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:

```
{ "Heading": { "localdatetime":"16-06-10-15:31", "timezone":"GMT Daylight Time", "gmt_time":"14:31", "part":1 }
,"14:32:03": { "event":"LoadGame", "Commander":"HRC1", "Ship":"SideWinder" }
,"14:32:03": { "event":"Rank", "Combat":0, "Trade":0, "Explore":1, "Empire":0, "Federation":0, "CQC":0 }
,"14:32:03": { "event":"Progress", "Combat":0, "Trade":0, "Explore":73, "Empire":0, "Federation":0, "CQC":0 }
,"14:32:15": { "event":"Location", "StarSystem":"Asellus Primus", "StarPos":[-23.938,40.875,-1.344] }
,"14:32:16": { "event":"Docked", "StationName":"Beagle 2 Landing", "StationType":"Coriolis" }
,"14:32:38": { "event":"RefuelAll", "Cost":12, "Amount":0.234493 }
,"14:34:25": { "event":"Undocked", "StationName":"Beagle 2 Landing", "StationType":"Coriolis" }
,"14:35:00": { "event":"FSDJump", "StarSystem":"HIP 78085", "StarPos":[120.250,40.219,268.594],
"JumpDist":36.034 }
,"14:35:22": { "event":"Scan", "BodyName":"HIP 78085 A", "StarType":"G" }
,"14:36:10": { "event":"FSDJump", "StarSystem":"Praea Euq NW-W b1-3", "StarPos":[120.719,34.188,271.750],
"JumpDist":6.823 }
,"14:36:42": { "event":"Scan", "BodyName":"Praea Euq NW-W b1-3", "StarType":"M" }
,"14:38:50": { "event":"Scan", "BodyName":"Praea Euq NW-W b1-3 3", "Description":"Icy body with neon rich
atmosphere and major water geysers volcanism" }
,"14:39:08": { "event":"Scan", "BodyName":"Praea Euq NW-W b1-3 3 a", "Description":"Tidally locked Icy body" }
,"14:41:03": { "event":"FSDJump", "StarSystem":"Asellus Primus", "StarPos":[-23.938,40.875,-1.344],
"JumpDist":39.112 }
,"14:41:26": { "event":"SupercruiseExit", "StarSystem":"Asellus Primus", "Body":"Beagle 2 Landing" }
,"14:41:29": { "event":"Docked", "StationName":"Beagle 2 Landing", "StationType":"Coriolis" }
,"14:41:58": { "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 File Format

The Player Journal is written in JSON format (see www.json.org), to provide a standard format for ease of machine parsing, while still being intelligible to the human reader.

Each Journal file is one Json object, containing one “Heading” entry, followed by a number of time-stamped journal entries. Each of these entries is a Json object, which may contain further nested objects or arrays.

2.1 File Location

The journal files are (currently) written into the same folder as the network log files.

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:

- localdatetime: the time and date in local timezone
- timezone: the player’s timezone
- gmt_time: the time in GMT
- part: the file part number

Example:
"Heading": { "localdatetime":"16-06-03-17:02", "timezone":"GMT Daylight Time", "gmt_time":"16:02", "part":1 }

(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)

2.3 Event Records

Each event record has the timestamp as as Json key, and a json object as the value

The object has 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.

3 Startup

3.1 ClearSavedGame

When written: If you should ever reset your game

Parameters:

- Name: commander name

Example:

"15:08:42": { "event":"ClearSavedGame", "Name":"HRC1" }

3.2 NewCommander

When written: Creating a new commander

Parameters:

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

Example:

"10:42:47": { "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
- StartLanded: true (only present if landed)
- StartDead:true (only present if starting dead: see “Resurrect”)

Example:

"15:54:36": { "event":"LoadGame", "Commander":"HRC1", "Ship":"CobraMkIII" }

3.4 Progress

When written: at startup

Parameters:

- Combat: percent progress to next rank
- Trade: “
- Explore: “
- Empire: “

- Federation: “
- CQC: “

Example:

```
"15:54:36": { "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:

```
"15:54:36": { "event":"Rank", "Combat":2, "Trade":2, "Explore":5, "Empire":1, "Federation":3,
"CQC":0 }
```

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’

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
- CockpitBreach:true (only if landing with breached cockpit)

Example:
16:06:20": { "event":"Docked", "StationName":"Jameson Memorial", "StationType":"Orbis" }

4.2 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
- BoostUsed: whether FSD boost was used

Example:
"16:03:49": { "event":"FSDJump", "StarSystem":"Shinrarta Dezhra",
“StarPos":[55.719,17.594,27.156], "JumpDist":39.224 }

4.3 Liftoff

When written: when taking off from planet surface

Parameters: none

Example:
"14:16:07": { "event":"Liftoff" }

4.4 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

Example:
"14:12:45": { "event":"Location", "StarSystem":"Khun", "StarPos":[-171.594,19.969,-56.969] }

4.5 SupercruiseEntry

When written: entering supercruise from normal space

Parameters:

- Starsystem

Example:

"13:35:18": { "event":"SupercruiseEntry", "StarSystem":"Yuetu" }

4.6 SupercruiseExit

When written: leaving supercruise for normal space

Parameters:

- Starsystem
- Body

Example:

"13:37:30": { "event":"SupercruiseExit", "StarSystem":"Yuetu", "Body":"Yuetu B" }

4.7 Touchdown

When written: landing on a planet surface

Parameters: none

Example:

"14:47:48": { "event":"Touchdown" }

4.8 Undocked

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

Parameters:

- StationName: name of station

Example:

"14:19:33": { "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:

```
"10:55:20": { "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:

```
"14:03:14": { "event":"Died", "KillerName":"$ShipName_Police_Independent;", "KillerShip":"viper",
"KillerRank":"Deadly" }
```

5.4 Died

When written: player was killed by a wing

Parameters:

- KillerNames: a JSON array of player names
- KillerShips: a JSON array of ship names
- KillerRanks: a JSON array of rank names

Example:

```
"14:17:21": { "event":"Died", "KillerNames":["Cmdr HRC1",”Cmdr HRC2”],
"KillerShips":["type6",”Python”], "KillerRanks":["Harmless",”Master”] }
```

5.5 EscapeInterdiction

When written: Player has escaped interdiction

Parameters: None

Example:

```
"14:47:48": { "event":"EscapeInterdiction" }
```

5.6 FactionKillBond

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

Parameters:

- Reward
- AwardingFaction
- VictimFaction

Example:

```
"09:33:10": { "event":"FactionKillBond",  “Reward”: 500, "AwardingFaction":"Jarildekald Public Industry", “VictimFaction”: “Lencali Freedom Party” }
```

5.7 Interdicted

When written: player was interdicted by player or npc

Parameters:

- Submitted: true or false

Example:

```
"12:33:01": { "event":"interdicted", “Submitted”:false }
```

5.8 Interdiction

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

Parameters:

- Success

Example:

```
"11:07:13": { "event":"interdiction", “Success”:true }
```

6 Exploration

6.1 Scan

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

Parameters:

- Bodyname: name of body
- Description: description (for planet or moon)
- StarType: Stellar classification (for a star)
- Landable: true (if landable)
- Materials: JSON object with material names and percentage occurrence

Examples:

```
"14:36:42": { "event":"Scan", "BodyName":"Praea Euq NW-W b1-3", "StarType":"M" }
```

```
"13:49:26": { "event":"Scan", "BodyName":"HIP 16121 4", "Description":"High metal content body",
"Landable":"true", "Materials":{ "iron":  23.8, "nickel":  18.0, "sulphur":  16.7, "carbon":  14.1,
"phosphorus":   9.0, "vanadium":   5.9, "germanium":   5.0, "selenium":   2.6, "cadmium":   1.9,
"molybdenum":   1.6, "ruthenium":   1.5 } }
```

6.2 MaterialCollected

When Written: whenever materials are collected

Parameters:

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

Examples:

```
“13:18:11": { "event":"MaterialCollected", "Category":"Raw", "Name":"sulphur" }
"11:44:58": { "event":"MaterialCollected", "Category":"Encoded", "Name":"disruptedwakeechoes" }
```

6.3 MaterialDiscarded

When Written: if materials are discarded

Parameters:

- Category
- Name
- Count

Example:

```
“13:18:11": { "event":"MaterialDiscarded", "Category":"Raw", "Name":"sulphur", “Count”: 5 }
```

6.4 MaterialDiscovered

When Written: when a new material is discovered

Parameters:

- Category
- Name
- DiscoveryNumber

Example:
"14:52:40": { "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:
"11:49:22": { "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:
"14:41:58": { "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:

```
"09:26:22": { "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:

```
"10:54:10": { "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:

```
"10:47:57": { "event":"CollectCargo", "Type":"agriculturalmedicines", "Stolen":0 }
```

7.3 EjectCargo

When Written:

Parameters:

- Type: cargo type
- Count: number of units
- Abandoned: whether ‘abandoned’

Example:

```
"10:13:57": { "event":"EjectCargo", "Type":"tobacco", "Count":1, "Abandoned":1 }
```

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:

```
"09:57:03": { "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:

```
"08:40:12": { "event":"MarketSell", "Type":"agriculturalmedicines", "Count":3, "SellPrice":1360,
"TotalSale":4080, "AvgPricePaid":304 }
```

```
"13:32:09": { "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:

```
"10:53:50": { "event":"MiningRefined", “Type:”Gold” }
```

8 Station Services

8.1 BuyAmmo

When Written: when purchasing ammunition

Parameters:

- Cost

Example:

"10:59:01": { "event":"BuyAmmo", "Cost":80 }

8.2 BuyDrones

When Written: when purchasing drones

Parameters:

- Type
- Count
- BuyPrice
- TotalCost

Example:

"16:20:58": { "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 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:

"11:00:00": { "event":"EngineerApply", "Engineer":"Elvira Martuuk", "Blueprint":"ShieldGenerator_Reinforced", “Level”:1 }

8.6 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:

"11:05:51": { "event":"EngineerCraft", "Engineer":"Elvira Martuuk", "Blueprint":"FSD_LongRange", “Level”:2, "Ingredients":{"praseodymium":1, "disruptedwakeechoes":3, "chemicalprocessors":2, "arsenic":2 } }

8.7 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:

"14:53:15": { "event":"EngineerProgress", "Progress":"Unlocked", "Engineer":"Elvira Martuuk" }
"18:04:09": { "event":"EngineerProgress", "Engineer":"Elvira Martuuk", "Rank":2 }

8.8 MissionAbandoned

When Written: when a mission has been abandoned

Parameters:

- Name: name of mission

Example:

"13:15:00": { "event":"MissionAbandoned", "Name":"Mission_Collect_name" }

8.9 MissionAccepted

When Written: when starting a mission

Parameters:

- Name: name of mission
- Faction: faction offering mission

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

Example:

```
"12:26:55": { "event":"MissionAccepted", "Faction":"Revolutionary Thiin Liberals",
"Name":"Mission_Collect", "Commodity":"AnimalMonitors", "Count":2 }
```

8.10 MissionCompleted

When Written: when a mission is completed

Parameters:

- Name: mission type
- Faction: faction name

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:

```
"12:33:34": { "event":"MissionCompleted", "Name":"Mission_Delivery_name",
"Commodity":$Beer_Name;, "Faction":"Lencali Freedom Party", "Reward":76258 }
```

8.11 MissionFailed

When Written: when a mission has failed

Parameters:

- Name: name of mission

8.12 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

If replacing an existing module:

- SellItem: item being sold
- SellPrice: sale price

Example:

```
"14:02:48": { "event":"ModuleBuy", "Slot":"MediumHardpoint2",
"SellItem":"hpt_pulselaser_fixed_medium", "SellPrice":0,
"BuyItem":"hpt_multicannon_gimbal_medium", "BuyPrice":50018, "Ship":"cobramkiii" }
```

8.13 ModuleSell

When Written: when selling a module in outfitting

Parameters:

- Slot
- SellItem
- SellPrice
- Ship

Example:

```
"10:48:30": { "event":"ModuleSell", "Slot":"Slot06_Size2", "SellItem":"int_cargorack_size1_class1",
"SellPrice":877, "Ship":"asp" }
```

8.14 ModuleSwap

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

Parameters:

- FromSlot
- ToSlot
- FromItem
- ToItem
- Ship

Examples:

```
"14:03:03": { "event":"ModuleSwap", "FromSlot":"MediumHardpoint1",
"ToSlot":"MediumHardpoint2", "FromItem":"hpt_pulselaser_fixed_medium",
"ToItem":"hpt_multicannon_gimbal_medium", "Ship":"cobramkiii" }

"14:05:20": { "event":"ModuleSwap", "FromSlot":"SmallHardpoint2", "ToSlot":"SmallHardpoint1",
"FromItem":"hpt_pulselaserburst_fixed_small_scatter", "ToItem":"Null", "Ship":"cobramkiii" }
```

8.15 PayFines

When Written: when paying fines

Parameters:

- Amount

Example:

```
"13:25:11": { "event":"PayFines", "Amount":1791 }
```

8.16 PayLegacyFines

When Written: when paying legacy fines

Parameters:

- Amount

8.17 RedeemVoucher

When Written: when claiming payment for combat bounties and bonds

Parameters:

- Type
- Amount

Example:

```
"10:59:24": { "event":"RedeemVoucher", "Type":"bounty", "Amount":1000 }
```

8.18 RefuelAll

When Written: when refuelling (full tank)

Parameters:

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

Example:

```
"09:24:51": { "event":"RefuelAll", "Cost":317, "Amount":6.322901 }
```

8.19 RefuelPartial

When Written: when refuelling (10%)

Parameters:

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

Example:

```
"09:24:42": { "event":"RefuelPartial", "Cost":83, "Amount":1.649000 }
```

8.20 Repair

When Written: when repairing the ship

Parameters:

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

Example:
"14:26:53": { "event":"Repair", "Item":"int_powerplant_size3_class5", "Cost":1100 }

8.21 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

Example:
"09:42:56": { "event":"RestockVehicle", “Type”:”SRV”, "Loadout":"starter", "Cost":1030 }

8.22 SellDrones

When Written: when selling unwanted drones back to the market

Parameters:

- Type
- Count
- SellPrice
- TotalSale

Example:
"16:20:58": { "event":"SellDrones", "Type":"Drones", "Count":1, "SellPrice":91, "TotalSale":91 }

8.23 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
- SellOldShip: (if selling current ship) ship type being sold
- SellPrice: (if selling current ship) ship sale price

Example:
"15:28:14": { "event":"ShipyardBuy", "ShipType":"anaconda", "ShipPrice":114636172, "StoreOldShip":"SideWinder" }

8.24 ShipyardSell

When Written: when selling a ship stored in the shipyard

Parameters:

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

Example:

```
"09:02:35": { "event":"ShipyardSell", "ShipType":"SideWinder", "ShipPrice":1718561 }
```

8.25 ShipyardTransfer

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

Parameters:

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

Example:

```
"13:30:15": { "event":"ShipyardTransfer", "ShipType":"Eagle",  "System":"Asellus Primus",
"Distance":4.482060, "TransferPrice":84 }
```

8.26 ShipyardSwap

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

Parameters:

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

Example

```
"16:24:32": { "event":"ShipyardSwap", "ShipType":"cobramkiii", "StoreOldShip":"Type6" }
```

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:

```
"09:11:51": { "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:

```
"09:09:09": { "event":"PowerplayDefect", "FromPower":"Zachary Hudson", "ToPower":"Li Yong-Rui"
}
```

9.3 PowerplayDeliver

When written: when delivering powerplay commodities

Parameters:

- Power
- Type
- Count

Example:

```
"09:17:37": { "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:

"10:10:00": { "event":"PowerplayJoin", "Power":"Zachary Hudson" }

9.6 PowerplayLeave

When written: when leaving a power

Parameters:

- Power

Example:

"10:10:00": { "event":"PowerplayLeave", "Power":"Li Yong-Rui" }

9.7 PowerplayNominate

When written: when nominating a system for expansion

Parameters:

- Power
- System
- Votes
-

9.8 PowerplaySalary

When written: when receiving salary payment from a power

Parameters:

- Power
- Amount
-

9.9 PowerplayVote

When written: when voting for a system expansion

Parameters:

- Power
- Votes
- System

9.10 PowerplayVoucher

When written: when receiving payment for powerplay combat

Parameters:

- Power
- Systems:[name,name]

10 Other Events

10.1 CockpitBreached

When written: when cockpit canopy is breached

Parameters: none

Example:
"14:16:17": { "event":"CockpitBreached" }

10.2 CommitCrime

When written: when a crime is recorded against the player

Parameters:

- CrimeType
- Faction

Optional parameters (depending on crime)

- Victim
- Fine
- Bounty

Examples:
"11:28:09": { "event":"CommitCrime", "CrimeType":"assault", "Faction":"The Pilots Federation", "Victim":"Potapinski", "Bounty":210 }
"09:29:11": { "event":"CommitCrime", "CrimeType":"fireInNoFireZone", "Faction":"Jarildekald Public Industry", "Fine":100 }

10.3 DatalinkScan

When written: when scanning a data link

Parameters:

- Message: message from data link

10.4 DockFighter

When written: when docking a fighter back with the mothership

Parameters: none

Example:
"09:52:12": { "event":"DockFighter" }

10.5 DockSRV

When written: when docking an SRV with the ship

Parameters: none

10.6 FuelScoop

When written: when scooping fuel from a star

Parameters:

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

Example:

```
"09:14:24": { "event":"FuelScoop", "Scooped":0.498700, "Total":16.000000 }
```

10.7 LaunchFighter

When written: when launching a fighter

Parameters:

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

```
"09:03:22": { "event":"LaunchFighter", "Loadout":"starter", "PlayerControlled":1 }
```

10.8 LaunchSRV

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

Parameters:

- Loadout

10.9 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:

```
"11:01:02": { "event":"Promotion", "Explore":2 }
```

10.10 RebootRepair

When written: when the ‘reboot repair’ function is used

Parameters:

- Modules: JSON array of names of modules repaired

Example:

```
"14:23:08": { "event":"RebootRepair", "Modules":[ “MainEngines”, “TinyHardpoint1” ] }
```

10.11 ReceiveText

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

Parameters:

- From
- Message

10.12 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

10.13 SelfDestruct

When written: when the ‘self destruct’ function is used

Parameters: none

10.14 SendText

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

Parameters:

- To
- Message

Example:

```
"09:58:16": { "event":"SendText", "To":"HRC-2", "Message":"zoom" }
```

10.15 Synthesis

When written: when synthesis is used to repair or rearm

Parameters:

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

Example:

```
"12:00:59": { "event":"Synthesis", "Name":"Repair Basic", "Materials":{ "iron":2, "nickel":1 } }
```

10.16 VehicleSwitch

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

Parameters:

- To: (Mothership/Fighter)

Examples:

```
"09:00:05": { "event":"VehicleSwitch", "To":"Fighter" }
"09:01:16": { "event":"VehicleSwitch", "To":"Mothership" }
```

10.17 WingAdd

When written: another player has joined the wing

Parameters:

- Name

Example:

```
"14:36:33": { "event":"WingAdd", "Name":"HRC-2" }
```

10.18 WingJoin

When written: this player has joined a wing

Parameters:

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

Example:

```
"14:50:10": { "event":"WingJoin", "Others":[ "HRC1" ] }
```

10.19 WingLeave

When written: this player has left a wing

Parameters: none

Example:

```
"15:06:08": { "event":"WingLeave" }
```

Appendix - Features not yet implemented

Module transfer and storage

Hiring NPC pilots

Index