Elite:Dangerous
Player Journal

1 Introduction

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

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

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

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

A short example of a player journal file (*out of date, some events may have additional data*):

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

1.1 ChangeLog

Version 4 *published 19/Sep/2016 (for 2.2 beta 1)*
- Add extra parameters to MissionAccepted events: destination info, and passenger info
- Interdiction events IsPlayer value is always a bool
- Clean up "smart quotes" and convert to "straight quotes"
- Add a note about the heading entry in every continuation of the file
- Add ApproachSettlement event

Version 3 *published 30/Aug/2016*
- Include ShipID in Module outfitting events
- Change some bool values from 1/0 to true/false
    - Resurrect/Bankrupt
    - Scan/TidalLock,Landable
    - Interdicted/Submitted
    - LaunchFighter/PlayerControlled
    - EjectCargo/Abandoned
    - CollectCargo/Stolen
    - ShieldState/ShieldsUp
- Include Major faction "Alliegance" in Location/FSDjump/Docked events
- Include surface gravity, pressure, temperature for a planet
- Include more info about rings when scanning star or planet
- Add events for NPC Crew interaction
- Localised text is in UTF8 encoding
- Added events DatalinkVoucher and DataScanned
- Added events JetConeBoost and JetConeDamage
- Added BrokerPercentage value to PayFines and RedeemVoucher
- Added ModuleStore and ModuleRetrieve
- Added the PVPKill event
- File saved in SavedGames folder
- Added "Continued" event
- Added MissionID parameter in mission events

Version2 *published 26/July/2016*
- File is formatted as line-delimited json
- Timestamp inside event object, ISO 8601 format
- Fileheader format changed
- Include faction info and faction state, for Starsystem and Station
- New event for dropping out of supercruise at a USS
- Interdiction events include extra info about the other player/NPC
- Remove PowerplayNominate (duplicate for PowerplayVote)
- Include gameplay mode, and credit balance in LoadGame
- Include station name and type in Location event if docked at startup
- Include Economy, Government and Security info for Starsystem on jump
- Include Economy, Government and Security info for Station when docking
- Include ship ID in shipyard entries
- Reorganised format for data when killed by a wing of players
- Record latitude and longitude when landing on planet
- Automatic localisation of text symbols
- Improved granularity of data, and additional info, for star and planet scans
- Planet Scan: Landable property is now 0 or 1, not a quoted string

- New HeatWarning and HeatDamage events
- New ShieldState and HullDamage events
- Report fuel used and fuel level on each jump
- RestockVehicle: added 'count' property for purchasing multiple vehicles
- Add events for DockingRequested, Denied, Granted etc
- Add mission expiry time

Version 1 was published 20/July/2016

2   File Format

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

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

2.1   File Location

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

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

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

2.2   Heading entry

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

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

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

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

2.3 Event Records

Each event record is a json object.

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

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

2.4 Localisation

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

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

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

"Government":"$government_PrisonColony;", "Government_Localised":"Colonie pénitentiaire"

3 Startup

3.1 ClearSavedGame

When written: If you should ever reset your game

Parameters:
- Name: commander name

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

3.2 NewCommander

When written: Creating a new commander

Parameters:
- Name: (new) commander name
- Package: selected starter package

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

3.3 LoadGame

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

Parameters:
- Commander: commander name
- Ship: current ship type
- ShipID: ship id number
- 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

Example:

{ "timestamp":"2016-06-10T14:32:03Z", "event":"LoadGame", "Commander":"HRC1", "Ship":"CobraMkIII", "ShipID":1, "GameMode":"Group", "Group":"Mobius", "Credits":600120, "Loan":0 }

3.4   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 }

3.5   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 }

4 Travel

4.1 Docked

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

Parameters:
- StationName: name of station
- StationType: type of station
- StarSystem: name of system
- CockpitBreach:true (only if landing with breached cockpit)
- Faction: station's controlling faction
- FactionState
- Allegiance
- Economy
- Government
- Security

Example:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Docked", "StationName":"Kotov Refinery", "StationType":"Outpost", "StarSystem":"Wolf 289", "Faction":"Wolf 289 Gold Federal Industry", "FactionState":"CivilWar", "Allegiance":"Federation", "Economy":"$economy_Extraction", "Government":"$government_Corporate", "Security":"$SYSTEM_SECURITY_high_anarchy;" }

4.2 DockingCancelled

When written: when the player cancels a docking request

Parameters:
- StationName: name of station

4.3 DockingDenied

When written: when the station denies a docking request

Parameters:
- StationName: name of station
- Reason: reason for denial

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

4.4 DockingGranted

When written: when a docking request is granted

Parameters:
- StationName: name of station
- LandingPad: pad number
-

4.5 DockingRequested

When written: when the player requests docking at a station

Parameters:
- StationName: name of station

4.6 DockingTimeout

When written: when a docking request has timed out

Parameters:
- StationName: name of station

4.7 FSDJump

When written: when jumping from one star system to another

Parameters:
- StarSystem: name of destination starsystem
- 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
- Faction: system controlling faction
- FactionState
- Allegiance
- Economy
- Government
- Security

Example:
{ "timestamp":"2016-07-21T13:16:49Z", "event":"FSDJump", "StarSystem":"LP 98-132", "StarPos":[-26.781,37.031,-4.594], "Economy":"$economy_Extraction;", "Allegiance":"Federation", "Government":"$government_Anarchy;", "Security":"$SYSTEM_SECURITY_high_anarchy;", "JumpDist":5.230, "FuelUsed":0.355614, "FuelLevel":12.079949, "Faction":"Brotherhood of LP 98-132", "FactionState":"Outbreak" }

4.8 Liftoff

When written: when taking off from planet surface

Parameters:
- Latitude
- Longitude

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

4.9 Location

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

Parameters:
- StarSystem: name of destination starsystem
- StarPos: star position, as a Json array [x, y, z], in light years
- Body: star's body name
- Docked: true (if docked)
- StationName: station name, (if docked)
- StationType: (if docked)
- Faction: star system controlling faction
- FactionState
- Allegiance
- Economy
- Government
- Security

Example:
{ "timestamp":"2016-07-21T13:14:25Z", "event":"Location", "Docked":1, "StationName":"Azeban City", "StationType":"Coriolis", "StarSystem":"Eranin", "StarPos":[-22.844,36.531,-1.188], "Allegiance":"Alliance", "Economy":"$economy_Agri;", "Government":"$government_Communism;", "Security":$SYSTEM_SECURITY_medium;, "Faction":"Eranin Peoples Party" }

4.10 SupercruiseEntry

When written: entering supercruise from normal space

Parameters:
- Starsystem

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

4.11 SupercruiseExit

When written: leaving supercruise for normal space

Parameters:
- Starsystem
- Body

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

4.12 Touchdown

When written: landing on a planet surface

Parameters:
- Latitude
- Longitude

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

4.13 Undocked

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

Parameters:
- StationName: name of station

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

5 Combat

5.1 Bounty

When written: player is awarded a bounty for a kill

Parameters:
- Faction: the faction awarding the bounty
- Reward: the reward value
- VictimFaction: the victim's faction
- SharedWithOthers: whether shared with other players

Example:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Bounty", "Faction":"$faction_Federation;",
"Target":"Skimmer", "Reward":1000, "VictimFaction":"MMU" }

5.2 CapShipBond

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

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

5.3 Died

When written: player was killed

Parameters:
- KillerName
- KillerShip
- KillerRank

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

5.4 Died

When written: player was killed by a wing

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

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

5.5 EscapeInterdiction

When written: Player has escaped interdiction

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

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

5.6 FactionKillBond

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

Parameters:
- Reward
- AwardingFaction
- VictimFaction

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

5.7 HeatDamage

When written: when taking damage due to overheating

Parameters:none

5.8 HeatWarning

When written: when heat exceeds 100%

Parameters: none

5.9 HullDamage

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

Parameters:
- Health

Example:
{ "timestamp":"2016-07-25T14:46:23Z", "event":"HullDamage", "Health":0.798496 }
{ "timestamp":"2016-07-25T14:46:23Z", "event":"HullDamage", "Health":0.595611 }
{ "timestamp":"2016-07-25T14:46:23Z", "event":"HullDamage", "Health":0.392725 }
{ "timestamp":"2016-07-25T14:46:26Z", "event":"HullDamage", "Health":0.188219 }

5.10 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" }

5.11 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 }

5.12 PVPKill

When written: when this player has killed another player

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

5.13 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 }

6   Exploration

6.1   Scan

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

Parameters(star)
- Bodyname: name of body
- DistanceFromArrivalLS
- StarType: Stellar classification (for a star)
- StellarMass: mass as multiple of Sol's mass
- Radius
- AbsoluteMagnitude
- OrbitalPeriod (seconds)
- RotationPeriod (seconds)
- Rings: [array] – if present

Parameters(Planet/Moon)
- Bodyname: name of body
- DistanceFromArrivalLS
- TidalLock: 1 if tidally locked
- TerraformState: Terraformable, Terraforming, Terraformed, or null
- PlanetClass
- Atmosphere
- Volcanism
- SurfaceGravity
- SurfaceTemperature
- SurfacePressure
- Landable: true (if landable)
- Materials: JSON object with material names and percentage occurrence
- OrbitalPeriod (seconds)
- RotationPeriod (seconds)
- Rings: [array of info] – if rings present

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

Examples:
{ "timestamp":"2016-07-25T10:02:38Z", "event":"Scan", "BodyName":"Alnitak", "DistanceFromArrivalLS":0.000000, "StarType":"O", "StellarMass":26.621094, "Radius":2305180672.000000, "AbsoluteMagnitude":-5.027969, "OrbitalPeriod":5755731.500000, "RotationPeriod":90114.937500 }

{ "timestamp":"2016-07-27T14:40:04Z", "event":"Scan", "BodyName":"HIP 4420 1", "DistanceFromArrivalLS":151.984283, "StarType":"Y", "StellarMass":0.019531,

"Radius":54144908.000000, "AbsoluteMagnitude":20.959091, "OrbitalPeriod":4977483.500000, "RotationPeriod":67481.585938, "Rings":[{ "Name":"HIP 4420 1 A Ring", "RingClass":"eRingClass_Rocky", "MassMT":3.040e11, "InnerRad":8.933e7, "OuterRad":1.361e8 }, { "Name":"HIP 4420 1 B Ring", "RingClass":"eRingClass_MetalRich", "MassMT":1.355e13, "InnerRad":1.362e9, "OuterRad":6.796e8 }] }

6.2   MaterialCollected

When Written: whenever materials are collected

Parameters:
- Category: type of material (Raw/Encoded/Manufactured)
- Name: name of material

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

6.3   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 }

6.4   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 }

6.5   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 }

6.6 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

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 }

6.7 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

Example:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"Screenshot", "Filename":"_Screenshots/Screenshot_0151.bmp", "Width":1600, "Height":900, "System":"Shinrarta Dezhra", "Body":"Founders World" }

7 Trade

7.1 BuyTradeData

When Written: when buying trade data in the galaxy map

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

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

7.2 CollectCargo

When Written: when scooping cargo from space or planet surface

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

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

7.3 EjectCargo

When Written:

Parameters:
- Type: cargo type
- Count: number of units
- Abandoned: whether 'abandoned'

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

7.4 MarketBuy

When Written: when purchasing goods in the market

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

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

7.5 MarketSell

When Written: when selling goods in the market

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

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

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

7.6 MiningRefined

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

Parameters:
- Type: cargo type

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

8   Station Services

8.1   BuyAmmo

When Written: when purchasing ammunition

Parameters:
- Cost

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

8.2   BuyDrones

When Written: when purchasing drones

Parameters:
- Type
- Count
- BuyPrice
- TotalCost

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

8.3   CommunityGoalJoin

When Written: when signing up to a community goal

Parameters:
- Name
- System

8.4   CommunityGoalReward

When Written: when receiving a reward for a community goal

Parameters:
- Name
- System
- Reward

8.5 CrewAssign

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

Parameters:
- Name
- Role

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

8.6 CrewFire

When written: when dismissing a member of crew

Parameters:
- Name

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

8.7 CrewHire

When written: when engaging a new member of crew

Parameters:
- Name
- Faction
- Cost
- Combat Rank

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

8.8 EngineerApply

When Written: when applying an engineer's upgrade to a module

Parameters:
- Engineer: name of engineer
- Blueprint: blueprint being applied
- Level: crafting level
- Override: whether overriding special effect

Example:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"EngineerApply", "Engineer":"Elvira Martuuk", "Blueprint":"ShieldGenerator_Reinforced", "Level":1 }

8.9   EngineerCraft

When Written: when requesting an engineer upgrade

Parameters:
- Engineer: name of engineer
- Blueprint: name of blueprint
- Level: crafting level
- Ingredients: JSON object with names and quantities of materials required

Example:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"EngineerCraft", "Engineer":"Elvira Martuuk", "Blueprint":"FSD_LongRange", "Level":2, "Ingredients":{"praseodymium":1, "disruptedwakeechoes":3, "chemicalprocessors":2, "arsenic":2 } }

8.10 EngineerProgress

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

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

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

8.11 MissionAbandoned

When Written: when a mission has been abandoned

Parameters:
- Name: name of mission
- MissionID

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

8.12 MissionAccepted

When Written: when starting a mission

Parameters:
- Name: name of mission
- Faction: faction offering mission
- MissionID
Optional Parameters (depending on mission type)
- Commodity: commodity type

- Count: number required / to deliver
- Target: name of target
- TargetType: type of target
- TargetFaction: target's faction
- Expiry: mission expiry time, in ISO 8601
- DestinationSystem
- DestinationStation
- PassengerCount
- PassengerVIPs: bool
- PassengerWanted: bool
- PassengerType: eg Tourist, Soldier, Explorer,...

Example:
{ "timestamp":"2016-07-26T11:36:44Z", "event":"MissionAccepted", "Faction":"Tsu Network", "Name":"Mission_Collect", "MissionID":65343026, "Commodity":"$Fish_Name;", "Commodity_Localised":"Fish", "Count":2, "Expiry":"2016-07-27T15:56:23Z" }

8.13 MissionCompleted

When Written: when a mission is completed

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

Optional parameters (depending on mission type)
- Commodity
- Count
- Target
- TargetType
- TargetFaction
- Reward: value of reward
- Donation: donation offered (for altruism missions)
- PermitsAwarded:[] (names of any permits awarded, as a JSON array)

Example:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"MissionCompleted", "Name":"Mission_Delivery_name", "MissionID":65343027, "Commodity":$Beer_Name;, "Faction":"Lencali Freedom Party", "Reward":76258 }

8.14 MissionFailed

When Written: when a mission has failed

Parameters:
- Name: name of mission
- MissionID

8.15 ModuleBuy

When Written: when buying a module in outfitting

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

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 }

8.16 ModuleRetrieve

When written: when fetching a previously stored module

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

8.17 ModuleSell

When Written: when selling a module in outfitting

Parameters:
- 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 }

8.18 ModuleStore

When written: when storing a module in Outfitting

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

8.19 ModuleSwap

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

Parameters:
- 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 }

8.20 PayFines

When Written: when paying fines

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

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

8.21 PayLegacyFines

When Written: when paying legacy fines

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

8.22 RedeemVoucher

When Written: when claiming payment for combat bounties and bonds

Parameters:
- Type
- Amount: (Net amount received, after any broker fee)
- BrokerPercenentage (if redeemed through a broker)

Example:
{ "timestamp":"2016-06-10T14:32:03Z", "event":"RedeemVoucher", "Type":"bounty", "Amount":1000 }

8.23 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 }

8.24 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 }

8.25 Repair

When Written: when repairing the ship

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

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

8.26 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 }

8.27 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 }

8.28 ShipyardBuy

When Written: when buying a new ship in the shipyard

Parameters:
- 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 }

8.29 ShipyardNew

When written: after a new ship has been purchased

Parameters:
- ShipType
- ShipID

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

8.30 ShipyardSell

When Written: when selling a ship stored in the shipyard

Parameters:
- 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" }

8.31 ShipyardTransfer

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

Parameters:
- ShipType: type of ship
- ShipID
- System: where it is
- Distance: how far away
- TransferPrice: cost of transfer

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

8.32 ShipyardSwap

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

Parameters:
- 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 }

9 Powerplay

9.1 PowerplayCollect

When written: when collecting powerplay commodities for delivery

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

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

9.2 PowerplayDefect

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

Parameters:
- FromPower
- ToPower

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

9.3 PowerplayDeliver

When written: when delivering powerplay commodities

Parameters:
- Power
- Type
- Count

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

9.4 PowerplayFastTrack

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

Parameters:
- Power
- Cost

9.5 PowerplayJoin

When written: when joining up with a power

Parameters:
- Power

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

9.6 PowerplayLeave

When written: when leaving a power

Parameters:
- Power

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

9.7 PowerplaySalary

When written: when receiving salary payment from a power

Parameters:
- Power
- Amount

9.8 PowerplayVote

When written: when voting for a system expansion

Parameters:
- Power
- Votes
- System

9.9 PowerplayVoucher

When written: when receiving payment for powerplay combat

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

10 Other Events

10.1 ApproachSettlement

When written: when approaching a planetary settlement

Parameters:
- Name

10.2 CockpitBreached

When written: when cockpit canopy is breached

Parameters: none

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

10.3 CommitCrime

When written: when a crime is recorded against the player

Parameters:
- CrimeType
- Faction

Optional parameters (depending on crime)
- Victim
- Fine
- Bounty

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

10.4 Continued

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

Parameters:
- Part: next part number

10.5 DatalinkScan

When written: when scanning a data link

Parameters:
- Message: message from data link

10.6 DatalinkVoucher

When written: when scanning a datalink generates a reward

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

10.7 DataScanned

When written: when scanning some types of data links

Parameters:
- Type

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

10.8 DockFighter

When written: when docking a fighter back with the mothership

Parameters: none

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

10.9 DockSRV

When written: when docking an SRV with the ship

Parameters: none

10.10 FuelScoop

When written: when scooping fuel from a star

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

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

10.11 JetConeBoost

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

Parameters:
- BoostValue

10.12 JetConeDamage

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

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

10.13 LaunchFighter

When written: when launching a fighter

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

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

10.14 LaunchSRV

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

Parameters:
- Loadout

10.15 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

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

10.16 RebootRepair

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

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

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

10.17 ReceiveText

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

Parameters:
- From
- Message

10.18 Resurrect

When written: when the player restarts after death

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

10.19 SelfDestruct

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

Parameters: none

10.20 SendText

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

Parameters:
- To
- Message

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

10.21 Synthesis

When written: when synthesis is used to repair or rearm

Parameters:
- Name: synthesis blueprint
- Materials: JSON object listing materials used and quantities

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

10.22 USSDrop

When written: when dropping from Supercruise at a USS

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

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

10.23 VehicleSwitch

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

Parameters:
- To: (Mothership/Fighter)

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

10.24 WingAdd

When written: another player has joined the wing

Parameters:
- Name

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

10.25 WingJoin

When written: this player has joined a wing

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

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

10.26 WingLeave

When written: this player has left a wing

Parameters: none

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

11Appendix

Ranks

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

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

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

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

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

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

Index