Pac-Lite Service

Pac-LiteService 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.

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)
* Next is a packet type used to differentiate how it is forwarded. Each documented packet will have one attatched ( 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 apposed 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

PacketType

SNP\_ID

Reserved

Json Payload

NULL

Packets Types used by Pac-Lite

* + Id 1). Index Summary Packet
    - Required fields { Machine, Line, Good, Bad ,Empty, Indexes, UOM, and NAED};
    - Optional Fields { };
    - Triggers a thruput 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.
  + 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.
  + Id 3). Short Time Statistic Packet
    - Required fields { Machine, Good, Bad ,Empty, Indexes, UOM, 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
  + 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.
  + 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.
  + Id 254). Create Machine Packet
    - Required fields { Machine,Theo,Line};
    - Optional Fields { };
    - Creates SQL databases for the machine enteredwith the default errors and naming scheme. Also addes a machine info entry with the above information and the pac-lite id it comes from

Field Information

* Machine
  + Key Machine
  + Value string, should be the name of the machine it is interfaceing 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 Mesure 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.