Pac-Lite Service

Pac-Lite Service Structure

Subscribes to SNP.Inbound topic and receives packets from there. Routes packets through service out to active MQ, out UDP ports to MDE or MDI, through remote procedure calls for gRPC, through a tcp ip connection for Camstar or through a sql querry to the Engineering Database.

Using ActiveMQ is the best way to send Control signals ( create edit delete machine).just copy the packet from here ( replace the weird quotes with normal quotes) then change the machine to fit your needs.

ENG\_DB

ActiveMQ

MDE

Camstar

Producing App

Producing App

Producing App

Producing App

gRPC

Pac-Lite Service

Stomp, JMS, MQTT, Etc.

UDP

SQL Queries

gRPC Calls

TCP/IP

MQTT

ActiveMQ

Producing App

Producing App

Producing App

Producing App

Pac-Lite Packets

Packet Structure

* 1 way communication with all information being contained within one packet.
* Separate Packet header Per application ( Pac-lite uses header 1 (soh signal not ascii 1, Scada Box Environment monitoring uses header 2)
* Next is a packet type used to differentiate how it is forwarded. Each documented packet will have one attached (this is stored in a byte as a character)
* Next is the device id. This differentiates different Pac-lite machines or other devices to the service. (one way to ensure that yours is unique is to use your unique mqtt identifier as your device id. Pac-Lite machines do this.
* Following that is 5 reserved bytes, if you are no using these bytes leave them as the number 32 (ascii space) doing this as opposed to leaving it null allows the packet to be treated as a string!
* Next is up to 246 bytes of json encoded data. This data will be explained more below but it is important that after these 246 bytes it is terminated with a null as this allows the entire packet to be treated as a string in a plc and higher-level language.
* 260-byte max packet size with a max payload of 246 bytes (~5% packet overhead)

1 Byte

246 Byte

1 Byte

1 Byte

5 Byte

1 Byte

1 Byte

Packet Length

Packet Header

Packet Type

SNP\_ID

Reserved

Json Payload

NULL

Packets Types used by Pac-Lite

* + Id 1). Index Summary Packet
    - Required fields {Machine, Good, Bad, Empty, Indexes, UOM, and NAED};
    - Optional Fields {};
    - Triggers a throughput packet to Camstar and a packet to the ENG Database for recording the information
    - Sent every 15 minutes by Pac-lite. Summarizes what has happened in those 15 minutes in terms of good bad empty and number of indexes run.
    - Sample
  + {"Machine": "TestMachine",  "Good":"42”, "Bad":"2", "Empty":"4", "Indexes":"48", "UOM":"EA", "NAED":"31474"}
  + Id 2). Downtime Packet
    - Required fields { Machine,Status,Time,MReason,UReason, NAED};
    - Optional Fields { };
    - Triggers a downtime packet to Camstar and a packet to the Eng. Database for recording the information sent out when no index is received for 60 seconds or a user/machine error stops it.
    - Sample
  + {"Machine": "TestMachine", "Status":"1”, "Time":"19-11-11-08:31:26”, "MReason":"Preventing Wrenching", "UReason":"oiling up wrench launcher", "NAED":"Wrench.NAED
  + Id 3). Short Time Statistic Packet
    - Required fields {Machine, Good, Bad, Empty, Indexes, UOM, Other, and NAED};
    - Optional Fields {Any Errors you are reporting};
    - Triggers a UDP packet to MDE or MDI as well as a packet to the Eng. Database for recording the information. Sent out each index by Pac-lite
    - Sample
  + {"Machine": "TestMachine", "Good":"0", "Bad":"1","Empty":"0","Attempt":"0","Error1":"0","Error2":"0","Error3":"0","Error4":"0","Other":"1","HeadNumber":"1","Theo":"4800"}
  + Id 252). Delete Machine Packet
    - Required fields {Machine};
    - Optional Fields {Theo, Line};
    - Triggers the removal of the databases and Machine Info Entry for whichever machine is entered.
    - Sample
  + ü {"Machine": "TestMachine"}
  + Id 253). Edit Machine Packet
    - Required fields { Machine,Theo,Line };
    - Optional Fields {};
    - Edits the Machine Info entry for the machine to match the rest of the statement and the Pac-lite id it comes from.
    - Cannot rename machine must delete then remake.
    - Sample
  + ý {"Machine": "TestMachine","Theo":"4800", "Line”: “Straight Base"}
  + Id 254). Create Machine Packet
    - Required fields { Machine,Theo,Line};
    - Optional Fields {};
    - Creates SQL databases for the machine entered with the default errors and naming scheme. Also adds a machine info entry with the above information and the Pac-lite id it comes from
    - Sample
  + þ {"Machine": "TestMachine","Theo":"4800", "Line”: “Straight Base", "Errors”: “Error1,Error2,Error3,Error4,Other"}

Packets Types used by Environmental Monitoring Program

* + Id 1). Index Summary Packet
    - Required fields {Temperature, ChangeOver5Seconds, Location};
    - Optional Fields {TimeStamp, Humidity, FlowRate };
    - Triggers a recording packet to sql to record the data shown above^ if the Time stamp is missing it will be taken server side and inserted. If humidity or flow rate are missing they are assumed 0.
    - Send every 10 seconds by the BaseTec Machines
    - Sample
  + {" Temperature ": 87.5, "Location":"StraightBaseCabinet" , "ChangeOver5Seconds":"2.12", "Humidity":"76.6",  "FlowRate":"2112.2", "TimeStamp":”19-11-11-08:31:26”}
  + Id 1). Index Summary Packet
    - Required fields {Temperature, ChangeOver5Seconds, Location};
    - Triggers a recording packet to sql to record the data shown above^ if the Time stamp is missing it will be taken server side and inserted.
    - Send every 10 seconds by the BaseTec Machines
    - Sample
  + {"Warning": “Help Im on fire”, "Location":"StraightBaseCabinet" , "Urgency":8, "TimeStamp":”19-11-11-08:31:26”}

Packets Types used by The service Itself

* + Id 1). LoggingLevel
    - Required fields {LoggingLevel};
    - Optional Fields {IntTimeInSeconds};
    - Sets the logging level of the service. If intTimeInSeconds is assigned and not 0 it will revert to the previous loggingLevel after that time has passed.
    - Sample
  + {"LoggingLevel": 5, "IntTimeInSeconds":"320"}
  + Id 2). Silence
    - Required fields {Sendbool};
    - Optional Fields { IntTimeInSeconds };
    - Sets weather the service will output to any outputs. If 0 it will not send outputs if 1 it will. If IntTimeInSeconds is assigned and not 0 it will revert to the vrevious logging level after that time has passed.
    - Sample
  + {"SendBool ": 1, "IntTimeInSeconds":"320"}
  + Id 3). Listen
    - Required fields {ListenBool};
    - Optional Fields {IntTimeInSeconds };
    - Sets weather the service will listen to any inputs. If 0 it will not listen to inputsif 1 it will. If IntTimeInSeconds is assigned and not 0 it will revert to the previous logging level after that time has passed. All control packets will still be heard even when deafened for hopefully obvious reasons
    - Sample
  + {"ListenBool ": 1, "IntTimeInSeconds":"320"}

Field Information

* Machine
  + Key Machine
  + Value string, should be the name of the machine it is interfacing with. For Pac-lite this is used to name tables and communicate with a Camstar resource named the same.
* Line
  + Key Line
  + Value string, not used by Pac-lite however recorded as part of the machine information and may be used by other applications as it is recorded to an sql database.
* Good
  + Key Good
  + Value int, a count of the number of good products produced.
* Bad
  + Key Bad
  + Value int, a count of the number of Bad products produced.
* Empty
  + Key Empty
  + Value int, a count of the number of Indexes that the head was empty during.
* Indexes
  + Key Indexes
  + Value int, a count of how many times the machine has indexed
* UOM
  + Key UOM
  + Value string, Unit of Measure for the NAED produced.
* NAED
  + Key NAED
  + Value string, The NAED of the product being produced.
* Time
  + Key TIME
  + Value string, The time whatever occurred, in the format yy-mm-dd-hh:mm:ss
* Status
  + Key Status
  + Value int, current status of the machine. 0 being unscheduled downtime 1 being PM and 2 being running
* MReason
  + Key MReason
  + Value int, Reason the machine went down. Either in the form of a description or error code.
* UReason
  + Key Good
  + Value int, UReason User description as to why the machine when down.
* Any Errors
  + Key Error/Column name
  + Value int, 0 for false 1 for true reports whatever the error is under sql for the error name in the short time statistics page and to mde as the mde error bit it is set to use.
* Theo
  + Key Theo
  + Value int, Theoretical index count of a machine.
* HeadNumber
  + Key Theo
  + Value int, Reports what head produced the last part for short time statistics.
* IntTimeInSeconds
  + Key IntTimeInSeconds
  + Value int, The number of times in seconds a setting is to last. If 0 it is ignored. It is also an optional field any time it is missing or 0 it is assumed permanent and will only be changed by another packet being sent or the app.config being changed.
* Sendbool
  + Key Sendbool
  + Value int, 1 allows packets to be processed and outputs dispatched. 0 The service will deque packets but will not process or dispatch any messages.
* LoggingLevel
  + Key LoggingLevel
  + Value int, 1: will only report errors 2: will only report errors and connection attempts 3: medium level reporting and diagnostic 4:Everything Currently 5:anything in the future that gets added that is deemed mostly useless but possibly needed.
* ListenBool
  + Key ListenBool
  + Value int, 1 will set the service to listen to the configuared input while 0 will disable input for the designated time. Does not affect control packets