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A short example of a player journal file (*out of date, some events may have additional data*):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

2.1 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.

2.2 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*

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

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

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

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

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

3.2 ClearSavedGame

When written: If you should ever reset your game

Parameters:
- Name: commander name
- FID: player id

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

3.3 Commander

When written: at the start of the LoadGame process

This is written before the inventory, loadout, etc

Parameters:
- Name: commander name

- FID: player id

3.4 Loadout

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

Parameters:
- Ship: current ship type
- ShipID: ship id number (indicates which of your ships you are in)
- ShipName: user-defined ship name
- ShipIdent: user-defined ship ID string
- HullValue – may not always be present
- ModulesValue – may not always be present
- HullHealth
- UnladenMass – Mass of Hull and Modules, excludes fuel and cargo
- FuelCapacity: { Main: , Reserve: }
- CargoCapacity
- MaxJumpRange: (based on zero cargo, and just enough fuel for 1 jump)
- Rebuy
- Hot: (if wanted at startup – may not always be present)

- Modules: array of installed items, each with:
  - Slot: slot name
  - Item: module name - lowercase
  - 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 §14.11 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
- FID: player id
- Package: selected starter package

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

3.8 LoadGame

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

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

Example:

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

3.9 Passengers

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

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

3.10 Powerplay

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

Parameters:
- Power: name

- Rank
- Merits
- Votes
- TimePledged (time in seconds)

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

3.11 Progress

When written: at startup

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

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

3.12 Rank

When written: at startup

Parameters:

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

Example:

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

3.13 Reputation

When written: at startup (after Rank and Progress)

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

Parameters:
- Empire
- Federation
- Independent
- Alliance

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

3.14 Statistics

When written: at startup
This line contains the information displayed in the statistics panel on the right side of the cockpit

Parameters:
- Bank_Account
  - Current_Wealth Spent_On_Ships
  - Spent_On_Outfitting Spent_On_Repairs
  - Spent_On_Fuel
  - Spent_On_Ammo_Consumables
  - Insurance_Claims
  - Spent_On_Insurance
- Combat
  - Bounties_Claimed
  - Bounty_Hunting_Profit
  - Combat_Bonds
  - Combat_Bond_Profits
  - Assassinations
  - Assassination_Profits
  - Highest_Single_Reward
  - Skimmers_Killed
- Crime
  - Fines
  - Total_Fines
  - Bounties_Received
  - Total_Bounties
  - Highest_Bounty
- Smuggling
  - Black_Markets_Traded_With
  - Black_Markets_Profits
  - Resources_Smuggled
  - Average_Profit
  - Highest_Single_Transaction
- Trading
  - Markets_Traded_With
  - Market_Profits
  - Resources_Traded
  - Average_Profit
  - Highest_Single_Transaction

- Mining
  - Mining_Profits
  - Quantity_Mined
  - Materials_Collected
- Exploration
  - Systems_Visited
  - Fuel_Scooped
  - Fuel_Purchased
  - Exploration_Profits
  - Planets_Scanned_To_Level_2
  - Planets_Scanned_To_Level_3
  - Highest_Payout
  - Total_Hyperspace_Distance
  - Total_Hyperspace_Jumps
  - Greatest_Distance_From_Start
  - Time_Played
- Passengers
  - Passengers_Missions_Bulk
  - Passengers_Missions_VIP
  - Passengers_Missions_Delivered
  - Passengers_Missions_Ejected
- Search_And_Rescue
  - SearchRescue_Traded
  - SearchRescue_Profit
  - SearchRescue_Count
- Crafting
  - Spent_On_Crafting
  - Count_Of_Used_Engineers
  - Recipes_Generated
  - Recipes_Generated_Rank_1
  - Recipes_Generated_Rank_2
  - Recipes_Generated_Rank_3
  - Recipes_Generated_Rank_4
  - Recipes_Generated_Rank_5
  - Recipes_Applied
  - Recipes_Applied_Rank_1
  - Recipes_Applied_Rank_2
  - Recipes_Applied_Rank_3
  - Recipes_Applied_Rank_4
  - Recipes_Applied_Rank_5
  - Recipes_Applied_On_Previously_Modified_Modules
- Crew
  - NpcCrew_TotalWages
  - NpcCrew_Hired
  - NpcCrew_Fired
  - NpcCrew_Died
- Multicrew
  - Multicrew_Time_Total
  - Multicrew_Gunner_Time_Total
  - Multicrew_Fighter_Time_Total
  - Multicrew_Credits_Total
  - Multicrew_Fines_Total

Note times are in seconds

Example:

{ "timestamp":"2017-09-25T15:18:31Z", "event":"Statistics", "Bank_Account":{ "Current_Wealth":148827050, "Spent_On_Ships":14499177, "Spent_On_Outfitting":30785093, "Spent_On_Repairs":17716, "Spent_On_Fuel":1647, "Spent_On_Ammo_Consumables":0, "Insurance_Claims":4, "Spent_On_Insurance":1397620 }, "Combat":{ "Bounties_Claimed":0, "Bounty_Hunting_Profit":0, "Combat_Bonds":0, "Combat_Bond_Profits":0, "Assassinations":0, "Assassination_Profits":0, "Highest_Single_Reward":0, "Skimmers_Killed":0 }, "Crime":{ "Fines":0, "Total_Fines":0, "Bounties_Received":0, "Total_Bounties":0, "Highest_Bounty":0 }, "Smuggling":{ "Black_Markets_Traded_With":0, "Black_Markets_Profits":0, "Resources_Smuggled":0, "Average_Profit":0, "Highest_Single_Transaction":0 }, "Trading":{ "Markets_Traded_With":3, "Market_Profits":40700, "Resources_Traded":23, "Average_Profit":4070, "Highest_Single_Transaction":17961 }, "Mining":{ "Mining_Profits":0, "Quantity_Mined":0, "Materials_Collected":100 }, "Exploration":{ "Systems_Visited":228, "Fuel_Scooped":111, "Fuel_Purchased":0, "Exploration_Profits":304469, "Planets_Scanned_To_Level_2":39, "Planets_Scanned_To_Level_3":15, "Highest_Payout":52503, "Total_Hyperspace_Distance":844927, "Total_Hyperspace_Jumps":295, "Greatest_Distance_From_Start":65222.47204614, "Time_Played":651060 }, "Passengers":{ "Passengers_Missions_Bulk":0, "Passengers_Missions_VIP":0, "Passengers_Missions_Delivered":0, "Passengers_Missions_Ejected":0 }, "Search_And_Rescue":{ "SearchRescue_Traded":12, "SearchRescue_Profit":19467, "SearchRescue_Count":8 }, "Crafting":{ "Spent_On_Crafting":0, "Count_Of_Used_Engineers":2, "Recipes_Generated":28, "Recipes_Generated_Rank_1":9, "Recipes_Generated_Rank_2":6, "Recipes_Generated_Rank_3":9, "Recipes_Generated_Rank_4":4, "Recipes_Generated_Rank_5":0, "Recipes_Applied":21, "Recipes_Applied_Rank_1":8, "Recipes_Applied_Rank_2":5, "Recipes_Applied_Rank_3":7, "Recipes_Applied_Rank_4":1, "Recipes_Applied_Rank_5":0, "Recipes_Applied_On_Previously_Modified_Modules":0 }, "Crew":{ "NpcCrew_TotalWages":0, "NpcCrew_Hired":0, "NpcCrew_Fired":0, "NpcCrew_Died":0 }, "Multicrew":{ "Multicrew_Time_Total":23327, "Multicrew_Gunner_Time_Total":14241, "Multicrew_Fighter_Time_Total":6070, "Multicrew_Credits_Total":0, "Multicrew_Fines_Total":0 } }

4   Travel

4.1 ApproachBody

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

Parameters:
- StarSystem
- SystemAddress
- Body
- BodyID

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

4.2 Docked

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

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

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

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

"StationEconomy_Localised":"Refinery", "StationEconomies":[{ "Name":"$economy_Refinery;", "Name_Localised":"Refinery", "Proportion":0.760000 }, { "Name":"$economy_Extraction;", "Name_Localised":"Extraction", "Proportion":0.240000 }], "DistFromStarLS":10.061876 }

StationServices can include:

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

New in v3.7: shop, carriermanagement, carrierfuel, carriervendor, livery, modulepacks, voucherredemption

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
  - Name
  - FactionState
- SystemAllegiance
- SystemEconomy
- SystemSecondEconomy
- SystemGovernment
- SystemSecurity
- Population
- Wanted
- Factions: an array of info for the local minor factions
  - Name
  - FactionState
  - Government
  - Influence
  - Happiness
  - MyReputation
  - PendingStates: array (if any) with State name and Trend value
  - RecovingStates: array (if any)with State name and Trend value
  - ActiveStates: array with State names (Note active states do not have a Trend value)
  - SquadronFaction:true (if player is in squadron aligned to this faction)
  - HappiestSystem:true (if player squadron faction, and this is happiest system)
  - HomeSystem:true(if player squadron faction, and this is home system)
- Conflicts: an array of info about local conflicts (if any)
  - WarType

- o Status
- o Faction1: { Name, Stake, WonDays }
- o Faction2: { Name, Stake, WonDays }

If the player is pledged to a Power in Powerplay, and the star system is involved in powerplay,
- Powers: a json array with the names of any powers contesting the system, or the name of the controlling power
- PowerplayState: the system state – one of ("InPrepareRadius", "Prepared", "Exploited", "Contested", "Controlled", "Turmoil", "HomeSystem")
-

Example:

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

Example:

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

Example Conflict data:
"Conflicts": [{ "WarType": "war", "Status": "active",
"Faction1": {"Name": "Movement for LHS 3163 League","Stake": "","WonDays": 0},
"Faction2": {"Name": "Official i Bootis Liberty Party","Stake": "Red Bronco Farms","WonDays": 1}
}]

4.9 FSDTarget

When written: when selecting a star system to jump to

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

Parameters:
- Starsystem
- Name
- RemainingJumpsInRoute
- StarClass

4.10 LeaveBody

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

Parameters:
- StarSystem
- SystemAddress
- Body
- BodyID

4.11 Liftoff

When written: when taking off from planet surface

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

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

4.12 Location

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

Parameters:
- StarSystem: name of destination starsystem
- SystemAddress
- StarPos: star position, as a Json array [x, y, z], in light years
- Body: star or planet's body name
- BodyID
- BodyType
- DistFromStarLS: (unless close to main star)
- Docked: (bool)
- Latitude (If landed)
- Longitude (if landed)
- StationName: station name, (if docked)
- StationType: (if docked)

- MarketID: (if docked)
- SystemFaction: star system controlling faction
  - Name
  - FactionState
- SystemAllegiance
- SystemEconomy
- SystemSecondEconomy
- SystemGovernment
- SystemSecurity
- Wanted
- Factions: an array with info on local minor factions (similar to FSDJump)
- Conflicts: an array with info on local conflicts (similar to FSDJump)

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

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

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

If starting docked in a station, also include:

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

4.13 StartJump

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

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

4.14 SupercruiseEntry

When written: entering supercruise from normal space

Parameters:
- Starsystem

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

4.15 SupercruiseExit

When written: leaving supercruise for normal space

Parameters:
- Starsystem
- Body
- BodyID
- BodyType

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

4.16 Touchdown

When written: landing on a planet surface

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

The NearestDestination is included if within 50km of a location listed in the nav panel

Example:
{ "timestamp":"2019-05-13T13:20:18Z", "event":"Touchdown", "PlayerControlled":true, "Latitude":10.503607, "Longitude":102.789810, "NearestDestination":"$SAA_Unknown_Signal:#type=$SAA_SignalType_Geological;:#index=9;", "NearestDestination_Localised":"Surface signal: Geological (9)" }

4.17 Undocked

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

Parameters:
- StationName: name of station
- MarketID

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

4.18 Route

When plotting a multi-star route, the file "Route.json" is written in the same directory as the journal, with a list of stars along that route

Example:

{ "timestamp":"2020-04-27T08:02:52Z", "event":"Route", "Route":[

{ "StarSystem":1733120004818, "StarPos":[-19.75000,41.78125,-3.18750], "StarClass":"K" },

{ "StarSystem":5068732704169, "StarPos":[-15.25000,39.53125,-2.25000], "StarClass":"M" }

 ] }

Note this may be changed for final 3.7 release to use tag name "SystemAddress" instead of "StarSystem" for better consistency

5 Combat

5.1 Bounty

When written: player is awarded a bounty for a kill

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

When the bounty is for a skimmer, it's slightly different:
- Faction: faction paying bounty
- Target: type of target
- Reward: amount
- VictimFaction

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

5.2 CapShipBond

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

Parameters:
- Reward: value of award
- AwardingFaction
- VictimFaction

5.3 Died

When written: player was killed

Parameters:
- KillerName
- KillerShip
- KillerRank

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

5.4 Died

When written: player was killed by a wing

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

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

5.5 EscapeInterdiction

When written: Player has escaped interdiction

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

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

5.6 FactionKillBond

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

Parameters:
- Reward
- AwardingFaction
- VictimFaction

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

5.7 FighterDestroyed

When written: when a ship-launched fighter is destroyed

Parameters: none

5.8 HeatDamage

When written: when taking damage due to overheating

Parameters:

- ID

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
- Power [*]

[*] when targetting another ship, if the player is aligned in powerplay, and the target is also aligned to a power, include the info

5.16 SRVDestroyed

When written: when the player's SRV is destroyed

Parameters:

- ID

5.17 UnderAttack

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

Parameters:
- Target: (Fighter/Mothership/You)

6   Exploration

6.1 CodexEntry

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

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

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

The "NearestDestination" is added if within 50km of a location listed in the navigation panel

Example:
{ "timestamp":"2019-05-13T13:28:51Z", "event":"CodexEntry", "EntryID":1400159, "Name":"$Codex_Ent_IceFumarole_CarbonDioxideGeysers_Name;", "Name_Localised":"Carbon Dioxide Ice Fumarole", "SubCategory":"$Codex_SubCategory_Geology_and_Anomalies;", "SubCategory_Localised":"Geology and anomalies", "Category":"$Codex_Category_Biology;", "Category_Localised":"Biological and Geological", "Region":"$Codex_RegionName_18;", "Region_Localised":"Inner Orion Spur", "System":"Hermitage", "SystemAddress":5363877956440, "NearestDestination":"$SAA_Unknown_Signal:#type=$SAA_SignalType_Geological;:#index=9;", "NearestDestination_Localised":"Surface signal: Geological (9)", "IsNewEntry":true, "VoucherAmount":50000 }

6.2 DiscoveryScan

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

Parameters:
- SystemAddress
- Bodies: number of new bodies discovered

6.3 Scan

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

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

Parameters(star)
- ScanType
- StarSystem: name
- SystemAddress
- Bodyname: name of body
- BodyID
- DistanceFromArrivalLS
- StarType: Stellar classification (for a star) – see §14.2
- Subclass: Star's heat classification 0..9
- StellarMass: mass as multiple of Sol's mass
- Radius
- AbsoluteMagnitude
- RotationPeriod (seconds)
- SurfaceTemperature
- Luminosity – see §14.9
- Age_MY: age in millions of years
- Rings: [array] – if present
- WasDiscovered: bool
- WasMapped: bool

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 §14.3
- * Atmosphere – see §14.4
- * AtmosphereType
- * AtmosphereComposition: [array of info]
- * Volcanism – see §14.5
- SurfaceGravity
- * SurfaceTemperature
- * SurfacePressure
- * Landable: true (if landable)
- * Materials: JSON array with objects with material names and percentage occurrence
- *Composition: structure containing info on solid composition
  - Ice
  - Rock
  - Metal
- * Rings: [array of info] – if rings present
- * ReserveLevel: (Pristine/Major/Common/Low/Depleted) – if rings present

If rotating:
- RotationPeriod (seconds)
- Axial tilt

Orbital Parameters for any Star/Planet/Moon (except main star of single-star system)
- SemiMajorAxis
- Eccentricity
- OrbitalInclination
- Periapsis
- OrbitalPeriod

Rings properties*
- Name
- RingClass
- MassMT – ie in megatons
- InnerRad
- OuterRad

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

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

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

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

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

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

6.4 FSSAllBodiesFound

When written: after having identified all bodies in the system

Parameters:
- SystemName

- SystemAddress
- Count

6.5 FSSDiscoveryScan

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

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

6.6 FSSSignalDiscovered

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

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

6.7 MaterialCollected

When Written: whenever materials are collected

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

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

6.8 MaterialDiscarded

When Written: if materials are discarded

Parameters:
- Category
- Name
- Count

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

6.9 MaterialDiscovered

When Written: when a new material is discovered

Parameters:
- Category
- Name
- DiscoveryNumber

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

6.10 MultiSellExplorationData

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

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

Example

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

6.11 NavBeaconScan

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

Parameters:
- NumBodies
- SystemAddress

6.12 BuyExplorationData

When Written: when buying system data via the galaxy map

Parameters:
- System

- Cost

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

6.13 SAAScanComplete

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

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

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

6.14 SAASignalsFound

When written: when using SAA scanner on a planet or rings

Parameters:
- SystemAddress
- BodyName
- BodyID
- Signals: (array)
  - Type
  - Count

Examples:
{ "timestamp":"2019-04-17T13:38:18Z", "event":"SAASignalsFound", "BodyName":"Hermitage 4 A Ring", "SystemAddress":5363877956440, "BodyID":11, "Signals":[{ "Type":"LowTemperatureDiamond", "Type_Localised":"Low Temperature Diamonds", "Count":1 }, { "Type":"Alexandrite", "Count":1 }] }
{ "timestamp":"2019-04-17T13:40:39Z", "event":"SAASignalsFound", "BodyName":"Hermitage 4 b", "SystemAddress":5363877956440, "BodyID":13, "Signals":[{ "Type":"$SAA_SignalType_Geological;", "Type_Localised":"Geological", "Count":14 }] }

6.15 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 }

6.16 Screenshot

When Written: when a screen snapshot is saved

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

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

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

7 Trade

7.1 AsteroidCracked

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

Parameters:
- Body: name of nearest body

7.2 BuyTradeData

When Written: when buying trade data in the galaxy map

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

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

7.3 CollectCargo

When Written: when scooping cargo from space or planet surface

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

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

7.4 EjectCargo

When Written:

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

If the cargo is related to powerplay delivery *from outlying systems back to the centre*:
- PowerplayOrigin: starsystem name

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

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

7.5 MarketBuy

When Written: when purchasing goods in the market

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

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

7.6 MarketSell

When Written: when selling goods in the market

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

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

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

7.7 MiningRefined

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

Parameters:
- Type: cargo type

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

8 Station Services

8.1 BuyAmmo

When Written: when purchasing ammunition

Parameters:
- Cost

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

8.2 BuyDrones

When Written: when purchasing drones

Parameters:
- Type
- Count
- BuyPrice
- TotalCost

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

8.3 CargoDepot

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

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

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

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

Example:

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

8.4 CommunityGoal

When written: when checking the status of a community goal

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

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

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

If the community goal has reached the first success tier:
  - TierReached
  - Bonus

Example:

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

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

8.5 CommunityGoalDiscard

When written: when opting out of a community goal

Parameters:
- CGID
- Name
- System

8.6 CommunityGoalJoin

When Written: when signing up to a community goal

Parameters:
- CGID
- Name
- System

8.7 CommunityGoalReward

When Written: when receiving a reward for a community goal

Parameters:
- CGID
- Name
- System
- Reward

8.8 CrewAssign

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

Parameters:
- Name
- CrewID
- Role

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

8.9 CrewFire

When written: when dismissing a member of crew

Parameters:

- Name
- CrewID

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

8.10 CrewHire

When written: when engaging a new member of crew

Parameters:
- Name
- CrewID
- Faction
- Cost
- Combat Rank

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

8.11 EngineerApply

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

8.12 EngineerContribution

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

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

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

8.13 EngineerCraft

When Written: when requesting an engineer upgrade

Parameters:
- Engineer: name of engineer
- EngineerID
- BlueprintName: name of blueprint
- BlueprintID
- Level: crafting level
- Quality: float
- ApplyExperimentalEffect: (when applying a new effect)
- Ingredients: JSON array of objects with names and quantities of materials required
- Modifiers: JSON array of modification data
  - Label: string – (see §14.11 below)
  - Value: float
  - OriginalValue: float
  - LessIsGood: bool
- Some modifications have string values:
  - Label: string
  - ValueStr: string

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

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

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

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

8.14 EngineerLegacyConvert

When written: when converting a pre-2.4 engineered module

This is generated when converting, or previewing a conversion of a legacy module to the new system. Due to the nature of the changes made 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

Also a summary of progress for all engineers at startup

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

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

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

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

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

8.16 FetchRemoteModule

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

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

8.17 Market

When written: when accessing the commodity market in a station

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

Parameters:
- MarketID
- StationName
- StarSystem

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

Example: in the journal:
{ "timestamp":"2017-10-05T10:11:38Z", "event":"Outfitting", "MarketID":128678535, "StationName":"Black Hide", "StarSystem":"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
    - Slot
    - Name
    - Hot
    - EngineerModifications (only present if modified)
    - Level
    - Quality

8.19 MaterialTrade

When written: when exchanging materials at the Material trader contact

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

Example: (note categories simplified)

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

8.20 MissionAbandoned

When Written: when a mission has been abandoned

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

Example:

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

8.21 MissionAccepted

When Written: when starting a mission

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

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

Examples:

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

8.22 MissionCompleted

When Written: when a mission is completed

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

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

TrendValue can be UpGood, UpBad, DownGood, or DownBad
Reputation and *Influence* are a set of + indicators (eg "+++")

Example:
{ "timestamp":"2018-12-19T21:41:09Z", "event":"MissionCompleted", "Faction":"Inara Nexus", "Name":"Mission_Courier_Elections_name", "MissionID":442511682, "TargetFaction":"Tougeir Blue Clan", "DestinationSystem":"Tougeir", "DestinationStation":"Janes Dock", "Reward":10000, "FactionEffects":[
{ "Faction":"Tougeir Blue Clan", "Effects":[{ "Effect":"$MISSIONUTIL_Interaction_Summary_EP_up;", "Effect_Localised":"The economic status of $#MinorFaction; has improved in the $#System; system.", "Trend":"UpGood" }
], "Influence":[{ "SystemAddress":5067927397769, "Trend":"UpGood", "Influence":"++++" }], "ReputationTrend":"UpGood", "Reputation":"" },

{ "Faction":"Inara Nexus", "Effects":[{ "Effect":"$MISSIONUTIL_Interaction_Summary_EP_up;",
"Effect_Localised":"The economic status of $#MinorFaction; has improved in the $#System;
system.", "Trend":"UpGood" }
], "Influence":[ ], "ReputationTrend":"UpGood", "Reputation":"+" }] }

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

8.23 MissionFailed

When Written: when a mission has failed

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

8.24 MissionRedirected

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

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

Example:

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

8.25 ModuleBuy

When Written: when buying a module in outfitting

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

If existing module is stored:
- StoredItem: item being stored

If replacing an existing module:
- SellItem: item being sold
- SellPrice: sale price

Example:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"ModuleBuy", "Slot":"MediumHardpoint2", "SellItem":"hpt_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 PayBounties

When written: when paying off bounties

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

Example:

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

8.33 PayFines

When Written: when paying fines

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

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

8.34 PayLegacyFines

When Written: when paying legacy fines

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

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

8.35 RedeemVoucher

When Written: when claiming payment for combat bounties and bonds

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

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

8.36 RefuelAll

When Written: when refuelling (full tank)

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

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

8.37 RefuelPartial

When Written: when refuelling (10%)

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

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

8.38 Repair

When Written: when repairing the ship

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

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

Note when repairing on a FleetCarrier, you can now get a list of the modules repaired:

{ "timestamp":"2020-03-31T13:39:42Z", "event":"Repair", "Items":["$hpt_dumbfiremissilerack_fixed_large_name;", "$hpt_beamlaser_gimbal_medium_name;", "$hpt_railgun_fixed_medium_name;", "$hpt_beamlaser_gimbal_medium_name;", "$hpt_dumbfiremissilerack_fixed_large_name;"], "Cost":34590 }

8.39 RepairAll

When written: when repairing everything

Parameters:
- Cost
-

8.40 RestockVehicle

When Written: when purchasing an SRV or Fighter

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

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

8.41 ScientificResearch

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

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

8.42 SearchAndRescue

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

Parameters:
- MarketID
- Name
- Count
- Reward

8.43 SellDrones

When Written: when selling unwanted drones back to the market

Parameters:
- Type
- Count
- SellPrice
- TotalSale

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

8.44 SellShipOnRebuy

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

Parameters:
- ShipType
- System
- SellShipId
- ShipPrice

Example:

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

8.45 SetUserShipName

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

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

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

8.46 Shipyard

When written: when accessing shipyard in a station

Parameters:
- MarketID
- StationName
- StarSystem

The full price list is written to a separate file, in the same folder as the journal, *Shipyard.json*
- Horizons: bool
- AllowCobraMkIV: bool
- Pricelist: array of objects
  - ShipType
  - ShipPrice

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

In the separate shipyard.json file:

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

8.47 ShipyardBuy

When Written: when buying a new ship in the shipyard

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

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

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

8.48 ShipyardNew

When written: after a new ship has been purchased

Parameters:
- ShipType
- NewShipID

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

8.49 ShipyardSell

When Written: when selling a ship stored in the shipyard

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

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

8.50 ShipyardTransfer

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

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

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

8.51 ShipyardSwap

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

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

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

8.52 StoredModules

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

Parameters:
- MarketID: current market
- Items: (array of objects)
  - Name
  - StarSystem
  - MarketID: where the module is stored
  - StorageSlot
  - TransferCost

- o TransferTime
- o Hot
- o EngineerModifications: (recipe name)
- o Level
- o Quality
- o InTransit:bool

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

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

8.53 StoredShips

When written: when visiting shipyard

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

If the ship is in transit:
- o InTransit: true

If the ship is not in transit:
- o StarSystem
- o ShipMarketID
- o TransferPrice
- o TransferType

Example:
{ "timestamp":"2017-10-04T10:07:21Z", "event":"StoredShips", "StationName":"Jameson
Memorial", "StarSystem":"Shinrarta Dezhra", "ShipsHere":[{ "ShipID":64, "ShipType":"sidewinder",

"Value":567962 }, { "ShipID":20, "ShipType":"empire_eagle", "Value":6373956 }], "ShipsRemote":[{ "ShipID":0, "ShipType":"CobraMkIII", "StarSystem":"Beta-1 Tucanae", "TransferPrice":3777, "TransferTime":1590, "Value":9464239 }] }

8.54 TechnologyBroker

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

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

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

9 Powerplay

9.1 PowerplayCollect

When written: when collecting powerplay commodities for delivery

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

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

9.2 PowerplayDefect

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

Parameters:
- FromPower
- ToPower

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

9.3 PowerplayDeliver

When written: when delivering powerplay commodities

Parameters:
- Power
- Type
- Count

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

9.4 PowerplayFastTrack

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

Parameters:
- Power
- Cost

9.5 PowerplayJoin

When written: when joining up with a power

Parameters:
- Power

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

9.6 PowerplayLeave

When written: when leaving a power

Parameters:
- Power

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

9.7 PowerplaySalary

When written: when receiving salary payment from a power

Parameters:
- Power
- Amount

9.8 PowerplayVote

When written: when voting for a system expansion

Parameters:
- Power
- Votes
- System

9.9 PowerplayVoucher

When written: when receiving payment for powerplay combat

Parameters:
- Power
- Systems:[name,name]

10 Squadrons

10.1 AppliedToSquadron

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

10.2 DisbandedSquadron

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

10.3 InvitedToSquadron

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

10.4 JoinedSquadron

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

10.5 KickedFromSquadron

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

10.6 LeftSquadron

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

10.7 SharedBookmarkToSquadron

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

10.8 SquadronCreated

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

10.9 SquadronDemotion

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

10.10   SquadronPromotion

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

10.11   SquadronStartup

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

Parameters:
- SquadronName
- CurrentRank

10.12   WonATrophyForSquadron

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

11 Fleet Carriers

11.1 CarrierJump

This is similar to FSDJump and Location, but it is written if the player is online and docked at a fleet carrier when the carrier jumps. Note it does not include the distance jumped, or fuel used.

Example:

{ "timestamp":"2020-03-25T15:55:56Z", "event":"CarrierJump", "Docked":true, "StationName":"FC L14X1J", "StationType":"FleetCarrier", "MarketID":3700005632, "StationFaction":{ "Name":"FleetCarrier" }, "StationGovernment":"$government_Carrier;", "StationGovernment_Localised":"Private Ownership ", "StationServices":["dock", "autodock", "blackmarket", "commodities", "outfitting", "crewlounge", "rearm", "refuel", "repair", "shipyard", "tuning", "engineer", "flightcontroller", "stationoperations", "stationMenu", "carriermanagement", "carrierfuel", "voucherredemption"], "StationEconomy":"$economy_Extraction;", "StationEconomy_Localised":"Extraction", "StationEconomies":[{ "Name":"$economy_Extraction;", "Name_Localised":"Extraction", "Proportion":1.000000 }], "StarSystem":"Hermitage", "SystemAddress":5363877956440, "StarPos":[-28.75000,25.00000,10.43750], "SystemAllegiance":"", "SystemEconomy":"$economy_None;", "SystemEconomy_Localised":"None", "SystemSecondEconomy":"$economy_None;", "SystemSecondEconomy_Localised":"None", "SystemGovernment":"$government_None;", "SystemGovernment_Localised":"None", "SystemSecurity":"$GAlAXY_MAP_INFO_state_anarchy;", "SystemSecurity_Localised":"Anarchy", "Population":0, "Body":"Hermitage", "BodyID":0, "BodyType":"Star", "SystemFaction":{ "Name":"FleetCarrier" } }

11.2 CarrierBuy

Player has bought a fleet carrier

- BoughtAtmarket: marketid
- CarrierID : marketid
- Location: starsystem name
- Price: number
- Variant: string
- Callsign: string

{ "timestamp":"2020-03-11T15:31:46Z", "event":"CarrierBuy", "CarrierID":3700029440, "BoughtAtMarket":3221301504, "Location":"Kakmbutan", "SystemAddress":3549513615723, "Price":4875000000, "Variant":"CarrierDockB", "Callsign":"P07-V3L" }

11.3 CarrierStats

When owner opens carrier management

- CarrierID: marketid
- Callsign: string
- Name: string
- DockingAccess: all/none/friends/squadron/squadronfriends
- AllowNotorious: bool
- FuelLevel: int
- JumpRangeCurr: float
- JumpRangeMax: float
- PendingDecommission: bool
- SpaceUsage { TotalCapacity, Crew, Cargo, CargoSpaceReserved, ShipPacks, ModulePacks, FreeSpace}

- Finance { CarrierBalance, ReserveBalance, AvailableBalance, ReservePercent, TaxRate }
- Crew [{ CrewRole, Activated, Enabled, CrewName },...]
- ShipPacks [{ PackTheme, packTier },...]
- ModulePacks [{PackTheme, packTier },...]

{ "timestamp":"2020-03-27T09:42:04Z", "event":"CarrierStats", "CarrierID":3700005632, "Callsign":"L14-X1J", "Name":"Spirula", "DockingAccess":"all", "AllowNotorious":false, "FuelLevel":63, "JumpRangeCurr":81.079422, "JumpRangeMax":500.000000, "PendingDecommission":false, "SpaceUsage":{ "TotalCapacity":25000, "Crew":5450, "Cargo":440, "CargoSpaceReserved":44, "ShipPacks":774, "ModulePacks":913, "FreeSpace":17379 }, "Finance":{ "CarrierBalance":10000000, "ReserveBalance":1800000, "AvailableBalance":8171946, "ReservePercent":18, "TaxRate":3 }, "Crew":[{ "CrewRole":"BlackMarket", "Activated":true, "Enabled":true, "CrewName":"Aliza Baldwin" }, { "CrewRole":"Captain", "Activated":true, "Enabled":true, "CrewName":"Herbert Benson" }, { "CrewRole":"Refuel", "Activated":true, "Enabled":false, "CrewName":"Maricela White" }, { "CrewRole":"Repair", "Activated":true, "Enabled":false, "CrewName":"Zayla Clements" }, { "CrewRole":"Rearm", "Activated":true, "Enabled":false, "CrewName":"Bill Lambert" }, { "CrewRole":"Commodities", "Activated":true, "Enabled":true, "CrewName":"Lizeth Morales" }, { "CrewRole":"VoucherRedemption", "Activated":true, "Enabled":false, "CrewName":"Phillip Gjoni" }, { "CrewRole":"Shipyard", "Activated":true, "Enabled":false, "CrewName":"Simon Rhodes" }, { "CrewRole":"Outfitting", "Activated":true, "Enabled":false, "CrewName":"Eugene Johnson" }, { "CrewRole":"CarrierFuel", "Activated":true, "Enabled":true, "CrewName":"Orlando York" }], "ShipPacks":[{ "PackTheme":"Zorgon Peterson - Cargo", "PackTier":1 }], "ModulePacks":[{ "PackTheme":"ExplosiveWeaponry", "PackTier":2 }] }

11.4 CarrierJumpRequest

At the time the player requests the jump, not the jump itself

- CarrierID: marketid
- SystemName: starsystem name
- SystemID: systemaddress
- Body: (name)
- BodyID

{ "timestamp":"2020-04-20T09:30:58Z", "event":"CarrierJumpRequest", "CarrierID":3700005632, "SystemName":"Paesui Xena", "Body":"Paesui Xena A", "SystemAddress":7269634680241, "BodyID":1 }

11.5 CarrierDecommission

At the time the player requests decommission

- CarrierID: marketid
- ScrapRefund: number
- Scraptime: timestamp

{ "timestamp":"2020-03-11T15:12:26Z", "event":"CarrierDecommission", "CarrierID":3700005632, "ScrapRefund":1746872629, "ScrapTime":1584601200 }

11.6 CarrierCancelDecommission

- CarrierID: marketid

{ "timestamp":"2020-03-11T15:12:38Z", "event":"CarrierCancelDecommission", "CarrierID":3700005632 }

11.7 CarrierBankTransfer

Player transfers credits to/from carrier

- CarrierID: marketid
- Deposit, or Withdraw
- PlayerBalance: (after transfer)
- CarrierBalance: (after transfer)

{ "timestamp":"2020-03-24T15:34:46Z", "event":"CarrierBankTransfer", "CarrierID":3700005632, "Deposit":80000, "PlayerBalance":717339604128, "CarrierBalance":3020010 }

11.8 CarrierDepositFuel

Any player giving fuel to the carrier

- CarrierID: marketid
- Amount: tons
- Total: total amount of fuel after donation

{ "timestamp":"2020-03-19T09:17:29Z", "event":"CarrierDepositFuel", "CarrierID":3700005632, "Amount":45, "Total":112 }

11.9 CarrierCrewServices

Changes to crew

- CarrierID: marketid
- Operation: (activate/deactivate/pause/resume/replace)
- CrewRole: string
- CrewName: string

{ "timestamp":"2020-03-17T12:38:54Z", "event":"CarrierCrewServices", "CarrierID":3700005632, "CrewRole":"Outfitting", "Operation":"Activate", "CrewName":"Eugene Johnson" }

11.10   CarrierFinance

Change to tax rate or reserve

- CarrierID: marketid
- TaxRate: number
- CarrierBalance: number
- ReserveBalance: number
- AvailableBalance: number
- ReservePercent: number

{ "timestamp":"2020-03-26T10:36:32Z", "event":"CarrierFinance", "CarrierID":3700005632, "TaxRate":5, "CarrierBalance":3278186, "ReserveBalance":0, "AvailableBalance":475108, "ReservePercent":0 }

11.11 CarrierShipPack

CarrierShipPack/CarrierModulePack

- CarrierID: marketid
- Operation: buypack/sellpack/restockpack
- PackTheme: name
- PackTier: name
- Cost/Refund: int

{ "timestamp":"2020-03-16T09:25:39Z", "event":"CarrierShipPack", "CarrierID":3700005632, "Operation":"BuyPack", "PackTheme":"Zorgon Peterson - Cargo", "PackTier":1, "Cost":1668880 }

11.12 CarrierModulePack

Same format as CarrierShipPack

11.13 CarrierTradeOrder

The carrier owner has requested the carrier buys or sells goods (or cancels such an order)

- CarrierID: marketid
- BlackMarket: bool
- Commodity: name
- PurchaseOrder: quantity
- or SaleOrder: quantity (current stock)
- or CancelTrade: true
- Price: int

{ "timestamp":"2020-03-16T14:52:36Z", "event":"CarrierTradeOrder", "CarrierID":3700005632, "BlackMarket":false, "Commodity":"mineraloil", "Commodity_Localised":"Mineral Oil", "PurchaseOrder":70, "Price":228 }

11.14 CarrierDockingPermission

The carrier owner has changed the docking permission criteria

- CarrierID: marketid
- DockingAccess: all/none/friends/squadron/squadronfriends
- AllowNotorious: bool

{ "timestamp":"2020-03-11T15:07:25Z", "event":"CarrierDockingPermission", "CarrierID":3700005632, "DockingAccess":"squadron", "AllowNotorious":true }

11.15 CarrierNameChanged

This is logged when a carrier's name is changed

- CarrierID
- Callsign
- Name

11.16 CarrierJumpCancelled

This is logged when a jump is cancelled

- CarrierID

12 Other Events

12.1 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 }

12.2 ApproachSettlement

When written: when approaching a planetary settlement

Parameters:
- Name
- MarketID
- Latitude
- Longitude
- SystemAddress
- BodyID
- BodyName

12.3 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)

12.4 CockpitBreached

When written: when cockpit canopy is breached

Parameters: none

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

12.5 CommitCrime

When written: when a crime is recorded against the player

Parameters:
- CrimeType – see §14.6
- 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 }

12.6 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

12.7 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
- ID

12.8 CrewMemberJoins

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

Parameters:
- Crew: player's commander name

12.9 CrewMemberQuits

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

Parameters:
- Crew: player's commander name

12.10 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" }

12.11 CrimeVictim

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

Parameters:
- Offender
- CrimeType
- Fine *or* Bounty

12.12 DatalinkScan

When written: when scanning a data link

Parameters:
- Message: message from data link

12.13 DatalinkVoucher

When written: when scanning a datalink generates a reward

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

12.14 DataScanned

When written: when scanning some types of data links

Parameters:
- Type

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

12.15 DockFighter

When written: when docking a fighter back with the mothership

Parameters:

- ID

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

12.16   DockSRV

When written: when docking an SRV with the ship

Parameters:

- ID

12.17   EndCrewSession

When written: when the captain in multicrew disbands the crew

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

12.18   FighterRebuilt

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

Parameters:
- Loadout
- ID

12.19   FuelScoop

When written: when scooping fuel from a star

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

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

12.20   Friends

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

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

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

12.21   JetConeBoost

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

Parameters:
- BoostValue

12.22   JetConeDamage

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

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

12.23   JoinACrew

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

Parameters:
- Captain: Helm player's commander name

12.24   KickCrewMember

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

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

12.25   LaunchDrone

When written: when using any type of drone/limpet

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

12.26   LaunchFighter

When written: when launching a fighter

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

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

12.27   LaunchSRV

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

Parameters:
- Loadout
- ID

12.28   ModuleInfo

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

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

Parameters: None

Example of ModulesInfo file:

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

12.29   Music

When written: when the game music 'mood' changes

Parameters:
- MusicTrack: (name)

Possible track names are: NoTrack, MainMenu, CQCMenu, SystemMap, GalaxyMap, GalacticPowers
CQC, DestinationFromHyperspace, DestinationFromSupercruise, Supercruise, Combat_Unknown
Unknown_Encounter, CapitalShip, CombatLargeDogFight, Combat_Dogfight, Combat_SRV
Unknown_Settlement, DockingComputer, Starport, Unknown_Exploration, Exploration

Note: Other music track names may be used in future

12.30   NpcCrewPaidWage

This is written when crew receive wages

Parameters:
- NpcCrewId
- NpcCrewName
- Amount

12.31   NpcCrewRank

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

Parameters:
- NpcCrewId
- NpcCrewName
- RankCombat

12.32   Promotion

When written: when the player's rank increases

Parameters: one of the following
- Combat: new rank
- Trade: new rank
- Explore: new rank
- CQC: new rank
- Federation:
- Empire:

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

12.33   ProspectedAsteroid

When using a prospecting drone

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

- Remaining: percentage

12.34 QuitACrew

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

Parameters:
- Captain: Helm player's commander name

12.35 RebootRepair

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

Parameters:
- Modules: JSON array of names of modules repaired

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

12.36 ReceiveText

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

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

12.37 RepairDrone

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

Parameters:
- HullRepaired
- CockpitRepaired
- CorrosionRepaired

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

12.38 ReservoirReplenished

When fuel is moved from one fuel tank to another

Parameters:
- FuelMain
- FuelReservoir

12.39   Resurrect

When written: when the player restarts after death

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

12.40   Scanned

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

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

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

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

12.41   SelfDestruct

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

Parameters: none

12.42   SendText

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

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

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

12.43   Shutdown

When written: on a clean shutdown of the game

Parameters: none

12.44 Synthesis

When written: when synthesis is used to repair or rearm

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

Example:

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

12.45 SystemsShutdown

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

Parameters: none

12.46 USSDrop

When written: when dropping from Supercruise at a USS

Parameters:
- USSType: description of USS
- USSThreat: threat level

Example:

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

12.47 VehicleSwitch

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

Parameters:
- To: (Mothership/Fighter)

Examples:

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

12.48 WingAdd

When written: another player has joined the wing

Parameters:
- Name

Example:

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

12.49   WingInvite

When written: when the player is invited to a wing

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

12.50   WingJoin

When written: this player has joined a wing

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

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

12.51   WingLeave

When written: this player has left a wing

Parameters: none

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

12.52   CargoTransfer

When transferring cargo between ship and fleet carrier, or between ship and SRV

- Transfers: [{Type, Count, Direction}, …]

{ "timestamp":"2020-03-12T12:49:54Z", "event":"CargoTransfer", "Transfers":[{ "Type":"tea", "Count":1, "Direction":"tocarrier" }, { "Type":"gold", "Count":1, "Direction":"toship" }] }

{ "timestamp":"2020-03-12T13:34:22Z", "event":"CargoTransfer", "Transfers":[{ "Type":"grain", "Count":2, "Direction":"tosrv" }] }

Index