies
- BaseValue: value of systems
- Bonus: bonus for first discoveries
- TotalEarnings: total credits received (including for example the 200% bonus if rank 5 with Li Yong Rui)

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"SellExplorationData", "Systems":[ "HIP 78085",  
"Praea Euq NW-W b1-3" ], "Discovered":[ "HIP 78085 A", "Praea Euq NW-W b1-3", "Praea Euq NW-  
W b1-3 3 a", "Praea Euq NW-W b1-3 3" ], "BaseValue":10822, "Bonus":3959, "TotalEarnings":44343 }
```

## 6.15 Screenshot

When Written: when a screen snapshot is saved

Parameters:

- Filename: filename of screenshot
- Width: size in pixels
- Height: size in pixels
- System: current star system
- Body: name of nearest body
- Latitude
- Longitude
- Altitude
- Heading

The latitude, longitude, altitude and heading will be included if on a planet or in low-altitude flight

Example:

```
{ "timestamp": "2018-01-17T09:48:26Z", "event": "Screenshot",  
  "Filename": "_Screenshots/Screenshot_0024.bmp", "Width": 1440, "Height": 900,  
  "System": "Nuenets", "Body": "Nuenets C 2", "Latitude": -60.799900, "Longitude": -74.059799,  
  "Heading": 39, "Altitude": 27502.876953 }
```



## 7 Trade

### 7.1 AsteroidCracked

When written: when the player has broken up a 'Motherlode' asteroid for mining

Parameters:

- Body: name of nearest body

### 7.2 BuyTradeData

When Written: when buying trade data in the galaxy map

Parameters:

- System: star system requested
- Cost: cost of data

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"BuyTradeData", "System":"i Bootis", "Cost":100 }
```

### 7.3 CollectCargo

When Written: when scooping cargo from space or planet surface

Parameters:

- Type: cargo type
- Stolen: whether stolen goods
- MissionID; (if relevant)

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"CollectCargo", "Type":"agriculturalmedicines",  
  "Stolen":false }
```

### 7.4 EjectCargo

When Written:

Parameters:

- Type: cargo type
- Count: number of units
- MissionID: (if relevant)
- Abandoned: whether 'abandoned'

If the cargo is related to powerplay delivery *from outlying systems back to the centre*:

- PowerplayOrigin: starsystem name

Examples:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"EjectCargo", "Type":"tobacco", "Count":1,  
  "Abandoned":true }
```

```
{ "timestamp":"2016-09-21T14:18:23Z", "event":"EjectCargo", "Type":"alliancelegaslativerecords",  
"Count":2, "Abandoned":true, "PowerplayOrigin":"Tau Bootis" }
```

## 7.5 MarketBuy

When Written: when purchasing goods in the market

Parameters:

- MarketID
- Type: cargo type
- Count: number of units
- BuyPrice: cost per unit
- TotalCost: total cost

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"MarketBuy", "Type":"foodcartridges", "Count":10,  
"BuyPrice":39, "TotalCost":390 }
```

## 7.6 MarketSell

When Written: when selling goods in the market

Parameters:

- MarketID
- Type: cargo type
- Count: number of units
- SellPrice: price per unit
- TotalSale: total sale value
- AvgPricePaid: average price paid
- IllegalGoods: (not always present) whether goods are illegal here
- StolenGoods: (not always present) whether goods were stolen
- BlackMarket: (not always present) whether selling in a black market

Examples:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"MarketSell", "Type":"agriculturalmedicines",  
"Count":3, "SellPrice":1360, "TotalSale":4080, "AvgPricePaid":304 }
```

```
{ "event":"MarketSell", "Type":"mineraloil", "Count":9, "SellPrice":72, "TotalSale":648,  
"AvgPricePaid":0, "StolenGoods":true, "BlackMarket":true }
```

## 7.7 MiningRefined

When Written: when mining fragments are converted unto a unit of cargo by refinery

Parameters:

- Type: cargo type

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"MiningRefined", "Type":"Gold" }
```

## 8 Station Services

### 8.1 BuyAmmo

When Written: when purchasing ammunition

Parameters:

- Cost

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"BuyAmmo", "Cost":80 }
```

### 8.2 BuyDrones

When Written: when purchasing drones

Parameters:

- Type
- Count
- BuyPrice
- TotalCost

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"BuyDrones", "Type":"Drones", "Count":2, "SellPrice":101, "TotalCost":202 }
```

### 8.3 CargoDepot

When written: when collecting or delivering cargo for a wing mission, or if a wing member updates progress

Parameters:

- MissionID:(int)
- UpdateType:(string) (one of: "Collect", "Deliver", "WingUpdate")
- CargoType
- Count
- StartMarketID(int)
- EndMarketID(int)
- ItemsCollected(int)
- ItemsDelivered(int)
- TotalItemsToDeliver(int)
- Progress:(float)

The CargoType and Count are included when you collect or deliver goods, they are not included for a wing update.

The Progress value actually represents pending progress for goods in transit: (ItemsCollected-ItemsDelievered)/TotalItemsToDeliver

Example:

```
{ "timestamp":"2018-03-07T15:47:03Z", "event":"CargoDepot", "MissionID":65394170,
"UpdateType":"Deliver", "CargoType":"BasicMedicines", "Count":8, "StartMarketID":3228867072,
"EndMarketID":3534964736, "ItemsCollected":16, "ItemsDelivered":16, "TotalItemsToDeliver":3020,
"Progress":0.000000 }
```

## 8.4 CommunityGoal

When written: when checking the status of a community goal

This event contains the current status of all community goals the player is currently subscribed to

Parameters:

- CurrentGoals: an array with an entry for each CG, containing:
  - CGID: a unique ID number for this CG
  - Title: the description of the CG
  - SystemName
  - MarketName
  - Expiry: time and date
  - IsComplete: Boolean
  - CurrentTotal
  - PlayerContribution
  - NumContributors
  - PlayerPercentileBand
  - TopTier:
    - Name: string
    - Bonus: string

If the community goal is constructed with a fixed-size top rank (ie max reward for top 10 players)

- TopRankSize: (integer)
- PlayerInTopRank: (Boolean)

If the community goal has reached the first success tier:

- TierReached
- Bonus

Example:

```
{ "timestamp":"2017-08-14T13:20:28Z", "event":"CommunityGoal", "CurrentGoals":[ { "CGID":726,
"Title":"Alliance Research Initiative – Trade", "SystemName":"Kaushpoos", "MarketName":"Neville
Horizons", "Expiry":"2017-08-17T14:58:14Z", "IsComplete":false, "CurrentTotal":10062,
```

```
"PlayerContribution":562, "NumContributors":101, "TopRankSize":10, "PlayerInTopRank":false,
"TierReached":"Tier 1", "PlayerPercentileBand":50, "Bonus":200000 } ] }
```

## 8.5 CommunityGoalDiscard

When written: when opting out of a community goal

Parameters:

- CGID
- Name
- System

## 8.6 CommunityGoalJoin

When Written: when signing up to a community goal

Parameters:

- CGID
- Name
- System

## 8.7 CommunityGoalReward

When Written: when receiving a reward for a community goal

Parameters:

- CGID
- Name
- System
- Reward

## 8.8 CrewAssign

When written: when changing the task assignment of a member of crew

Parameters:

- Name
- CrewID
- Role

Example:

```
{ "timestamp":"2016-08-09T08:45:31Z", "event":"CrewAssign", "Name":"Dannie Koller",
"Role":"Active" }
```

## 8.9 CrewFire

When written: when dismissing a member of crew

Parameters:

- Name
- CrewID

Example:

```
{ "timestamp":"2016-08-09T08:46:11Z", "event":"CrewFire", "Name":"Whitney Pruitt-Munoz" }
```

## 8.10 CrewHire

When written: when engaging a new member of crew

Parameters:

- Name
- CrewID
- Faction
- Cost
- Combat Rank

Example:

```
{ "timestamp":"2016-08-09T08:46:29Z", "event":"CrewHire", "Name":"Margaret Parrish",  
"Faction":"The Dark Wheel", "Cost":15000, "CombatRank":1 }
```

## 8.11 EngineerApply

This event is obsolete, and is no longer written (as from v3.0)

## 8.12 EngineerContribution

When written: when offering items cash or bounties to an Engineer to gain access

Parameters:

- Engineer: name of engineer
- EngineerID
- Type: type of contribution (Commodity, materials, Credits, Bond, Bounty)
- Commodity
- Material
- Faction (for bond or bounty)
- Quantity: amount offered this time
- TotalQuantity: total amount now donated

Example:

```
{ "timestamp":"2017-05-24T10:41:51Z", "event":"EngineerContribution", "Engineer":"Elvira  
Martuuk", "EngineerID":300160, "Type":"Commodity", "Commodity":"soontillrelics", "Quantity":2,  
"TotalQuantity":3 }
```

## 8.13 EngineerCraft

When Written: when requesting an engineer upgrade

Parameters:

- Engineer: name of engineer
- EngineerID
- **BlueprintName**: name of blueprint
- BlueprintID
- Level: crafting level
- Quality: float
- ApplyExperimentalEffect: (when applying a new effect)
- Ingredients: JSON array of objects with names and quantities of materials required
- Modifiers: JSON array of modification data
  - Label: string – (see §13.11 below)
  - Value: float
  - OriginalValue: float
  - LessIsGood: bool
- Some modifications have string values:
  - Label: string
  - ValueStr: string

The new "quality" value represents the quality or progress of the blueprint. The quality should increase from 0 to 1 as the blueprint is refined through further crafting, and once it reaches a certain value, the player will have the option to upgrade the blueprint to the next level of recipe.

When applying an experimental effect, the *ApplyExperimentalEffect* property will show the name of the effect applied, and the ingredient list will hold the ingredients for that effect.

Example:

```
{ "timestamp": "2018-03-04T07:08:27Z", "event": "EngineerCraft",
  "Slot": "Slot03_Size3", "Module": "int_dronecontrol_collection_size3_class5",
  "Ingredients": [ { "Name": "phosphorus", "Count": 1 } ], "Engineer": "Ram Tah",
  "EngineerID": 300110, "BlueprintID": 128731526,
  "BlueprintName": "Misc_LightWeight", "Level": 1, "Quality": 0.955000,
  "Modifiers": [ { "Label": "Mass", "Value": 4.436000, "OriginalValue": 8.000000,
    "LessIsGood": 1 }, { "Label": "Integrity", "Value": 81.000000,
    "OriginalValue": 90.000000, "LessIsGood": 0 } ] }
```

These modification types have string values: WeaponMode, DamageType, CabinClass (although I'm not sure if there are any blueprints that actually modify a cabin class!)

## 8.14 EngineerLegacyConvert

When written: when converting a pre-2.4 engineered module

This is generated when converting, or previewing a conversion of a legacy module to the new system. Due to the nature of the changes made for 3.0, modules generated in the old system are not compatible with the new crafting system, so players will be unable to craft with them. However, players will be given the opportunity to convert their legacy modules to the new format with the caveat that converted modules will be a recipe level below what they were before the conversion. The EngineerLegacyConvert journal entry is generated when converting a recipe, or just previewing a conversion, so some of our creative third party developers out there may be able to make tools to show how a ship loadout compares before and after converting their modules. The entry itself is the same as the EngineerCraft entry, minus the ingredients data (since no materials are required to convert), but plus an "IsPreview" bool to indicate whether this entry has been generated from a conversion, or just a preview.

## 8.15 EngineerProgress

When Written: when a player increases their access to an engineer

Also a summary of progress for all engineers at startup

Parameters (summary at startup)

- Engineers: array
  - Engineer: name
  - EngineerID
  - Rank: rank reached (when unlocked)
  - Progress: progress stage (Invited/Acquainted/Unlocked/Barred)
  - RankProgress: percentage towards next rank

Parameters (update for one engineer)

- Engineer: name
- EngineerID
- Rank: rank reached (when unlocked)
- Progress: progress stage (Invited/Acquainted/Unlocked/Barred)
- RankProgress: percentage towards next rank

Example (update)

```
{ "timestamp": "2018-01-16T09:34:36Z", "event": "EngineerProgress", "Engineer": "Zacariah Nemo",  
  "EngineerID": 300050, "Rank": 4, "RankProgress": 0 }
```

Example (startup):

```
{ "timestamp": "2018-05-04T13:58:22Z", "event": "EngineerProgress", "Engineers": [ { "Engineer": "Zacariah  
Nemo", "EngineerID": 300050, "Progress": "Unlocked", "RankProgress": 0, "Rank": 5 }, { "Engineer": "Marco  
Qwent", "EngineerID": 300200, "Progress": "Unlocked", "RankProgress": 37, "Rank": 4 }, { "Engineer": "Hera Tani",  
"EngineerID": 300090, "Progress": "Unlocked", "RankProgress": 0, "Rank": 3 }, { "Engineer": "Tod 'The Blaster'  
McQuinn", "EngineerID": 300260, "Progress": "Unlocked", "RankProgress": 97, "Rank": 3 }, { "Engineer": "Selene  
Jean", "EngineerID": 300210, "Progress": "Known" }, { "Engineer": "Lei Cheung", "EngineerID": 300120,  
"Progress": "Known" }, { "Engineer": "Juri Ishmaak", "EngineerID": 300250, "Progress": "Known" }, {  
"Engineer": "Felicity Farseer", "EngineerID": 300100, "Progress": "Unlocked", "RankProgress": 0, "Rank": 5 }, {
```



```
"Engineer":"Professor Palin", "EngineerID":300220, "Progress":"Invited" }, { "Engineer":"Elvira Martuuk",
"EngineerID":300160, "Progress":"Unlocked", "RankProgress":0, "Rank":5 }, { "Engineer":"Lori Jameson",
"EngineerID":300230, "Progress":"Known" }, { "Engineer":"The Dweller", "EngineerID":300180,
"Progress":"Unlocked", "RankProgress":0, "Rank":5 }, { "Engineer":"Liz Ryder", "EngineerID":300080,
"Progress":"Unlocked", "RankProgress":93, "Rank":3 }, { "Engineer":"Ram Tah", "EngineerID":300110,
"Progress":"Unlocked", "RankProgress":31, "Rank":3 } ] }
```

## 8.16 FetchRemoteModule

When written: when requesting a module is transferred from storage at another station

Parameters:

- StorageSlot
- StoredItem
- ServerId
- TransferCost
- Ship
- ShipId
- TransferTime: (in seconds)

## 8.17 Market

When written: when accessing the commodity market in a station

A separate file **market.json** is written to the same folder as the journal, containing full market price info

Parameters:

- MarketID
- StationName
- StarSystem

The separate file also contains:

- Items: array of objects
  - id
  - Name
  - Category
  - BuyPrice
  - SellPrice
  - MeanPrice
  - StockBracket
  - DemandBracket
  - Stock
  - Demand
  - Consumer: bool
  - Producer: bool
  - Rare: bool

Example: in the journal:

```
{ "timestamp":"2017-10-05T10:11:38Z", "event":"Outfitting", "MarketID":128678535, "StationName":"Black
Hide", "StarSystem":"Wyrdr" }
```

Separate file:

```
{ "timestamp":"2017-10-05T10:10:34Z", "event":"Market", "MarketID":128678535, "StationName":"Black
Hide", "StarSystem":"Wyr", "Items":[
{ "id":128049152, "Name":"$platinum_name;", "Name_Localised":"Platinum",
"Category":"$MARKET_category_metals;", "Category_Localised":"Metals", "BuyPrice":0, "SellPrice":42220,
"MeanPrice":19756, "StockBracket":0, "DemandBracket":3, "Stock":0, "Demand":9182, "Consumer":true,
"Producer":false, "Rare":false },
{ "id":128049153, "Name":"$palladium_name;", "Name_Localised":"Palladium",
"Category":"$MARKET_category_metals;", "Category_Localised":"Metals", "BuyPrice":0, "SellPrice":13999,
"MeanPrice":13244, "StockBracket":0, "DemandBracket":3, "Stock":0, "Demand":123183, "Consumer":true,
"Producer":false, "Rare":false },
{ "id":128049154, "Name":"$gold_name;", "Name_Localised":"Gold",
"Category":"$MARKET_category_metals;", "Category_Localised":"Metals", "BuyPrice":0, "SellPrice":10831,
"MeanPrice":9373, "StockBracket":0, "DemandBracket":3, "Stock":0, "Demand":151492, "Consumer":true,
"Producer":false, "Rare":false },
{ "id":128049155, "Name":"$silver_name;", "Name_Localised":"Silver",
"Category":"$MARKET_category_metals;", "Category_Localised":"Metals", "BuyPrice":0, "SellPrice":5512,
"MeanPrice":4759, "StockBracket":0, "DemandBracket":3, "Stock":0, "Demand":266258, "Consumer":true,
"Producer":false, "Rare":false },
: (about 100 lines later)
:
}]}
```

## 8.18 MassModuleStore

When written: when putting multiple modules into storage

Parameters:

- MarketID
- Ship
- ShipId
- Items: Array of records
  - Slot
  - Name
  - Hot
  - EngineerModifications (only present if modified)
  - Level
  - Quality

## 8.19 MaterialTrade

When written: when exchanging materials at the Material trader contact

Parameters:

- MarketID
- TraderType
- Paid
  - Material
  - Category
  - Quantity
- Received
  - Material
  - Category
  - Quantity

Example: (note categories simplified)

```
{ "timestamp": "2018-02-21T15:23:49Z", "event": "MaterialTrade", "MarketID": 3221397760,
  "TraderType": "encoded", "Paid": { "Material": "scandatabanks", "Material_Localised": "Classified Scan Databanks",
  "Category": "Encoded", "Quantity": 6, "Category": "Encoded" }, "Received": {
  "Material": "encodedscandata", "Material_Localised": "Divergent Scan Data", "Quantity": 1 } }
```

## 8.20 MissionAbandoned

When Written: when a mission has been abandoned

Parameters:

- Name: name of mission
- MissionID
- Fine: (if relevant)

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "MissionAbandoned",
  "Name": "Mission_Collect_name", "MissionID": 65343025 }
```

## 8.21 MissionAccepted

When Written: when starting a mission

Parameters:

- Name: name of mission
- LocalisedName: the mission name as displayed to the user
- Faction: faction offering mission
- MissionID
- Influence: effect on influence (None/Low/Med/High)
- Reputation: effect on reputation (None/Low/Med/High)
- Reward: expected cash reward
- Wing: bool

Optional Parameters (depending on mission type)

- Commodity: commodity type
- Count: number required / to deliver
- Donation: contracted donation (as string) (for altruism missions)
- Donated: actual donation (as int)
- Target: name of target
- TargetType: type of target
- TargetFaction: target's faction
- KillCount: number of targets
- Expiry: mission expiry time, in ISO 8601
- DestinationSystem
- DestinationStation
- NewDestinationSystem (if it has been redirected)
- NewDestinationStation (if redirected)
- PassengerCount
- PassengerVIPs: bool
- PassengerWanted: bool
- PassengerType: eg Tourist, Soldier, Explorer,...

Examples:

```
{ "timestamp": "2018-02-28T12:06:37Z", "event": "MissionAccepted", "Faction": "Official i Bootis Liberty Party", "Name": "Mission_DeliveryWing_Agriculture", "LocalisedName": "Agricultural supply run: 2280 units of Tea", "Commodity": "$Tea_Name;", "Commodity_Localised": "Tea", "Count": 2280, "TargetFaction": "Ovid Vision & Co", "DestinationSystem": "Ovid", "DestinationStation": "Shriver Platform", "Expiry": "2018-03-01T12:05:53Z", "Wing": true, "Influence": "Med", "Reputation": "Med", "Reward": 2686155, "MissionID": 65393626 }
```

## 8.22 MissionCompleted

When Written: when a mission is completed

Parameters:

- Name: mission type
- Faction: faction name
- MissionID

Optional parameters (depending on mission type)

- Commodity
- Count
- Target
- TargetType
- TargetFaction
- Reward: value of reward
- Donation: contracted donation (as string) (for altruism missions)
- Donated: actual donation (as int)
- PermitsAwarded:[] (names of any permits awarded, as a JSON array)
- CommodityReward:[] (names and counts of any commodity rewards)
- MaterialsReward:[] ( name, category and count)
- FactionEffects: array of records
  - Faction
  - Effects: array of Effect and Trend value pairs
  - Influence: array of SystemAddress and Trend value pairs
  - Reputation: Trend value

*TrendValue* can be UpGood, UpBad, DownGood, or DownBad

Example:

```
{ "timestamp": "2018-01-30T10:00:20Z", "event": "MissionCompleted", "Faction": "Farack Stakeholders", "Name": "Mission_Collect_name", "MissionID": 65391731, "Commodity": "$AgriculturalMedicines_Name;", "Count": 12, "DestinationSystem": "Farack", "DestinationStation": "Mitra Estate", "Reward": 10000, "FactionEffects": [ { "Faction": "Farack Stakeholders", "Effects": [ { "Effect": "$MISSIONUTIL_Interaction_Summary_outbreak_down;", "Effect_Localised": "Outbreak Down", "Trend": "DownGood" } ], "Influence": [ { "System": 3932143293138, "Trend": "UpGood" } ], "Reputation": "UpGood" } ] }
```

Example of MaterialsReward:

```
... "MaterialsReward": [ { "Name": "DisruptedWakeEchoes", "Name_Localised": "Atypical Disrupted Wake Echoes", "Category": "$MICRORESOURCE_CATEGORY_Encoded;", "Category_Localised": "Encoded", "Count": 4 } ] ...
```

## 8.23 MissionFailed

When Written: when a mission has failed

Parameters:

- Name: name of mission
- MissionID
- Fine: (if relevant)

## 8.24 MissionRedirected

When written: when a mission is updated with a new destination

Parameters

- MissionID
- Name
- NewDestinationStation
- OldDestinationStation
- NewDestinationSystem
- OldDestinationSystem

Example:

```
{ "timestamp": "2017-08-01T09:04:07Z", "event": "MissionRedirected", "MissionID": 65367315,
  "NewDestinationStation": "Metcalf Orbital", "OldDestinationStation": "Cuffey Orbital",
  "NewDestinationSystem": "Cemiess", "OldDestinationSystem": "Vequess" }
```

## 8.25 ModuleBuy

When Written: when buying a module in outfitting

Parameters:

- MarketID
- Slot: the outfitting slot
- BuyItem: the module being purchased
- BuyPrice: price paid
- Ship: the players ship
- ShipID

If existing module is stored:

- StoredItem: item being stored

If replacing an existing module:

- SellItem: item being sold
- SellPrice: sale price

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"ModuleBuy", "Slot":"MediumHardpoint2",  
  "SellItem":"hpt_pulse_laser_fixed_medium", "SellPrice":0,  
  "BuyItem":"hpt_multicannon_gimbal_medium", "BuyPrice":50018, "Ship":"cobramkiii", "ShipID":1 }
```

## 8.26 ModuleRetrieve

When written: when fetching a previously stored module

Parameters:

- MarketID
- Slot
- Ship
- ShipID
- RetrievedItem
- Hot
- EngineerModifications: name of modification blueprint, if any
- Level
- Quality
- SwapOutItem (if slot was not empty)
- Cost

## 8.27 ModuleSell

When Written: when selling a module in outfitting

Parameters:

- MarketID
- Slot
- SellItem
- SellPrice
- Ship
- ShipID

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"ModuleSell", "Slot":"Slot06_Size2",  
  "SellItem":"int_cargorack_size1_class1", "SellPrice":877, "Ship":"asp", "ShipID":1 }
```

## 8.28 ModuleSellRemote

When written: when selling a module in storage at another station

Parameters:

- StorageSlot
- SellItem
- ServerId
- SellPrice
- Ship
- ShipId

## 8.29 ModuleStore

When written: when storing a module in Outfitting

Parameters:

- MarketID
- Slot
- Ship
- ShipID
- StoredItem
- Hot
- EngineerModifications: name of modification blueprint, if any
- Level
- Quality
- ReplacementItem (if a core module)
- Cost (if any)

Example:

```
{ "timestamp":"2018-01-31T10:55:15Z", "event":"ModuleStore", "MarketID":128676487,
  "Slot":"TinyHardpoint1", "StoredItem":"$hpt_shieldbooster_size0_class3_name;",
  "StoredItem_Localised":"Shield Booster", "Ship":"empire_courier", "ShipID":11, "Hot":false,
  "EngineerModifications":"ShieldBooster_Explosive", "Level":1, "Quality":0.527100 }
```

## 8.30 ModuleSwap

When Written: when moving a module to a different slot on the ship

Parameters:

- MarketID
- FromSlot
- ToSlot
- FromItem
- ToItem
- Ship
- ShipID

Examples:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"ModuleSwap", "FromSlot":"MediumHardpoint1",
  "ToSlot":"MediumHardpoint2", "FromItem":"hpt_pulselaser_fixed_medium",
  "ToItem":"hpt_multicannon_gimbal_medium", "Ship":"cobramkiii", "ShipID":1 }
```

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"ModuleSwap", "FromSlot":"SmallHardpoint2",
  "ToSlot":"SmallHardpoint1", "FromItem":"hpt_pulselaserburst_fixed_small_scatter",
  "ToItem":"Null", "Ship":"cobramkiii", "ShipID":1 }
```

## 8.31 Outfitting

Written when accessing the outfitting menu

The full parts pricelist is written to a separate file ***Outfitting.json***

Parameters:

- MarketID
- StationName
- StarSystem

The separate file also contains

- Horizons: bool
- Items: array of objects
  - id
  - Name
  - BuyPrice

Example: (in journal)

```
{ "timestamp":"2017-10-05T10:11:38Z", "event":"Outfitting", "MarketID":128678535, "StationName":"Black Hide", "StarSystem":"Wyrd" }
```

(In separate file)

```
{ "timestamp":"2017-10-05T10:11:38Z", "event":"Outfitting", "MarketID":128678535, "StationName":"Black Hide", "StarSystem":"Wyrd", "Horizons":true, "Items":[  
  { "id":128049382, "Name":"hpt_pulselaser_fixed_medium", "BuyPrice":16731 },  
  { "id":128049383, "Name":"hpt_pulselaser_fixed_large", "BuyPrice":66924 },  
  { "id":128049385, "Name":"hpt_pulselaser_gimbal_small", "BuyPrice":6275 },  
  { "id":128049386, "Name":"hpt_pulselaser_gimbal_medium", "BuyPrice":33653 },  
  { "id":128049388, "Name":"hpt_pulselaser_turret_small", "BuyPrice":24717 },  
  { "id":128681995, "Name":"hpt_pulselaser_gimbal_huge", "BuyPrice":834269 },  
  :  
  : (50 lines later)  
  :  
  ] }
```

## 8.32 PayBounties

When written: when paying off bounties

Parameters:

- Amount: (total amount paid , including any broker fee)
- BrokerPercentage (present if paid via a Broker)
- AllFines: bool
- Faction
- ShipID

Example:

```
{ "timestamp":"2018-03-19T10:25:10Z", "event":"PayBounties", "Amount":400,  
  "Faction":"$faction_Federation;", "Faction_Localised":"Federation", "ShipID":9,  
  "BrokerPercentage":25.000000 }
```



### 8.33 PayFines

When Written: when paying fines

Parameters:

- Amount: (total amount paid , including any broker fee)
- BrokerPercentage (present if paid via a Broker)
- AllFines: bool
- Faction: (if paying off an individual faction's fines)
- ShipID

Example:

```
{ "timestamp": "2018-03-19T10:24:21Z", "event": "PayFines", "Amount": 250, "AllFines": false, "Faction": "Batz Transport Commodities", "ShipID": 9 }
```

### 8.34 PayLegacyFines

When Written: when paying legacy fines

(This is now obsolete, as of v3.0)

Parameters:

- Amount (total amount paid, including any broker fee)
- BrokerPercentage (present if paid through a broker)

### 8.35 RedeemVoucher

When Written: when claiming payment for combat bounties and bonds

Parameters:

- Type: (CombatBond/Bounty/Trade/Settlement/Scannable)
- Amount: (Net amount received, after any broker fee)
- Faction: name of faction (*for types other than Bounty*)
- BrokerPercentage (if redeemed through a broker)
- Factions: array of faction/amount pairs (*for Type=Bounty*)

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "RedeemVoucher", "Type": "bounty", "Factions": [ { "Faction": "Ed's 38", "Amount": 1000 }, { "Faction": "Zac's Lads", "Amount": 2000 } ] }
```

### 8.36 RefuelAll

When Written: when refuelling (full tank)

Parameters:

- Cost: cost of fuel
- Amount: tons of fuel purchased

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "RefuelAll", "Cost": 317, "Amount": 6.322901 }
```

### 8.37 RefuelPartial

When Written: when refuelling (10%)

Parameters:

- Cost: cost of fuel
- Amount: tons of fuel purchased

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "RefuelPartial", "Cost": 83, "Amount": 1.649000 }
```

### 8.38 Repair

When Written: when repairing the ship

Parameters:

- Item: all, wear, hull, paint, or name of module
- Cost: cost of repair

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "Repair", "Item": "int_powerplant_size3_class5", "Cost": 1100 }
```

### 8.39 RepairAll

When written: when repairing everything

Parameters:

- Cost
- 

### 8.40 RestockVehicle

When Written: when purchasing an SRV or Fighter

Parameters:

- Type: type of vehicle being purchased (SRV or fighter model)
- Loadout: variant
- Cost: purchase cost
- Count: number of vehicles purchased

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "RestockVehicle", "Type": "SRV", "Loadout": "starter", "Cost": 1030, "Count": 1 }
```

### 8.41 ScientificResearch

When written: when contributing materials to a "research" community goal

Parameters:

- MarketID
- Name: material name
- Category

- Count
- 

#### 8.42 SearchAndRescue

When written: when delivering items to a Search and Rescue contact

Parameters:

- MarketID
- Name
- Count
- Reward

#### 8.43 SellDrones

When Written: when selling unwanted drones back to the market

Parameters:

- Type
- Count
- SellPrice
- TotalSale

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"SellDrones", "Type":"Drones", "Count":1,
  "SellPrice":91, "TotalSale":91 }
```

#### 8.44 SellShipOnRebuy

When written: When selling a stored ship to raise funds when on insurance/rebuy screen

Parameters:

- ShipType
- System
- SellShipId
- ShipPrice

Example:

```
{ "timestamp":"2017-07-20T08:56:39Z", "event":"SellShipOnRebuy", "ShipType":"Dolphin",
  "System":"Shinrarta Dezhra", "SellShipId":4, "ShipPrice":4110183 }
```

## 8.45 SetUserShipName

When written: when assigning a name to the ship in Starport Services

Parameters:

- Ship: Ship model (eg CobraMkIII)
- ShipID: player's ship ID number
- UserShipName: selected name
- UserShipId: selected ship id

Example:

```
{ "timestamp": "2017-01-24T10:42:38Z", "event": "SetUserShipName", "Ship": "cobramkiii",  
  "ShipID": 2, "UserShipName": "Enterprise", "UserShipId": "NCC 1701" }
```

## 8.46 Shipyard

When written: when accessing shipyard in a station

Parameters:

- MarketID
- StationName
- StarSystem

The full price list is written to a separate file, in the same folder as the journal, *Shipyard.json*

- Horizons: bool
- AllowCobraMkIV: bool
- Pricelist: array of objects
  - ShipType
  - ShipPrice

Example: in the Journal:

```
{ "timestamp": "2017-10-04T10:01:38Z", "event": "Shipyard", "MarketID": 128122104,  
  "StationName": "Seven Holm", "StarSystem": "Tamor", }
```

In the separate shipyard.json file:

```
{ "timestamp": "2017-10-04T10:01:38Z", "event": "Shipyard", "MarketID": 128122104, "StationName": "Seven  
Holm", "StarSystem": "Tamor", "Horizons": true, "AllowCobraMkIV": true, "PriceList": [  
  { "id": 128049249, "ShipType": "sidewinder", "ShipPrice": 24336 },  
  { "id": 128049255, "ShipType": "eagle", "ShipPrice": 34071 },  
  { "id": 128049261, "ShipType": "hauler", "ShipPrice": 40094 },  
  { "id": 128049267, "ShipType": "adder", "ShipPrice": 66779 },  
  { "id": 128672138, "ShipType": "empire_eagle", "ShipType_Localised": "Imperial Eagle", "ShipPrice": 84283 },  
  :  
] }
```

### 8.47 ShipyardBuy

When Written: when buying a new ship in the shipyard

Parameters:

- MarketID
- ShipType: ship being purchased
- ShipPrice: purchase cost
- StoreOldShip: (if storing old ship) ship type being stored
- StoreShipID
- SellOldShip: (if selling current ship) ship type being sold
- SellShipID
- SellPrice: (if selling current ship) ship sale price

Note: the new ship's ShipID will be logged in a separate event after the purchase

Example:

```
{ "timestamp": "2016-07-21T14:36:38Z", "event": "ShipyardBuy", "ShipType": "hauler",  
  "ShipPrice": 46262, "StoreOldShip": "SideWinder", "StoreShipID": 2 }
```

### 8.48 ShipyardNew

When written: after a new ship has been purchased

Parameters:

- ShipType
- NewShipID

Example:

```
{ "timestamp": "2016-07-21T14:36:38Z", "event": "ShipyardNew", "ShipType": "hauler",  
  "NewShipID": 4 }
```

### 8.49 ShipyardSell

When Written: when selling a ship stored in the shipyard

Parameters:

- MarketID
- ShipType: type of ship being sold
- SellShipID
- ShipPrice: sale price
- System: (if ship is in another system) name of system

Example:

```
{ "timestamp": "2016-07-21T15:12:19Z", "event": "ShipyardSell", "ShipType": "Adder", "SellShipID": 6,  
  "ShipPrice": 79027, "System": "Eranin" }
```

## 8.50 ShipyardTransfer

When Written: when requesting a ship at another station be transported to this station

Parameters:

- MarketID
- ShipType: type of ship
- ShipID
- System: where it is
- ShipMarketID
- Distance: how far away
- TransferPrice: cost of transfer
- TransferTime: time taken in seconds

Example:

```
{ "timestamp":"2016-07-21T15:19:49Z", "event":"ShipyardTransfer", "ShipType":"SideWinder",  
  "ShipID":7, "System":"Eranin", "Distance":85.639145, "TransferPrice":580 }
```

## 8.51 ShipyardSwap

When Written: when switching to another ship already stored at this station

Parameters:

- MarketID
- ShipType: type of ship being switched to
- ShipID
- StoreOldShip: (if storing old ship) type of ship being stored
- StoreShipID
- SellOldShip: (if selling old ship) type of ship being sold
- SellShipID

Example

```
{ "timestamp":"2016-07-21T14:36:06Z", "event":"ShipyardSwap", "ShipType":"sidewinder",  
  "ShipID":10, "StoreOldShip":"Asp", "StoreShipID":2 }
```

## 8.52 StoredModules

When written: when first visiting Outfitting, and when the set of stored modules has changed

Parameters:

- MarketID: current market
- Items: (array of objects)
  - Name
  - StarSystem
  - MarketID: where the module is stored
  - StorageSlot
  - TransferCost
  - TransferTime
  - Hot
  - EngineerModifications: (recipe name)
  - Level
  - Quality

- InTransit:bool

"EngineerModifications", "Level" and "Quality" only appear for an engineered module  
The InTransit value only appears (with value true) if the module is being transferred. In this case, the system, market, transfer cost and transfer time are not written.

```
{ "timestamp":"2018-01-31T10:55:16Z", "event":"StoredModules", "MarketID":128676487, "StationName":"Farseer Inc",
"StarSystem":"Deciat", "Items":[
{ "Name":"$int_engine_size3_class5_name;", "Name_Localised":"Thrusters", "StorageSlot":57, "StarSystem":"Deciat",
"MarketID":128676487, "TransferCost":0, "TransferTime":0, "BuyPrice":495215, "Hot":false,
"EngineerModifications":"Engine_Dirty", "Level":1, "Quality":0.000000 },
{ "Name":"$int_hyperdrive_size6_class5_name;", "Name_Localised":"FSD", "StorageSlot":59, "StarSystem":"Shinrarta
Dezhra", "MarketID":128666762, "TransferCost":79680, "TransferTime":1317, "BuyPrice":12620035, "Hot":false,
"EngineerModifications":"FSD_LongRange", "Level":5, "Quality":0.000000 } ] }
```

## 8.53 StoredShips

When written: when visiting shipyard

Parameters:

- MarketID
- StationName
- StarSystem
- ShipsHere: (array of objects)
  - ShipID
  - ShipType
  - Name (if named)
  - Value
  - Hot
- ShipsRemote: (array of objects)
  - ShipID
  - ShipType
  - Name (if named)
  - Value
  - Hot

If the ship is in transit:

- InTransit: true

If the ship is not in transit:

- StarSystem
- ShipMarketID
- TransferPrice
- TransferType

Example:

```
{ "timestamp":"2017-10-04T10:07:21Z", "event":"StoredShips", "StationName":"Jameson
Memorial", "StarSystem":"Shinrarta Dezhra", "ShipsHere":[ { "ShipID":64, "ShipType":"sidewinder",
"Value":567962 }, { "ShipID":20, "ShipType":"empire_eagle", "Value":6373956 } ], "ShipsRemote":[ {
"ShipID":0, "ShipType":"CobraMkIII", "StarSystem":"Beta-1 Tucanae", "TransferPrice":3777,
"TransferTime":1590, "Value":9464239 } ] }
```

## 8.54 TechnologyBroker

When written: when using the Technology Broker to unlock new purchasable technology

Parameters:

- BrokerType
- MarketID
- ItemsUnlocked: the name(s) of the new item unlocked (available in Outfitting)
- Commodities:
  - Name: name of item
  - Count: number of items used
- Materials:
  - Name
  - Count
  - Category

Example:

```
{ "timestamp": "2018-03-02T11:28:44Z", "event": "TechnologyBroker", "BrokerType": "Human",  
  "MarketID": 128151032, "ItemsUnlocked": [{ "Name": "Hpt_PlasmaShockCannon_Fixed_Medium",  
    "Name_Localised": "Shock Cannon" }], "Commodities": [{ "Name": "iondistributor",  
    "Name_Localised": "Ion Distributor", "Count": 6 }], "Materials": [ { "Name": "vanadium", "Count": 30,  
    "Category": "Raw" }, { "Name": "tungsten", "Count": 30, "Category": "Raw" }, { "Name": "rhenium",  
    "Count": 36, "Category": "Raw" }, { "Name": "technetium", "Count": 30, "Category": "Raw" } ] }
```



## 9 Powerplay

### 9.1 PowerplayCollect

When written: when collecting powerplay commodities for delivery

Parameters:

- Power: name of power
- Type: type of commodity
- Count: number of units

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "PowerplayCollect", "Power": "Li Yong-Rui",  
  "Type": "siriusfranchisepackage", "Count": 10 }
```

### 9.2 PowerplayDefect

When written: when a player defects from one power to another

Parameters:

- FromPower
- ToPower

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "PowerplayDefect", "FromPower": "Zachary  
Hudson", "ToPower": "Li Yong-Rui" }
```

### 9.3 PowerplayDeliver

When written: when delivering powerplay commodities

Parameters:

- Power
- Type
- Count

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "PowerplayDeliver", "Power": "Li Yong-Rui",  
  "Type": "siriusfranchisepackage", "Count": 10 }
```

### 9.4 PowerplayFastTrack

When written: when paying to fast-track allocation of commodities

Parameters:

- Power
- Cost

## 9.5 PowerplayJoin

When written: when joining up with a power

Parameters:

- Power

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"PowerplayJoin", "Power":"Zachary Hudson" }
```

## 9.6 PowerplayLeave

When written: when leaving a power

Parameters:

- Power

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"PowerplayLeave", "Power":"Li Yong-Rui" }
```

## 9.7 PowerplaySalary

When written: when receiving salary payment from a power

Parameters:

- Power
- Amount

## 9.8 PowerplayVote

When written: when voting for a system expansion

Parameters:

- Power
- Votes
- System

## 9.9 PowerplayVoucher

When written: when receiving payment for powerplay combat

Parameters:

- Power
- Systems:[name,name]

## 10 Squadrons

### 10.1 AppliedToSquadron

```
{ "timestamp": "2018-10-17T16:17:55Z", "event": "AppliedToSquadron",  
  "SquadronName": "TestSquadron" }
```

### 10.2 DisbandedSquadron

```
{ "timestamp": "2018-10-17T16:17:55Z", "event": "DisbandedSquadron",  
  "SquadronName": "TestSquadron" }
```

### 10.3 InvitedToSquadron

```
{ "timestamp": "2018-10-17T16:17:55Z", "event": "InvitedToSquadron",  
  "SquadronName": "TestSquadron" }
```

### 10.4 JoinedSquadron

```
{ "timestamp": "2018-10-17T16:17:55Z", "event": "JoinedSquadron",  
  "SquadronName": "TestSquadron" }
```

### 10.5 KickedFromSquadron

```
{ "timestamp": "2018-10-17T16:17:55Z", "event": "KickedFromSquadron",  
  "SquadronName": "TestSquadron" }
```

### 10.6 LeftSquadron

```
{ "timestamp": "2018-10-17T16:17:55Z", "event": "LeftSquadron", "SquadronName": "TestSquadron" }
```

### 10.7 SharedBookmarkToSquadron

```
{ "timestamp": "2018-10-17T16:17:55Z", "event": "SharedBookmarkToSquadron",  
  "SquadronName": "TestSquadron" }
```

### 10.8 SquadronCreated

```
{ "timestamp": "2018-10-17T16:17:55Z", "event": "SquadronCreated",  
  "SquadronName": "TestSquadron" }
```

### 10.9 SquadronDemotion

```
{ "timestamp":"2018-10-17T16:17:55Z", "event":"SquadronDemotion",  
  "SquadronName":"TestSquadron", "OldRank":3, "NewRank":2 }
```

### 10.10 SquadronPromotion

```
{ "timestamp":"2018-10-17T16:17:55Z", "event":"SquadronPromotion",  
  "SquadronName":"TestSquadron", "OldRank":2, "NewRank":3 }
```

### 10.11 SquadronStartup

Written at game startup, to identify the player's squadron

Parameters:

- SquadronName
- CurrentRank

### 10.12 WonATrophyForSquadron

```
{ "timestamp":"2018-10-17T16:17:55Z", "event":"WonATrophyForSquadron",  
  "SquadronName":"TestSquadron" }
```

## 11 Other Events

### 11.1 AfmuRepairs

When written: when repairing modules using the Auto Field Maintenance Unit (AFMU)

Parameters:

- Module: module name
- FullyRepaired: (bool)
- Health; (float 0.0..1.0)

If the AFMU runs out of ammo, the module may not be fully repaired.

Example:

```
{ "timestamp":"2017-08-14T15:41:50Z", "event":"AfmuRepairs",  
  "Module":"$modularcargobaydoor_name;", "Module_Localised":"Cargo Hatch",  
  "FullyRepaired":true, "Health":1.000000 }
```

### 11.2 ApproachSettlement

When written: when approaching a planetary settlement

Parameters:

- Name
- MarketID
- Latitude
- Longitude

### 11.3 ChangeCrewRole

When written: when in a crew on someone else's ship, player switched crew role

Parameters:

- Role: name of selected role (Idle, FireCon, FighterCon)

### 11.4 CockpitBreached

When written: when cockpit canopy is breached

Parameters: none

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"CockpitBreached" }
```

## 11.5 CommitCrime

When written: when a crime is recorded against the player

Parameters:

- CrimeType – see §13.6
- Faction

Optional parameters (depending on crime)

- Victim
- Fine
- Bounty

Examples:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"CommitCrime", "CrimeType":"assault",  
  "Faction":"The Pilots Federation", "Victim":"Potapinski", "Bounty":210 }  
{ "timestamp":"2016-06-10T14:32:03Z", "event":"CommitCrime", "CrimeType":"fireInNoFireZone",  
  "Faction":"Jarildekald Public Industry", "Fine":100 }
```

## 11.6 Continued

When written: if the journal file grows to 500k lines, we write this event, close the file, and start a new one

Parameters:

- Part: next part number

## 11.7 CrewLaunchFighter

When written: when in multicrew, in Helm player's log, when a crew member launches a fighter

Parameters:

- Crew: name of crew member launching in fighter
- 

## 11.8 CrewMemberJoins

When written: When another player joins your ship's crew

Parameters:

- Crew: player's commander name

## 11.9 CrewMemberQuits

When written: When another player leaves your ship's crew

Parameters:

- Crew: player's commander name

### 11.10 CrewMemberRoleChange

When written: in Multicrew, Helm's log, when another crew player changes role

Parameters:

- Crew: player name
- Role: selected role

Example:

```
{ "timestamp":"2017-02-22T14:56:54Z", "event":"CrewMemberRoleChange", "Crew":"HRC1",  
  "Role":"FireCon" }
```

### 11.11 CrimeVictim

When written: when another player commits a crime against the current player

Parameters:

- Offender
- CrimeType
- Fine *or* Bounty

### 11.12 DatalinkScan

When written: when scanning a data link

Parameters:

- Message: message from data link

### 11.13 DatalinkVoucher

When written: when scanning a datalink generates a reward

Parameters:

- Reward: value in credits
- VictimFaction
- PayeeFaction

### 11.14 DataScanned

When written: when scanning some types of data links

Parameters:

- Type

Type will typically be one of "DataLink", "DataPoint", "ListeningPost", "AbandonedDataLog", "WreckedShip", etc

### 11.15 DockFighter

When written: when docking a fighter back with the mothership

Parameters: none

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"DockFighter" }
```

### 11.16 DockSRV

When written: when docking an SRV with the ship

Parameters: none

### 11.17 EndCrewSession

When written: when the captain in multicrew disbands the crew

Parameters:

- OnCrime: (bool) true if crew disbanded as a result of a crime in a lawful session

### 11.18 FighterRebuilt

When written: when a ship's fighter is rebuilt in the hangar

Parameters:

- Loadout

### 11.19 FuelScoop

When written: when scooping fuel from a star

Parameters:

- Scooped: tons fuel scooped
- Total: total fuel level after scooping

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"FuelScoop", "Scooped":0.498700, "Total":16.000000 }
```

### 11.20 Friends

When written: when receiving information about a change in a friend's status

Also written at startup for friends who are already online (new in v2.4)

Parameters:

- Status: one of the following: *Requested, Declined, Added, Lost, Offline, Online*
- Name: the friend's commander name

### 11.21 JetConeBoost

When written: when enough material has been collected from a solar jet code (at a white dwarf or neutron star) for a jump boost



Parameters:

- BoostValue

### 11.22 JetConeDamage

When written: when passing through the jet code from a white dwarf or neutron star has caused damage to a ship module

Parameters:

- Module: the name of the module that has taken some damage

### 11.23 JoinACrew

When written: When you join another player ship's crew

Parameters:

- Captain: Helm player's commander name

### 11.24 KickCrewMember

When written: When you force another player to leave your ship's crew

Parameters:

- Crew: player's commander name
- OnCrime: (bool) true if player is automatically kicked for committing a crime in a lawful session

### 11.25 LaunchDrone

When written: when using any type of drone/limpet

Parameters:

- Type: one of:

"Hatchbreaker", "FuelTransfer", "Collection", "Prospector", "Repair", "Research", "Decontamination"

### 11.26 LaunchFighter

When written: when launching a fighter

Parameters:

- Loadout
- PlayerControlled: whether player is controlling the fighter from launch

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "LaunchFighter", "Loadout": "starter",  
  "PlayerControlled": true }
```

### 11.27 LaunchSRV

When written: deploying the SRV from a ship onto planet surface

Parameters:

- Loadout

## 11.28 ModuleInfo

When written: when looking at the cockpit RHS modules info panel, if data has changed

This also writes a ModulesInfo.json file alongside the journal, listing the modules in the same order as displayed

Parameters: None

Example of ModulesInfo file:

```
{ "timestamp": "2018-01-10T14:35:08Z", "event": "ModuleInfo", "Modules": [
{ "Slot": "MainEngines", "Item": "int_engine_size3_class5", "Power": 3.720000, "Priority": 0 },
{ "Slot": "MediumHardpoint1", "Item": "hpt_beamlaser_turret_medium", "Power": 0.930000, "Priority": 0 },
{ "Slot": "Slot03_Size2", "Item": "int_shieldgenerator_size2_class1", "Power": 0.900000, "Priority": 2 },
{ "Slot": "MediumHardpoint2", "Item": "hpt_multicannon_gimbal_medium", "Power": 0.640000, "Priority": 0 },
{ "Slot": "CargoHatch", "Item": "modularcargobaydoor", "Power": 0.600000, "Priority": 2 },
{ "Slot": "MediumHardpoint3", "Item": "hpt_pulselaser_gimbal_medium", "Power": 0.600000, "Priority": 0 },
{ "Slot": "PowerDistributor", "Item": "int_powerdistributor_size3_class5", "Power": 0.600000, "Priority": 0 },
{ "Slot": "FrameShiftDrive", "Item": "int_hyperdrive_size3_class5", "Power": 0.450000, "Priority": 0 },
{ "Slot": "Slot04_Size2", "Item": "int_fuelscoop_size2_class5", "Power": 0.390000, "Priority": 0 },
{ "Slot": "LifeSupport", "Item": "int_lifesupport_size1_class1", "Power": 0.320000, "Priority": 2 },
{ "Slot": "Slot05_Size2", "Item": "int_buggybay_size2_class1", "Power": 0.250000, "Priority": 0 },
{ "Slot": "Radar", "Item": "int_sensors_size2_class2", "Power": 0.210000, "Priority": 0 },
{ "Slot": "ShipCockpit", "Item": "empire_courier_cockpit", "Power": 0.000000 },
{ "Slot": "PowerPlant", "Item": "int_powerplant_size4_class2", "Power": 0.000000 },
{ "Slot": "Slot01_Size3", "Item": "int_cargorack_size2_class1", "Power": 0.000000 },
{ "Slot": "Slot02_Size3", "Item": "int_cargorack_size2_class1", "Power": 0.000000 },
{ "Slot": "Slot06_Size1", "Item": "int_stellarbodydiscoveryscanner_standard", "Power": 0.000000 },
{ "Slot": "DataLinkScanner", "Item": "hpt_shipdatalinkscanner", "Power": 0.000000, "Priority": 0 }
] }
```

## 11.29 Music

When written: when the game music 'mood' changes

Parameters:

- MusicTrack: (name)

Possible track names are: NoTrack, MainMenu, CQCMenu, SystemMap, GalaxyMap, GalacticPowers CQC, DestinationFromHyperspace, DestinationFromSupercruise, Supercruise, Combat\_Unknown Unknown\_Encounter, CapitalShip, CombatLargeDogFight, Combat\_Dogfight, Combat\_SRV Unknown\_Settlement, DockingComputer, Starport, Unknown\_Exploration, Exploration

Note: Other music track names may be used in future

### 11.30 NpcCrewPaidWage

This is written when crew receive wages

Parameters:

- NpcCrewId
- NpcCrewName
- Amount

### 11.31 NpcCrewRank

This is written when a crew member's combat rank increases

Parameters:

- NpcCrewId
- NpcCrewName
- RankCombat

### 11.32 Promotion

When written: when the player's rank increases

Parameters: one of the following

- Combat: new rank
- Trade: new rank
- Explore: new rank
- CQC: new rank
- Federation:
- Empire:

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Promotion", "Explore":2 }
```

### 11.33 ProspectedAsteroid

When using a prospecting drone

Parameters:

- Materials: (array of Name and Proportion)
- Content: (a string representing High/Medium/Low)
- MotherlodeMaterial: name (if it's a motherlode)
- Remaining: percentage

### 11.34 QuitACrew

When written: When you leave another player ship's crew

Parameters:

- Captain: Helm player's commander name

### 11.35 RebootRepair

When written: when the 'reboot repair' function is used

Parameters:

- Modules: JSON array of names of modules repaired

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "RebootRepair", "Modules": [ "MainEngines", "TinyHardpoint1" ] }
```

### 11.36 ReceiveText

When written: when a text message is received from another player or npc

Parameters:

- From
- Message
- Channel: (wing/local/voicechat/friend/player/npc/squadron/starsystem)

### 11.37 RepairDrone

When written: when the player's ship has been repaired by a repair drone

Parameters:

- HullRepaired
- CockpitRepaired
- CorrosionRepaired

Each of these is a number indicating the amount of damage that has been repaired

### 11.38 ReservoirReplenished

When fuel is moved from one fuel tank to another

Parameters:

- FuelMain
- FuelReservoir

### 11.39 Resurrect

When written: when the player restarts after death

Parameters:

- Option: the option selected on the insurance rebuy screen
- Cost: the price paid
- Bankrupt: whether the commander declared bankruptcy

### 11.40 Scanned

When written: when the player's ship has been scanned

(note the "Scan Detected" indication is at the start of the scan, this is written at the end of a successful scan)

Parameters:

- ScanType: Cargo, Crime, Cabin, Data or Unknown

Example:

```
{ "timestamp":"2017-02-13T12:30:09Z", "event":"Scanned", "ScanType":"Cargo" }
```

### 11.41 SelfDestruct

When written: when the 'self destruct' function is used

Parameters: none

### 11.42 SendText

When written: when a text message is sent to another player

Parameters:

- To: may be player name, or channel name
- Message

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"SendText", "To":"HRC-2", "Message":"zoom" }
```

### 11.43 Shutdown

When written: on a clean shutdown of the game

Parameters: none

### 11.44 Synthesis

When written: when synthesis is used to repair or rearm

Parameters:

- Name: synthesis blueprint
- Materials: JSON array with objects listing materials used and quantities

Example:

```
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Synthesis", "Name":"Repair Basic", "Materials":[  
  { "Name":"iron", "Count":2 }, { "Name":"nickel", "Count":1 } ] }
```

### 11.45 SystemsShutdown

When written: when the player's ship systems shut down (eg in a Thargoid encounter)

Parameters: none

### 11.46 USSDrop

When written: when dropping from Supercruise at a USS

Parameters:

- USSType: description of USS
- USSThreat: threat level

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "USSDrop", "USSType": "Disrupted wake echoes",  
  "USSThreat": 0 }
```

### 11.47 VehicleSwitch

When written: when switching control between the main ship and a fighter

Parameters:

- To: ( Mothership/Fighter)

Examples:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "VehicleSwitch", "To": "Fighter" }  
{ "timestamp": "2016-06-10T14:32:03Z", "event": "VehicleSwitch", "To": "Mothership" }
```

### 11.48 WingAdd

When written: another player has joined the wing

Parameters:

- Name

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "WingAdd", "Name": "HRC-2" }
```

### 11.49 WingInvite

When written: when the player is invited to a wing

Parameters:

- Name: the commander name of the player inviting to the wing

### **11.50 WingJoin**

When written: this player has joined a wing

Parameters:

- Others: JSON array of other player names already in wing

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "WingJoin", "Others": [ "HRC1" ] }
```

### **11.51 WingLeave**

When written: this player has left a wing

Parameters: none

Example:

```
{ "timestamp": "2016-06-10T14:32:03Z", "event": "WingLeave" }
```

## 12Status File

In addition to the journal file, which is written incrementally, there is now (in v3.0) a new file **Status.json** which is updated every few seconds, with some information about the current state of the game.

This has a similar format to a line in the journal, but the whole file is replaced every time. It has a timestamp like the journal, and "event":"Status"

Parameters:

- Flags: multiple flags encoded as bits in an integer (see below)
- Pips: an array of 3 integers representing energy distribution (in half-pips)
- Firegroup: the currently selected firegroup number
- GuiFocus: the selected GUI screen
- Fuel: { FuelMain, FuelReservoir} – both mass in tons
- Cargo: mass in tons
- Latitude (if on or near a planet)
- Altitude
- Longitude
- Heading

Flags:

Bit	Value	Hex	Meaning
0	1	0000 0001	Docked, (on a landing pad)
1	2	0000 0002	Landed, (on planet surface)
2	4	0000 0004	Landing Gear Down
3	8	0000 0008	Shields Up
4	16	0000 0010	Supercruise
5	32	0000 0020	FlightAssist Off
6	64	0000 0040	Hardpoints Deployed
7	128	0000 0080	In Wing
8	256	0000 0100	LightsOn
9	512	0000 0200	Cargo Scoop Deployed
10	1024	0000 0400	Silent Running,
11	2048	0000 0800	Scooping Fuel
12	4096	0000 1000	Srv Handbrake
13	8192	0000 2000	Srv using Turret view
14	16384	0000 4000	Srv Turret retracted (close to ship)
15	32768	0000 8000	Srv DriveAssist
16	65536	0001 0000	Fsd MassLocked
17	131072	0002 0000	Fsd Charging
18	262144	0004 0000	Fsd Cooldown
19	524288	0008 0000	Low Fuel ( < 25% )
20	1048576	0010 0000	Over Heating ( > 100% )
21	2097152	0020 0000	Has Lat Long
22	4194304	0040 0000	IsInDanger
23	8388608	0080 0000	Being Interdicted
24	16777216	0100 0000	In MainShip
25	33554432	0200 0000	In Fighter
26	67108864	0400 0000	In SRV
27	134217728	0800 0000	Hud in Analysis mode
28	268435456	1000 0000	Night Vision

Examples:



```
{ "timestamp":"2017-12-07T10:31:37Z", "event":"Status", "Flags":16842765, "Pips":[2,8,2], "FireGroup":0, "Fuel":{"FuelMain":15.146626, "FuelReservoir":0.382796 }, "GuiFocus":5 }
```

```
{ "timestamp":"2017-12-07T12:03:14Z", "event":"Status", "Flags":18874376, "Pips":[4,8,0], "FireGroup":0, "Fuel":{"FuelMain":15.146626, "FuelReservoir":0.382796 }, "GuiFocus":0, "Latitude":-28.584963, "Longitude":6.826313, "Heading":109, "Altitude": 404 }
```

In the first example above 16842765 (0x0101000d) has flags 24, 16, 3, 2, 0: In main ship, Mass locked, Shields up, Landing gear down, Docked

GuiFocus values:

- |    |                                 |
|----|---------------------------------|
| 0  | NoFocus                         |
| 1  | InternalPanel (right hand side) |
| 2  | ExternalPanel (left hand side)  |
| 3  | CommsPanel (top)                |
| 4  | RolePanel (bottom)              |
| 5  | StationServices                 |
| 6  | GalaxyMap                       |
| 7  | SystemMap                       |
| 8  | Orrery                          |
| 9  | FSS mode                        |
| 10 | SAA mode                        |
| 11 | Codex                           |

The latitude or longitude need to change by 0.02 degrees to trigger an update when flying, or by 0.0005 degrees when in the SRV

## 13Appendix

### 13.1 Ranks

**Combat ranks:** 0='Harmless', 1='Mostly Harmless', 2='Novice', 3='Competent', 4='Expert', 5='Master', 6='Dangerous', 7='Deadly', 8='Elite'

**Trade ranks:** 0='Penniless', 1='Mostly Penniless', 2='Peddler', 3='Dealer', 4='Merchant', 5='Broker', 6='Entrepreneur', 7='Tycoon', 8='Elite'

**Exploration ranks:** 0='Aimless', 1='Mostly Aimless', 2='Scout', 3='Surveyor', 4='Explorer', 5='Pathfinder', 6='Ranger', 7='Pioneer', 8='Elite'

**Federation ranks:** 0='None', 1='Recruit', 2='Cadet', 3='Midshipman', 4='Petty Officer', 5='Chief Petty Officer', 6='Warrant Officer', 7='Ensign', 8='Lieutenant', 9='Lt. Commander', 10='Post Commander', 11='Post Captain', 12='Rear Admiral', 13='Vice Admiral', 14='Admiral'

**Empire ranks:** 0='None', 1='Outsider', 2='Serf', 3='Master', 4='Squire', 5='Knight', 6='Lord', 7='Baron', 8='Viscount', 9='Count', 10='Earl', 11='Marquis', 12='Duke', 13='Prince', 14='King'

**CQC ranks:** 0='Helpless', 1='Mostly Helpless', 2='Amateur', 3='Semi Professional', 4='Professional', 5='Champion', 6='Hero', 7='Legend', 8='Elite'

### 13.2 Star Descriptions

(Main sequence:) O B A F G K M L T Y

(Proto stars:) TTS AeBe

(Wolf-Rayet:) W WN WNC WC WO

(Carbon stars:) CS C CN CJ CH CHd

MS S

(white dwarfs:) D DA DAB DAO DAZ DAV DB DBZ DBV DO DOV DQ DC DCV DX

N (=Neutron)

H (=Black Hole)

X (=exotic)

SupermassiveBlackHole

A\_BlueWhiteSuperGiant

F\_WhiteSuperGiant

M\_RedSuperGiant

M\_RedGiant

K\_OrangeGiant

RoguePlanet

Nebula

StellarRemnantNebula

### 13.3 Planet Classes

Metal rich body  
High metal content body  
Rocky body  
Icy body  
Rocky ice body  
Earthlike body  
Water world  
Ammonia world  
Water giant  
Water giant with life  
Gas giant with water based life  
Gas giant with ammonia based life  
Sudarsky class I gas giant (also class II, III, IV, V)  
Helium rich gas giant  
Helium gas giant

### 13.4 Atmosphere Classes

No atmosphere  
Suitable for water-based life  
Ammonia and oxygen  
Ammonia  
Water  
Carbon dioxide  
Sulphur dioxide  
Nitrogen  
Water-rich  
Methane-rich  
Ammonia-rich  
Carbon dioxide-rich  
Methane  
Helium  
Silicate vapour  
Metallic vapour  
Neon-rich  
Argon-rich  
Neon  
Argon  
Oxygen

### 13.5 Volcanism classes

(all with possible 'minor' or 'major' qualifier)

None  
Water Magma  
Sulphur Dioxide Magma  
Ammonia Magma  
Methane Magma  
Nitrogen Magma  
Silicate Magma  
Metallic Magma  
Water Geysers  
Carbon Dioxide Geysers  
Ammonia Geysers  
Methane Geysers  
Nitrogen Geysers  
Helium Geysers  
Silicate Vapour Geysers

### 13.6 Crime types

Assault  
Murder  
Piracy  
Interdiction  
IllegalCargo  
DisobeyPolice  
FireInNoFireZone  
FireInStation  
DumpingDangerous  
DumpingNearStation  
DockingMinor\_BlockingAirlock  
DockingMajor\_BlockingAirlock  
DockingMinor\_BlockingLandingPad  
DockingMajor\_BlockingLandingPad  
DockingMinor\_Trespass  
DockingMajor\_Trespass  
CollidedAtSpeedInNoFireZone  
CollidedAtSpeedInNoFireZone\_HullDamage

### 13.7 BodyType values

"Null" (eg the barycentre of a binary star system)  
"Star"  
"Planet"  
"PlanetaryRing"  
"StellarRing"  
"Station"  
"AsteroidCluster"

## 13.8 Gases in AtmosphereComposition

Water  
Oxygen  
CarbonDioxide  
SulphurDioxide  
Ammonia  
Methane  
Nitrogen  
Hydrogen  
Helium  
Neon  
Argon  
Silicates  
Iron

## 13.9 Star Luminosity classes

0,  
I,  
Ia0,  
Ia,  
Ib,  
Iab,  
II,  
IIa,  
IIab,  
IIb,  
III,  
IIIa,  
IIIab,  
IIIb,  
IV,  
IVa,  
IVab,  
IVb,  
V,  
Va,  
Vab,  
Vb,  
Vz,  
VI,  
VII,

### 13.10 Engineer IDs

300000	Didi Vatermann
300010	Bill Turner
300030	Broo Tarquin
300040	The Sarge
300050	Zachariah Nemo
300080	Liz Ryder
300090	Hera Tani
300100	Felicity Farseer
300110	Ram Tah
300120	Lei Cheung
300140	Col. Bris Dekker
300160	Elvira Martuuk
300180	The Dweller
300200	Marco Quent
300210	Selene Jean
300220	Prof. Palin
300230	Lori Jameson
300250	Juri Ishmaak
300260	Tod 'The Blaster' McQuinn
300270	Tiana Fortune

## 13.11 Module Attributes

These are the named attributes on modules, but not all are modifiable via engineering:

(note, based on enum eltemInfoFieldType)

Size,	EngineMinPerformance,	DroneMinJettisonedCargo,
Class,	EngineOptPerformance,	DroneMaxJettisonedCargo,
Mass,	EngineMaxPerformance,	FuelScoopRate,
Integrity,	EngineHeatRate,	FuelCapacity,
PowerDraw,	PowerCapacity,	OxygenTimeCapacity,
BootTime,	HeatEfficiency,	RefineryBins,
ShieldBankSpinUp,	WeaponsCapacity,	AFMRepairCapacity,
ShieldBankDuration,	WeaponsRecharge,	AFMRepairConsumption,
ShieldBankReinforcement,	EnginesCapacity,	AFMRepairPerAmmo,
ShieldBankHeat,	EnginesRecharge,	MaxRange,
DamagePerSecond,	SystemsCapacity,	SensorTargetScanAngle,
Damage,	SystemsRecharge,	Range,
DistributorDraw,	DefenceModifierHealthMultiplier,	VehicleCargoCapacity,
ThermalLoad,	DefenceModifierHealthAddition,	VehicleHullMass,
ArmourPenetration,	DefenceModifierShieldMultiplier,	VehicleFuelCapacity,
MaximumRange,	DefenceModifierShieldAddition,	VehicleArmourHealth,
ShotSpeed,	KineticResistance,	VehicleShieldHealth,
RateOfFire,	ThermicResistance,	FighterMaxSpeed,
BurstRateOfFire,	ExplosiveResistance,	FighterBoostSpeed,
BurstSize,	CausticResistance,	FighterPitchRate,
AmmoClipSize,	FSDInterdictorRange,	FighterDPS,
AmmoMaximum,	FSDInterdictorFacingLimit,	FighterYawRate,
RoundsPerShot,	ScannerRange,	FighterRollRate,
ReloadTime,	DiscoveryScannerRange,	CabinCapacity,
BreachDamage,	DiscoveryScannerPassiveRange,	CabinClass,
MinBreachChance,	MaxAngle,	DisruptionBarrierRange,
MaxBreachChance,	ScannerTimeToScan,	DisruptionBarrierChargeDuration,
Jitter,	ChaffJamDuration,	DisruptionBarrierActivePower,
WeaponMode,	ECMRange,	DisruptionBarrierCooldown,
DamageType,	ECMTimeToCharge,	WingDamageReduction,
ShieldGenMinimumMass,	ECMActivePowerConsumption,	WingMinDuration,
ShieldGenOptimalMass,	ECMHeat,	WingMaxDuration,
ShieldGenMaximumMass,	ECMCooldown,	ShieldSacrificeAmountRemoved,
ShieldGenMinStrength,	HeatSinkDuration,	ShieldSacrificeAmountGiven,
ShieldGenStrength,	ThermalDrain,	FSDJumpRangeBoost,
ShieldGenMaxStrength,	NumBuggySlots,	FSDFuelUseIncrease,
RegenRate,	CargoCapacity,	BoostSpeedMultiplier,
BrokenRegenRate,	MaxActiveDrones,	BoostAugmenterPowerUse,
EnergyPerRegen,	DroneTargetRange,	ModuleDefenceAbsorption,
FSDOptimalMass,	DroneLifeTime,	FalloffRange,
FSDHeatRate,	DroneSpeed,	DSS_RangeMult,
MaxFuelPerJump,	DroneMultiTargetSpeed,	DSS_AngleMult,
EngineMinimumMass,	DroneFuelCapacity,	DSS_RateMult,
EngineOptimalMass,	DroneRepairCapacity,	
MaximumMass,	DroneHackingTime,	

The following module attributes have enum values, written as a string, all others are float  
 WeaponMode should be one of: ForwardFire, Gimbal, TargetOnly, FireAtWill, Manual, NoMode  
 DamageType should be one of: Kinetic, Thermal, Explosive, Collision, Thargoid, AntiThargoid  
 CabinClass should be one of: Prison, Economy, Standard, Business, First, Luxury

# Index

AfmuRepairs, 61

AppliedToSquadron, 59

ApproachBody, 14

ApproachSettlement, 61

AsteroidCracked, 33

Atmosphere Classes, 75

BodyType values, 76

Bounty, 21

BuyAmmo, 35

BuyDrones, 35

BuyExplorationData, 30

BuyTradeData, 33

CapShipBond, 21

Cargo, 6

CargoDepot, 35

ChangeCrewRole, 61

ChangeLog, 2

ClearSavedGame, 6

CockpitBreached, 61

CodexEntry, 26

CollectCargo, 33

Combat, 21

Commander, 6

CommitCrime, 62

CommunityGoal, 36

CommunityGoalDiscard, 37

CommunityGoalJoin, 37

CommunityGoalReward, 37

Continued, 62

CrewAssign, 37

CrewFire, 37

CrewHire, 38

CrewLaunchFighter, 62

CrewMemberJoins, 62

CrewMemberQuits, 62

CrewMemberRoleChange, 63

Crime types, 76

CrimeVictim, 63

DatalinkScan, 63

DatalinkVoucher, 63

DataScanned, 63

Died, 21, 22

DisbandedSquadron, 59

DiscoveryScan, 26

Docked, 14

DockFighter, 63

DockingCancelled, 15

DockingDenied, 15

DockingGranted, 15

DockingRequested, 15

DockingTimeout, 16

DockSRV, 64



EjectCargo, 33	Interdicted, 23
EndCrewSession, 64	Interdiction, 23
Engineer IDs, 78	Introduction, 1
EngineerApply, 38	InvitedToSquadron, 59
EngineerContribution, 38	JetConeBoost, 65
EngineerCraft, 39	JetConeDamage, 65
EngineerLegacyConvert, 40	JoinACrew, 65
EngineerProgress, 40	JoinedSquadron, 59
EscapeInterdiction, 22	KickCrewMember, 65
Event Records, 5	KickedFromSquadron, 59
Exploration, 26	LaunchDrone, 65
FactionKillBond, 22	LaunchFighter, 65
FetchRemoteModule, 41	LaunchSRV, 66
FighterDestroyed, 22	LeaveBody, 17
FighterRebuilt, 64	LeftSquadron, 59
File Location, 4	Liftoff, 18
Friends, 64	LoadGame, 8
FSDJump, 16	Loadout, 7
FSDTarget, 17	Location, 18
FSSAllBodiesFound, 28	Market, 41
FSSDiscoveryScan, 28	MarketBuy, 34
FSSSignalDiscovered, 29	MarketSell, 34
FuelScoop, 64	MassModuleStore, 42
Heading, 4	MaterialCollected, 29
HeatDamage, 22	MaterialDiscarded, 29
HeatWarning, 23	MaterialDiscovered, 30
HullDamage, 23	Materials, 7

MaterialTrade, 42	Planet Classes, 75
MiningRefined, 34	Powerplay, 9, 57
MissionAbandoned, 43	PowerplayCollect, 57
MissionAccepted, 43	PowerplayDefect, 57
MissionCompleted, 44	PowerplayDeliver, 57
MissionFailed, 45	PowerplayFastTrack, 57
MissionRedirected, 45	PowerplayJoin, 58
Missions, 8	PowerplayLeave, 58
Module Attributes, 79	PowerplaySalary, 58
ModuleBuy, 45	PowerplayVote, 58
ModuleInfo, 66	PowerplayVoucher, 58
ModuleRetrieve, 46	Progress, 10
ModuleSell, 46	Promotion, 67
ModuleSellRemote, 46	ProspectedAsteroid, 67
ModuleStore, 47	PVPKill, 24
ModuleSwap, 47	QuitACrew, 67
MultiSellExplorationData, 30	Rank, 10
Music, 66	Ranks, 74
NavBeaconScan, 30	RebootRepair, 68
NewCommander, 8	ReceiveText, 68
NpcCrewPaidWage, 67	RedeemVoucher, 49
NpcCrewRank, 67	RefuelAll, 49
Outfitting, 48	RefuelPartial, 50
Passengers, 9	Repair, 50
PayBounties, 48	RepairAll, 50
PayFines, 49	RepairDrone, 68
PayLegacyFines, 49	Reputation, 10

ReservoirReplenished, 68	SquadronPromotion, 60
RestockVehicle, 50	SquadronStartup, 60
Resurrect, 68	SRVDestroyed, 24
SAAScanComplete, 31	Star Descriptions, 74
Scan, 26	StartJump, 19
Scanned, 69	Startup, 6
ScientificResearch, 50	Station Services, 35
Screenshot, 32	Statistics, 11
SearchAndRescue, 51	Status File, 72
SelfDestruct, 69	StoredModules, 54
SellDrones, 51	StoredShips, 55
SellExplorationData, 31	SupercruiseEntry, 19
SellShipOnRebuy, 51	SupercruiseExit, 19
SendText, 69	Synthesis, 69
SetUserShipName, 52	SystemsShutdown, 70
SharedBookmarkToSquadron, 59	TechnologyBroker, 56
ShieldState, 24	Touchdown, 20
ShipTargetted, 24	Trade, 33
Shipyards, 52	Travel, 14
ShipyardsBuy, 53	UnderAttack, 25
ShipyardsNew, 53	Undocked, 20
ShipyardsSell, 53	USSDrop, 70
ShipyardsSwap, 54	VehicleSwitch, 70
ShipyardsTransfer, 54	Volcanism classes, 76
Shutdown, 69	WingAdd, 70
SquadronCreated, 59	WingInvite, 70
SquadronDemotion, 60	WingJoin, 71

WingLeave, 71

WonATrophyForSquadron, 60