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A short example of a player journal file (*out of date, some events may have additional data*):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 16 – for v3.0.2 (5th March 2018)
- MissionCompleted: if rewards include materials, add category
- MaterialTrade: include category
- Status.json: when driving SRV, lat and long changes of 0.0005deg trigger update (in ship remains at 0.02deg)
- EngineerCraft: some modifications have string values, not float
- MissionAccepted: add a flag indicating if it is a wing mission
- New "CargoDepot" event for progress in a wing delivery mission
- Also fixed:
  - Fix localisation of strings where we don't have any substitution parameters to insert into them
  - Materials inventory at startup: omit items with a count of zero
  - SellDrones: fix sell price (was previously using buy price)

Version 15 – for v3.0 – beta3 (6th Feb 2018)
- Add a property in the "Location" or "FSDJump" event indicating if the player is wanted locally
- Ensure all ShipType strings written to journal get localisation (so we get "ShipType_Localised":"Alliance Chieftain" as well as just "ShipType":"TypeX")
- Add a "Powerplay" event in the journal at startup to log a player's powerplay allegiance, rank, merits etc
- Adding crew name in events that only had ID, adding ID in events that only had Name
- In Loadout, When writing engineered module info to Journal, include the name of the "Experimental Effect" if any
- Note ship loadout, when written at any time after startup, may not include HullValue, ModulesValue
- indicate in LoadGame whether the player has Horizons access, also in Shipyard and Outfitting files
- Include more information about the results of a mission in the MissionCompleted event
- Improve the data written to the journal info when applying an experimental effect to a module in engineering
- When writing commodity price lists to Market.json, include unlocalised and localised item name, also include category name
- ModuleStore, ModuleRetrieve, MassModuleStore, StoredModules: add Level and Quality to EngineerModifications; add "Hot" flag; add "BuyPrice" in list of stored items
- In list of stored ships, indicate whether any are 'Hot'
- Include some extra information in the Journal's Scan event, to show each body's parent body info

Version 14 – for v3.0 – beta1 (25/Jan/2018)
- Commodity names and Material names are now localised
- Added NpcCrewRank and NpcCrewPaidWages events
- Added ShipTargetted event
- Added Commander at startup before other loadgame events
- Added Shutdown event
- Fixed station name in "Docking Denied" event
- Added solid composition data when scanning a planet
- Include community goal ID in events: CommunityGoalJoin, CommunityGoalReward, CommunityGoalDiscard

- Added some info about the name of a community goal's top tier, and the global bonus info (if available)
- Added Reputation event, to provide info on player's rep with superpowers
- Include fines in MissionFailed and MissionAbandoned events, where appropriate
- Add Statistics event at startup
- Include StationType info in DockingRequested, DockingGranted, DockingDenied, DockingCancelled, DockingTimeout
- Include UnderAttack event
- Include SystemAddress 64bit id value in FSDJump, Location, Docked, StartJump, NavBeaconScan
- Add StoredModules, and StoredShips
- Added Missions list at startup
- Added info to Cargo/Inventory to show whether cargo is stolen
- Report results of DiscoveryScan to the journal
- Include ScanType in Scan event
- Add FighterDestroyed and FighterRebuilt events
- Add LaunchDrone event
- Add Shipyard Pricelist, and Outfitting pricelist, Market pricelist, written to separate files
- Include BodyID in Location, SupercruiseExit, Scan, ApproachBody, LeaveBody
- Include MarketID in many events where relevant
- Include ship's HullValue, ModulesValue, and Rebuy price in Loadout event
- Fix an error in the ShipyardNew event description
- Note EngineerApply event is no longer generated
- Added "Quality" and "BlueprintID" properties and "Modifiers" array to EngineerCraft
- Added "EngineerLegacyConvert" event
- Added "EngineerID" property to EngineerContribution, EngineerCraft, EngineerProgress events
- Include Engineering data for modified modules in the Loadout event
- Include list of possible modifiable module attributes in appendix
- Added MaterialTrade and TechnologyBroker events
- Added info on the real-time Status.Json file
- Include TotalEarnings in SellExplorationData
- Added ModuleInfo event
- Note Loadout event after using outfitting
- Note Altitude and Heading in Screenshot event
- Added SystemsShutdown event
- Added SRVDestroyed event
- Added Wanted flag in Docked event

Version 13 - In 2.4 Open beta *(24th Aug 2017)*
Version 12 - In 2.4 beta1 *(17th 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 son.org and 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*.<datestamp>.<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 8.45

Status.json – contains frequently-changing info as displayed in the cockpit gui. This is updated when the data changes. See section 11

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 iocalised 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*

Parameters:
- Inventory: array of cargo,
  - Name
  - Count
  - Stolen

Example:
{ "timestamp":"2017-02-10T14:25:51Z", "event":"Cargo", "Inventory":[{ "Name":"syntheticmeat", "Count":2, "Stolen": 0 }, { "Name":"evacuationshelter", "Count":1, "Stolen": 0 }, { "Name":"progenitorcells", "Count":3, "Stolen": 3 }, { "Name":"bioreducinglichen", "Count":1, "Stolen": 0 }, { "Name":"neofabricinsulation", "Count":2, "Stolen": 0 }] }

3.2 ClearSavedGame

When written: If you should ever reset your game

Parameters:
- Name: commander name

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

3.3 Commander

When written: at the start of the LoadGame process

This is written before the inventory, loadout, etc

Parameters:
- Name: commander name

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
- Rebuy

- 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 12.11 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":"refinedfocuscrystals", "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
- Package: selected starter package

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

3.8 LoadGame

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

Parameters:
- Commander: commander name
- 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", "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

- o Passengers_Missions_Delivered
- o Passengers_Missions_Ejected
- Search_And_Rescue
  - o SearchRescue_Traded
  - o SearchRescue_Profit
  - o SearchRescue_Count
- Crafting
  - o Spent_On_Crafting
  - o Count_Of_Used_Engineers
  - o Recipes_Generated
  - o Recipes_Generated_Rank_1
  - o Recipes_Generated_Rank_2
  - o Recipes_Generated_Rank_3
  - o Recipes_Generated_Rank_4
  - o Recipes_Generated_Rank_5
  - o Recipes_Applied
  - o Recipes_Applied_Rank_1
  - o Recipes_Applied_Rank_2
  - o Recipes_Applied_Rank_3
  - o Recipes_Applied_Rank_4
  - o Recipes_Applied_Rank_5
  - o Recipes_Applied_On_Previously_Modified_Modules
- Crew
  - o NpcCrew_TotalWages
  - o NpcCrew_Hired
  - o NpcCrew_Fired
  - o NpcCrew_Died
- Multicrew
  - o Multicrew_Time_Total
  - o Multicrew_Gunner_Time_Total
  - o Multicrew_Fighter_Time_Total
  - o Multicrew_Credits_Total
  - o 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
- StationGovernment
- DistFromStarLS
- StationServices: (Array of strings)
- Wanted: (only if docking when wanted locally)

Example:
{ "timestamp":"2017-08-18T10:52:26Z", "event":"Docked", "StationName":"Goddard Hub", "StationType":"Coriolis", "StarSystem":"HR 3499", "StationFaction":"Labour of HR 3499", "StationGovernment":"$government_Democracy;", "StationGovernment_Localised":"Democracy", "StationAllegiance":"Federation", "StationServices":["Dock", "Autodock", "BlackMarket", "Commodities", "Contacts", "Exploration", "Missions", "Outfitting", "CrewLounge", "Rearm", "Refuel", "Repair", "Shipyard", "Tuning", "MissionsGenerated", "FlightController", "StationOperations", "Powerplay", "SearchAndRescue"], "StationEconomy":"$economy_Industrial;", "StationEconomy_Localised":"Industrial", "DistFromStarLS":129.454132 }

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
- SystemGovernment
- SystemSecurity
- Population
- Wanted
- Factions: an array of info for the local minor factions
  - Name
  - FactionState
  - Government
  - Influence
  - PendingStates: array (if any) with State name and Trend value
  - RecovingStates: array (if any)with State name and Trend value

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:

{ "timestamp":"2017-02-27T15:37:47Z", "event":"FSDJump", "StarSystem":"HR 3316", "StarPos":[100.719,19.813,-51.125], "SystemAllegiance":"Independent", "SystemEconomy":"$economy_Colony;", "SystemEconomy_Localised":"Colony", "SystemGovernment":"$government_Democracy;", "SystemGovernment_Localised":"Democracy", "SystemSecurity":"$SYSTEM_SECURITY_medium;", "SystemSecurity_Localised":"Medium Security", "JumpDist":20.326, "FuelUsed":1.260775, "FuelLevel":12.872868, "Factions":[{ "Name":"Independent HR 3316 Liberals", "FactionState":"Outbreak", "Government":"Democracy", "Influence":0.550000 }, { "Name":"Jet Natural Partners", "FactionState":"None", "Government":"Corporate", "Influence":0.150000 }, { "Name":"Camorra of HR 3316", "FactionState":"None", "Government":"Anarchy", "Influence":0.090000 }, { "Name":"HR 3316 Nobles",

"FactionState":"None", "Government":"Feudal", "Influence":0.210000 }], "SystemFaction":"Independent HR 3316 Liberals", "FactionState":"Outbreak" }

Examples of trending states:
... "Factions":[{ "Name":"Inupiates Patrons of Law", "FactionState":"Lockdown", "Government":"Patronage", "Influence":0.550000, "Allegiance":"Empire", "PendingStates":[{ "State":"Boom", "Trend":0 }, { "State":"CivilUnrest", "Trend":0 }] }, ...

... "Factions":[{ "Name":"IV Comae Berenices Purple Creative", "FactionState":"CivilWar", "Government":"Corporate", "Influence":0.550000, "Allegiance":"Independent", "RecoveringStates":[{ "State":"Boom", "Trend":0 }] }, ...

4.9 LeaveBody

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

Parameters:
- StarSystem
- SystemAddress
- Body
- BodyID

4.10 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 }

4.11 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
- 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")

4.12 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)

4.13 SupercruiseEntry

When written: entering supercruise from normal space

Parameters:
- Starsystem

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

4.14 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" }

4.15 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 }

4.16 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
- 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

Example:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Bounty", "Rewards": [{"Faction":"Federation", "Reward":1000 }, {"Faction":"Nuenets Corp.", "Reward": 10280}],"Target":"Skimmer", "TotalReward":11280, "VictimFaction":"MMU" }

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 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

6.2 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 12.2
- StellarMass: mass as multiple of Sol's mass
- Radius
- AbsoluteMagnitude
- RotationPeriod (seconds)
- SurfaceTemperature
- Luminosity – see 11.9
- 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 12.3
- * Atmosphere – see 12.4
- * AtmosphereType
- * AtmosphereComposition: [array of info]
- * Volcanism – see 12.5
- 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

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 }

6.3 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 }

6.4 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 }

6.5 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 }

6.6 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

6.7 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 }

6.8 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 }

6.9 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 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 }

7.2 CollectCargo

When Written: when scooping cargo from space or planet surface

Parameters:
- Type: cargo type
- Stolen: whether stolen goods

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

7.3 EjectCargo

When Written:

Parameters:
- Type: cargo type
- Count: number of units
- 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" }

7.4 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 }

7.5 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 }

7.6 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")
- StartMarketID(int)
- EndMarketID(int)
- ItemsCollected(int)
- ItemsDelivered(int)
- TotalItemsToDeliver(int)
- Progress:(float)

Example:
{ "timestamp":"2018-02-28T12:07:22Z", "event":"CargoDepot", "MissionID":65393626, "StartMarketID":128005888, "EndMarketID":3535249920, "ItemsCollected":128, "ItemsDelivered":0, "TotalItemsToDeliver":2280, "Progress":0.056140 }

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
- Blueprint: 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

- o Label: string – (see 12.11 below)
- o Value: float
- o OriginalValue: float
- o LessIsGood: bool
- Some modifications have string values:
  - o Label: string
  - o 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":"2017-12-04T16:38:50Z", "event":"EngineerCraft", "Engineer":"Elvira Martuuk", "EngineerID":300160, "Blueprint":"ShieldGenerator_Kinetic", "BlueprintID":128673820, "Level":1, "Slot":"Slot02_Size4", "Quality":0.207000, "Module":"int_shieldgenerator_size3_class5", "Ingredients":[{ "Name":"shieldcyclerecordings", "Name_Localised":"Distorted Shield Cycle Recordings", "Count":1 }], "Modifiers":[{ "Label":"Mass", "Value":4.159000, "OriginalValue":5.000000, "LessIsGood":1 }, { "Label":"Integrity", "Value":73.150002, "OriginalValue":77.000000, "LessIsGood":0 }, { "Label":"PowerDraw", "Value":2.062620, "OriginalValue":2.520000, "LessIsGood":1 }, { "Label":"ShieldGenOptimalMass", "Value":161.699997, "OriginalValue":165.000000, "LessIsGood":0 }, { "Label":"ShieldGenStrength", "Value":122.172012, "OriginalValue":120.000008, "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 for3.0, 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

Parameters
- Engineer: name of engineer
- EngineerID
- Rank: rank reached (when unlocked)
- Progress: progress stage (Invited/Acquainted/Unlocked/Barred)

Examples:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"EngineerProgress", "Progress":"Unlocked", "Engineer":"Elvira Martuuk", "EngineerID":300160 }
{ "timestamp":"2016-06-10T14:32:03Z", "event":"EngineerProgress", "Engineer":"Elvira Martuuk", "EngineerID":300160, "Rank":2 }

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

- o Demand
- o Consumer: bool
- o Producer: bool
- o Rare: bool

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

Separate file:

{ "timestamp":"2017-10-05T10:10:34Z", "event":"Market", "MarketID":128678535, "StationName":"Black Hide", "StarSystem":"Wyrd", "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
  - o Slot
  - o Name
  - o Hot
  - o EngineerModifications (only present if modified)
  - o Level
  - o 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:
{ "timestamp":"2018-02-21T15:23:49Z", "event":"MaterialTrade", "MarketID":3221397760, "TraderType":"encoded", "Paid":{ "Material":"scandatabanks", "Material_Localised":"Classified Scan Databanks", "Category":"$MICRORESOURCE_CATEGORY_Encoded;", "Category_Localised":"Encoded", "Quantity":6, "Category":"$MICRORESOURCE_CATEGORY_Encoded;", "Category_Localised":"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
- 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
- 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
- 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: donation offered (for altruism missions)
- 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
- MissionName
- 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_pulselaser_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 PayFines

When Written: when paying fines

Parameters:
- Amount: (total amount paid , including any broker fee)
- BrokerPercentage (present if paid via a Broker)

Example:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"PayFines", "Amount":1791 }

8.33 PayLegacyFines

When Written: when paying legacy fines

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

8.34 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 }] }

8.35 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 }

8.36 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 }

8.37 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 }

8.38 RepairAll

When written: when repairing everything

Parameters:
- Cost
-

8.39 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 }

8.40 ScientificResearch

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

Parameters:
- MarketID
- Name: material name
- Category
- Count
-

8.41 SearchAndRescue

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

Parameters:
- MarketID
- Name
- Count
- Reward

8.42 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 }

8.43 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 }

8.44 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" }

8.45 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 },
  :
] }
```

8.46 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 }
```

8.47 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 }

8.48 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" }

8.49 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 }

8.50 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 }

8.51 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 }] }

8.52 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
  - 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 }] }

8.53 TechnologyBroker

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

Parameters:
- ItemUnlocked: the name of the new item unlocked (available in Outfitting)
- Ingredients: array of objects
  - Name: name of item
  - Count: number of items used

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 Other Events

10.1 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 }

10.2 ApproachSettlement

When written: when approaching a planetary settlement

Parameters:
- Name
- MarketID

10.3 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)

10.4 CockpitBreached

When written: when cockpit canopy is breached

Parameters: none

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

10.5 CommitCrime

When written: when a crime is recorded against the player

Parameters:
- CrimeType – see 12.6
- 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 }

10.6 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

10.7 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
-

10.8 CrewMemberJoins

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

Parameters:
- Crew: player's commander name

10.9 CrewMemberQuits

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

Parameters:
- Crew: player's commander name

10.10   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" }

10.11   DatalinkScan

When written: when scanning a data link

Parameters:
- Message: message from data link

10.12   DatalinkVoucher

When written: when scanning a datalink generates a reward

Parameters:
- Reward: value in credits
- VictimFaction
- PayeeFaction

10.13   DataScanned

When written: when scanning some types of data links

Parameters:
- Type

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

10.14   DockFighter

When written: when docking a fighter back with the mothership

Parameters: none

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

10.15   DockSRV

When written: when docking an SRV with the ship

Parameters: none

10.16   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

10.17   FighterRebuilt

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

Parameters:
- Loadout

10.18   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 }

10.19   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

10.20   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

10.21   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

10.22   JoinACrew

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

Parameters:
- Captain: Helm player's commander name

10.23   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

10.24   LaunchDrone

When written: when using any type of drone/limpet

Parameters:
- Type: one of:
"Hatchbreaker", "FuelTransfer", "Collection", "Prospector", "Repair", "Research", "Decontamination"

10.25   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 }

10.26   LaunchSRV

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

Parameters:
- Loadout

10.27   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 }
] }

10.28   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

10.29   NpcCrewPaidWage

This is written when crew receive wages

Parameters:
- NpcCrewId

- NpcCrewName
- Amount

10.30   NpcCrewRank

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

Parameters:
- NpcCrewId
- NpcCrewName
- RankCombat

10.31   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 }

10.32   QuitACrew

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

Parameters:
- Captain: Helm player's commander name

10.33   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"] }

10.34   ReceiveText

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

Parameters:
- From

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

10.35   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

10.36   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

10.37   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" }

10.38   SelfDestruct

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

Parameters: none

10.39   SendText

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

Parameters:
- To

- Message

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

10.40 Shutdown

When written: on a clean shutdown of the game

Parameters: none

10.41 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 }] }

10.42 SystemsShutdown

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

Parameters: none

10.43 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 }

10.44 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" }

10.45   WingAdd

When written: another player has joined the wing

Parameters:
- Name

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

10.46   WingInvite

When written: when the player is invited to a wing

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

10.47   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"] }

10.48   WingLeave

When written: this player has left a wing

Parameters: none

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

11 Status 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:
- o Flags: multiple flags encoded as bits in an integer (see below)
- o Pips: an array of 3 integers representing energy distribution (in half-pips)
- o Firegroup: the currently selected firegroup number
- o GuiFocus: the selected GUI screen
- o Latitude (if on or near a planet)
- o Altitude
- o Longitude
- o 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 Turret
14	16384	0000 4000	Srv UnderShip
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

Examples:

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

{ "timestamp":"2017-12-07T12:03:14Z", "event":"Status", "Flags":18874376, "Pips":[4,8,0], "FireGroup":0, "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

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

12 Appendix

12.1 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 Pennliess', 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'

12.2 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

12.3 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

12.4 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

12.5 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

12.6 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

12.7 BodyType values

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

12.8 Gases in AtmosphereComposition

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

12.9 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,

12.10 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

12.11   Module Attributes

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

(note, based on enum eItemInfoFieldType)

Size,
Class,
Mass,
Integrity,
PowerDraw,
BootTime,
ShieldBankSpinUp,
ShieldBankDuration,
ShieldBankReinforcement,
ShieldBankHeat,
DamagePerSecond,
Damage,
DistributorDraw,
ThermalLoad,
ArmourPenetration,
MaximumRange,
ShotSpeed,
RateOfFire,
BurstRateOfFire,
BurstSize,
AmmoClipSize,
AmmoMaximum,
RoundsPerShot,
ReloadTime,
BreachDamage,
MinBreachChance,
MaxBreachChance,
Jitter,
WeaponMode,
DamageType,
ShieldGenMinimumMass,
ShieldGenOptimalMass,
ShieldGenMaximumMass,
ShieldGenMinStrength,
ShieldGenStrength,
ShieldGenMaxStrength,
RegenRate,
BrokenRegenRate,
EnergyPerRegen,
FSDOptimalMass,
FSDHeatRate,
MaxFuelPerJump,
EngineMinimumMass,
EngineOptimalMass,
MaximumMass,

EngineMinPerformance,
EngineOptPerformance,
EngineMaxPerformance,
EngineHeatRate,
PowerCapacity,
HeatEfficiency,
WeaponsCapacity,
WeaponsRecharge,
EnginesCapacity,
EnginesRecharge,
SystemsCapacity,
SystemsRecharge,
DefenceModifierHealthMultiplier,
DefenceModifierHealthAddition,
DefenceModifierShieldMultiplier,
DefenceModifierShieldAddition,
KineticResistance,
ThermicResistance,
ExplosiveResistance,
CausticResistance,
FSDInterdictorRange,
FSDInterdictorFacingLimit,
ScannerRange,
DiscoveryScannerRange,
DiscoveryScannerPassiveRange,
MaxAngle,
ScannerTimeToScan,
ChaffJamDuration,
ECMRange,
ECMTimeToCharge,
ECMActivePowerConsumption,
ECMHeat,
ECMCooldown,
HeatSinkDuration,
ThermalDrain,
NumBuggySlots,
CargoCapacity,
MaxActiveDrones,
DroneTargetRange,
DroneLifeTime,
DroneSpeed,
DroneMultiTargetSpeed,
DroneFuelCapacity,
DroneRepairCapacity,
DroneHackingTime,

DroneMinJettisonedCargo,
DroneMaxJettisonedCargo,
FuelScoopRate,
FuelCapacity,
OxygenTimeCapacity,
RefineryBins,
AFMRepairCapacity,
AFMRepairConsumption,
AFMRepairPerAmmo,
MaxRange,
SensorTargetScanAngle,
Range,
VehicleCargoCapacity,
VehicleHullMass,
VehicleFuelCapacity,
VehicleArmourHealth,
VehicleShieldHealth,
FighterMaxSpeed,
FighterBoostSpeed,
FighterPitchRate,
FighterDPS,
FighterYawRate,
FighterRollRate,
CabinCapacity,
CabinClass,
DisruptionBarrierRange,
DisruptionBarrierChargeDuration,
DisruptionBarrierActivePower,
DisruptionBarrierCooldown,
WingDamageReduction,
WingMinDuration,
WingMaxDuration,
ShieldSacrificeAmountRemoved,
ShieldSacrificeAmountGiven,
FSDJumpRangeBoost,
FSDFuelUseIncrease,
BoostSpeedMultiplier,
BoostAugmenterPowerUse,
ModuleDefenceAbsorption,
FalloffRange,
DSS_RangeMult,
DSS_AngleMult,
DSS_RateMult,

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