1. overview

The BlueFire API for Xamarin development is available on GitHub at <https://github.com/BlueFire-LLC/BlueFire-API-for-Xamarin>. It consists of an API Demo app and the libraries needed to perform custom application development.

1. General Methods and Properties
   1. BlueFire (Constructor )

Parameters: An App Event Handler

Description: Constructs the API and hooks up the Event Handler for the App to receive API connection events. The App Event Handler needs to receive and process the ConnectionState (see Enums below).

* 1. Initialize

Parameters: None

Description: Initialize the API. This is an asynchronous task and must be called directly after the API constructor.

* 1. APIVersion

Type: String

Data: API version number

Description: Can be used to verify current API version. Format is “nn.nn”.

* 1. Dispose

Parameters: None

Description: Disconnects the API from the Adapter and disposes the API.

1. Methods and Properties before Adapter Connection
   1. UseBT21

Type: Boolean

Data: Connect using Bluetooth 2.1 (Bluetooth Classic).

Description: Instructs the API to connect to a Bluetooth Classis Adapter. If UseBT21 and UseBLE are both not set, the API will automatically determine the type of Adapter and set the appropriate property (UseBT21 or UseBLE).

* 1. UseBLE

Type: Boolean

Data: Connect using BLE (Bluetooth Low Energy).

Description: Instructs the API to connect to a BLE Adapter. If UseBLE and UseBT21 are both not set, the API will automatically determine the type of Adapter and set the appropriate property (UseBLE or UseBT21). Note, there are limitations to the BLE Adapter on Android. See the Read.me on GitHub for more information.

* 1. IgnoreJ1939

Type: Boolean

Data: Indicates if the J1939 Databus is to be ignored.

Description: This must be set prior to connecting to the Adapter.

* 1. IgnoreJ1708

Type: Boolean

Data: Indicates if the J1708 Databus is to be ignored.

Description: It is recommended that if J1708/J1587 data is not required this should be set to ignore. This must be set prior to connecting to the Adapter.

* 1. MinInterval

Type: Integer

Data: The minimum interval for retrieving PGNs (in milliseconds).

Description: Sets the Adapter's minimum data transmit interval. This will reduce the Bluetooth data traffic and may help with Adapter connection issues. This is set by default to 500 milliseconds (.5 seconds) for BLE Adapters due to the limitations of Android BLE.

* 1. PerformanceMode

Type: Boolean

Data: The Adapter’s performance mode setting, on (true) or off (false).

Description: Setting this true will improve the retrieval of slower PGNs (Brake Pressures, Odometer, Temps, etc.) by interrupting faster PGNs (RPM, Speed, Fuel Economy, etc.).

* 1. SyncTimeout

Type: Integer

Data: The timeout value (in milliseconds) for synchronous truck data methods.

Description: Sets the timeout value that will cause the truck data methods to timeout if data is not retrieved (see RetrievalMethods.Synchronized in the Enums section). The default is 1000 milliseconds (one second).

* 1. AdapterIdFilter

Type: Array List

Data: Contains Adapter Ids that are to be ignored when establishing an Adapter connection.

Description: Any Adapter Id in the filter list will be ignored when connecting to the Adapter.

* 1. UseInsecureConnection

Type: Boolean

Data: Indicates whether to use an insecure connection when connecting with Bluetooth 2.1.

Description: Set this true to force the API to connect to the Adapter using an insecure Bluetooth connection. Some older Android devices will not connect to the Adapter using a secured Bluetooth connection but only with an insecure connection. Note that there is no security risk with using an insecure connection.

* 1. MaxConnectAttempts

Type: Byte

Data: The maximum number of retries the API should attempt during an Adapter connection (default is 10)

Description: Each connection attempt can take several seconds so this value should be set carefully and accurately. Issuing a Disconnect command will stop the connection.

* 1. MaxReconnectAttempts

Type: Byte

Data: The maximum number of retries the API should attempt during an Adapter reconnection (default is 10)

Description: Each reconnection attempt can take several seconds so this value should be set carefully and accurately. Issuing a Disconnect command will stop the reconnection.

* 1. AdvertisementTimeout

Type: Byte

Data: The time in seconds that the API will wait for a BlueFire BLE adapter to be found during its advertisement scan. The default is 5 seconds.

Description: Normally this is very quick but if there are many BLE devices in the general vicinity, finding a BlueFire adapter could take longer.

* 1. DiscoveryTimeout

Type: Byte

Data: The time in seconds that the API should wait for the Bluetooth Discovery process to find an Adapter (in milliseconds, default is 30 seconds.

Description: This will only occur if the Adapter is not initially paired manually and will only occur once for each Adapter the API connects to.

* 1. ConnectTimeout

Type: Integer

Data: The timeout value (in milliseconds) for synchronous Adapter connection.

Description: Sets the timeout value that will cause the Connect method to timeout if the App fails to connect to the Adapter. The default is 1000 milliseconds (one second).

* 1. Connect

Parameters:

Synchronized (Boolean - optional, default is false)

Returns:

Connection Status (Boolean)

Description: Initiate a connection attempt to the Adapter. If not synchronized then this is an asynchronous task so call it appropriately. If Synchronized is set, the method will block until either a connection is established, the connection fails (eg. authentication), or a timeout occurs. If the Connect method fails, the last state of the connection can be examined with the ConnectionState (see Enums ConnectionStates). The timeout value can be set with the ConnectTimeout property.

1. Methods and Properties after Adapter Connection
   1. ConnectionState

Type: ConnectionStates

Data: The state of the connection during and after an Adapter connection.

Description: This is the same value that is passed to the App in the Data Event Handler. It can be used when connecting to the Adapter synchronously.

* 1. AdapterCount

Type: Integer

Data: The number of adapters that are paired with the computer.

Description: Windows UWP caches the BLE information (Device, Services, etc.) for the first adapter paired and any additional adapters that are paired will not be connected. In order to connect to multiple adapters, all previous adapters must be removed so that the adapter to be connected is the first and only adapter paired.

* 1. SignalStrength

Type: Integer

Data: The signal strength (RSSI) of the Adapter.

Description: This is the raw RSSI of the BLE device. If there is no signal the value will be -127.

* 1. IsCompatible

Type: Boolean

Data: Indicates if the Adapter is compatible with the API.

Description: The Adapter must have Firmware Version 3.1 or greater.

* 1. FirmwareVersion

Type: String

Data: The Adapter’s Firmware Version

Description: Format is “nn.nn”.

* 1. HardwareVersion

Type: String

Data: The Adapter’s Hardware Version

Description: Format is “nn.nn”.

* 1. SerialNo

Type: String

Data: The Adapter’s Serial Number.

Description: The serial number assigned at manufacturing.

* 1. IsCANAvailable

Type: Boolean

Data: Indicates if the J1939 CAN bus is available.

Description: In most trucks the CAN bus is detectable only when the key is on.

* 1. IsReconnecting

Type: Boolean

Data: Indicates that the API has lost connection to the Adapter and is attempting to reconnect.

Description: See the API Demo for best use for this.

* 1. ReconnectReason

Type: String

Data: The reason the API had to reconnect to the adapter.

Description: This should be logged for later analysis.

* 1. LedBrightness

Type: Integer

Data: The brightness setting for the Adapter LEDs.

Description: This must be in the range of 5 to 100. Note, the brightness level is not allowed to be less than 5 due to the importance of the state of the LEDs. The API must be connected to the Adapter before setting this property.

* 1. SleepMode

Type: SleepModes

Data: Sleep mode of the Adapter

Description: The Adapter can be set for NoSleep or WakeOnApp. When WakeOnApp is used, the Adapter will go into sleep mode two minutes after no activity and will wake up when the API initiates a connection attempt. The API must be connected to the Adapter before setting this property.

* 1. AdapterId

Type: String

Description: Sets the Bluetooth Device Id that the API will use to compare with the connecting Adapter to verify a valid connection. This check only occurs if ConnectToLastAdapter or SecureAdapter is set to true. This should be called before connecting to the Adapter.

* 1. ConnectToLastAdapter

Type: Boolean

Data: Indicates whether to connect to the Adapter using the last Bluetooth Device Id.

Description: Setting this true will cause the API to only connect to an Adapter that has its Bluetooth Device Id equal to the AdapterId that was saved the last time a connection was successful.

* 1. UserName

Type: String

Data: The security user name used for authenticating with the Adapter.

Description: If security authentication is used this is the user name assigned to the Adapter.

* 1. Password

Type: String

Data: The security password used for authenticating with the Adapter.

Description: If security authentication is used this is the password assigned to the Adapter.

* 1. UpdateSecurity

Parameters:

SecureAdapter (Boolean)

UserName (String, 0-20 characters)

Password (String, 0-12 characters)

Description: SecureAdapter will secure the adapter to the app and not allow any other app to connect to the adapter. Likewise the app will not be able to connect to any other adapter. The UserName and Password are case sensitive. Adapter authentication occurs at connection time so these must be set prior to connecting to the Adapter. If security authentication is not being used this method is not necessary.

* 1. SetAdapterTime

Parameters: None

Description: Sets the Adapters clock to the current App’s UTC date and time. This is primarily used for ELD recording.

* 1. ErrorMessage

Type: String

Data: An error message from the adapter.

Description: This is a system error message. It should be logged for later analysis.

* 1. ErrorException

Type: Exception

Data: An error exception from the adapter.

Description: This is an exception thrown by the API. It should be logged for later analysis.

* 1. ResetAdapter

Parameters: None

Description: Resets the Adapter to factory settings. The API will disconnect from the Adapter and start the factory reset which will take approximately 45 seconds. A re-connection should not be attempted until the factory reset is complete. To manually reset the Adapter follow the instructions in the Appendix.

* 1. RebootAdapter

Parameters: None

Description: Initiates a reboot of the Adapter. The API must be connected to the Adapter before calling this method. When the Adapter reboots, the API will detect a disconnection and will attempt to reconnect.

* 1. Disconnect

Parameters:

WaitForDisconnect (Boolean - optional, default is false)

Description: Disconnects the API from the Adapter. If the WaitForDisconnect parameter is true, the call will block until the Adapter has disconnected. This is an asynchronous task so call it appropriately.

* 1. Dispose

Parameters: None

Description: Disconnects the API from the Adapter and disposes the API.

1. Truck Data Methods and Properties after Adapter Connection
   1. Truck

Type: Class

Data: See data retrieval methods below.

Description: Contains all the truck related information retrieved from the Adapter.

* 1. Faults

Type: Class

Data: See the GetFaults method below.

Description: Contains active fault information retrieved from the Adapter.

* 1. ClearData

Parameters: None

Description: Clears all the truck data retrieval requests from the Adapter and any truck data already retrieved from the API.

* 1. GetVehicleId

Parameters: None

Returns: Vehicle Id (VIN)

Description: Retrieves asynchronously the vehicle id (VIN) from the Adapter. The VIN will be returned in the Truck Data class. The API must be connected to the Adapter before calling this method.

* 1. GetVehicleIdSync

Parameters: None

Returns:

False if unable to retrieve the VIN, otherwise the VIN will be returned in the Truck Data class.

Description: Retrieves synchronously the vehicle id (VIN) from the Adapter. The API must be connected to the Adapter before calling this method. Note, this is a blocking method.

* 1. GetVehicleData

Parameters: None

Returns:

Make

Model

Serial Number

Unit Number

Description: Retrieves asynchronously the Engine and Truck (Cab/Body) data from the Adapter. The API must be connected to the Adapter before calling this method.

* 1. GetEngineData1

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

RPM

Percent Torque

Driver Torque

Torque Mode

Description: Retrieves engine data from the Adapter. The API must be connected to the Adapter before calling this method.

* 1. GetEngineData2

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Percent Load

Accelerator Pedal Position

Description: Retrieves engine data from the Adapter. The API must be connected to the Adapter before calling this method.

* 1. GetEngineData3

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Max Speed

Vehicle Speed

Brake Switch

Clutch Switch

Park Brake Switch

Cruise Control Switches

Description: Retrieves engine data from the Adapter. The API must be connected to the Adapter before calling this method.

* 1. GetDistance

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Engine Distance – high resolution if available otherwise low resolution

High Resolution Distance,

Low Resolution Distance,

OEM Odometer – high resolution if available otherwise low resolution

High Resolution Distance,

Low Resolution Distance,

Description: Retrieves Engine Distance (high resolution and/or low resolution) and Odometer from the Adapter. Note, Engine Distance is retrieved from the Engine ECM and Odometer is retrieved from the OEM ECM (if available).

* 1. GetOdometer

Same as GetDistance.

* 1. GetTemps

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Oil Temp

Coolant Temp

Intake Manifold Temp

Description: Retrieves engine temperature data from the Adapter. The API must be connected to the Adapter before calling this method.

* 1. GetPressures

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Oil Pressure

Coolant Pressure

Intake Manifold (Boost) Pressure

Description: Retrieves engine temperatures from the Adapter. The API must be connected to the Adapter before calling this method.

* 1. GetCoolantLevel

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Coolant Level

Description: Retrieves the engine coolant level from the Adapter. The API must be connected to the Adapter before calling this method.

* 1. GetFuelData

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Fuel Rate

Throttle Position

Instant Fuel Economy

Average Fuel Economy

Total Low Res Fuel Used

Total High Res Fuel Used

Total Idle Fuel Used

Description: Retrieves engine fuel data from the Adapter. The API must be connected to the Adapter before calling this method.

* 1. GetEngineHours

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Total Idle Hours

Total Engine Hours

Description: Retrieves engine hours from the Adapter. The API must be connected to the Adapter before calling this method.

* 1. GetBrakeData

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Brake Application Pressure

Brake Primary Pressure

Brake Secondary Pressure

Description: Retrieves brake data from the Adapter. The API must be connected to the Adapter before calling this method.

* 1. GetTransmissionGears

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Current Gear

Selected Gear

Description: Retrieves the transmission gears from the Adapter. This will only be available if the transmission ECM provides the data. The API must be connected to the Adapter before calling this method.

* 1. GetBatteryVoltage

Parameters:

RetrievalMethod (RetrievalMethods - see Enums, optional, default is OnChange)

Interval (Integer – in milliseconds, optional, required for RetrievalMethod OnInterval).

Returns:

False if unable to retrieve synchronized data, otherwise the following data will be returned in the Truck Data class (asynchronously if not synchronizing):

Battery Potential (Voltage

Description: The API must be connected to the Adapter before calling this method.

* 1. GetFaults

Parameters: None

Returns:

SPN

FMI

Conversion

Occurrence

Description

Description: Retrieves all active faults from the Adapter. Use the Faults class to retrieve the Fault data. The API must be connected to the Adapter before calling this method.

* 1. ResetFaults

Parameters: None

Description: Resets any active faults. Note, the API sends the appropriate Fault Reset command to all ECUs. It is up to the ECUs to accept and process the Reset command. The API must be connected to the Adapter before calling this method.

1. J1939 and J1708 Methods and Properties
   1. PGNData

Type: Class

Data:

PGN (Integer)

Source (Integer)

Data (Byte [8])

Description: Contains the PGN data that is returned by the Adapter. This is used when requesting a PGN with the SendPGN method.

* 1. MonitorPGN

Parameters:

Source (Integer)

PGN (Integer)

Interval (Integer) – in milliseconds, optional, default is 0 (on data change).

IsOnRequestPGN (Boolean) – optional, default is false.

Description: Monitors a PGN. Data will be returned based on the Interval. If IsOnRequestPGN is true, the PGN will be sent to the CAN Bus as an on-request PGN. The maximum number of concurrent monitoring PGNS is 20. Use the StopMonitoringPGN method to remove a PGN from this count. The API must be connected to the Adapter before calling this method.

* 1. StopMonitoringPGN

Parameters:

Source (Integer)

PGN (Integer)

Description: Stops monitoring a PGN. This will remove the PGN from the maximum number of monitoring PGNs (20). The API must be connected to the Adapter before calling this method.

* 1. RequestPGN

Parameters:

Source (Integer)

PGN (Integer)

IsOnRequestPGN (Boolean – optional, default is false)

Description: Request PGN data once only. If IsOnRequestPGN is false, the PGN will be monitored and will return the first data received. If IsOnRequestPGN is true, the PGN will be sent to the CAN Bus as an on-request PGN. The API must be connected to the Adapter before calling this method.

* 1. SendPGN

Parameters:

PGN (Integer)

Priority (Byte - optional, default is 6)

Source (Integer - optional, default is 43)

PGN Data (Byte Array, size is 8)

Description: Send a non-standard API PGN to the Adapter. Any response can be retrieved with the PGNData property. The API must be connected to the Adapter before calling this method.

* 1. RequestPID

Parameters:

MID (Integer)

PID (Integer)

IsOnRequestPGN (Boolean – optional, default is false)

Description: Request J1587 PID data once only. If IsOnRequestPGN is false, the PID will be monitored and will return the first data received. If IsOnRequestPGN is true, the PID will be sent to the J1708 Bus as an on-request PID. The API must be connected to the Adapter before calling this method.

* 1. MonitorPID

Parameters:

MID (Integer)

PID (Integer)

Interval (Integer – in milliseconds, optional, default is 0, on data change)

IsOnRequestPGN (Boolean – optional, default is false)

Description: Monitors a J1587 PID. Data will be returned based on the Interval. If IsOnRequestPGN is true, the PID will be sent to the J1708 Bus as an on-request PID. The API must be connected to the Adapter before calling this method..

* 1. StopMonitoringPID

Parameters:

MID (Integer)

PID (Integer)

Description: Stops monitoring a J1587 PID. The API must be connected to the Adapter before calling this method.

1. Enums
   1. RetrievalMethods

**OnChange** (default)

Data will be retrieved when it changes. Use this with caution with Android as BLE may become unstable if the data rate is high.

**OnInterval**

Data will be retrieved on the specified interval only if it changes. If the interval is less than MinInterval, MinInterval will be used.

**Synchronized**

Data will be retrieved immediately. This will force a blocking call. Timeout occurs after the SyncTimeout value expires.

* 1. ConnectionStates

**NotConnected**

Initial state of the Adapter

**Initializing**

**Initialized**

Occurs during

**Discovering**

Occurs if Bluetooth Discovery is invoked

**Connecting**

**Connected**

Occurs when connecting to the Adapter and a Bluetooth connected is established.

**Authenticating**

**Authenticated**

**NotAuthenticated**

Occurs after connecting to the Adapter and the API is authenticating the Adapter version and the App security.

**Ready**

**RetrievingData**

Occurs after the App is authenticated with the Adapter and the Adapter settings are being retrieved.

**KeyIsTurnedOn**

Occurs when the ignition key if turned on.

**KeyIsTurnedOff**

Occurs when the ignition key is turned off.

**Disconnecting**

**Disconnected**

Occurs when the App or the API is disconnecting from the Adapter.

**Reconnecting**

**Reconnected**

**NotReconnected**

Occurs when the API is reconnecting to the Adapter after a loss of connection.

**DataAvailable**

Occurs when there is Truck data available for the App to process.

**CANFilterFull**

Occurs when too many data requests have been sent to the Adapter. This applies specifically to J1939 PGNs.

**DataError**

Occurs when an Adapter data error is detected by the API. This will cause the API to disconnect and reconnect the Adapter. The ErrorMessage property will contain the data error message.

**CommTimeout**

**ConnectTimeout**

**AdapterTimeout**

Occurs when a timeout happens between the Adapter and the App. The Adapter will be disconnected.

**NoAdapter**

Indicates that an Adapter was not detected when attempting to connect. This will also occur if pairing is canceled.

**BluetoothNA**

Indicates that Bluetooth is not available on the connecting device.

**IncompatibleVersion**

Occurs then the API is authenticating the Adapter version and discovers an incompatible firmware version (firmware version 3.9+ is required).

**SystemError**

Occurs when the API encounters a code exception. The ErrorMessage and ErrorException properties will contain the exception information.

1. truck data
   1. RPM

Type: Integer

Data: Engine RPM

* 1. Speed

Type: Float

Data: Vehicle Road Speed (metric, kph)

* 1. AccelPedal

Type: Float

Data: Accelerator Pedal Position (0-100%)

* 1. ThrottlePos

Type: Float

Data: Throttle Position (0-100%)

* 1. MaxSpeed

Type: Integer

Data: Maximum Set Speed (kph)

* 1. HiResMaxSpeed

Type: Float

Data: High Resolution Maximum Set Speed (kph)

* 1. Distance

Type: Float

Data: High/Low Resolution Engine Distance (meters)

* 1. HiResDistance

Type: Float

Data: High Resolution Engine Distance (meters)

* 1. LoResDistance

Type: Float

Data: Low Resolution Engine Distance (kilometers)

* 1. Odometer

Type: Float

Data: High/Low Resolution OEM Distance (meters)

* 1. HiResOdometer

Type: Float

Data: High Resolution OEM Distance (meters)

* 1. LoResOdometer

Type: Float

Data: Low Resolution OEM Distance (kilometers)

* 1. TotalHours

Type: Float

Data: Total Engine Hours

* 1. IdleHours

Type: Float

Data: Total Engine Idle Hours

* 1. PctLoad

Type: Integer

Data: Percent Load

* 1. PctTorque

Type: Integer

Data: Percent Torque

* 1. DrvPctTorque

Type: Integer

Data: Drivers Percent Torque

* 1. TorqueMode

Type: TorqueModes

Data: Torque Mode

* 1. FuelRate

Type: Float

Data: Fuel Rate (liters / hour)

* 1. FuelUsed

Type: Float

Data: Total Fuel Used (liters)

* 1. HiResFuelUsed

Type: Float

Data: High Resolution Total Fuel Used (liters)

* 1. IdleFuelUsed

Type: Float

Data: Total Idle Fuel Used (liters)

* 1. AvgFuelEcon

Type: Float

Data: Average Fuel Economy (kilometers / liter)

* 1. InstFuelEcon

Type: Float

Data: Instant Fuel Economy (kilometers / liter)

* 1. BrakeAppPressure

Type: Float

Data: Brake Application Pressure (kPa)

* 1. Brake1AirPressure

Type: Float

Data: Brake Primary Air Pressure (kPa)

* 1. Brake2AirPressure

Type: Float

Data: Brake Secondary Air Pressure (kPa)

* 1. OilTemp

Type: Float

Data: Oil Temperature (Celsius)

* 1. OilPressure

Type: Integer

Data: Oil Pressure (kPa)

* 1. IntakeTemp

Type: Float

Data: Intake Temperature (Celsius)

* 1. IntakePressure

Type: Float

Data: Intake (Boost) Pressure (kPa)

* 1. CoolantTemp

Type: Float

Data: Coolant Temperature (Celsius)

* 1. CoolantLevel

Type: Float

Data: Coolant Level (0-100%)

* 1. CoolantPressure

Type: Float

Data: Coolant Pressure (kPa)

* 1. TransCurrentGear

Type: Float

Data: Transmission’s Current Gear

* 1. TransSelectedGear

Type: Float

Data: Transmission’s Selected Gear

* 1. BatteryPotential

Type: Float

Data: Battery Potential (Voltage)

* 1. EngineVIN

Type: String

Data: Vehicle Identification Number from the engine

* 1. CabBodyVIN

Type: String

Data: Vehicle Identification Number from the truck (cab/body).

* 1. Engine.Make

Type: String

Data: Engine Make

* 1. Engine.Model

Type: String

Data: Engine Model

* 1. Engine.SerialNo

Type: String

Data: Engine Serial Number

* 1. Engine.UnitNo

Type: String

Data: Engine Unit Number

* 1. CabBody.Make

Type: String

Data: Truck Make

* 1. CabBody.Model

Type: String

Data: Truck Model

* 1. CabBody.SerialNo

Type: String

Data: Truck Serial Number

* 1. CabBody.UnitNo

Type: String

Data: Truck Unit Number

* 1. CruiseState

Type: CruiseControlStates

Data: Cruise Control State

* 1. CruiseSpeed

Type: Integer

Data: Cruise Control Set Speed (kph)

* 1. CruiseSwitches

Type: SwitchStates

Data: Cruise Control Switches

CruiseActive

CruiseSwitch

CruiseSet

CruiseCoast

CruiseResume

CruiseAccel

* 1. ActiveFaultsCount

Type: Integer

Data: Number of Active Faults

* 1. Faults.Items[x].Source

Type: Byte

Data: Fault Source

* 1. Faults.Items[x].SPN

Type: Integer

Data: Fault SPN

* 1. Faults.Items[x].FMI

Type: Integer

Data: Fault FMI

* 1. Faults.Items[x].Conversion

Type: Boolean

Data: Fault Conversion

* 1. Faults.Items[x].Occurrence

Type: Integer

Data: Fault Occurrence

* 1. Faults.Items[x].LampType

Type: LampTypes

Data: Fault Lamp

* 1. Faults.Items[x].Code

Type: String

Data: Fault SPN and FMI

* 1. Faults.Items[x].Description

Type: String

Data: Fault SPN Description

* 1. Faults.Items[x].SourceDescription

Type: String

Data: Fault Source Description

* 1. Faults.Items[x].ToString()

Type: String

Data: Full description including Source, SPN, and FMI descriptions.

1. Truck Enums
   1. SleepModes

NoSleep

NA

WakeOnApp

* 1. SwitchStates

Off

On

Error

NA

* 1. CruiseControlStates

Off

Hold,

Accelerate,

Decelerate,

Resume,

Set,

AccelOverride,

NA

* 1. TorqueModes

LowIdleGovernor,

AccelPedal,

CruiseControl,

PTOGovernor,

RoadSpeedGovernor,

ASRControl,

TransControl,

ABSControl,

TorqueLimiting,

HighSpeedGovernor,

BrakingSystem,

RemoteAccelerator,

ServiceProcedure,

NotDefined,

Other,

NA

1. ELD Methods and Properties
   1. ELD

Type: Class

Description: This is the ELD class that is instantiated by the API. All methods and properties are based in this class.

* 1. Connect

Parameters: None

Description: Connects the App to the Adapter for ELD recording. This must be called after the Adapter is connected.

* 1. IsCompatible

Type: Boolean

Data: Indicates whether the adapter is compatible with ELD recording.

Description: ELD recording requires Adapter firmware 3.10+. If the Adapter firmware is not at this level this value will be set to false and ELD recording will not be allowed.

* 1. DriverId

Type: String (0-22 characters)

Data: The driver id that will be recorded.

Description: The driver id is optional and if set will be included as the first ELD record when ELD recording is started. Note, the driver id is not persistent in the Adapter so it is the App’s responsibility to save it.

* 1. ELDInterval

Type: Integer

Data: The interval for ELD recording (in minutes).

Description: Set this to record ELD records at a specific interval. The default is 60 minutes.

* 1. AlignELD

Type: Boolean

Data: Indicates whether ELD records should be aligned to the hour.

Description: Aligning the ELD records will ensure that an ELD record will be recorded at the top of the hour. An example of this would be an ELD interval of 15 minutes would record at the 15, 30, 45 minute mark and at the top of the hour. An interval that greater than an hour or one that cannot be aligned to the hour is not allowed if this option is selected.

* 1. RecordIFTA

Type: Boolean

Data: Indicates whether IFTA records should be recorded by the Adapter.

Description: Set this to have the Adapter record IFTA records along with ELD records. Note that recording IFTA records will reduce the available ELD recording time (ie. the number of records available for ELD recording).

* 1. IFTAInterval

Type: Integer

Data: The interval for IFTA recording (in minutes).

Description: Set this to record IFTA records at a specific interval. The default is 1 minute.

* 1. AlignIFTA

Type: Boolean

Data: Indicates whether IFTA records should be aligned to the hour.

Description: See AlignELD Description.

* 1. RecordStats

Type: Boolean

Data: Indicates whether Statistical records should be recorded by the Adapter.

Description: Set this to have the Adapter record Statistical records along with ELD records. Note that recording Statistical records will reduce the available ELD recording time (ie. the number of records available for ELD recording).

* 1. StatsInterval

Type: Integer

Data: The interval for Statistical recording (in minutes).

Description: Set this to record Statistical records at a specific interval. The default is 60 minutes.

* 1. AlignStats

Type: Boolean

Data: Indicates whether Statistical records should be aligned to the hour.

Description: See AlignELD Description.

* 1. IsHourAligned

Parameters:

Interval (Float)

Type: Boolean

Data: Returns whether the interval can be aligned to the hour.

Description: This is a helpful method to ensure that the intervals and aligns are correctly set.

* 1. IsStreaming

Type: Boolean

Data: Indicates whether the Adapter is to stream the ELD records to the App.

Description: This is a setting that is saved by the API for the App to reference. Note, this does not start or stop streaming, only the Start/Stop Streaming methods do that.

* 1. IsSecured

Type: Boolean

Data: Indicates whether the ELD recording is secured to the App and Adapter.

Description: Instructs the Adapter to secure the ELD recording and not allow any other App to stop recording or upload or delete records.

* 1. IsAccessSecured

Type: Boolean

Data: Indicates whether ELD recording has been secured by another App.

Description: If this is set then the ELD recording has been started by another App and that App has secured the recording. You will only be allowed to stream the ELD records.

* 1. RecordingMode

Type: RecordingModes

Data: The RecordingMode for recording ELD data.

Description: This is the RecordingMode (see Enums) that determines if and how the ELD records will be recorded by the Adapter.

* 1. SetRecordingMode

Parameters:

RecordConnected (Boolean)

RecordDisconnected (Boolean)

Description: Sets the RecordingMode based on the RecordConnected and RecordDisconnected parameters. This is a helper method for setting the RecordingMode.

* 1. IsRecordingLocally

Type: Boolean

Data: Indicates that the RecordingMode is set to record ELD records locally (App only).

Description: This is a helper property that references RecordingMode to indicate that the ELD records are only to be recorded by the App. The Adapter will not be doing any ELD recording. This is the same as IsStreaming true and RecordingMode set to RecordNever or RecordDisconnected.

* 1. IsRecordingConnected

Type: Boolean

Data: Indicates that the RecordingMode is set to record ELD only when the App is connected to the Adapter.

Description: This is a helper property that references RecordingMode to indicate that the ELD records are to be recorded by the Adapter only when it is connected to the App. This is the same as RecordingMode set to Always or RecordConnected.

* 1. IsRecordingDisconnected

Type: Boolean

Data: Indicates that the RecordingMode is set to record ELD only when the App is disconnected from the Adapter.

Description: This is a helper property that references RecordingMode to indicate that the ELD records are to be recorded by the Adapter only when it is disconnected from the App. This is the same as RecordingMode set to Always or RecordDisconnected.

* 1. IsStarted

Type: Boolean

Data: Indicates whether the Adapter has started ELD recording.

Description: This is set by the Start/Stop Recording methods. This will also indicated if the Adapter is recording at the time the App connects to it which will occur if the App disconnects without stopping ELD recording.

* 1. LocalRecordNo

Type: Integer

Data: The current local ELD record number.

Description: This is the record number of the current (ie. last) ELD record that was retrieved when streaming or recording locally. This will be updated whenever an ELD record is retrieved with IsStreaming true and RecordingMode set to RecordNever or RecordDisconnected (see property IsRecordingLocally).

* 1. CurrentRecordNo

Type: Integer

Data: The current Adapter ELD record number.

Description: This is the record number of the current (ie. last) ELD record that was recorded by the Adapter. This will be updated whenever an ELD record is retrieved with IsStreaming true and RecordingMode set to RecordConnected or RecordAlways.

* 1. RemainingTime

Type: Float

Data: The remaining time (in minutes) that is available for ELD recording.

Description: This is the duration that is remaining for all ELD recordings including IFTA and Statistical records. This is equivalent to the amount of ELD memory that is remaining. Note, that when there is no time remaining, ELD recording will stop. It is the responsibility of the application to monitor this and perform a Reset when necessary (see the Reset method). This will be updated whenever an ELD record is retrieved (streaming or direct request).

* 1. RemainingPercent

Type: Float

Data: The percentage of memory that is available for ELD recording.

Description: This is the percentage of ELD memory that is remaining for ELD recording (including IFTA and Statistical records). When the percentage is 10% or less, the adapter will pulsate a magenta color. When the percentage reaches zero the adapter will go solid magenta and stop recording. This will be updated whenever an ELD record is retrieved (streaming or direct request).

* 1. IsDataRetrieved

Type: Boolean

Data: Indicates that ELD records have been retrieved and ready for processing.

Description: Use this to process ELD records after retrieving them by streaming or direct record request.

* 1. GetRecords

Parameters:

Start Record Number (Integer)

End Record Number (Integer)

Description: Retrieves the requested ELD records. The data will be returned in the ELD data property and an ELD IsDataRetrieved event will be raised.

* 1. DeleteRecords

Parameters:

Number of Records (Integer)

Description: Deletes the specified number of records starting at record number 1. Note that this does not erase the record from memory but just moves the first ELD record to the record after those deleted. Note also that because of the sequential nature of the ELD recordings, it is not possible to delete records in the middle of the recordings.

* 1. Reset

Parameters: None

Description: Deletes all ELD records and erases memory. This method should only be called when there is not enough ELD memory to store the required ELD records within the time remaining (see the RemainingTime and RemainingPercent properties). Caution, calling this method too often can affect the life of the ELD memory.

* 1. StartRecording

Parameters: None

Description: Instructs the Adapter to start ELD recording. Note, all recording parameters must be set prior to starting ELD recording.

* 1. StopRecording

Parameters: None

Description: Instructs the Adapter to stop ELD recording.

* 1. StartStreaming

Parameters: None

Description: Instructs the Adapter to start streaming the ELD records as they are being recorded. The data will be returned in the ELD data property and an ELD IsDataRetrieved event will be raised for each record received.

* 1. StopStreaming

Parameters: None

Description: Instructs the Adapter to stop streaming the ELD records.

1. ELD Data
   1. Record Number

Type: Integer

Data: The ELD record number (relative to 1)

Record Types: All

* 1. Record Id

Type: RecordIds

Data: Record Id (see RecordIds)

Record Types: All

* 1. Time

Type: DateTime

Data: The DateTime stamp of the record. When the App starts ELD recording, the current date-time is sent to the Adapter. This will persist in the Adapter for as long as the Adapter is powered. If the Adapter is power cycled, the current date-time will be set to 1/1/2042 UTC.

Record Types: All

* 1. Driver Id

Type: String

Data: Driver Identification – from the App

Record Types: ID

* 1. VIN

Type: String

Data: Engine VIN

Record Types: ID

* 1. Distance

Type: Single

Data: Vehicle Distance (miles) – from the Engine

Record Types: ELD, IFTA, Statistics

* 1. Odometer

Type: Single

Data: Truck Distance (miles) – from the Instrument Cluster

Record Types: ELD, IFTA

* 1. TotalHours

Type: Single

Data: Total Engine Hours

Record Types: ELD, Statistics

* 1. IdleHours

Type: Single

Data: Total Engine Idle Hours

Record Types: Statistics

* 1. TotalFuel

Type: Single

Data: Total Fuel Used (gallons)

Record Types: IFTA, Statistics

* 1. IdleFuel

Type: Single

Data: Total Idle Fuel Used (gallons)

Record Types: Statistics

* 1. Latitude

Type: Double

Data: GPS Latitude – This is only populated if GPS coordinates are available on the CAN bus. Note, some OEMs have on-board GPS and some off-board manufacturers such as Qualcomm may also publish GPS coordinates.

Record Types: ELD, IFTA

* 1. Longitude

Type: Double

Data: GPS Longitude – See Latitude.

Record Types: ELD, IFTA

* 1. Custom

Type: Byte Array

Data: 22 bytes.

Record Types: Custom

1. ELD enums
   1. RecordingModes

**RecordNever** (default)

ELD data will not be recorded by the Adapter (same as when RecordConnected and RecordDisconnected are both cleared.

**RecordConnected**

ELD data will be recorded by the Adapter only when the App is connected to the Adapter.

**RecordDisconnected**

ELD data will be recorded by the Adapter only when the App is disconnected from the Adapter.

**RecordAlways**

ELD data will always be recorded by the Adapter (same as when RecordConnected and RecordDisconnected are both set.

* 1. RecordId (Record Type)

NA (internal)

Used internally by the API.

DriverId (ID)

Received from the App before ELD is started.

VIN (ID)

Retrieved by the adapter when ELD is started.

StartEngine (ELD)

RPM > 0 and Distance and Total Hours have been retrieved by the adapter.

StartDriving (ELD)

Engine has started and speed >= 5 mph.

Driving (ELD)

Started driving or is driving and ELD interval has elapsed.

StopDriving (ELD)

Was driving and speed = 0 for at least 3 seconds.

StopEngine (ELD)

Stopped driving and RPM = 0.

IFTA (IFTA)

Started driving or is driving and IFTA interval has elapsed.

Stats (Statistics)

Started engine or is driving and Statistics interval has elapsed.

Waiting (internal)

App started streaming and no records have been streamed.

Custom (custom)

App defined record id. Custom record ids start at 128 (ie. custom record id 1 will be recorded as ELD record id 129.

1. Adapter LEDs
   1. Normal Operation

Blue Blinking

The Adapter is waiting for the App to connect.

Blue Solid

The App is connected and the Adapter is receiving ECM data.

Green Blinking

The App is connected but the Adapter is waiting for the ECMs. Most likely this is because the key is off.

Red Blinking

The Adapter is in a user reset mode. This will occur one minute after plugging it in and not connecting to it. Unplug the adapter while it is blinking red, plug it back in, wait for it to blink red again, unplug it again, plug it back in and the adapter will reset itself to factory settings.

Red Solid

The Adapter is writing data to flash memory. DO NOT unplug it or you risk causing a factory reset.

* 1. ELD Operation

This only applies when the Adapter is recording data (RecordingMode set to RecordAlways, RecordConnected, or RecordDisconnected). If recording locally, the LEDs will function normally.

Blue Solid with Magenta Flashes

The App is connected and the Adapter is recording ELD data. The LEDs will flash Magenta each time a record is recorded.

Green Solid with Magenta Flashes

The App is not connected and the Adapter is recording ELD data. The LEDs will flash Magenta each time a record is recorded.

Green Pulsating

The App is not connected and the Adapter is recording ELD data but waiting for ECM data. Most likely the key if off.

Magenta Pulsating with Blue Flashes

The App is not connected and the Adapter is recording ELD data but the ELD date-time is not set. This is caused by power cycling the adapter after ELD recording has been started. It is recommended to connect the App to set the date-time. If the LEDs have been dimmed they will be brought to full brightness.

Magenta Pulsating with Cyan Flashes

This indicates that the ELD memory is almost full (90%). If the LEDs have been dimmed they will be brought to full brightness.

Magenta Solid

The ELD memory is full and ELD recording has stopped. If the LEDs have been dimmed they will be brought to full brightness.

1. Appendix
   1. Common J1939 Sources

0 = Engine

3 = Transmission

11 = Brakes

23 = Instruments

25 = Climate Control

33 = Body

49 = Cab

85 = AfterTreatment

255 = Global

* 1. Common J1939 PGNs

57344 = Cab Message (1 s)

61443 = Engine Controller 2 (50 ms)

61444 = Engine Controller 1 (50 ms)

64777 = High Resolution Fuel Consumption (1 s)

64920 = AfterTreatment Information (On Request)

65110 = DEF Tank (1 s)

65198 = Air Pressure (1 s)

65203 = Fuel Information (On Request)

65213 = Fan Drive (1s)

65217 = High Resolution Vehicle Distance (1 s)

65226 = DM1 (Faults 1 s)

65227 = DM2 (InActive Faults On Request)

65228 = DM3 (InActive Fault Reset)

65229 = DM4 (Freeze Frames On Request)

65235 = DM11 (Active Fault Reset)

65242 = Software Id (On Request)

65244 = Idle Operation (On Request)

65248 = Vehicle Distance (100 ms)

65253 = Engine Hours Revolutions (On Request)

65257 = Fuel Consumption (On Request)

65259 = Component Id (On Request)

65260 = Vehicle Id (On Request)

65262 = Engine Temperature (1 s)

65261 = Cruise Control Speed Setup (On Request)

65263 = Engine Fluid Level Pressure (500 ms)

65265 = Cruise Control Vehicle Speed (100 ms)

65266 = Fuel Economy (100 ms)

65269 = Ambient Conditions (1 s)

65270 = Inlet Exhaust Condition (500 ms)

65271 = Vehicle Electrical Power (1 s)

65272 = Transmission Fluids (1 s)

65274 = Brakes (1 s)

65276 = Dash Display (1 s)

* 1. Common J1587 MIDs

0 = Global

128 = Engine

130 = Transmission

136 = Brakes

140 = Instruments

142 = Vehicle Management

146 = Climate Control

171 = Driver Information

249 = Body Controller

* 1. Common J1587 PIDs

41 = Cruise Switches

44 = Fault Lamps

51 = Throttle Position

68 = Percent Torque

70 = Parking Brake

74 = Max Speed

84 = Speed

85 = Cruise Control

86 = Cruise Set Speed

91 = Accelerator Pedal Position

92 = Percent Load

96 = Fuel Level

100 = Oil Pressure

102 = Intake Pressure

105 = Intake Temperature

108 = Barometric Pressure

110 = Coolant Temperature

111 = Coolant Level

116 = Brake Application Pressure

117 = Brake Primary Pressure

118 = Brake Secondary Pressure

128 = Component Request

168 = Charging Voltage

175 = Oil Temperature

177 = Transmission Temperature

183 = Fuel Rate

184 = Instant Fuel Economy

185 = Average Fuel Economy

190 = RPM

192 = Multi-Section Message

194 = Diagnostics (On Request)

195 = Clear Fault

196 = Clear Fault Response

234 = Software Id (On Request)

235 = Idle Hours (On Request)

236 = Idle Fuel Used (On Request)

237 = VIN (On Request)

243 = Component Id (On Request)

245 = Distance

247 = Total Hours (On Request)

250 = Fuel Used (On Request)

1. Version Changes
   1. Version 1.0
      1. Initial Publication
   2. Version 1.1
      1. Renamed the IsVersionIncompatible property to IsCompatible.
   3. Version 1.2
      1. Compatible with API version 1.3.
      2. Added GetEngineVIN method.
      3. Added RetrievalMethod parameter to Truck Data methods.
   4. Version 1.3
      1. Compatible with API version 1.4.
      2. Requires Adapter Firmware 3.10.7.
      3. Added enum RetrievalMethods.
      4. Added RetrievalMethods to Truck Data methods.
      5. Added property SyncTimeout.
      6. Added property AdapterIdFilter.
      7. Added property MaxReconnectAttempts.
      8. Added property ConnectionState.
      9. Added property ConnectTimeout.
      10. Added Synchronized option to the Connect method.
      11. Added ConnectionState CANFilterFull.
      12. Renamed property DiscoveryTimeOut to DiscoveryTimeout.
      13. Renamed property MaxConnectRetrys to MaxReconnectAttempts.
   5. Version 2.0
      1. Added ELD Recording.
      2. Fixed retrieving data bugs.
      3. Added Adapter LEDs section to the document.
      4. Rearranged the document sections.
   6. Version 2.1
      1. IsStreaming no longer starts or stops streaming.
      2. Added SignalStrength property.
   7. Version 2.2
      1. Changed property VIN to EngineVIN.
      2. Added property CabBodyVIN.
      3. Added properties CabBody.Make, Model, SerialNo, and UnitNo.
      4. Renamed method GetEngineVIN to GetVehicleId and removed all parameters.
      5. Added a GetVehicleIdSync method.
      6. Added property AdvertisementTimeout.
      7. Updated Demo app with above changes.
      8. For BLE adapters, if the ConnectToLastAdapter and UpdateSecurity (SecureAdapter) are not set, the API will connect to the adapter with the strongest signal.
      9. Compatible with Adapter Firmware 3.11.
      10. Updated with the latest BlueFire 4.6 core libraries.
   8. Version 2.3
      1. Added HiRes and LoRes Distance and Odometer to Truck Data.
      2. Added GetDistance method which is the same as GetOdometer.
      3. GetDistance (and GetOdometer) will return the Engine Distance and the OEM Odometer (if available).
      4. Updated with the latest BlueFire 4.8 core libraries.