Sign In or Register

CODEMASTERS COMMUNITY

BROWSE ACTIVITY

Forums Clubs Calendar Staff Online Users Leaderboard Search...

Home > F1 Games > F1® 2019 Game Forum > Technical Assistance > All Activity

NVIDIA Graphics Card? Crashing? Click here for solution - UPDATED 27/04

Want to report an issue? Have a bug to report? Please look here first!!!!

F1 2019 UDP Specification

Sign in to follow this FOLLOWERS 3

By Faya, June 20, 2019 in Technical Assistance

★★★★★

1 2 3 4 5 6 NEXT » Page 1 of 6

Faya

Members
1,189
1,269 posts

Posted June 20, 2019

The F1 series of games support the output of certain game data across UDP connections. This data can be used supply race information to external applications, or to drive certain hardware (e.g. motion platforms, force feedback steering wheels and LED devices).

The following information summarise this data structures so that developers of supporting hardware or software are able to configure these to work correctly with the F1 game.

If you cannot find the information that you require then please contact community@codemasters.com and a member of the dev team will respond to your query as soon as possible.

Packet Information

Packet Types

Each packet can now carry different types of data rather than having one packet which contains everything. A header has been added to each packet as well so that versioning can be tracked and it will be easier for applications to check they are interpreting the incoming data in the correct way. Please note that all values are encoded using Little Endian format. All data is packed.

The following data types are used in the structures:

Type	Description
uint8	Unsigned 8-bit integer
int8	Signed 8-bit integer
uint16	Unsigned 16-bit integer

int16	Signed 16-bit integer
float	Floating point (32-bit)
uint64	Unsigned 64-bit integer

Packet Header

Each packet has the following header:

```
struct PacketHeader
{
    uint16    m_packetFormat;         // 2019
    uint8     m_gameMajorVersion;     // Game major version - "X.00"
    uint8     m_gameMinorVersion;     // Game minor version - "1.XX"
    uint8     m_packetVersion;        // Version of this packet type, all start fr
    uint8     m_packetId;             // Identifier for the packet type, see belo
    uint64    m_sessionUID;           // Unique identifier for the session
    float     m_sessionTime;          // Session timestamp
    uint      m_frameIdentifier;      // Identifier for the frame the data was re
    uint8     m_playerCarIndex;       // Index of player's car in the array
};
```

Packet IDs

The packets IDs are as follows:

Packet Name	Value	Description
Motion	0	Contains all motion data for player's car – only sent while player is in control
Session	1	Data about the session – track, time left
Lap Data	2	Data about all the lap times of cars in the session
Event	3	Various notable events that happen during a session
Participants	4	List of participants in the session, mostly relevant for multiplayer
Car Setups	5	Packet detailing car setups for cars in the race
Car Telemetry	6	Telemetry data for all cars
Car Status	7	Status data for all cars such as damage

Motion Packet

The motion packet gives physics data for all the cars being driven. There is additional data for the car being driven with the goal of being able to drive a motion platform setup.

N.B. For the normalised vectors below, to convert to float values divide by 32767.0f – 16-bit signed values are used to pack the data and on the assumption that direction values are always between -1.0f and 1.0f.

Frequency: Rate as specified in menus

Size: 1343 bytes

Version: 1

```
struct CarMotionData
{
    float         m_worldPositionX;           // World space X position
    float         m_worldPositionY;           // World space Y position
    float         m_worldPositionZ;           // World space Z position
    float         m_worldVelocityX;           // Velocity in world space X
    float         m_worldVelocityY;           // Velocity in world space Y
    float         m_worldVelocityZ;           // Velocity in world space Z
    int16         m_worldForwardDirX;         // World space forward X direction (
    int16         m_worldForwardDirY;         // World space forward Y direction (
    int16         m_worldForwardDirZ;         // World space forward Z direction (
    int16         m_worldRightDirX;           // World space right X direction (n
    int16         m_worldRightDirY;           // World space right Y direction (n
    int16         m_worldRightDirZ;           // World space right Z direction (n
    float         m_gForceLateral;            // Lateral G-Force component
    float         m_gForceLongitudinal;       // Longitudinal G-Force component
    float         m_gForceVertical;           // Vertical G-Force component
    float         m_yaw;                      // Yaw angle in radians
    float         m_pitch;                    // Pitch angle in radians
    float         m_roll;                     // Roll angle in radians
};

struct PacketMotionData
{
    PacketHeader    m_header;                   // Header

    CarMotionData   m_carMotionData[20];        // Data for all cars on track

    // Extra player car ONLY data
    float         m_suspensionPosition[4];       // Note: All wheel arrays have th
    float         m_suspensionVelocity[4];       // RL, RR, FL, FR
    float         m_suspensionAcceleration[4];  // RL, RR, FL, FR
    float         m_wheelSpeed[4];              // Speed of each wheel
    float         m_wheelSlip[4];                // Slip ratio for each wheel
    float         m_localVelocityX;             // Velocity in local space
    float         m_localVelocityY;             // Velocity in local space
    float         m_localVelocityZ;             // Velocity in local space
    float         m_angularVelocityX;           // Angular velocity x-component
    float         m_angularVelocityY;            // Angular velocity y-component
    float         m_angularVelocityZ;            // Angular velocity z-component
    float         m_angularAccelerationX;        // Angular velocity x-component
    float         m_angularAccelerationY;       // Angular velocity y-component
    float         m_angularAccelerationZ;        // Angular velocity z-component
    float         m_frontWheelsAngle;            // Current front wheels angle in
};
```

Session Packet

The session packet includes details about the current session in progress.

Frequency: 2 per second

Size: 149 bytes

Version: 1

```
struct MarshalZone
{
    float  m_zoneStart;   // Fraction (0..1) of way through the lap the marshal zo
    int8   m_zoneFlag;    // -1 = invalid/unknown, 0 = none, 1 = green, 2 = blue,
};

struct PacketSessionData
{
    PacketHeader    m_header;                   // Header

    uint8           m_weather;                  // Weather - 0 = clear, 1 = light
                                                // 3 = light rain, 4 = heavy rain,
    int8                    m_trackTemperature;         // Track temp. in degrees
    int8                    m_airTemperature;           // Air temp. in degrees ce
    uint8           m_totalLaps;                // Total number of laps in this ra
    uint16          m_trackLength;              // Track length in metres
    uint8           m_sessionType;              // 0 = unknown, 1 = P1, 2 = P2, 3
                                                // 5 = Q1, 6 = Q2, 7 = Q3, 8 = Sho
                                                // 10 = R, 11 = R2, 12 = Time Tria
    int8            m_trackId;                  // -1 for unknown, 0-21 for tracks
    uint8           m_formula;                          // Formula, 0 = F1 Modern,
// 3 = F1 Generic
    uint16          m_sessionTimeLeft;          // Time left in session in seconds
    uint16          m_sessionDuration;          // Session duration in seconds
    uint8           m_pitSpeedLimit;            // Pit speed limit in kilometres p
    uint8           m_gamePaused;                // Whether the game is paused
    uint8           m_isSpectating;             // Whether the player is spectatin
    uint8           m_spectatorCarIndex;        // Index of the car being spectate
    uint8           m_sliProNativeSupport;      // SLI Pro support, 0 = inactive,
    uint8           m_numMarshalZones;          // Number of marshal zones to foll
    MarshalZone     m_marshalZones[21];                 // List of marshal zones -
    uint8           m_safetyCarStatus;           // 0 = no safety car, 1 = full sa
                                                 // 2 = virtual safety car
    uint8           m_networkGame;               // 0 = offline, 1 = online
};
```

Lap Data Packet

The lap data packet gives details of all the cars in the session.

Frequency: Rate as specified in menus

Size: 843 bytes

Version: 1

```
struct LapData
{
    float       m_lastLapTime;          // Last lap time in seconds
    float       m_currentLapTime;       // Current time around the lap in seconds
    float       m_bestLapTime;          // Best lap time of the session in second
    float       m_sector1Time;          // Sector 1 time in seconds
    float       m_sector2Time;          // Sector 2 time in seconds
    float       m_lapDistance;          // Distance vehicle is around current lap
                                        // be negative if line hasn't been crosse
    float       m_totalDistance;                // Total distance travelled in se
                                        // be negative if line hasn't been crosse
    float       m_safetyCarDelta;        // Delta in seconds for safety car
    uint8       m_carPosition;          // Car race position
    uint8       m_currentLapNum;                // Current lap number
    uint8       m_pitStatus;                    // 0 = none, 1 = pitting, 2 = in 
    uint8       m_sector;                       // 0 = sector1, 1 = sector2, 2 = 
    uint8       m_currentLapInvalid;            // Current lap invalid - 0 = vali
    uint8       m_penalties;                    // Accumulated time penalties in 
    uint8       m_gridPosition;                 // Grid position the vehicle star
    uint8       m_driverStatus;                 // Status of driver - 0 = in gara
// 2 = in lap, 3 = out lap, 4 = on track
    uint8       m_resultStatus;          // Result status - 0 = invalid, 1 = inac
// 3 = finished, 4 = disqualified, 5 = not classified
// 6 = retired
};


struct PacketLapData
{
    PacketHeader    m_header;              // Header

    LapData         m_lapData[20];         // Lap data for all cars on track
};
```

Event Packet

This packet gives details of events that happen during the course of a session.

Frequency: When the event occurs

Size: 32 bytes

Version: 1

```
// The event details packet is different for each type of event.
// Make sure only the correct type is interpreted.
union EventDataDetails
{
    struct
    {
        uint8   vehicleIdx; // Vehicle index of car achieving fastest lap
        float   lapTime;    // Lap time is in seconds
    } FastestLap;

    struct
    {
        uint8   vehicleIdx; // Vehicle index of car retiring
    } Retirement;

    struct
    {
        uint8   vehicleIdx; // Vehicle index of team mate
    } TeamMateInPits;

    struct
    {
        uint8   vehicleIdx; // Vehicle index of the race winner
    } RaceWinner;
};

struct PacketEventData
{
    PacketHeader        m_header;               // Header

uint8                   m_eventStringCode[4];   // Event string code, see below
EventDataDetails        m_eventDetails;         // Event details - should be inter
// for each type
};
```

Event String Codes

Event	Code	Description
Session Started	“SSTA”	Sent when the session starts
Session Ended	“SEND”	Sent when the session ends
Fastest Lap	“FTLP”	When a driver achieves the fastest lap
Retirement	“RTMT”	When a driver retires
DRS enabled	“DRSE”	Race control have enabled DRS
DRS disabled	“DRSD”	Race control have disabled DRS
Team mate in pits	“TMPT”	Your team mate has entered the pits
Chequered flag	“CHQF”	The chequered flag has been waved
Race Winner	“RCWN”	The race winner is announced

Participants Packet

This is a list of participants in the race. If the vehicle is controlled by AI, then the name will be the driver name. If this is a multiplayer game, the names will be the Steam Id on PC, or the LAN name if appropriate.

N.B. on Xbox One, the names will always be the driver name, on PS4 the name will be the LAN name if playing a LAN game, otherwise it will be the driver name.

The array should be indexed by vehicle index.

Frequency: Every 5 seconds

Size: 1104 bytes

Version: 1

```
struct ParticipantData
{
    uint8      m_aiControlled;           // Whether the vehicle is AI (1) or Huma
    uint8      m_driverId;              // Driver id - see appendix
    uint8      m_teamId;                 // Team id - see appendix
    uint8      m_raceNumber;             // Race number of the car
    uint8      m_nationality;            // Nationality of the driver
    char       m_name[48];               // Name of participant in UTF-8 format –
                                        // Will be truncated with … (U+2026) if t
    uint8      m_yourTelemetry;          // The player's UDP setting, 0 = restric
};

struct PacketParticipantsData
{
    PacketHeader    m_header;            // Header

    uint8           m_numActiveCars;    // Number of active cars in the data – sh
// cars on HUD
    ParticipantData m_participants[20];
};
```

Car Setups Packet

This packet details the car setups for each vehicle in the session. Note that in multiplayer games, other player cars will appear as blank, you will only be able to see your car setup and AI cars.

Frequency: 2 per second

Size: 843 bytes

Version: 1

```
struct CarSetupData
{
    uint8     m_frontWing;                // Front wing aero
    uint8     m_rearWing;                 // Rear wing aero
    uint8     m_onThrottle;               // Differential adjustment on throttle (
    uint8     m_offThrottle;              // Differential adjustment off throttle
    float     m_frontCamber;              // Front camber angle (suspension geome
    float     m_rearCamber;               // Rear camber angle (suspension geomet
    float     m_frontToe;                 // Front toe angle (suspension geometry
    float     m_rearToe;                  // Rear toe angle (suspension geometry)
    uint8     m_frontSuspension;          // Front suspension
    uint8     m_rearSuspension;           // Rear suspension
    uint8     m_frontAntiRollBar;         // Front anti-roll bar
    uint8     m_rearAntiRollBar;          // Front anti-roll bar
    uint8     m_frontSuspensionHeight;    // Front ride height
    uint8     m_rearSuspensionHeight;     // Rear ride height
    uint8     m_brakePressure;            // Brake pressure (percentage)
    uint8     m_brakeBias;                // Brake bias (percentage)
    float     m_frontTyrePressure;        // Front tyre pressure (PSI)
    float     m_rearTyrePressure;         // Rear tyre pressure (PSI)
    uint8     m_ballast;                  // Ballast
    float     m_fuelLoad;                 // Fuel load
};

struct PacketCarSetupData
{
    PacketHeader    m_header;            // Header

    CarSetupData    m_carSetups[20];
};
```

Car Telemetry Packet

This packet details telemetry for all the cars in the race. It details various values that would be recorded on the car such as speed, throttle application, DRS etc.

Frequency: Rate as specified in menus

Size: 1347 bytes

Version: 1

```
struct CarTelemetryData
{
    uint16    m_speed;                    // Speed of car in kilometres per hour
    float     m_throttle;                 // Amount of throttle applied (0.0 to 1.
    float     m_steer;                    // Steering (-1.0 (full lock left) to 1.
    float     m_brake;                    // Amount of brake applied (0.0 to 1.0)
    uint8     m_clutch;                   // Amount of clutch applied (0 to 100)
    int8      m_gear;                     // Gear selected (1-8, N=0, R=-1)
    uint16    m_engineRPM;                // Engine RPM
    uint8     m_drs;                      // 0 = off, 1 = on
    uint8     m_revLightsPercent;         // Rev lights indicator (percentage)
    uint16    m_brakesTemperature[4];     // Brakes temperature (celsius)
    uint16    m_tyresSurfaceTemperature[4]; // Tyres surface temperature (celsius,
    uint16    m_tyresInnerTemperature[4]; // Tyres inner temperature (celsius)
    uint16    m_engineTemperature;        // Engine temperature (celsius)
    float     m_tyresPressure[4];         // Tyres pressure (PSI)
    uint8     m_surfaceType[4];           // Driving surface, see appendices
};

struct PacketCarTelemetryData
{
    PacketHeader        m_header;              // Header

    CarTelemetryData    m_carTelemetryData[20];

    uint32              m_buttonStatus;        // Bit flags specifying which butt
                                               // currently - see appendices
};
```

Car Status Packet

This packet details car statuses for all the cars in the race. It includes values such as the damage readings on the car.

Frequency: Rate as specified in menus

Size: 1143 bytes

Version: 1

```
struct CarStatusData
{
    uint8       m_tractionControl;          // 0 (off) - 2 (high)
    uint8       m_antiLockBrakes;           // 0 (off) - 1 (on)
    uint8       m_fuelMix;                  // Fuel mix - 0 = lean, 1 = standard,
    uint8       m_frontBrakeBias;           // Front brake bias (percentage)
    uint8       m_pitLimiterStatus;         // Pit limiter status - 0 = off, 1 =
    float       m_fuelInTank;               // Current fuel mass
float       m_fuelCapacity;             // Fuel capacity
float       m_fuelRemainingLaps;        // Fuel remaining in terms of laps (value
    uint16      m_maxRPM;                   // Cars max RPM, point of rev limiter
    uint16      m_idleRPM;                  // Cars idle RPM
    uint8       m_maxGears;                 // Maximum number of gears
    uint8       m_drsAllowed;               // 0 = not allowed, 1 = allowed, -1 =
    uint8       m_tyresWear[4];             // Tyre wear percentage
    uint8       m_actualTyreCompound;      // F1 Modern - 16 = C5, 17 = C4, 18 =
                                           // 7 = inter, 8 = wet
                                           // F1 Classic - 9 = dry, 10 = wet
                                           // F2 – 11 = super soft, 12 = soft, 13
                                           // 15 = wet
uint8       m_tyreVisualCompound;       // F1 visual (can be different from actual
   // 16 = soft, 17 = medium, 18 = hard, 7 = inter, 8 = wet
   // F1 Classic – same as above
   // F2 – same as above
    uint8       m_tyresDamage[4];           // Tyre damage (percentage)
    uint8       m_frontLeftWingDamage;      // Front left wing damage (percentage
    uint8       m_frontRightWingDamage;     // Front right wing damage (percentage
    uint8       m_rearWingDamage;           // Rear wing damage (percentage)
    uint8       m_engineDamage;             // Engine damage (percentage)
    uint8       m_gearBoxDamage;            // Gear box damage (percentage)
    int8        m_vehicleFiaFlags;         // -1 = invalid/unknown, 0 = none, 1 =
                                            // 2 = blue, 3 = yellow, 4 = red
    float       m_ersStoreEnergy;           // ERS energy store in Joules
    uint8       m_ersDeployMode;            // ERS deployment mode, 0 = none, 1 =
                                           // 3 = high, 4 = overtake, 5 = hotlap
    float       m_ersHarvestedThisLapMGUK;  // ERS energy harvested this lap by M
    float       m_ersHarvestedThisLapMGUH;  // ERS energy harvested this lap by M
    float       m_ersDeployedThisLap;       // ERS energy deployed this lap
};

struct PacketCarStatusData
{
    PacketHeader        m_header;          // Header

    CarStatusData       m_carStatusData[20];
};
```

Restricted data (Your Telemetry setting)

There is some data in the UDP that you may not want other players seeing if you are in a multiplayer game. This is controlled by the “Your Telemetry” setting in the Telemetry options. The options are:

- Restricted (Default) – other players viewing the UDP data will not see values for your car
- Public – all other players can see all the data for your car

Note: You can always see the data for the car you are driving regardless of the setting.

The following data items are set to zero if the player driving the car in question has their “Your Telemetry” set to “Restricted”:

Car status packet

- m_fuelInTank
- m_fuelCapacity
- m_fuelMix
- m_fuelRemainingLaps
- m_frontBrakeBias
- m_frontLeftWingDamage
- m_frontRightWingDamage
- m_rearWingDamage
- m_engineDamage
- m_gearBoxDamage
- m_tyresWear (All four wheels)
- m_tyresDamage (All four wheels)
- m_ersDeployMode
- m_ersStoreEnergy
- m_ersDeployedThisLap
- m_ersHarvestedThisLapMGUK
- m_ersHarvestedThisLapMGUH

3

Faya

Members
1,189
1,269 posts

Posted June 20, 2019

FAQS

How do I enable the UDP Telemetry Output?

In F1 2019, UDP telemetry output is controlled via the in-game menus. To enable this, enter the options menu from the main menu (triangle / Y), then enter the settings menu - the UDP option will be at the bottom of the list. From there you will be able to enable / disable the UDP output, configure the IP address and port for the receiving application, toggle broadcast mode and set the send rate. Broadcast mode transmits the data across the network subnet to allow multiple devices on the same subnet to be able to receive this information. When using broadcast mode it is not necessary to set a target IP address, just a target port for applications to listen on.

Advanced PC Users: You can additionally edit the game’s configuration XML file to configure UDP output. The file is located here (after an initial boot of the game):

...\Documents\My Games\<game_folder>\hardwaresettings\hardware_settings_config.xml

You should see the tag:

```
<motion>

  ...

  <udp enabled="false" broadcast=”false” ip="127.0.0.1" port="20777" sendRate=”20

           ...

</motion>
```

Here you can set the values manually. Note that any changes made within the game when it is running will overwrite any changes made manually. Note the enabled flag is now a state.

What has changed since last year?

F1 2019 sees the following changes to the UDP specification:

- Updated driver list and included F2 drivers
- Added terrain tyre contact type to motion packet
- Removed exhaust damage
- Added game version to packet header
- Renamed num cars to num active cars
- Increased accuracy of steering, brakes and accelerator values
- Added new event types
- Added visual tyre compound to deal with new F1 tyres
- Added fuel remaining in laps – value on MFD
- Added restrictions to sensitive data via “Your Telemetry” setting

What is the order of the wheel arrays?

All wheel arrays are in the following order:

```
   0 – Rear Left (RL)
   1 – Rear Right (RR)
   2 – Front Left (FL)
   3 – Front Right (FR)
```

Do the vehicle indices change?

During a session, each car is assigned a vehicle index. This will not change throughout the session and all the arrays that are sent use this vehicle index to dereference the correct piece of data.

What encoding format is used?

All values are encoded using Little Endian format.

Are the data structures packed?

Yes, all data is packed, there is no padding used.

Will there always be 20 cars in the data structures?

Most structures contain 20 cars – this is the maximum that can be in a race in the game. There is a data item called m_numActiveCars in the participants packet which tells you how many cars are active in the race. However, you should check the individual result status of each car in the lap data to see if that car is actively providing data. If it is not "Invalid" or "Inactive" then the corresponding vehicle index has valid data.

How often are updated packets sent?

For the packets which get updated at "Rate as specified in the menus" you can be guaranteed that on the frame that these get sent they will all get sent together and will never be separated across frames. This of course relies on the reliability of your network as to whether they are received correctly as everything is sent via UDP. Other packets that get sent at specific rates can arrive on any frame.

If you are connected to the game when it starts transmitting the first frame will contain the following information to help initialise data structures on the receiving application:

Packets sent on Frame 1: (All packets sent on this frame have "Session timestamp" 0.000)

- Session
- Participants
- Car Setups
- Lap Data
- Motion Data
- Car Telemetry
- Car Status

As an example, assuming that you are running at 60Hz with 60Hz update rate selected in the menus then you would expect to see the following packets and timestamps:

Packets sent on Frame 2: (All packets sent on this frame have "Session timestamp" 0.016)

- Lap Data
- Motion Data
- Car Telemetry
- Car Status
- …

Packets sent on Frame 31: (All packets sent on this frame have "Session timestamp" 0.5)

- Session (since 2 updates per second)
- Car Setups (since 2 updates per second)
- Lap Data
- Motion Data
- Car Telemetry
- Car Status

Will my old app still work with F1 2019?

F1 2019 uses a new format for the UDP data. However, earlier formats of the data are still supported so that most older apps implemented using the previous data formats should work with little or no change from the developer. To use the old formats, please enter the UDP options menu and set "UDP Format" to either "F1 2018" or "legacy" (for F1 2017 and earlier).

Specifications for the legacy format can be seen here: http://forums.codemasters.com/discussion/53139/f1-2017-d-box-and-udp-output-specification/p1

Specifications for the F1 2018 format can be seen here:

Hoo created a topic in Technical Assistance
August 23, 2018

F1 2018 UDP Specification

OVERVIEW The F1 series of games support the outputting of key game data via a UDP data stream. This data can be interpreted by external apps or connected peripherals for a range ...

284 replies

How do I enable D-BOX output?

D-BOX output is currently supported on the PC platform. In F1 2019, the D-BOX activation can be controlled via the menus. Navigate to `Game Options->Settings->UDP Telemetry Settings->D-BOX` to activate this on your system.

Advanced PC Users: It is possible to control D-BOX by editing the games' configuration XML file. The file is located here (after an initial boot of the game):

```
...\Documents\My Games\<game_folder>\hardwaresettings\hardware_settings_config.xm
```

You should see the tag:

```
  <motion>
    <dbox enabled="false" />
    ...
  </motion>
```

Set the "`enabled`" value to "`true`" to allow the game to output to your D-BOX motion platform. Note that any changes made within the game when it is running will overwrite any changes made manually.

How can I disable in-game support for LED device?

The F1 game has native support for some of the basic features supported by some external LED devices, such as the *Leo Bodnar SLI Pro* and the *Fanatec* steering wheels. To avoid conflicts between Codemasters' implementation and any third-party device managers on the PC platform it may be necessary to disable the native support. This is done using the following `led_display` flags in the `hardware_settings_config.xml`. The file is located here (after an initial boot of the game):

```
...\Documents\My Games\<game_folder>\hardwaresettings\hardware_settings_config.xm
```

The flags to enabled/disable LED output are:

```
<led_display fanatecNativeSupport="true" sliProNativeSupport="true" />
```

The `sliProNativeSupport` flag controls the output to SLI Pro devices. The `fanatecNativeSupport` flag controls the output to Fanatec (and some related) steering wheel LEDs. Set the values for any of these to “`false`” to disable them and avoid conflicts with your own device manager.

Please note there is an additional flag to manually control the LED brightness on the SLI Pro:

```
<led_display sliProForceBrightness="127" />
```

This option (using value in the range 0-255) will be ignored when setting the `sliProNativeSupport` flag to “`false`”.

Also note it is now possible to edit these values on the fly via the `Game Options->Settings->UDP Telemetry Settings` menu.

Can I configure the UDP output using an XML File?

PC users can edit the game’s configuration XML file to configure UDP output. The file is located here (after an initial boot of the game):

```
...\Documents\My Games\<game_folder>\hardwaresettings\hardware_settings_config.xml
```

You should see the tag:

```
   <motion>
     ...
     <udp enabled="false" broadcast=”false” ip="127.0.0.1" port="20777" sendRate=”
     ...
   </motion>
```

Here you can set the values manually. Note that any changes made within the game when it is running will overwrite any changes made manually.

Faya

Members
1,189
1,269 posts

Posted June 20, 2019

Appendices

Here are the values used for the team ID, driver ID and track ID parameters.

N.B. Driver IDs in network games differ from the actual driver IDs. All the IDs of human players start at 100 and are unique within the game session, but don’t directly correlate to the player.

Team IDs

ID	Team	ID	Team	ID	Team
0	Mercedes	21	Red Bull 2010	42	Art GP ’19
1	Ferrari	22	Ferrari 1976	43	Campos ’19

2	Red Bull Racing	23	ART Grand Prix	44	Carlin ’19
3	Williams	24	Campos Vexatec Racing	45	Sauber Junior Charouz ’19
4	Racing Point	25	Carlin	46	Dams ’19
5	Renault	26	Charouz Racing System	47	Uni-Virtuosi ‘19
6	Toro Rosso	27	DAMS	48	MP Motorsport ‘19
7	Haas	28	Russian Time	49	Prema ’19
8	McLaren	29	MP Motorsport	50	Trident ’19
9	Alfa Romeo	30	Pertamina	51	Arden ’19
10	McLaren 1988	31	McLaren 1990	63	Ferrari 1990
11	McLaren 1991	32	Trident	64	McLaren 2010
12	Williams 1992	33	BWT Arden	65	Ferrari 2010
13	Ferrari 1995	34	McLaren 1976		
14	Williams 1996	35	Lotus 1972		
15	McLaren 1998	36	Ferrari 1979		
16	Ferrari 2002	37	McLaren 1982		
17	Ferrari 2004	38	Williams 2003		
18	Renault 2006	39	Brawn 2009		
19	Ferrari 2007	40	Lotus 1978		

Driver IDs

ID	Driver	ID	Driver	ID	Driver
0	Carlos Sainz	37	Peter Belousov	69	Ruben Meijer
1	Daniil Kvyat	38	Klimek Michalski	70	Rashid Nair
2	Daniel Ricciardo	39	Santiago Moreno	71	Jack Tremblay
6	Kimi Räikkönen	40	Benjamin Coppens	74	Antonio Giovinazzi

7	Lewis Hamilton	41	Noah Visser	75	Robert Kubica
9	Max Verstappen	42	Gert Waldmuller	78	Nobuharu Matsushita
10	Nico Hulkenburg	43	Julian Quesada	79	Nikita Mazepin
11	Kevin Magnussen	44	Daniel Jones	80	Guanya Zhou
12	Romain Grosjean	45	Artem Markelov	81	Mick Schumacher
13	Sebastian Vettel	46	Tadasuke Makino	82	Callum Ilott
14	Sergio Perez	47	Sean Gelael	83	Juan Manuel Correa
15	Valtteri Bottas	48	Nyck De Vries	84	Jordan King
19	Lance Stroll	49	Jack Aitken	85	Mahaveer Raghunathan
20	Arron Barnes	50	George Russell	86	Tatiana Calderon
21	Martin Giles	51	Maximilian Günther	87	Anthoine Hubert
22	Alex Murray	52	Nirei Fukuzumi	88	Guiliano Alesi
23	Lucas Roth	53	Luca Ghiotto	89	Ralph Boschung
24	Igor Correia	54	Lando Norris		
25	Sophie Levasseur	55	Sérgio Sette Câmara		
26	Jonas Schiffer	56	Louis Delétraz		
27	Alain Forest	57	Antonio Fuoco		
28	Jay Letourneau	58	Charles Leclerc		
29	Esto Saari	59	Pierre Gasly		
30	Yasar Atiyeh	62	Alexander Albon		
31	Callisto Calabresi	63	Nicholas Latifi		
32	Naota Izum	64	Dorian Boccolacci		
33	Howard Clarke	65	Niko Kari		
34	Wilheim Kaufmann	66	Roberto Merhi		
35	Marie Laursen	67	Arjun Maini		

36	Flavio Nieves	68	Alessio Lorandi		

Track IDs

ID	Track
0	Melbourne
1	Paul Ricard
2	Shanghai
3	Sakhir (Bahrain)
4	Catalunya
5	Monaco
6	Montreal
7	Silverstone
8	Hockenheim
9	Hungaroring
10	Spa
11	Monza
12	Singapore
13	Suzuka
14	Abu Dhabi
15	Texas
16	Brazil
17	Austria
18	Sochi
19	Mexico
20	Baku (Azerbaijan)
21	Sakhir Short

22	Silverstone Short
23	Texas Short
24	Suzuka Short

Nationality IDs

ID	Nationality	ID	Nationality	ID	Nationality
1	American	31	Greek	61	Panamanian
2	Argentinean	32	Guatemalan	62	Paraguayan
3	Australian	33	Honduran	63	Peruvian
4	Austrian	34	Hong Konger	64	Polish
5	Azerbaijani	35	Hungarian	65	Portuguese
6	Bahraini	36	Icelander	66	Qatari
7	Belgian	37	Indian	67	Romanian
8	Bolivian	38	Indonesian	68	Russian
9	Brazilian	39	Irish	69	Salvadoran
10	British	40	Israeli	70	Saudi
11	Bulgarian	41	Italian	71	Scottish
12	Cameroonian	42	Jamaican	72	Serbian
13	Canadian	43	Japanese	73	Singaporean
14	Chilean	44	Jordanian	74	Slovakian
15	Chinese	45	Kuwaiti	75	Slovenian
16	Colombian	46	Latvian	76	South Korean
17	Costa Rican	47	Lebanese	77	South African
18	Croatian	48	Lithuanian	78	Spanish
19	Cypriot	49	Luxembourger	79	Swedish
20	Czech	50	Malaysian	80	Swiss

21	Danish	51	Maltese	81	Thai
22	Dutch	52	Mexican	82	Turkish
23	Ecuadorian	53	Monegasque	83	Uruguayan
24	English	54	New Zealander	84	Ukrainian
25	Emirian	55	Nicaraguan	85	Venezuelan
26	Estonian	56	North Korean	86	Welsh
27	Finnish	57	Northern Irish		
28	French	58	Norwegian		
29	German	59	Omani		
30	Ghanaian	60	Pakistani		

Surface types

These types are from physics data and show what type of contact each wheel is experiencing.

ID	Surface
0	Tarmac
1	Rumble strip
2	Concrete
3	Rock
4	Gravel
5	Mud
6	Sand
7	Grass
8	Water
9	Cobblestone
10	Metal
11	Ridged

Button flags

These flags are used in the telemetry packet to determine if any buttons are being held on the controlling device. If the value below logical ANDed with the button status is set then the corresponding button is being held.

Bit Flag	Button
0x0001	Cross or A
0x0002	Triangle or Y
0x0004	Circle or B
0x0008	Square or X
0x0010	D-pad Left
0x0020	D-pad Right
0x0040	D-pad Up
0x0080	D-pad Down
0x0100	Options or Menu
0x0200	L1 or LB
0x0400	R1 or RB
0x0800	L2 or LT
0x1000	R2 or RT
0x2000	Left Stick Click
0x4000	Right Stick Click

Bannish

Members
3
26 posts

Posted June 21, 2019

I have one question regarding the surface type.

When is the surface type set to 8 (water)? As soon as there is standing water on the track or does that not affect the surface type field?

AlexTT

Posted June 22, 2019

Members
⊕ 488
786 posts

Nice! Fuel laps like on MFD!

Oasis81

O

Members
⊕ 4
49 posts

Posted June 23, 2019

HI

The Sectors in the laptime, for every cars, are stored this year or not?

in the 2018 I had to save it everytime because after the lap they disappeared, and it was not so simple

it will be good if every laptime in Q and Race are saved with their sector time... or stored when the lap is complete and overwrite in the new lap.

rafitagd

R

Members
⊕ 2
5 posts

Posted June 23, 2019

@Retornik there is a "special" version of the game (press version) which codemasters give access to for some people (brand ambassadors, YouTubers,known people in sim racing etc.)

Normally those persons get a copy of the game like a month before lunch.

They can't post anything (prohibited) about the game for a time. But usually week or two before release the game developer allows this users to start posting info and videos about the game.

For F1 2019 was last Friday, so since last Friday all people with this press game version where allowed by codemasters to share all type of media about the game.

This is an usual procedure with any game.

And thanks Codemasters to share the new specs before release for us devs to be able to have our apps ready for release day!

LonelyRacer

L

Members
⊕ 8
60 posts

Posted June 25, 2019

@Hoo,

Any change, that at some upcoming patches, the PacketEventData could send the DRS status (DRSD or DRSE) after the SSTA?

Now as I understand the situation, DRS is available on all other modes, except at Race Start. And in P/Q, if it rains, it should be disabled. Now it seems, the DRS is available in Practice even if it rains.

If you want to display the correct start of session DRSD/DRSE status in a tool, getting the DRSD / DRSE after SSTA would make things much easier.

Cheers,

Hoo

Codemasters Staff

Codemasters

163
1,199 posts

Posted June 25, 2019

> On 6/25/2019 at 3:14 AM, LonelyRacer said:
>
> @Hoo,
>
> Any change, that at some upcoming patches, the PacketEventData could send the DRS status (DRSD or DRSE) after the SSTA?
>
> Now as I understand the situation, DRS is available on all other modes, except at Race Start. And in P/Q, if it rains, it should be disabled. Now it seems, the DRS is available in Practice even if it rains.
>
> If you want to display the correct start of session DRSD/DRSE status in a tool, getting the DRSD / DRSE after SSTA would make things much easier.
>
> Cheers,

Hi,

I've checked the DRS here and it seems to be correctly disabled when in light or heavy rain during practice sessions. If it is very light rain then DRS will still be allowed, so please check whether it has hit the "light rain" threshold. Let us know if you are still seeing issues an we'll take another look.

The DRS events (DRSD / DRDE) should be triggering correctly when there is a change in status during the race. If DRS is permitted from the start of the session then it should already be set to enabled. Trying to send this message after the race start could get messy as the game would end up sending out a bunch of session info after the race has actually started. We'd prefer to avoid changing the timing of this unless it is causing you major problems.

If you are noticing the DRS-related messages triggering incorrectly then please let us know and we'll take a look.

Thanks,
Hoo.

navarreitor

N

Members
4
44 posts

Posted June 25, 2019 (edited)

@Hoo @Faya

I have a PS4 copy and I can't find where the Restricted/public telemetry data option is.

I hope on PS4 we can make this data public because for commentators all this info is very useful.

In fact, it would be fabulous if in the menus of the leagues it could be restricted to be public, just as the assistance that can be used in the races can be restricted when you create the league. In all the competitions in which I participate, we use models that I have provided to have that information to commentators and viewers of the races on youtube/twitch (links below)

> have provided to have that information to commentators and viewers of the races on youtube/twitch (links below)

Hi @navarreitor ,

The current version (1.03) contains the correct structure in the UDP packet, but the game doesn't yet use this field. This option should be available in our next game update (v1.04).

The intention is for leagues to manage the setting of the restricted flag with their members directly. The game will not override users' privacy settings based on game mode. If this causes too many issues with league management then let us know and we will see if there are any alternative approaches that we can take.

Thanks,
Hoo.

navarreitor

Members
4
44 posts

Posted June 25, 2019 (edited)

> On 6/25/2019 at 10:49 AM, Hoo said:
>
> Hi @navarreitor ,
>
> The current version (1.03) contains the correct structure in the UDP packet, but the game doesn't yet use this field. This option should be available in our next game update (v1.04).
>
> The intention is for leagues to manage the setting of the restricted flag with their members directly. The game will not override users' privacy settings based on game mode. If this causes too many issues with league management then let us know and we will see if there are any alternative approaches that we can take.
>
> Thanks,
> Hoo.

Thank you @Hoo

We can ask our participants to do public their data but would be better if we can manage that form the admin/owner menu. If not, it is possible that some pilot forget to put his data public and the only solution we would have is to sanction them.

BTW, League mode is great, but it would be best if it has a fourth profile as a guest (as well as owner, administrator, and participant).

That way, when not all the participants can run a race, instead of AI, they could be replaced by guest pilots for a grand prize. The AI is fine, but nothing like the human touch.

Another subject, it would be possible that "Stop and go" penalties sums in the penalties of a player until they are fulfilled?

Thank you again

Edited June 25, 2019 by navarreitor

trenamax

T

Members
● 0
22 posts

Posted June 25, 2019

> On 6/20/2019 at 4:51 PM, Faya said:
>
> # FAQS
>
> ## What has changed since last year?
>
> F1 2019 sees the following changes to the UDP specification:
>
> - Added terrain tyre contact type to motion packet

Hi @Faya

I can't see terrain tyre contact type in the motion packet table. It looks the same as the 2018 version.

trenamax

T

Members
● 0
22 posts

Posted June 25, 2019

> On 6/25/2019 at 8:28 PM, trenamax said:
>
> Hi @Faya
>
> I can't see terrain tyre contact type in the motion packet table. It looks the same as the 2018 version.

Ah, I'm guessing it's this in the car telemetry data packet.

```
uint8     m_surfaceType[4];
```

navarreitor

N

Members
⊕ 4
44 posts

Posted June 26, 2019 (edited)

Hi

Why 48 names if races only allow 20 + 2 spect?

```
   (m_name[48];)
```

Edit

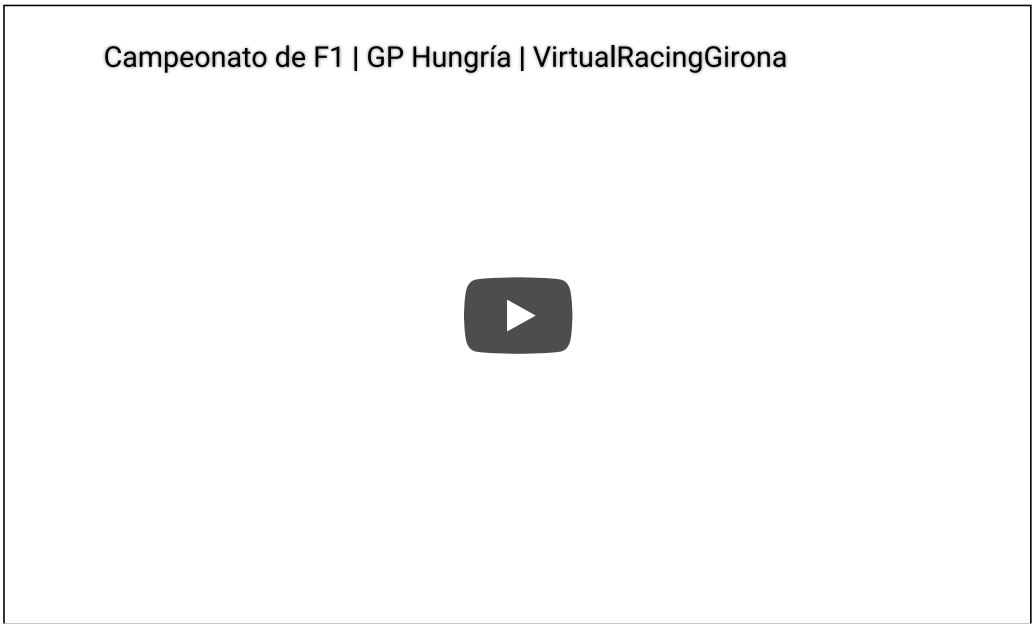
I just test it and now understand my mistake, are 48 characters for each name .... But on ps4 are P/L/A/Y/E/R always on multiplayer

Same last year personal data policy ?

Edited June 26, 2019 by navarreitor

♥ 1

VSL-1 || GP Canadá || F1 2018 || Sin ayudas / No assitance



Campeonato de F1 | GP Hungría | VirtualRacingGirona

Edited June 25, 2019 by navarreitor

Hoo

Codemasters Staff

Codemasters

163

1,199 posts

Posted June 25, 2019

> On 6/25/2019 at 10:05 AM, navarreitor said:
>
> @Hoo @Faya
>
> I have a PS4 copy and I can't find where the Restricted/public telemetry data option is.
>
> I hope on PS4 we can make this data public because for commentators all this info is very useful.
>
> In fact, it would be fabulous if in the menus of the leagues it could be restricted to be public, just as the assistance that can be used in the races can be restricted when you create the league. In all the competitions in which I participate, we use models that I

Oasis81

Members
4
49 posts

Posted June 28, 2019 (edited)

HI

the tyrecompund data ...what value is used in the game correctly? tyrecompound or tyreVisual?

Edited June 28, 2019 by Oasis81

Oasis81

Members
4
49 posts

Posted June 28, 2019

> On 6/22/2019 at 8:05 PM, Retornik said:
>
> I would like to know why many people can play F1 2019 and even make lives on the game since yesterday, yet the game only comes out on the 25/06/2019 Legendary, and normal on 28/06/2019

this is not the zone

pls user general zone.

game on pc was unlocked yesterday evening and on phisical store was sold before, same as other games.

CanTQuiT

Members
1
15 posts

Posted June 28, 2019 (edited)

In Multiplayer (unranked) all player names are "Player". (PC-Version)

Seriously?

Edited June 29, 2019 by CanTQuiT

Bannish

Posted July 1, 2019 (edited)

> On 6/20/2019 at 4:29 PM, Faya said:

B

Members
3
26 posts

> union EventDataDetails { struct { uint8 vehicleIdx; // *Vehicle index of car achieving fastest lap* float lapTime; // *Lap time is in seconds* } FastestLap; struct { uint8 vehicleIdx; // *Vehicle index of car retiring* } Retirement; struct { uint8 vehicleIdx; // *Vehicle index of team mate* } TeamMateInPits; struct { uint8 vehicleIdx; // *Vehicle index of the race winner* } RaceWinner; };

This might be a nooby question, but I am writing a Telemetry app in c# and afaik, there is no "union" type. What do I have to do in C# to get this to work?

Edit:

After some searching, I found out that people say to use the Explicit LayoutKind. so now i wanted to know, if this would work, or if I can make this easier

```
[StructLayout(LayoutKind.Explicit, Pack = 1)]
public struct EventDataDetails
{
    [FieldOffset(0)]
    public FastestLap ftlp;

    [FieldOffset(0)]
    public Retirement rtmt;

    [FieldOffset(0)]
    public TeamMateInPits tmpt;

    [FieldOffset(0)]
    public RaceWinner rcwn;
}

public struct FastestLap
{
    byte vehicleIdx; // Vehicle index of car achieving fastest lap
    float lapTime;    // Lap time is in seconds
}

public struct Retirement
{
    byte vehicleIdx; // Vehicle index of car retiring

}

public struct TeamMateInPits
{
    byte vehicleIdx; // Vehicle index of team mate
}

public struct RaceWinner
{
    byte vehicleIdx; // Vehicle index of the race winner
}
```

Edited July 1, 2019 by Bannish
maybe found a solution

Oasis81

O

Members
⊕ 4
49 posts

Posted July 1, 2019 (edited)

I really don't understand why you don't storage the sector!

in the packetlapdata only the bestlap is stored, the sector is always resetted.

but in the table during the Q every bestlap has is sector

I'done a lot of work for find a way to store the sector for every bestlap for every driver, but if i FFw the time obviously i lost that info.

if i come back to garage directly the time has a jump and i lost the info if some driver does his best lap.

why?! wehre are that info?!

Edited July 1, 2019 by Oasis81

rafitagd

R

Members
⊕ 2
5 posts

Posted July 2, 2019

> On 7/1/2019 at 11:02 AM, Bannish said:
>
> This might be a nooby question, but I am writing a Telemetry app in c# and afaik, there is no "union" type. What do I have to do in C# to get this to work?
>
> Edit:
>
> After some searching, I found out that people say to use the Explicit LayoutKind. so now i wanted to know, if this would work, or if I can make this easier
>
> ```
> [StructLayout(LayoutKind.Explicit, Pack = 1)]
> public struct EventDataDetails
> {
>     [FieldOffset(0)]
>     public FastestLap ftlp;
>
>     [FieldOffset(0)]
>     public Retirement rtmt;
>
>     [FieldOffset(0)]
>     public TeamMateInPits tmpt;
>
>     [FieldOffset(0)]
>     public RaceWinner rcwn;
> }
>
> public struct FastestLap
> {
>     byte vehicleIdx; // Vehicle index of car achieving fastest lap
>     float lapTime;    // Lap time is in seconds
> }
>
> public struct Retirement
> ```

```
public struct Retirement
{
    byte vehicleIdx; // Vehicle index of car retiring
}

public struct TeamMateInPits
{
    byte vehicleIdx; // Vehicle index of team mate
}

public struct RaceWinner
{
    byte vehicleIdx; // Vehicle index of the race winner
}
```

@Bannish You can't do much "better" than that.

You are right using structs and you are right using StructLayout, this way your structs will get populated with correct data from data packet, as long as the order of the fields is the same, since C# use the data type of the field to know how many bits should take from data packet for each field...
But YOU HAVE TO BE SURE, the order of your fields/props doesn't change/is the same as Codies posted... if it changes you will get "corrupted" data...

NoizeGamer

N

Members
● 0
2 posts

Posted July 3, 2019 (edited)

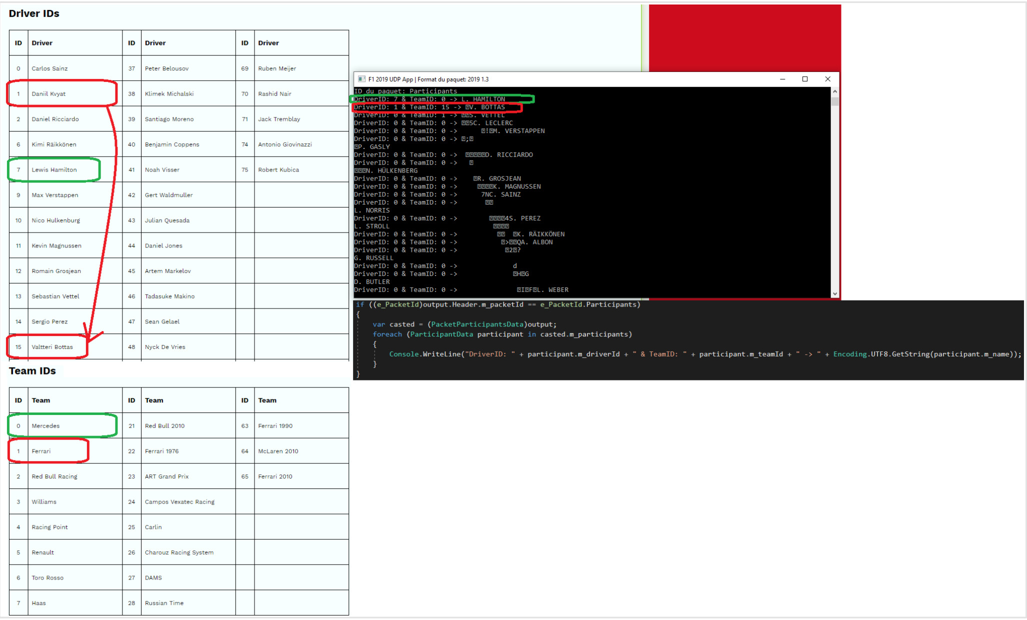
Hello,

I started a new projet (C#) yesterday but i can't show the participants' names correctly while i used UTF-8 format 😕

packet format = 2019

> On 6/20/2019 at 4:29 PM, Faya said:
>
> char m_name[48]; *// Name of participant in UTF-8 format – null terminated // Will be truncated with ... (U+2026) if too long*

and the driverId (from Participant packet, m_participants[X].m_driverId) when i receive it, it's not the same than table here...



Driver IDs

ID	Driver	ID	Driver	ID	Driver
0	Carlos Sainz	37	Peter Belousov	69	Ruben Meijer
1	Daniil Kvyat	38	Klimek Michalski	70	Rashid Nair
2	Daniel Ricciardo	39	Santiago Moreno	71	Jack Tremblay
6	Kimi Räikkönen	40	Benjamin Coppens	74	Antonio Giovinazzi
7	Lewis Hamilton	41	Noah Visser	75	Robert Kubica
9	Max Verstappen	42	Gert Waldmuller		
10	Nico Hulkenburg	43	Julian Quesada		
11	Kevin Magnussen	44	Daniel Jones		
12	Romain Grosjean	45	Artem Markelov		
13	Sebastian Vettel	46	Tadasuke Makino		
14	Sergio Perez	47	Sean Gelael		
15	Valtteri Bottas	48	Nyck De Vries		

Team IDs

ID	Team	ID	Team	ID	Team
0	Mercedes	21	Red Bull 2010	63	Ferrari 1990
1	Ferrari	22	Ferrari 1976	64	McLaren 2010
2	Red Bull Racing	23	ART Grand Prix	65	Ferrari 2010
3	Williams	24	Campos Vexatec Racing		
4	Racing Point	25	Carlin		
5	Renault	26	Charouz Racing System		
6	Toro Rosso	27	DAMS		
7	Haas	28	Russian Time		

```
ID du paquet: Participants
DriverID: 7 & TeamID: 0 -> L. HAMILTON
DriverID: 1 & TeamID: 15 -> ⍰V. BOTTAS
DriverID: 0 & TeamID: 1 -> ⍰⍰S. VETTEL
DriverID: 0 & TeamID: 0 -> ⍰⍰5C. LECLERC
DriverID: 0 & TeamID: 0 ->      ⍰!⍰M. VERSTAPPEN
DriverID: 0 & TeamID: 0 -> ⍰;⍰
⍰P. GASLY
DriverID: 0 & TeamID: 0 ->  ⍰⍰⍰⍰⍰D. RICCIARDO
DriverID: 0 & TeamID: 0 ->   ⍰
⍰⍰⍰N. HÜLKENBERG
DriverID: 0 & TeamID: 0 ->    ⍰R. GROSJEAN
DriverID: 0 & TeamID: 0 ->     ⍰⍰⍰⍰K. MAGNUSSEN
DriverID: 0 & TeamID: 0 ->      7NC. SAINZ
DriverID: 0 & TeamID: 0 ->       ⍰⍰
L. NORRIS
DriverID: 0 & TeamID: 0 ->        ⍰⍰⍰⍰4S. PEREZ
L. STROLL                          ⍰⍰⍰⍰
DriverID: 0 & TeamID: 0 ->          ⍰⍰  ⍰K. RÄIKKÖNEN
DriverID: 0 & TeamID: 0 ->           ⍰>⍰⍰QA. ALBON
DriverID: 0 & TeamID: 0 ->            ⍰2⍰?
G. RUSSELL
DriverID: 0 & TeamID: 0 ->             d
DriverID: 0 & TeamID: 0 ->             ⍰H⍰G
D. BUTLER
DriverID: 0 & TeamID: 0 ->              ⍰I⍰F⍰L. WEBER
```

```
if ((e_PacketId)output.Header.m_packetId == e_PacketId.Participants)
{
    var casted = (PacketParticipantsData)output;
    foreach (ParticipantData participant in casted.m_participants)
    {
        Console.WriteLine("DriverID: " + participant.m_driverId + " & TeamID: " + participant.m_teamId + " -> " + Encoding.UTF8.GetString(participant.m_name));
    }
}
```

Could someone clarify me ?

PS : sorry for my english.

Edited July 4, 2019 by NoizeGamer
new code

thetravisty

T

Members
● 0
2 posts

Posted July 5, 2019

Can anyone help me please?

Ive been able to read all the basic data but I don't understand how the multi-values work.

Example

m_brakesTemperature[4] is supposed to be a uint16 but I get 32 as the answer when i try to read it: readUInt16LE(43).toString()

Sorry for the noob question. Thank you in advance

DaveyGravy

D

Members
+ 10
53 posts

Posted July 6, 2019

> On 7/5/2019 at 11:52 PM, thetravisty said:
>
> Can anyone help me please?
>
> Ive been able to read all the basic data but I don't understand how the multi-values work.
>
> Example
>
> m_brakesTemperature[4] is supposed to be a uint16 but I get 32 as the answer when i try to read it: readUInt16LE(43).toString()
>
> Sorry for the noob question. Thank you in advance

Welcome to programming, you’ve picked a tough project to start with but it sounds like you’re going well. Unsigned 16 bit number is a value between 0 and 65535 so 32 isn’t wrong. Also, I’ve noticed that the brake temps seem to come through as 32 at the start of a session so I think if you wait a bit longer you’ll get the full values correctly. In terms of the multi-value aspect, this just means it’s an array of 4 items which starts at 0; so you can access each item by using m_brakesTemperature[0], m_brakesTemperature[1], etc.

Hope this helps.

🏆 1

thetravisty

T

Members

● 0

2 posts

Posted July 8, 2019

@DaveyGravy Thank you for that information. Ive been able to get more data from the packets because of it. Now I need to figure out why the engine temp is always 89....

Im actually an ex-programmer but I mostly worked in .net - I never came across structs and udp during that time.

1 2 3 4 5 6 NEXT » Page 1 of 6

Create an account or sign in to comment

You need to be a member in order to leave a comment

CREATE AN ACCOUNT

Sign up for a new account in our community. It's easy!

Register a new account

SIGN IN

Already have an account? Sign in here.

Sign In Now

GO TO TOPIC LISTING
Technical Assistance

Home › F1 Games › F1® 2019 Game Forum › Technical Assistance ›

All Activity