



MAV IQ Intelligent ANPR Camera

STORE AND FORWARD MODULE

Store and Forward Module

Contents

About this Guide	3
Icons Used in this Manual.....	3
Copyright Notice	3
Trademarks	3
Technical Support.....	3
Introduction	4
Storage Management Module	5
Storage Processes.....	6
Channel Licensing	6
Storage Capacity	6
Storage Policy Settings.....	6
Maximum Records Setting	6
Historic Retrieval	6
Forwarding Processes	7
SAFID	8
Injection of SAFID into Decode Messages.....	8
General Forwarding Policy Settings.....	9
API Interface.....	10
SAFID	10
Purge	10
Retrieve.....	11
Web User Interface	12
Monitor Page	12
Configuration Settings	13
System Settings	13
Storage Settings.....	13
Generic Channel Settings	14
Configuration Parameters.....	15
System Parameters.....	15
Channel Settings.....	16
Generic Channel Settings.....	16
Appendix A – HTTP POST JSON	18
Web User Interface	18
Forwarding Process	18
Configuration Parameters	19
Appendix B – HTTP SOAP UTMC V1.2.....	21
Web User Interface	21
Configuration Parameters	21
Appendix C - FTP	23
FTP Configuration Interface	24
Forwarding Process	24

About this Guide

ICONS USED IN THIS MANUAL

These special messages refer to noteworthy information, and include a symbol for quick identification:



Alert: Important information that cautions about features affecting performance, security features, or causing potential problems with your Intelligent ANPR Software.



Tip: Useful information about the configuration of your Intelligent ANPR Firmware.



Note: Important information on a feature that requires callout for special attention.



Cross Reference: Provides a pointer to related information in this or other documentation.

COPYRIGHT NOTICE

This document is Copyright © MAV Systems Ltd, 2015. All rights reserved. The content of this document is licensed from Rudstone Technologies Ltd. Reproduction of any part of this guide in any form whatsoever requires express written permission from MAV. The contents of the guide are subject to change, and MAV can accept no responsibility for any errors, or their consequences.

TRADEMARKS

MAV Systems Ltd have a number of products and components as registered trademarks and the use of any of these registered trademarks for any purpose must be confirmed with MAV prior to release.

TECHNICAL SUPPORT

For technical support, please contact MAV using anpr@anprcameras.com for email requests or contact +44 01344 859753 for assignment of a point of contact.

Introduction

The standard ANPR push system for passing decode messages via JSON has a small volatile memory based buffer of decode history (decode buffers). These buffers are suitable for streaming decode messages to a “well behaved” host that does not introduce delays. (See “Tech Note” below). Any delays can, in some cases, cause the circular decode buffers to be overwritten.

Protocols which require acknowledgement, by their nature, can cause delays in transmission and the potential for decode messages to be lost. For this reason, the push mechanism is not acknowledged and provides only limited ability to recover from data loss through network issues. (See “Tech Note” below).

The Store and Forward module provides a controlled method of sending decodes to a host using acknowledged protocols and storing undelivered decodes for later retrieval. The Store and Forward module uses non-volatile storage of decode information for long term availability and copes with network/link loss and facilitates further features:-

- The ability to store decode history in non-volatile storage.
- The ability to implement protocols that require acknowledgements and/or to withstand delays in processing.
- The ability to attempt retransmission of messages that are unable to be sent or that are not acknowledged.
- The ability to retrieve historic data from storage.

The Store and Forward module consists of a number of components:-

- **Storage Management Component** that orchestrates the movement to decode messages into and out of the non-volatile storage.
- **Bearer Component** that implements a specific protocol to send the message to a host.
- **Web Component** that provides an API interface used by developers and/or the web page user interface.



The design of the TCP/IP communication system guarantees the delivery of messages.

It does this by acknowledging transmissions at a low level between sender and receiver.

The data is buffered on the sender in case it needs to be resent and the data is buffered on the receiver for delivery to the host application when the host application is ready to receive it.

If the communications is disrupted or the host application is not keeping up with the flow of data, the buffers can start to fill up.

At some point the sender's buffer (the camera) becomes full and the sending is paused until buffer space becomes available.

If the Push Service is paused for too long, then decodes will be lost as the oldest decode messages are overwritten by the newest.

The Store and Forward module works alongside the standard Push Service, but some bandwidth management will be introduced in the future to priorities applications using the network interface.

Measured throughput rates of 40 decodes per second have been seen during development and testing, however some pacing will be added so the Store and Forward system will not monopolise the available bandwidth.

The Store and Forward module also shares the use of the non-volatile storage system of the camera. The bandwidth restriction of the non-volatile storage means its use needs to be managed in order to reduce contention.

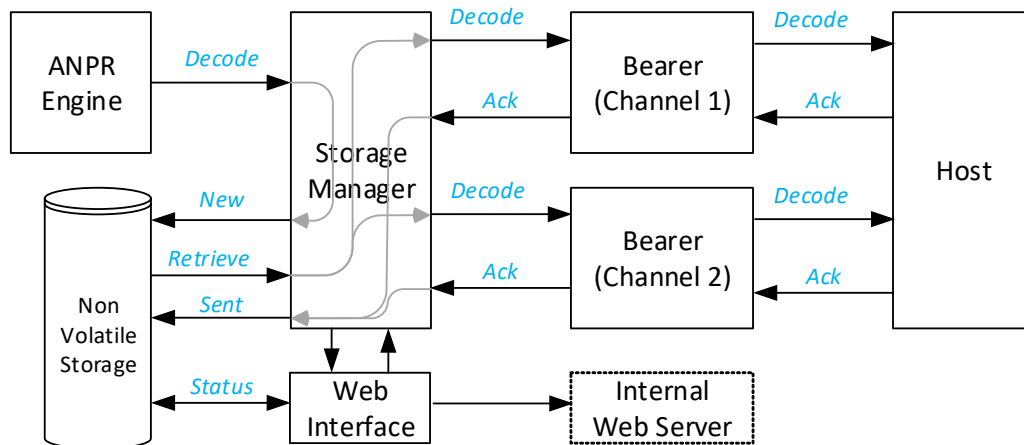
Storage Management Module

The core component of the Store and Forward system is the Storage Management module.

The Storage Management module facilitates the following functions:-

- Receives decodes from the ANPR Engine.
- Commits decodes to the non-volatile storage.
- Manages a "Live" queue of the 16 most recent decodes.
- Manages the non-volatile storage in line with the policy settings.
- Facilitates a selection mechanism so that "live" decodes are sent in preference to older decodes.
- Feeds decodes to the Bearer Component.
- Supports the simultaneous use of more than one Bearer module.
- Provides a Web Interface that allows an authorised user to manage the Store and Forward system.

The diagram below shows a basic dataflow between the components to the Store and Forward module.



Storage Process

The storage of decode messages can be enabled and disabled. By disabling the Storage, it stops the camera accumulating data. This might be desirable during any testing, commissioning or planned outages.

The enable/disable of the storage process is controlled by a configuration parameter accessible through the API or the Web User Interface.

CHANNEL LICENSING

The Store and Forward system has a single channel enabled with a second channel as an additional licensed option.

When decodes are stored, they are marked as “Unsent” for the first channel, but “not required” for the second if not licenced.

If the second channel is subsequently enabled, then new decodes will be marked as “unsent” for that channel. The message marked as “not required” are never sent on the second channel.

STORAGE CAPACITY

The Store and Forward system allocates 75% of the non-volatile storage capacity for its use. This would be normally no less than 4Gb.

The amount of data contained in a single decode will vary depending on the plate patch and overview image settings. It is therefore difficult to specify a capacity in “number of decodes”.

By way of example, a large decode with large, high quality image settings would be in the order of 70K whereas more typical settings would be in the order of 10K and could be as little as 3K.

STORAGE POLICY SETTINGS

The storage system has two policy settings:-

- The maximum number of records that can be stored in the system.
- If historic retrieval should be allowed.

MAXIMUM RECORDS SETTING

The total number of records stored in the system can be limited to a maximum quantity.

However, because of the structure of the storage system, the decode messages are batched into storage blocks. The storage system is designed to have a current “open” storage block so that new decode messages can be added to it.

The consequence is that on occasions the total record will exceed the configured maximum. The total record will drop down to the configured maximum as the older storage blocks are removed.

HISTORICAL RETRIEVAL

Once a decode message has been acknowledged by the destination, the message is marked as sent.

These sent messages can be left in storage until they are deleted.

Using the Retrieve API, it is possible to retrieve records still in storage.

System performance is improved by not having to scan or manage storage blocks in which all data has been sent.

The option "Allow Historic Retrieval of Records" when set to "disable" deletes storage blocks in which all the data has been sent.

When storage blocks are deleted, it is not possible to retrieve historic data.

Forwarding Processes

The communications of the stored decode messages to the host is the responsibility of the channel manager.

The Store and Forward module supports up to two independent channels, each having its own bearer. (The second channel is an additional licensed option.)

This would allow the ANPR System to send decode messages independently to two different locations, possibly using two different protocols.

The “Forward” processing, as a concept, has two queues:-

- The Live Queue which contains the last 16 decode messages generated¹.
- The Storage Queue which contains all decode messages held in non-volatile storage.

The channel manager is provided with most recent decodes from *Live Queue* in preference to those from the *Storage Queue*. Decode messages from the *Storage Queue* are delivered oldest first.

The consequence is that the latest decodes are sent as a priority.

Two bearer modules are available at this revision of software:-

- JSON decode messages sent using HTTP/POST
- Basic UTMC version 1.2 message over HTTP/SOAP

A channel can be configured with only one bearer, however the channel manager will start a number of Bearer clients in parallel to increase throughput. The number of bearers is implementation specific.

SAFID

The SAFID (“SAF ID”) is a unique reference for records that have been stored in the Store and Forward System.

The SAFID will wrap, but this is a function of the record limit imposed on configuration.

Part of the SAFID is incremented when a new storage unit is created. In the current configuration, the SAFID will wrap after 16383 storage units have been used.

If a “maximum number of records” of 1 record has been defined (see “**Error! Reference source not found.**” on page **Error! Bookmark not defined.**), then it will wrap after 16383 records. This value should be considered in the context that the Storage System has been configured to hold just 1 record, and he therefore been overwritten 16384 times before thr SAFID wraps.

If the “maximum number of records” is very large (over 1,000,000), this would wrap after approximately 23 million records in typical usage.

Whilst the current SAFID is an 8-character hex encoding of a 32 bit integer, developers should not rely on any meaning derived from these values.

The SAFID is sent as a string and may be up to 16 characters in the future and may not be hex encoded.

¹ This is an architectural decision to reduce the load on the non-volatile storage system. Developers should not rely on this feature in the future.

INJECTION OF SAFID INTO DECODE MESSAGES

For systems which receive data from the Store and Forward System as a JSON Decode message, the SAFID has been added to the decode message as a field called SAFID.

This allows the Store and Forward system to deliver smaller messages and for the downstream system to retrieve more data later using the SAFID and the API Retrieve function.

GENERAL FORWARDING POLICY SETTINGS

The channel manager has a number of settings that are common to all bearers:-

- The oldest record that can be sent over this channel.
- If repeated decode messages should be sent over this channel.
- The type of bearer that is to be used on this channel.

There are more details on these parameters in the "Web User Guide" section.

API Interface

The Store and Forward module contains a RESTfull web interface that can be used to interact with the module.

The interface allows the application:-

- To purge data from the non-volatile store.
- To retrieve data from non-volatile storage.



The non-volatile storage has limited bandwidth which needs to be managed and protected so as to not compromise parts of the system.
As such, functionality that scans the non-volatile storage has not been implemented.

SAFID

The Store and Forward system references records by a field called the “Store and Forward ID” – shortened to “SAFID”.

Developers will recognise that the SAFID is the hex encoding of an unsigned integer.

Developers should treat the SAFID as a character string, as future versions may use different encodings.

Applications should be prepared to store up to 16 characters which may span the whole printable ASCII character range.

PURGE

The purge command completely erases all stored data.

The command has the following syntax:-

[HTTP://.../SAF?func=Purge](http://.../SAF?func=Purge)



By design, the storage has no dependence on time because the date and time on the camera can be altered and this would negatively impact design of the storage system.
A consequence is that purge cannot be performed by date or age criteria.

MAV IQ Intelligent ANPR Camera Store and Forward Module

RETRIEVE

The Retrieve command is used to retrieve decode data from the non-volatile store.

Retrieval of data is paced at 2 requests per second to protect network bandwidth and to plan access to the non-volatile storage.

The command has the following syntax:-

[HTTP://.../SAF?func=Retrieve&SAFID=«safid»](http://.../SAF?func=Retrieve&SAFID=«safid»)

Optionally a template specification can be added.



See “Templates” in the manual “Intelligent ANPR Camera Module: Application Program Interface”.

[HTTP://.../SAF?func=Retrieve&SAFID=«safid»&Template=«templateno»](http://.../SAF?func=Retrieve&SAFID=«safid»&Template=«templateno»)

The response will be one of the following JSON Objects:-

```
{“decodes”:[«decode data»]}
```

Returns a single decode object in the decode array.



See the manual “Intelligent ANPR Camera Module: Application Program Interface” for more information on decode messages.

```
{“error”:{“code”:100,“text”:“Required parameter missing”}}
```

If the SAFID is not present in the query.

```
{“error”:{“code”:101,“text”:“Bad parameter value”}}
```

The SAFID does not parse as a hex number
If a template is specified, the value is not between 0 and 4

```
{“error”:{“code”:102,“text”:“SaF Module not enabled”}}
```

SAF module is not licensed on this camera.

MAV IQ Intelligent ANPR Camera Store and Forward Module

Web User Interface

When Store and Forward is added to the camera, a Store and Forward Monitor page and a Configuration page are added to the Web User Interface.

MONITOR PAGE

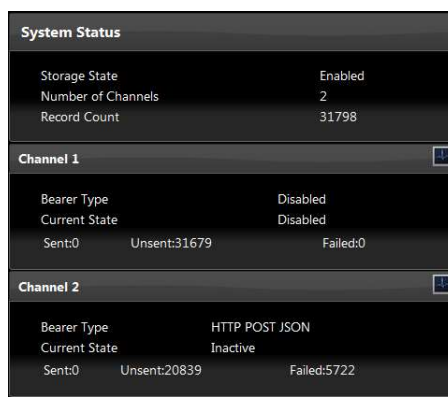
A sample monitor page is shown below with a second licensed channel active. Normally, "Channel 2" would not be seen.

SYSTEM STATUS

Storage State - indicates if the storage of decode messages is enabled.

Number of Channels - shows the number of channels the product is licensed for.

Record Count - the total record count in the storage system.



The screenshot shows a 'System Status' section with the following data:

Storage State	Enabled
Number of Channels	2
Record Count	31798

Below this are two channel sections, each with a diagnostics icon:

Channel 1

Bearer Type	Disabled	
Current State	Disabled	
Sent:0	Unsent:31679	Failed:0

Channel 2

Bearer Type	HTTP POST JSON	
Current State	Inactive	
Sent:0	Unsent:20839	Failed:5722

CHANNEL STATUS

Bearer Type - The configured bearer for the channel.

Current State - The current state of the bearer.

- Disabled – if the bearer is disabled.
- Active – if the bearer is ready to, or is sending data.
- Inactive – the connection has failed and the bearer is trying to connect again.

Sent - A count of the number of records sent since last powered up.

Unsent - A count of the number of records to be sent by this channel.

Failed - A count of the number of records that have failed to be sent. (Some bearers do not have the concept of "fail" so the records will remain "unsent".)

A diagnostics dialog for each channel can be displayed by clicking the corresponding  icon in the channel header bar.

Configuration Settings

The configure settings are divided between System settings, Storage Setting and settings for 1 or more channels.

SYSTEM SETTINGS

The **System** settings currently contains just one button to purge all data from storage.

More controls will be added to this panel in the future.

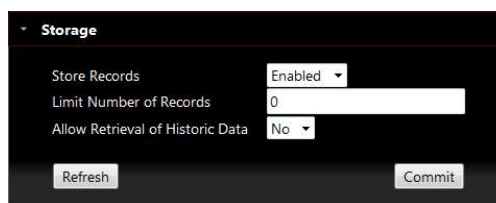


STORAGE SETTINGS

The storage of records can be enabled or disabled. New decode messages will only be stored when enabled.

The number of records that can be stored in the system can be limited.

By setting a value of zero, the limit is the maximum the system can hold.



See "Maximum Records Setting" on page 7 for more details.

If "Allow Retrieval of Historic Data" is set to "No", then storage blocks are deleted when all decode messages in that block have been sent.



See "Historic Retrieval" on page 7 for more details.

MAV IQ Intelligent ANPR Camera Store and Forward Module

GENERIC CHANNEL SETTINGS

Message selection for the channel can be limited to a maximum age. This feature would be used if data became irrelevant after a cut-off point.

A value of zero means there is no age -checking performed.



The screenshot shows a configuration window titled "Channel 1". It contains three settings:

- "Oldest Record Age to Send": A text input field containing "0" and a dropdown menu set to "Seconds".
- "Send repeated plates": A dropdown menu set to "No".
- "Bearer": A dropdown menu set to "None".

At the bottom of the window are two buttons: "Refresh" and "Commit".

In situations where the back-office system or the network was not operational for an extended period of time, then the Store can accumulate data which would be irrelevant because it was too old for the downstream system.

If a VRM has been in view for an extended period of time, it may be reported as a repeated decode. This repeated information may not be relevant for some systems, so this setting allows repeated decodes to be sent or not sent.

By selecting different bearer types, the available settings change.

The settings for the specific bearers are in the Appendix relevant to the bearer.

Configuration Parameters

SYSTEM PARAMETERS

ID	Access Level	Short Name	Data Type
090004	2	SaFStoreEnable	Boolean

Enables the storage of decode messages

Value	Function
True	The System will store decode messages
False	The System will not store decode messages.

ID	Access Level	Short Name	Data Type
090000	2	SaFRecordLimit	Integer: 0 - 4294967295

Sets the limit for the number of records in storage

Value	Function
0	The number of records is limited by the available storage
Other	Sets the number point at which old storage blocks will be purged. By design, the number of records will exceed this limit but will return below this number as old storage blocks are deleted.

ID	Access Level	Short Name	Data Type
090003	2	SaFDeleteFiles	Boolean

Enables the deletion of storage block in which all decode messages have been sent. This will inhibit the ability to retrieve historic data if enabled.

Value	Function
True	Storage blocks that contain only sent messages are deleted.
False	Storage blocks will only be deleted when the system recycles storage.

Store and Forward Module

CHANNEL SETTINGS

Each channel has settings stored at a base location:-

- 092000 for Channel 1
- 094000 for Channel 2

The ID values in the following tables are offset from these values. All ID values are hex (base 16).

GENERIC CHANNEL SETTINGS

ID	Access Level	Short Name	Data Type
+0	2	BearerType	Integer: 0 - 4294967295

Selected the bearer for a given channel

Value	Function
0	Disabled
1	HTTP POST JSON bearer
2	HTTP SOAP UTM V1.2 bearer
Other	Disabled

ID	Access Level	Short Name	Data Type
+4	2	BearerOldest	Integer: 0 - 4294967295

A factor to multiply *BearerOldest* by to get seconds

Value	Function
0	No age checking is performed.
other	The value is multiplied by <i>BearerAgeUnits</i> (+5) to give a value in seconds. Decodes which are greater than this age are marked as "expired" and are not sent.

Store and Forward Module

ID	Access Level	Short Name	Data Type
+5	2	BearerAgeUnits	Integer: as per list

A factor to multiply *BearerOldest* by to get seconds

Value	Function
1	Seconds
60	Minutes
3600	Hours
86400	Days
Others	Setting other values will make the Web User Interface display a blank value for this parameter.

ID	Access Level	Short Name	Data Type
+6	2	BearerSendRepeats	Boolean

A factor to multiply *BearerOldest* by to get seconds

Value	Function
True	Decodes marked as repeat decodes are sent
False	Decodes marked as repeat decodes are not sent

Appendix A – HTTP POST JSON

The HTTP POST JSON Bearer sends the JSON decode message (see API Manual) using the HTTP POST method.

The bearer contains 4 HTTP Clients that send data simultaneously to the destination.

The HTTP Clients use the HTTP/1.1 protocol.

WEB USER INTERFACE

URL - The URL parameter is used to specify the destination end point for the HTTP Connection. Where no path is required a trailing “/” is expected.

Template ID - A template (see API Document) can be specified for creating the JSON for non-repeated decodes and repeated decodes using different templates.

The screenshot shows a configuration form with the following fields:

- URL: http://127.0.0.1:8080/jsonapi
- Template ID: 0 (dropdown), For repeats: 0 (dropdown)
- Connection Timeout: 120
- Make a Persistent Connection: Disabled (dropdown)
- Use 100 Continue: Disabled (dropdown)
- User ID: (empty text field)
- Password: (empty text field)

Connection Timeout - The number of seconds the HTTP Client waits for a connection before going into *Reconnect Mode* (see *Reconnect Mode description later*).

Make Persistent Connection - If enabled, the HTTP Client will not drop the connection between transactions. By not having to establish a connection to the server for each transaction, the throughput is (in some cases) increased. However, even with this option enabled, the server may unilaterally drop the connection.

Use 100 Continue - some servers will require this to be enabled. The HTTP Client sends the header to the server, giving the server the understanding of what will be sent. The server can send “100 Continue” if it agrees to receive the data or some other error if it does not.

User ID & Password - the username and password if required by the server.

FORWARDING PROCESS

The HTTP POST JSON bearer will forward decode messages as they are given to the bearer.

If “200 OK” is returned, then the bearer will ask that the record is marked as “Sent” in the storage system.

The Bearer will respond to redirect and authentication requests.

Any other status code will cause the bearer to go into *Recovery Mode*.

In recovery mode, the other HTTP Clients are allowed to finish their transactions and are then disabled. The remaining HTTP Client continues to try and send the next decode until it receives “200 OK” at which point the other HTTP Clients are released.

This mechanism stops the camera flooding the server if there is a problem.

Currently, there is no status code that would cause a record to be marked as failed.

MAV IQ Intelligent ANPR Camera Store and Forward Module

CONFIGURATION PARAMETERS

Each channel has its own base id for its configuration parameters.

- Channel 1 – 092000
- Channel 2 – 094000

The configuration parameters described below document their offset with each channel. The base address and offset have to be added together to get the actual parameter id for the specific parameter on a specific channel.

Note, all ID and OFFSET values are in base 16.

ID	Access Level	Short Name	Data Type
+0	2	BearerURL	string

Is the URL text used to connect to the remote service.
Only HTTP is supported.
Host can be addressed by DNS name or IP address.
Port numbers are allowed.
A path must be specified, even if it is just “/”.

ID	Access Level	Short Name	Data Type
+1	2	BearerTemplateId	Integer:0-4

Selects the template number when sending non-repeated decodes.

ID	Access Level	Short Name	Data Type
+2	2	BearerTimeout	Integer:1 - 4294967295

When multiplied by *BearerAgeUnits*, the product is the number of seconds to wait for connection to the server.

ID	Access Level	Short Name	Data Type
+7	2	BearerTemplateIdRepeat	Integer:0 – 4

Selects the template number when sending repeated decodes.

Store and Forward Module

ID	Access Level	Short Name	Data Type
+201	2	BearerHTTPUse100	Boolean

“Expect 100 continue” is added to the HTTP Headers and processing of the response is enabled.

ID	Access Level	Short Name	Data Type
+202	2	BearerHTTPUserName	Boolean

The username provided to the server if requested.

ID	Access Level	Short Name	Data Type
+203	2	BearerHTTPPassword	Boolean

The password provided to the server if requested.

Appendix B- HTTP SOAP UTMC V1.2

The current bearer is a very basic version of the UTMC protocol delivered over SOAP.

The SOAP standard is well documented on www.w3.org

Currently, the bearer only delivers a *plate read array* containing a single plate read, and an image array containing a single overview image and a single plate patch image.

In this release, the UTMC bearer does not respond to commands sent by the server. UTMC functional support will increase in future versions.

The UTMC Bearer takes a response of “200 OK” as consumption of the data and marks it as “sent”.

The Bearer will respond to redirect and authentication requests.

Any other status code will cause the bearer to go into *Recovery Mode*.

In *Recovery Mode*, the other HTTP Clients are allowed to finish their transactions and are then disabled. The remaining HTTP Client continues to try and send the next decode until it receives “200 OK” at which point the other HTTP Clients are released.

This mechanism stops the camera flooding the server if there is a problem.

WEB USER INTERFACE

URL - The URL parameter is used to specify the destination end point for the HTTP Connection. Where no path is required a trailing “/” must be given as per RFC 2616.

Connection Timeout -The number of seconds the HTTP Client waits for a connection before going into *Recovery Mode*.

Make Persistent Connection - If enabled, the HTTP Client will not drop the connection between transactions. By not having to establish a connection to the server for each transaction, the throughput is (in some cases) increased. However, even with this option enabled, the server may unilaterally drop the connection.

Use 100 Continue - some servers will require this to be enabled. The HTTP Client sends the header to the server, giving the server the understanding of what will be sent. The server can send “100 Continue” if it agrees to receive the data or some other error if it does not.

User ID & Password - the username and password if required by the server.

URL	<input type="text" value="http://127.0.0.1:8080/jsonapi"/>
Connection Timeout	<input type="text" value="120"/>
Make a Persistent Connection	<input type="text" value="Disabled"/>
Use 100 Continue	<input type="text" value="Disabled"/>
User ID	<input type="text"/>
Password	<input type="text"/>

CONFIGURATION PARAMETERS

Each channel has its own base id for its configuration parameters.

- Channel 1 – 092000
- Channel 2 – 094000

The configuration parameters described below document their offset with each channel. The base address and offset have to be added together to get the actual parameter id, for the specific parameter on a specific channel.

Note, all ID and OFFSET values are in base 16.

MAV IQ Intelligent ANPR Camera Store and Forward Module

ID	Access Level	Short Name	Data Type
+0	2	BearerURL	String

Is the URL text used to connect to the remote service.
Only HTTP is supported.
Host can be addressed by DNS name or IP address.
Port numbers are allowed.
A path must be specified, even if it is just "/".

ID	Access Level	Short Name	Data Type
+2	2	BearerTimeout	Integer:1 - 4294967295

When multiplied by *BearerAgeUnits*, the product is the number of seconds to wait for connection to the server.

ID	Access Level	Short Name	Data Type
+2	2	BearerHTTPersistentCon	Boolean

Controls if connection are closed after each transaction.
Note, the server can still unilaterally close the connection at anytime.

ID	Access Level	Short Name	Data Type
+201	2	BearerHTTPUse100	String

"Expect 100 continue" is added to the HTTP Headers and processing of the response is enabled.

ID	Access Level	Short Name	Data Type
+202	2	BearerHTTPUserName	String

The username provided to the server if requested.

ID	Access Level	Short Name	Data Type
+203	2	BearerHTTPPassword	String

The password provided to the server if requested.

Appendix C- FTP

The FTP Bearer sends the JSON decode message in a FTP format.

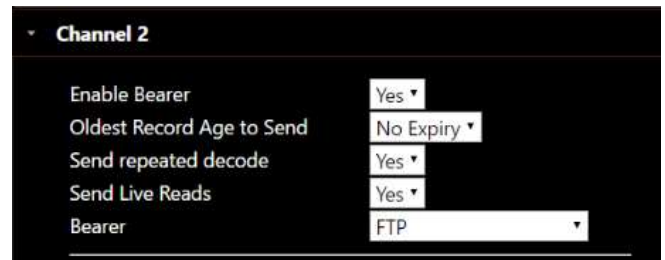
Each FTP read can consist 3 files.

A Text data file,

A Jpg of the Patch plate

A jpg of the Overview Image

The file names of these files are created from the Json data fields.



Example Text data file

[ref]_[cameraName]_[VRM]_[localtime.date.jp]_[localtime.time.jp]_[motion].txt

Example Patch jpg file

[ref]_[cameraName]_[VRM]_[localtime.date.jp]_[localtime.time.jp]_[motion]_patch.jpeg

Example Overview jpg file

[ref]_[cameraName]_[VRM]_[localtime.date.jp]_[localtime.time.jp]_[motion]_OV.jpeg

We recommend that FTP naming convention allows simple searches by using date/time/MS (jp format) and VRM - can also include direction for motion filtering (e.g. if static then these images could be reviewed to set in/out manually)

Important if any field names are left blank no image/file will be sent

Example Content of the text data file, this can include a full templated JSON or specific fields combined with free text

[countryCode]

[vrm]

[spacedVRM]

[FTPPath]

[localtime.date.jp]

[localtime.time.jp].[localtime.ms]

MAV IQ Intelligent ANPR Camera Store and Forward Module

FTP CONFIGURATION INTERFACE

FTP Host Name - The URL parameter is used to specify the destination end point for the FTP Connection. This could be a IP address or URL

FTP Port No - Default port number is 21, we do not support secure FTPs

FTP Username - FTP Username

FTP Password - Password

FTP Base Folder - Directory list is from root of destination FTP as applied by FTP server.

FORWARDING PROCESS

The FTP JSON bearer will forward decode messages as they are given to the bearer.

If no access to the FTP server then the bearer will store reads until the service is available.

The screenshot shows a configuration window titled 'Storage' with a sub-section 'Channel 1'. The interface includes several settings:

- Enable Bearer:** A dropdown menu set to 'No'.
- Oldest Record Age to Send:** A numeric input field set to '0' followed by a 'Seconds' label and a dropdown arrow.
- Send repeated decode:** A dropdown menu set to 'Yes'.
- Send Live Reads:** A dropdown menu set to 'Yes'.
- Send No Reads:** A dropdown menu set to 'Yes'.
- Bearer:** A dropdown menu set to 'FTP'.
- FTP Host Name:** An empty text input field.
- FTP Port No:** A text input field containing the number '0'.
- FTP Username:** An empty text input field.
- FTP Password:** An empty text input field.
- FTP Base folder:** An empty text input field.
- Data Filename Template:** An empty text input field.
- Patch Filename Template:** An empty text input field.
- Overview Filename Template:** An empty text input field.
- Data Content Template:** A large empty text area.

At the bottom of the window, there are two buttons: 'Refresh' and 'Commit'.