

M2Mapp.Ini Parameter Definitions

Table Of Contents

| | |
|---|----|
| AVIDIRECTOR.MODEL | 5 |
| AVID.SERIAL | 5 |
| SYSTEM.APP_NAME | 5 |
| SYSTEM.OPTIONS..... | 5 |
| SYSTEM.MAPSERIALPORT | 6 |
| SYSTEM.RDD_PWR_ON | 6 |
| SYSTEM.DEBUG | 6 |
| SYSTEM.NOLOGGING..... | 6 |
| SYSTEM.TIMESTAMP_LOG | 7 |
| SYSTEM.LOG_DEBUG..... | 7 |
| SYSTEM.REBOOT_ON_EXIT | 7 |
| SYSTEM.CLASSLOAD_ERRORS_FATAL..... | 7 |
| SYSTEM.LOG_FATAL_ERROR | 7 |
| SYSTEM.LOGFILE..... | 8 |
| SYSTEM.LOGFILENAME | 8 |
| SYSTEM.LOGFILE_MAXSIZE | 8 |
| SYSTEM.INTERACTIVE | 8 |
| SYSTEM.GC_RUN_INTERVAL | 8 |
| SYSTEM.GC_WATERMARK | 9 |
| SYSTEM.WATCHDOG..... | 9 |
| SYSTEM.WATCHDOG_TIME..... | 9 |
| SYSTEM.WATCHDOG_HWRESET..... | 9 |
| SYSTEM.DEVICE_UUID..... | 9 |
| M2MXML.DEVICE_UUID (Alias for SYSTEM.DEVICE_UUID) | 9 |
| SL.DEVICE_UID (Alias for SYSTEM.DEVICE_UUID)..... | 9 |
| SYSTEM.MAX_NOCOMM_TIME | 10 |
| M2MXML.MAX_NOCOMM_TIME (Alias for SYSTEM.MAX_NOCOMM_TIME) | 10 |
| M2MXML.VERSION | 10 |
| M2MLET.CLASSNAME | 10 |
| M2MLET.LOAD | 10 |
| M2MLET.START | 10 |
| NO_COMMUNICATOR | 11 |
| ACTIVE_COMMUNICATOR..... | 11 |
| MODEM_MODEL | 11 |
| SERIAL_PORT | 11 |
| MAPSERIALPORT | 12 |
| SERIAL_PORT_BAUDRATE | 12 |
| SERIAL_PORT_HANDSHAKE..... | 12 |
| SERIAL_PORT_TIMEOUT..... | 12 |
| AT_INIT | 13 |
| IP_ADDRESS..... | 13 |
| PHONE_NUMBER | 13 |
| NETWORK_NAME | 13 |
| BAND..... | 13 |
| SLOT | 14 |
| SERVER_IP_ADDRESS | 14 |
| SERVER_PORT..... | 14 |

M2Mapp.Ini Parameter Definitions

| | |
|---|----|
| SERVER_HTTP_M2MXML_SEND_PAGE | 14 |
| SERVER_HTTP_USE_HTTP_11 | 15 |
| SERVER_HTTP_SEND_USER_AGENT | 15 |
| SERVER_HTTP_M2MXML_POLL_PAGE | 15 |
| SERVER_M2MXML_POLL_TIME | 15 |
| SERVER_M2MXML_UDP_PORT | 15 |
| SERVER_SMS_NUMBER | 16 |
| SMS_STORE | 16 |
| NEW_SMS_MESSAGE_MODE | 16 |
| DONT_POWER_OFF | 16 |
| POWER_ON_WAIT_TIME | 17 |
| INVERT_RADIO_PWRON | 17 |
| RESET_RADIO | 17 |
| MAX_FATAL_ERRORS | 17 |
| MAX_RADIO_NOT_READY_ERRORS | 17 |
| GPRS_APN | 18 |
| CARRIER | 18 |
| SMSC | 18 |
| REG_MODE | 18 |
| REG_CARRIER | 19 |
| PPP_USERNAME | 19 |
| PPP_PASSWORD | 19 |
| PPP_GATEWAY | 19 |
| PPP_PASSIVE | 19 |
| PPP_DNS_PRIMARY | 20 |
| PPP_DNS_SECONDARY | 20 |
| DNS_TIMEOUT | 20 |
| DNS_RETRIES | 20 |
| DNS_LOOKUP_ERRORS | 20 |
| SOCKET_TIMEOUT | 21 |
| PPP_AUTHENTICATION | 21 |
| PPP_MRU | 21 |
| PPP_DIALNUMBER | 21 |
| PPP_DIALTIME | 21 |
| PPP_CONNECTTIME | 22 |
| PPP_HANGTIME | 22 |
| KEEP_CONNECTION_UP | 22 |
| CONNECTION_TYPE (IO270, GPRS radio) | 22 |
| CONNECTION_TYPE (for bluetooth radio) | 22 |
| DEVICE_TYPE (BLUETOOTH Only) | 23 |
| SERVICE_NAME (BLUETOOTH Only) | 23 |
| DYNDNS_HOSTNAME | 23 |
| DYNDNS_USERNAME | 23 |
| DYNDNS_PASSWORD | 23 |
| DYNDNS_DNS_TYPE | 24 |
| DYNDNS_AUTOUPDATE | 24 |
| SIM_PIN | 24 |
| SIM_PIN2 | 24 |

M2Mapp.Ini Parameter Definitions

| | |
|--|----|
| ENCRYPT_DESKEY (Aerocomm, AerocommZigbee) | 24 |
| USE_RDD_XMIT_CMD_MODE (Aerocomm, AerocommZigbee) | 25 |
| USE_RDD_AUX_CMD_MODE (Aerocomm, AerocommZigbee) | 25 |
| DESTINATION_ADDRESS | 25 |
| MAX_RETRY_COUNT | 25 |
| MAX_QUEUE_SIZE | 25 |
| DELETE_QUEUE_FULL | 25 |
| LIFO..... | 26 |
| IODEVICE.SERIAL_PORT | 26 |
| IODEVICE.SERIAL_PORT_BAUD_RATE | 26 |
| IODEVICE.SERIAL_PORT_TIMEOUT..... | 26 |
| IODEVICE.ENABLE | 26 |
| IODEVICE.MAPSERIALPORT | 27 |
| IODEVICE.RDD_PWR_ON..... | 27 |
| IODEVICE.M2MXML_ADDRESS..... | 27 |
| IODEVICE.NUM_AVG_READINGS | 27 |
| IODEVICE.TIMEZONE | 28 |
| Time Zone Table..... | 28 |
| IODEVICE.POLLTIME..... | 28 |
| IODEVICE.TIME | 29 |
| M2MIODEVICE | 29 |
| SETPOINT..... | 29 |
| DATA | 29 |
| VALUE..... | 29 |
| IOTYPE | 29 |
| IPOLLING | 30 |
| READING_PROCESSOR_CLASSNAME | 30 |
| ABSOLUTE_REPORT_TIME | 30 |
| PERIODIC_REPORT_TIME | 30 |
| DIGITAL_TRIGGER_HIGH | 30 |
| INVERT | 30 |
| PULLUP_INPUT..... | 31 |
| PULLDOWN_INPUT..... | 31 |
| DURATION..... | 31 |
| PULSEHIGH_TIME | 31 |
| PULSELOW_TIME | 31 |
| TIMER_HIGHINTERVAL_TIME | 31 |
| TIMER_LOWINTERVAL_TIME | 32 |
| TIMER_HIGHABSOLUTE_TIME | 32 |
| TIMER_LOWABSOLUTE_TIME..... | 32 |
| DIGITAL_TRANSISTION_ALARM_LEVEL | 32 |
| ANALOG_OFFSET..... | 32 |
| ANALOG_FACTOR..... | 33 |
| ANALOG_ABSOLUTE_HIGH..... | 33 |
| ANALOG_ABSOLUTE_LOW | 33 |
| ANALOG_DEADBAND_HIGH..... | 33 |
| ANALOG_DEADBAND_LOW..... | 33 |
| LOAD..... | 34 |

M2Mapp.Ini Parameter Definitions

| | |
|---|----|
| LOAD ALL..... | 34 |
| USE_SERIAL2 | 34 |
| USE_TTL2023..... | 34 |
| USE_EXT14 | 34 |
| GPS.GPS_INIT_STRING | 34 |
| GPS.GPS_INIT_NMEA_STRING | 35 |
| GPS.GPS_START_READING | 35 |
| GPS.MAX_BAD_READINGS | 35 |
| GPS. CONTINUOUS..... | 35 |
| GPS. SONY..... | 35 |
| GPS. USGGLOBALSAT..... | 35 |
| GPS. GPS_TYPE | 36 |
| GPS. REQUIRED | 36 |
| GPS. INITIAL_BAUD_RATE | 36 |
| GPS. ENHANCED_GPS | 36 |
| GPS. FASTPOLL_TIME | 36 |
| GPS. TRIGGER_FASTPOLL_MPH..... | 36 |
| GPS. TRIGGER_FASTPOLL_KPH | 37 |
| GPS. TRIGGER_FASTPOLL_METERS..... | 37 |
| *SERIAL_PORT_BAUDRATE | 37 |
| SMARTSENSOR. CONTINUOUS..... | 37 |
| SMARTSENSOR.SNUM | 37 |
| SMARTSENSOR. MIN_THRESHOLD | 37 |
| SMARTSENSOR. MIN_THRESHOLD2 | 38 |
| SMARTSENSOR. MAX_THRESHOLD..... | 38 |
| SMARTSENSOR. MAX_THRESHOLD2 | 38 |
| SMARTSENSOR. DISPLAY_ALL_READINGS..... | 38 |
| SMARTSENSOR. SEND..... | 38 |
| SMARTSENSOR. SEND (for individual sensor)..... | 38 |
| SMARTSENSOR. IODEVICE_READING_PROCESSOR_CLASS | 39 |
| SMARTSENSOR. DELTA | 39 |
| SMARTSENSOR. DELTA2 | 39 |
| SMARTSENSOR. OFFSET | 39 |
| SMARTSENSOR. OFFSET2..... | 39 |
| SMARTSENSOR. FACTOR | 39 |
| SMARTSENSOR. FACTOR2 | 40 |
| SMARTSENSOR. NUM_AVG_READINGS2..... | 40 |
| SIRIT. TAGTYPE..... | 40 |
| SIRIT. READ_REPEAT | 40 |
| THERMISTOR. RFIX..... | 40 |
| THERMISTOR. MAXA2D | 40 |

M2Mapp.Ini Parameter Definitions

AVIDIRECTOR.MODEL

Implemented: Yes

Description: Configuration key to set the AVIDdirector model

Type of Value: String

Default: 100

Units: NA

Range of Values: 100, 200/M2M

Example: AVIDIRECTOR.MODEL=200

AVID.SERIAL

Implemented: Yes

Description: Configuration Key to set the serial number of the AVIDdirector

Type of Value: String

Default: 100

Units: NA

Range of Values: 100, 200/M2M

Example: AVIDIRECTOR.MODEL=200

SYSTEM.APP_NAME

Implemented: Yes

Description: Configuration key to set the Application name

Type of Value: String

Default: M2MApp

Units: NA

Range of Values: (Can be a customized name)

Example: SYSTEM.APP_NAME=MYSPECIALAPP

SYSTEM.OPTIONS

Implemented: Yes

Description: Configuration key name to set the Options for the M2MApp framework - may be entered in hex

Type of Value: Integer

Default: 0

Units: NA

Range of Values: 0-2147483647

Example: SYSTEM.OPTIONS=78

SYSTEM.MAPSERIAL - (Alias for MAPSERIALPORT)

Implemented: Yes

Description: See SYSTEM.MAPSERIALPORT

Type of Value: String

Default: Null

Units: NA

Range of Values: Format: serialN=portName,serialN=portName. SerialN {serial0, serial1,serial2,serial3} & PortName = {CONSOLE, RJ12, RS232, RADIO, EXT, RS232_2, RADIO2, TTL34}

Example: SYSTEM.MAPSERIAL=serial1=radio,serial2=radio_b

M2Mapp.Ini Parameter Definitions

SYSTEM.MAPSERIALPORT

Implemented: Yes

Description: Configuration file entry SYSTEM.MAPSERIALPORT to map the serial ports. The format is 'serialN=portname,serialN=portname' where 'serialN' is serial0 (note this will direct your console to this port!), serial1, serial2, serial3 and 'portname' is one of CONSOLE, RJ12, RS232, RADIO, EXT, RS232_2, RADIO2, TTL34

Type of Value: String **Default:** Null **Units:** NA

Range of Values: Format: serialN=portName,serialN=portName. SerialN {serial0, serial1, serial2, serial3} & PortName = {CONSOLE, RJ12, RS232, RADIO, EXT, RS232_2, RADIO2, TTL34}

Example: SYSTEM.MAPSERIALPORT=serial1=radio,serial2=radio_b

SYSTEM.RDD_PWR_ON

Implemented: Yes

Description: Configuration file entry to set the power on for the radio board independent of radios.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: SYSTEM.RDD_PWR_ON=true

SYSTEM.FORCE_RDD_PWR_ON

Implemented: Yes

Description: Configuration key to set the boolean value used to turn on the radio power independent of any radios and keep it on.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: SYSTEM.FORCE_RDD_PWR_ON=true

SYSTEM.DEBUG

Implemented: Yes

Description: Configuration file entry to set the debug level. This is set to either: 0 (no debug), 1 (some debug), 2 (all debug messages), nn or 0xnn Mask for the 32 bit debug flag value. Level 1 is the normal setting, unless you want all debug turned off. Note that Logger provides an interactive changing of this debug setting. Note you will see noticeable slower response time with level 1 debug enabled and increased slower response time with level 2 enabled.

Type of Value: Integer **Default:** 1 **Units:** NA

Range of Values: 0,1,2,NN-Mask for the 32 bit debug flag value

Example: SYSTEM.DEBUG=0

SYSTEM.NOLOGGING

Implemented: Yes

Description: Configuration key to set the boolean property name for no logging to the console - turns off display of log messages on the console, True or false (default false).

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: SYSTEM.NOLOGGING=true

M2Mapp.Ini Parameter Definitions

SYSTEM.TIMESTAMP_LOG

Implemented: Yes

Description: Configuration key for enabling display of timestamps on our debug and log messages, True or false (default false).

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: SYSTEM.TIMESTAMP_LOG=true

SYSTEM.LOG_DEBUG

Implemented: Yes

Description: Configuration key to set the property name for copying the debug data to the log file. True or false (default false).

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: SYSTEM.LOG_DEBUG=true

SYSTEM.REBOOT_ON_EXIT

Implemented: Yes

Description: Configuration key to set the property to force a reboot of the system of we exit our M2Mapp Framework. Set to True or false. (default true). This will ensure fail safe operation if there is an error.

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: SYSTEM.REBOOT_ON_EXIT=false

SYSTEM.CLASSLOAD_ERRORS_FATAL

Implemented: Yes

Description: Configuration key to set the property that will treat a class loading error (we can't load a class) as a fatal error. Set to True or false. (default true). This will ensure fail safe operation if there is an error

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: SYSTEM.CLASSLOAD_ERRORS_FATAL=false

SYSTEM.LOG_FATAL_ERROR

Implemented: Yes

Description: Configuration key to set the property for logging our fatal error log. This is all the log messages before we reboot. True or false. We still write the LastFatalError.log file containing the Fatal error message regardless of this.

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: SYSTEM.LOG_FATAL_ERROR=false

M2Mapp.Ini Parameter Definitions

SYSTEM.LOGFILE

Implemented: Yes

Description: Configuration key to set the property for the Logging level which is set to a non zero number for logging, The settings are: 0 disables logging to the log file (default), 1 will log to a new log file, 2 will log and append to the log file, If this is a negative number then all logging is disabled

Type of Value: Integer

Default: 0

Units: NA

Range of Values: 0,1,2

Example: SYSTEM.LOGFILE=1

SYSTEM.LOGFILENAME

Implemented: Yes

Description: Configuration key to set the property for the Logging filename. This defaults to '/tmp/M2MApp.log'.

Type of Value: String

Default: /tmp/M2MApp.log

Units: NA

Range of Values: Valid filename with path

Example: SYSTEM.LOGFILENAME=/root/MyApp.log

SYSTEM.LOGFILE_MAXSIZE

Implemented: Yes

Description: Configuration key to set the property that defines the maximum size of our log file before we will break it into a backup log. This defaults to the Logger.logfileMaxSize value, which is currently 64000 bytes

Type of Value: Integer

Default: 64000

Units: Bytes

Range of Values: 1024-128000

Example: SYSTEM.LOGFILE_MAXSIZE=96000

SYSTEM.INTERACTIVE

Implemented: Yes

Description: Configuration key to set the property for Interactive display of M2MApp application, True or false

Type of Value: Boolean

Default: false

Units:

Range of Values: true/false

Example: SYSTEM.INTERACTIVE = true

SYSTEM.GC_RUN_INTERVAL

Implemented: Yes

Description: Configuration key to set the property for how often the memory is checked to see if the Garbage Collector needs to be run, in seconds. Default is 1 minute. Any Value above 10000 is assumed to be Milliseconds.

Type of Value: Integer

Default: 60

Units: Seconds/Milliseconds

Range of Values: 0-2147483647

Example: SYSTEM.GC_RUN_INTERVAL=120

M2Mapp.Ini Parameter Definitions

SYSTEM.GC_WATERMARK

Implemented: Yes

Description: Configuration key to set the property for how low the free heap ram can get to before it triggers a garbage collection (GC). Default is 150,000 bytes.

Type of Value: Integer **Default:** 1000001

Units: NA **Range of Values:** 0-2147483647

Example: SYSTEM.GC_WATERMARK=2000001

SYSTEM.WATCHDOG

Implemented: Yes

Description: Configuration key to set the property for enabling the watchdog timer. If enabled the application must continue to pet it or it will reset.

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: SYSTEM.WATCHDOG=true

SYSTEM.WATCHDOG_TIME

Implemented: Yes

Description: Configuration key to set the property for the Watch dog timer value in milliseconds.

Type of Value: Integer **Default:** 131000

Units: Milliseconds **Range of Values:** 0-131000

Example: SYSTEM.WATCHDOG_TIME=120000

SYSTEM.WATCHDOG_HWRESET

Implemented: Yes

Description: Configuration key to set the property to enable the hardware Watch dog timer value.

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: SYSTEM.WATCHDOG_HWRESET=false

SYSTEM.DEVICE_UUID

Implemented: Yes

Description: Configuration key to set the property for the Server/M2MXML Telemetry device Unique Device ID. Previous names (and still recognized) for this are M2MXML.DEVICE_UUID and SL.DEVICE_UUID.

Type of Value: String **Default:** Null

Units: NA **Range of Values:** Can be a user defined name

Example: SYSTEM.DEVICE_UUID=PowerMonitor

M2MXML.DEVICE_UUID (Alias for SYSTEM.DEVICE_UUID)

Implemented: Yes

Example: M2MXML.DEVICE_UUID=Temperature

SL.DEVICE_UUID (Alias for SYSTEM.DEVICE_UUID)

Implemented: Yes

Example: SL.DEVICE_UUID=Vibration

M2Mapp.Ini Parameter Definitions

SYSTEM.MAX_NOCOMM_TIME

Implemented: Yes

Description: Configuration key to set the property for how long we can allow no communication to the server. Default is 0 it is not checked.

Type of Value: Long

Default: 0

Units: Milliseconds

Range of Values: 1-2147483647

Example: SYSTEM.MAX_NOCOMM_TIME=180

M2MXML.MAX_NOCOMM_TIME (Alias for SYSTEM.MAX_NOCOMM_TIME)

Implemented: Yes

Example: M2MXML.MAX_NOCOMM_TIME=180

M2MXML.VERSION

Implemented: Yes

Description: Configuration key to set the property for the version of M2MXML we are using.

Type of Value: String

Default: BETA

Units: NA

Range of Values: 1.0, 1.1(recommended), 1.2

Example: M2MXML.VERSION = 1.1

M2MLET.CLASSNAME

Implemented: Yes

Description: Configuration key to set the property for a Class loader classname. Used to load IODevices, Radios and M2Mlets. This is a common endings for configuration file entries for multiple Class files

Type of Value: String

Default: Null

Units: NA

Range of Values: Full Classname

Example: M2MLET.CLASSNAME= com.avidwireless.avidirector.Demo_M2Mlet

M2MLET.LOAD

Implemented: Yes

Description: Configuration key to set the Load property for a class. If set to false the IODevice, Radio or M2Mlet, the class files won't be loaded. This is a common endings for configuration file entries for multiple Class files

Type of Value: Boolean

Default: true

Units: NA

Range of Values: true/false

Example: M2MLET.LOAD=FALSE

M2MLET.START

Implemented: Yes

Description: Configuration key to set the property that will run the start method or not. If set to false the IODevice, Radio or M2Mlet, they won't call the START method. This is a common endings for configuration file entries for multiple Class files

Type of Value: Boolean

Default: true

Units: NA

Range of Values: true/false

Example: M2MLET.START=false

M2Mapp.Ini Parameter Definitions

NO_COMMUNICATOR

Implemented: Yes

Description: Configuration key read to determine if we don't have a communicator associated with this radio. This is normally false, but setting it true will prevent the communicator message threads from running. The entry to load the Radio's class file reads the reset radio value using the key name for the class loader and appending .NO_COMMUNICATOR.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: XPORT.NO_COMMUNICTAOR=true

ACTIVE_COMMUNICATOR

Implemented: Yes

Description: Configuration key read to determine if this Radio is the active communicator. Normally the first radio read in the configuration file is used, but this allows the radio specifically to be specified as the active Communicator to our portal. The entry to load the Radio's class file reads the reset radio value using the key name for the class loader and appending .ACTIVE_COMMUNICATOR.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: XPORT.ACTIVE_COMMUNICATOR=true

MODEM_MODEL

Implemented: Yes

Description: Configuration key read to determine the Radio's model. This is for Similar board with multiple carriers or carrier specific features Known models are: Sprint CDMA, Verizon CDMA.

Type of Value: String **Default:** Null

Units: NA **Range of Values:** Verizon CDMA, Sprint CDMA

Example: CDMA.MODEM_MODEL=Sprint CDMA

SERIAL_PORT

Implemented: Yes

Description: Configuration key read to determine the Radio's serial port from the Configuration file. The entry to load the Radio's class file reads the serial port using the key name for the class loader and appending .SERIAL_PORT to read the serial port for that radio.

Type of Value: String **Default:** serial3

Units: NA **Range of Values:** serial1, serial2, serial3, serial0

Example: GPRS.SERIAL_PORT=serial1

M2Mapp.Ini Parameter Definitions

MAPSERIALPORT

Implemented: Yes

Description: Configuration key read to determine if we have to map this serial port to another hardware configuration. This allows the radios to map the serial port to a variety of hardware configurations. The entry to load the Radio's class file reads the serial port using the key name for the class loader and appending .MAPSERIALPORT to read the hardware for the serial port for that radio.

Type of Value: String

Default: Null

Units: NA

Range of Values: RS232, RADIO/RADIOA/RDD1, RADIOB/RDD2, GPS/RDD3, XPORT/RDD4, EXTB

Example: XPORT.MAPSERIALPORT=XPORT

SERIAL_PORT_BAUDRATE

Implemented: Yes

Description: Configuration key read to determine the Radio's serial port baud rate from the Configuration file. The entry to load the Radio's class file reads the serial port using the key name for the class loader and appending .SERIAL_PORT_BAUDRATE to read the serial port for that radio.

Type of Value: Integer

Default: 57600

Units: NA

Range of Values: 4800, 9600, 38400, 57600 (optimal), 115200

Example: GPRS.SERIAL_PORT_BAUDRATE=19200

SERIAL_PORT_HANDSHAKE

Implemented: Yes

Description: Configuration key read to determine the Radio's serial port hardware handshake configuration from the Configuration file. The entry to load the Radio's class file reads the serial port using the key name for the class loader and appending .SERIAL_PORT_HANDSHAKE to read the serial port for that radio.

Type of Value: String

Default: RTS/CTS

Units: NA

Range of Values: NONE, RTS/CTS

Example: CDMA.SERIALPORT_HANDSHAKE=NONE

SERIAL_PORT_TIMEOUT

Implemented: Yes

Description: Configuration key read to determine the Radio's serial port timeout from the Configuration file. The entry to load the Radio's class file reads the serial timeout using the key name for the class loader and appending .SERIAL_PORT_TIMEOUT to read the serial timeout for that radio.

Type of Value: Integer

Default: 3000

Units: Milliseconds

Range of Values: 10-180000

Example: MT_GPRS.SERIAL_PORT_TIMEOUT=5000

M2Mapp.Ini Parameter Definitions

AT_INIT

Implemented: Yes

Description: Configuration key used to read in additional AT initialization commands to send to the modem. These values are '|' (bar) delimited, so an example value might be "AT&K0|AT&R1". If the command starts with a "T=" then this will set a timeout for the commands going forward to wait for. The entry to load the Radio's class file reads the serial timeout using the key name for the class loader and appending .AT_INIT.

Type of Value: String

Default: Null

Units: NA

Range of Values: AT commands separated by '|'

Example: MT_GPRS.AT_INIT=ATE0|AT14

IP_ADDRESS

Implemented: Yes

Description: Configuration key read to determine the radio's IP address we use to communicate with the Portal from the Configuration file. This is usually not required since the Carrier assigns this to our device or is DHCP The entry to load the Radio's class file reads the Radio's IP address value using the key name for the class loader and appending .IP_ADDRESS

Type of Value: String

Default: Null

Units: NA

Range of Values: IP address string

Example: MT_GPRS.IP_ADDRESS=10.100.0.97

PHONE_NUMBER

Implemented: Yes

Description: Configuration key read to determine our radio's phone number. This is not normally used and can be read from the radio. The entry to load the Radio's class file reads the Phone number value using the key name for the class loader and appending .PHONE_NUMBER

Type of Value: String

Default: As Read from SIM

Units: NA

Range of Values: Unit's Phone Number

Example: MT_GPRS.PHONE_NUMBER=9725734000

NETWORK_NAME

Implemented: Yes

Description: Configuration key read to determine our modem's network name. The entry to load the Radio's class file reads the Network name value using the key name for the class loader and appending .NETWORK_NAME

Type of Value: String

Default: Null

Units: NA

Range of Values: NA

Example: MT_GPRS.NETWORKNAME=MYNETWORK

BAND

Implemented: Yes

Description: Configuration key read to determine our modem's network name. The entry to load the Radio's class file reads the Network name value using the key name for the class loader and appending .NETWORK_NAME

Type of Value: String

Default: Null

Units: NA

Range of Values: US,USA,Canada,850/1900,4,EUROPE,ROW,900/1800,5,6

Example: MT_GPRS.BAND=USA

M2Mapp.Ini Parameter Definitions

SLOT

Implemented: Yes

Description: Configuration key read to determine the Radio's slot (for AVIDdirector-XE) from the Configuration file. Slot for this radio modem; 1-4 for XE, 0 for M2M or -1 for external modem. The entry to load the Radio's class file reads the slot value using the key name for the class loader and appending .SLOT to read the slot that Radio is installed into

Type of Value: Integer

Default: 1

Units: NA

Range of Values: NA

Example: MT_GPRS.SLOT=0

SERVER_IP_ADDRESS

Implemented: Yes

Description: Configuration key read to determine the Gateway's IP address we use to communicate with the Portal from the Configuration file. The entry to load the Radio's class file reads the Server's IP address value using the key name for the class loader and appending .SERVER_IP_ADDRESS

Type of Value: String

Default: Null

Units: NA

Range of Values: IP address or name of the server

Example: XPORT.SERVER_IP_ADDRESS=aviddashboard.com

SERVER_PORT

Implemented: Yes

Description: Configuration key read to determine the Gateway's IP PORT number we use to communicate with the Portal from the Configuration file. The entry to load the Radio's class file reads the Server's PORT number value using the key name for the class loader and appending .SERVER_PORT.SERVER_IP_ADDRESS

Type of Value: Integer

Default: 0

Units: NA

Range of Values: Valid port number

Example: XPORT.SERVER_PORT=8088

SERVER_HTTP_M2MXML_SEND_PAGE

Implemented: Yes

Description: Configuration key read to get the HTTP page to send M2MXML to the Portal from the Configuration file. The entry to load the Radio's class file reads the HTTP page string value using the key name for the class loader and appending .SERVER_HTTP_M2MXML_SEND_PAGE

Type of Value: String

Default: Null

Units: NA

Range of Values: Valid send page URL

Example: XPORT.SERVER_HTTP_M2MXML_SEND_PAGE=/AVIDdirector/Host?uuid\=

M2Mapp.Ini Parameter Definitions

SERVER_HTTP_USE_HTTP_11

Implemented: Yes

Description: Configuration key read to get the option to send using HTTP 1.1 from the Configuration file. Otherwise HTTP 1.0 is used for communications. The entry to load the Radio's class file reads the HTTP User HTTP 1.1 value using the key name for the class loader and appending .SERVER_HTTP_USE_HTTP_11

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: XPORT.SERVER_HTTP_USE_HTTP11=true

SERVER_HTTP_SEND_USER_AGENT

Implemented: Yes

Description: Configuration key read to get the String to send as the User Agent String with the HTTP request. from the Configuration file. The entry to load the Radio's class file reads the HTTP User Agent send value using the key name for the class loader and appending .SERVER_HTTP_SEND_USER_AGENT

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: XPORT.SERVER_HTTP_SEND_USER_AGENT=true

SERVER_HTTP_M2MXML_POLL_PAGE

Implemented: Yes

Description: Configuration key read to get the HTTP page to poll for M2MXML from the Portal from the Configuration file. The entry to load the Radio's class file reads the HTTP page to poll string value using the key name for the class loader and appending .SERVER_HTTP_M2MXML_POLL_PAGE

Type of Value: String **Default:** Null

Units: NA **Range of Values:** Valid poll page URL

Example: XPORT.SERVER_HTTP_M2MXML_POLL_PAGE=/AVIDdirector/Host?uuid\=

SERVER_M2MXML_POLL_TIME

Implemented: Yes

Description: Configuration key read to get the interval (in milliseconds) to poll the M2MXML Portal from the Configuration file. The entry to load the Radio's class file reads the polling time value using the key name for the class loader and appending .SERVER_M2MXML_POLL_TIME

Type of Value: Long **Default:** 1

Units: Milliseconds **Range of Values:** 1-9223372036854769999

Example: XPORT.SERVER_M2MXML_POLL_TIME=1800000

SERVER_M2MXML_UDP_PORT

Implemented: Yes

Description: Configuration key read the UDP port to listen on for M2MXML shoulder taps from the server. The entry to load the Radio's class file reads the UDP port value using the key name for the class loader and appending .SERVER_M2MXML_UDP_PORT

Type of Value: Integer **Default:** 0

Units: NA **Range of Values:** 1-65535

Example: XPORT.SERVER_M2MXML_UDP_PORT=4326

M2Mapp.Ini Parameter Definitions

SERVER_SMS_NUMBER

Implemented: Yes

Description: Configuration key read to determine the Gateway's SMS number to dial that we use to communicate with the Portal from the Configuration file. The entry to load the Radio's class file reads the Server's SMS number value using the key name for the class loader and appending .SERVER_SMS_NUMBER

Type of Value: String

Default: Null

Units: NA

Range of Values: SMS Number

Example: MT_GPRS.SERVER_SMS_NUMBER=21455487654

SMS_STORE

Implemented: Yes

Description: Configuration key read to determine default SMS storage location for messages. Typically either SM for GPRS units and MT for CDMA. SM= SIM memory, BM= Volatile memory, SR= Status report storage. The entry to load the Radio's class file reads the default SMS storage value using the key name for the class loader and appending .SMS_STORE

Type of Value: String

Default: Null

Units: NA

Range of Values: SM, SR, BM, MT

Example: MT_GPRS.SMS_STORE=SM

NEW_SMS_MESSAGE_MODE

Implemented: Yes

Description: Configuration key read to determine if we are to just get notified for new SMS or receive them automatically. Set to 1 for notification and later read, 2=read immediately The entry to load the Radio's class file reads the new SMS message mode value using the key name for the class loader and appending .NEW_SMS_MESSAGE_MODE

Type of Value: Integer

Default: 0

Units: NA

Range of Values: 0,1,2

Example: MT_GPRS.NEW_SMS_MESSAGE_MODE=1

DONT_POWER_OFF

Implemented: Yes

Description: Configuration key used ensure we don't power off the radio when we stop it. This is mainly used when other radios are dependent on the Radio board power left on The entry to load the Radio's class file reads the serial timeout using the key name for the class loader and appending .DONT_POWER_OFF

Type of Value: Boolean

Default: false

Units: NA

Range of Values: true/false

Example: MT_GPRS.DON'T_POWER_OFF=true

M2Mapp.Ini Parameter Definitions

POWER_ON_WAIT_TIME

Implemented: Yes

Description: Configuration key used to read in how long in milliseconds to wait before talking to the radio after power initialization. The entry to load the Radio's class file reads the serial timeout using the key name for the class loader and appending .POWER_ON_WAIT_TIME

Type of Value: Integer

Default: 10000

Units: NA

Range of Values: 0-240000

Example: MT_GPRS.POWER_ON_WAIT_TIME=60000

INVERT_RADIO_PWRON

Implemented: Yes

Description: Configuration key read to invert the RDD_POWERON signal to the Radio board. The entry to load the Radio's class file reads the INVERT_RADIO_PWRON value using the key name for the class loader and appending .INVERT_RADIO_PWRON to read the boolean value.

Type of Value: Boolean

Default: false

Units: NA

Range of Values: true/false

Example: MT_GPRS.INVERT_RADIO_PWRON=true

RESET_RADIO

Implemented: Yes

Description: Configuration key that will be read in to determine if we need to reset the radio during initialization. This is set TRUE to reset the radio. The entry to load the Radio's class file reads the reset radio value using the key name for the class loader and appending .RESET_RADIO

Type of Value: Boolean

Default: false

Units: NA

Range of Values: true/false

Example: MT_GPRS.RESET_RADIO=true

MAX_FATAL_ERRORS

Implemented: Yes

Description: Configuration key read to determine the maximum errors before they are considered fatal, which triggers the device to reset itself The entry to load the Radio's class file reads the reset radio value using the key name for the class loader and appending .MAX_FATAL_ERRORS

Type of Value: Integer

Default: 5

Units: NA

Range of Values: 0-2147483647

Example: MT_GPRS.MAX_FATAL_ERRORS=20

MAX_RADIO_NOT_READY_ERRORS

Implemented: Yes

Description: Configuration key read to determine the maximum Radio Not Ready errors before we ignore them and continue the initialization. This primarily happens with GPRS radios with the status CME515. The default is 12 15 second tries (3 minutes) The entry to load the Radio's class file reads the reset radio value using the key name for the class loader and appending .MAX_RADIO_NOT_READY_ERRORS

Type of Value: Integer

Default: 12

Units: NA

Range of Values: 0-2147483647

Example: MT_GPRS.MAX_RADIO_NOT_READY_ERRORS=20

M2Mapp.Ini Parameter Definitions

GPRS_APN

Implemented: Yes

Description: Configuration key read to determine the Carrier's APN address we use to communicate with the Portal from the Configuration file. The entry to load the Radio's class file reads the Server's APN address value using the key name for the class loader and appending .GPRS_APN. <http://groups.google.com/group/aviddirector-m2m/web/m2m-apn-listings?version=5>

Type of Value: String

Default: Null

Units: NA

Range of Values: APN string

Example: MT_GPRS.GPRS_APN= WAP.CINGULAR

CARRIER

Implemented: Yes

Description: Configuration key read to determine the Carrier's information. We read the string and if it matches a known carrier in CarrierData then we set the APN, SMSC, PPP name and password and any other carrier specific data. The entry to load the Radio's class file reads the Carrier information using the key name for the class loader and appending .CARRIER

Type of Value: String

Default: Null

Units: NA

Range of Values: SPRINT, Cingular, Tmobile etc.

Example: MT_GPRS.CARRIER=CINGULAR

SMSC

Implemented: Yes

Description: Configuration key read to determine the Gateway's SMSC address we use to communicate with the Portal from the Configuration file. The entry to load the Radio's class file reads the Server's SMSC address value using the key name for the class loader and appending .SMSC

Type of Value: String

Default: Null

Units: NA

Range of Values: 11 digit SMSC number

Example: MT_GPRS.SMSC= 13123149810

REG_MODE

Implemented: Yes

Description: Configuration key read to determine the how we register with the carrier. Values are: 0 = Automatic mode, any carrier (default) 1 = Manual mode. Register only with the carrier specified in REG_CARRIER, 2 = Manual mode with fallback to automatic. First try to register with the carrier specified in REG_CARRIER. The entry to load the Radio's class file reads the Server's REG_MODE value using the key name for the class loader and appending .REG_MODE

Type of Value: Integer

Default: 0

Units: NA

Range of Values: 0,1,2

Example: MT_GPRS.REG_MODE=1

M2Mapp.Ini Parameter Definitions

REG_CARRIER

Implemented: Yes

Description: Configuration key read to determine the preferred carrier to register our device with. Only applies if REG_MODE has a value of 1 or 2 The entry to load the Radio's class file reads the REG_CARRIER value using the key name for the class loader and appending .REG_CARRIER

Type of Value: String

Default: Null

Units: NA

Range of Values: SPRINT, Cingular, Tmobile etc.

Example: MT_GPRS.REG_CARRIER=SPRINT

PPP_USERNAME

Implemented: Yes

Description: Configuration key read for the user name for the PPP connection. Default is blank. The entry to load the Radio's class file reads the Connection user name value using the key name for the class loader and appending .PPP_USERNAME

Type of Value: String

Default: Blank

Units: NA

Range of Values: NA

Example: MT_GPRS.PPP_USERNAME = WAP@CINGULARGPRS.COM

PPP_PASSWORD

Implemented: Yes

Description: Configuration key read for the password for the username for the PPP connection. Default is blank. The entry to load the Radio's class file reads the Connection password value using the key name for the class loader and appending .PPP_PASSWORD.

Type of Value: String

Default: Blank

Units: NA

Range of Values: NA

Example: MT_GPRS.PPP_PASSWORD = MYPW

PPP_GATEWAY

Implemented: Yes

Description: Configuration key read for the Server gateway to use for the PPP connection. Default is blank. The entry to load the Radio's class file reads the Connection password value using the key name for the class loader and appending .PPP_GATEWAY .

Type of Value: String

Default: Null

Units: NA

Range of Values: IP String

Example: MT_GPRS.PPP_GATEWAY=192.168.0.1

PPP_PASSIVE

Implemented: Yes

Description: Configuration key read to determine if our PPP connection is passive or active. Default is blank. The entry to load the Radio's class file reads the PPP Passive value using the key name for the class loader and appending .PPP_PASSIVE

Type of Value: Boolean

Default: true

Units: NA

Range of Values: true/false

Example: MT_GPRS.PPP_PASSIVE = false

M2Mapp.Ini Parameter Definitions

PPP_DNS_PRIMARY

Implemented: Yes

Description: Configuration key read for the primary DNS Server for the PPP connection. Default is blank. The entry to load the Radio's class file reads the Connection password value using the key name for the class loader and appending .PPP_DNS_PRIMARY

Type of Value: String

Default: Null

Units: NA

Range of Values: IP String

Example: MT_GPRS.PPP_DNS_PRIMARY=192.168.0.1

PPP_DNS_SECONDARY

Implemented: Yes

Description: Configuration key read for the secondary DNS Server for the PPP connection. Default is blank. The entry to load the Radio's class file reads the Connection password value using the key name for the class loader and appending .PPP_DNS_SECONDARY

Type of Value: String

Default: Null

Units: NA

Range of Values: IP String

Example: MT_GPRS.PPP_DNS_SECONDARY=192.168.0.1

DNS_TIMEOUT

Implemented: Yes

Description: Configuration file entry to set the DNS lookup timeout in milliseconds. The entry to load the Radio's class file reads the Network DNS timeout value using the key name for the radio's class loader and appending .DNS_TIMEOUT

Type of Value: Integer

Default: 15000

Units: Millisecond

Range of Values: 1000-120000

Example: MT_GPRS.DNS_TIMEOUT=30000

DNS_RETRIES

Implemented: Yes

Description: Configuration file entry to set the number of times to retry a DNS lookup before we say there is a DNS error. After DNS_LOOKUP_ERRORS setting (default to 1) we restart the PPP connection. The entry to load the Radio's class file reads the Network DNS retry value using the key name for the radio's class loader and appending .DNS_RETRIES

Type of Value: Integer

Default: 2

Units: NA

Range of Values: 2-2147483647

Example: XPORT.DNS_RETRIES=4

DNS_LOOKUP_ERRORS

Implemented: Yes

Description: Configuration file entry to set the number of DNS lookup errors before we drop the connection. The entry to load the Radio's class file reads the Network DNS retry value using the key name for the radio's class loader and appending .DNS_LOOKUP_ERRORS

Type of Value: Integer

Default: 1

Units: NA

Range of Values: 1-2147483647

Example: XPORT.DNS_LOOKUP_ERRORS=3

M2Mapp.Ini Parameter Definitions

SOCKET_TIMEOUT

Implemented: Yes

Description: Configuration file entry to set the TCP/IP Socket lookup timeout in milliseconds. The entry to load the Radio's class file reads the Network Socket timeout value using the key name for the radio's class loader and appending .SOCKET_TIMEOUT

Type of Value: Integer

Default: 30000

Units: Millisecond

Range of Values: 30000-120000

Example: XPORT.SOCKET_TIMEOUT=45000

PPP_AUTHENTICATION

Implemented: Yes

Description: Configuration key read for the type of authentication to use. PAP, CHAP or NONE/FALSE for none The entry to load the Radio's class file reads the Connection authentication value using the key name for the class loader and appending .PPP_AUTHENTICATION

Type of Value: String

Default: Null

Units: NA

Range of Values: PAP, CHAP

Example: XPORT.PPP_AUTHENTICATION=PAP

PPP_MRU

Implemented: Yes

Description: Configuration key read to determine the MRU (Maximum Receive Unit) value for our PPP connection. If not specified, then it defaults to the connection default. The entry to load the Radio's class file reads the PPP MRU size value using the key name for the class loader and appending .PPP_MRU

Type of Value: Integer

Default: 0

Units: NA

Range of Values: 512-1500

Example: XPORT.PPP_MRU=1400

PPP_DIALNUMBER

Implemented: Yes

Description: Configuration key read for the phone number to dial for the PPP connection. Default is blank. The entry to load the Radio's class file reads the PPP dialing number value using the key name for the class loader and appending .PPP_DIALNUMBER

Type of Value: String

Default: Null

Units: NA

Range of Values: NA

Example: XPORT.PPP_DIALNUMBER=*99***1#

PPP_DIALTIME

Implemented: Yes

Description: Configuration key read for how long to wait for the PPP server to dial. The entry to load the Radio's class file reads the PPP dial time using the key name for the class loader and appending .PPP_DIALTIME

Type of Value: Integer

Default: 10000

Units: Milliseconds

Range of Values: 0-180000

Example: XPORT.DIALTIME=15000

M2Mapp.Ini Parameter Definitions

PPP_CONNECTTIME

Implemented: Yes

Description: Configuration key read for how long wait to establish a PPP connection in startPPP(). The entry to load the Radio's class file reads the PPP Connection time value using the key name for the class loader and appending .PPP_CONNECTTIME

Type of Value: Integer

Default: 45000

Units: Milliseconds

Range of Values: 0-180000

Example: XPORT.CONNECTIONTIME=45000

PPP_HANGTIME

Implemented: Yes

Description: Configuration key read for how long to maintain a PPP connection after it is closed with stopPPP(). This allows reuse of an existing connection for multiple accesses. The entry to load the Radio's class file reads the PPP Connection Hang timer value using the key name for the class loader and appending .PPP_HANGTIME

Type of Value: Integer

Default: 15000

Units: Milliseconds

Range of Values: 0-240000

Example: XPORT.PPP_HANGTIME=60000

KEEP_CONNECTION_UP

Implemented: Yes

Description: Configuration key read if we are to keep the network connection alive or allow the radio to disconnect. This is useful for PPP radios such as GPRS devices that may have to disconnect to receive an SMS, or radios you want the PPP connection always maintained. The entry to load the Radio's class file reads the if we are to maintain the PPP connection up value using the key name for the class loader and appending .KEEP_CONNECTION_UP

Type of Value: Boolean

Default: false

Units: NA

Range of Values: true/false

Example: MT_GPRS.KEEP_CONNECTION_UP=true

CONNECTION_TYPE (IO270, GPRS radio)

Implemented: Yes

Description: Configuration key read for the type of connection to use to the server. The entry to load the Radio's class file reads the Connection type value using the key name for the class loader and appending .CONNECTION_TYPE

Type of Value: String

Default: HTTP

Units: NA

Range of Values: HTTP, SMS

Example: MT_GPRS.CONNECTION_TYPE=SMS

CONNECTION_TYPE (for bluetooth radio)

Implemented: Yes

Description: Configuration key read for the type of connection to use to the server: MASTER or SLAVE. The entry to load the Radio's class file reads the Connection type value using the key name for the class loader and appending .CONNECTION_TYPE

Type of Value: String

Default: MASTER

Units: NA

Range of Values: MASTER, SLAVE

Example: BLUETOOTH.CONNECTION_TYPE=SLAVE

M2Mapp.Ini Parameter Definitions

DEVICE_TYPE (BLUETOOTH Only)

Implemented: No

Description: Configuration key read for the type of device (or the class of device, in Bluetooth). The class of device (COD) we are. The entry to load the Radio's class file reads the Device type value using the key name for the class loader and appending .DEVICE_TYPE

Type of Value: Integer **Default:** 0x00000300

Units: NA **Range of Values:** 0-2147483647

Example: BLUETOOTH.DEVICE_TYPE=768

SERVICE_NAME (BLUETOOTH Only)

Implemented: No

Description: Configuration key read for the name of the available services offered by this device. The entry to load the Radio's class file reads the Service name value using the key name for the class loader and appending .SERVICE_NAME

Type of Value: String **Default:** COMO

Units: NA **Range of Values:**

Example: BLUETOOTH.SERVICE_NAME=COMO

DYNDNS_HOSTNAME

Implemented: Yes

Description: Configuration key read for the DynDNS.com hostname we want to track our IP address. There is not default entry. www.dyndns.com. The entry to load the Radio's class file reads the Service name value using the key name for the class loader and appending .DYNDNS_HOSTNAME

Type of Value: String **Default:** Null

Units: NA **Range of Values:** Server Name

Example: MT_GPRS.DYNDNS_HOSTNAME=test.dyndns.org

DYNDNS_USERNAME

Implemented: Yes

Description: Configuration key read for the DynDNS.com username for our account The entry to load the Radio's class file reads the Service name value using the key name for the class loader and appending .DYNDNS_USERNAME

Type of Value: String **Default:** Null

Units: NA **Range of Values:**

Example: MT_GPRS.DYNDNS_USERNAME=me@dnsServer.com

DYNDNS_PASSWORD

Implemented: Yes

Description: Configuration key read for the DynDNS.com password for our account The entry to load the Radio's class file reads the Service name value using the key name for the class loader and appending .DYNDNS_PASSWORD

Type of Value: String **Default:** Null

Units: NA **Range of Values:**

Example: MT_GPRS.DYNDNS_PASSWORD=mydnsServerpwd

M2Mapp.Ini Parameter Definitions

DYNDNS_DNS_TYPE

Implemented: Yes

Description: Configuration key read for the DynDNS.com password for our account The entry to load the Radio's class file reads the Service name value using the key name for the class loader and appending .DYNDNS_PASSWORD

Type of Value: String

Default: DYNAMIC

Units: NA

Range of Values: DYNAMIC, STATIC, CUSTOM

Example: MT_GPRS.DYNDNS_DNS_TYPE=CUSTOM

DYNDNS_AUTOUPDATE

Implemented: Yes

Description: Configuration key read for the DynDNS to automatically update the DynDNS server for any IP address changes. Defaults to false. This property is only used if DYNDNS hostname, username and password is set. The entry to load the Radio's class file reads the Service name value using the key name for the class loader and appending .DYNDNS_AUTOUPDATE

Type of Value: Boolean

Default: false

Units: NA

Range of Values: true/false

Example: MT_GPRS.DYNDNS_AUTOUPDATE=true

SIM_PIN

Implemented: Yes

Description: Configuration key read for the SIM PIN1 number. The entry to load the Radio's class file reads the Service name value using the key name for the class loader and appending .SIM_PIN

Type of Value: String

Default: Null

Units: NA

Range of Values: four-to-eight digit numeric or password

Example: MT_GPRS.SIM_PIN=1111

SIM_PIN2

Implemented: Yes

Description: Configuration key read for the SIM PIN2 number. The entry to load the Radio's class file reads the Service name value using the key name for the class loader and appending .SIM_PIN

Type of Value: String

Default: Null

Units: NA

Range of Values: four-to-eight digit numeric or password

Example: MT_GPRS.SIM_PIN2=2221

ENCRYPT_DESKEY (Aerocomm, AerocommZigbee)

Implemented: No

Description: Configuration value for the DES key (56 bit value). If we want to encrypt the communications, we will use the DES key.

Type of Value: Long

Default: 1

Units: NA

Range of Values: 1-9223372036854769999

Example: AEROCOMM.ENCRYPT_DESKEY=256

M2Mapp.Ini Parameter Definitions

USE_RDD_XMIT_CMD_MODE (Aerocomm, AerocommZigbee)

Implemented: Yes

Description: Configuration value for the using RDD_XMIT line for Command Mode in older radios

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: AEROCOMM.USE_RDD_XMIT_CMD_MODE=true

USE_RDD_AUX_CMD_MODE (Aerocomm, AerocommZigbee)

Implemented: Yes

Description: Configuration value for the using RDD_AUX line for new radios

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: AEROCOMM.USE_RDD_AUX_CMD_MODE=false

DESTINATION_ADDRESS

Implemented: Yes

Description: The property name for the Mobitex Gateway Address

Type of Value: String **Default:** 0

Units: NA **Range of Values:** Mobitex Gateway address

Example: SERIALPORTRADIO.DESTINATION_ADDRESS=13455

MAX_RETRY_COUNT

Implemented: Yes

Description: Configuration value for maximum retry counter

Type of Value: Integer **Default:** 3

Units: NA **Range of Values:** 0-2147483647

Example: MT_GPRS.MAX_RETRY_COUNT=10

MAX_QUEUE_SIZE

Implemented: Yes

Description: Configuration value for maximum queue size

Type of Value: Integer **Default:** 25

Units: NA **Range of Values:** 0-2147483647

Example: XPORT.MAX_QUEUE_SIZE=100

DELETE_QUEUE_FULL

Implemented: Yes

Description: Configuration value to delete the oldest message if the queue gets full

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: MT_GPRS.DELETE_QUEUE_FULL=true

M2Mapp.Ini Parameter Definitions

LIFO

Implemented: Yes

Description: Setting to turn the message queue into a Last In, First Out queue instead of the default First In First Out

Type of Value: Boolean

Default: false

Units: NA

Range of Values: true/false

Example: XPORT.LIFO=TRUE

IODEVICE.SERIAL_PORT

Implemented: Yes

Description: Constants for the configuration file for the serial port to use

Type of Value: String

Default: Device dependant

Units: NA

Range of Values: serial1, serial2, serial3 or serial0

Example: IODEVICE.SERIAL_PORT = serial1

IODEVICE.SERIAL_PORT_BAUD_RATE

Implemented: Yes

Description: Serial port baud rate for I/O devices.

Type of Value: Integer

Default: Device dependant

Units: NA

Range of Values: Standard baud rates from 1200 to 115200, 57600 (optimal), 115200

Example: IODEVICE.SERIAL_PORT_BAUD_RATE = 57600

IODEVICE.SERIAL_PORT_TIMEOUT

Implemented: Yes

Description: Serial port timeout when listening for reply.

Type of Value: Integer

Default: Device dependant

Units: Milliseconds

Range of Values: 10-180000

Example: IODEVICE.SERIAL_PORT_TIMEOUT = 100

IODEVICE.ENABLE

Implemented: Yes

Description: Defines if a particular I/O device is enabled or not.

Type of Value: Boolean

Default: Device dependant

Units: NA

Range of Values: true/false

Example: IODEVICE.ENABLE = true

M2Mapp.Ini Parameter Definitions

IODEVICE.MAPSERIALPORT

Implemented: Yes

Description: Configuration key read to determine if we have to map this serial port to another hardware configuration. This allows the IODevices to map the serial port to a variety of hardware configurations. The entry to load the IODevice's class file reads the serial port using the key name for the class loader and appending .MAPSERIALPORT to read the hardware for the serial port for that radio.

Type of Value: String **Default:** Device dependant **Units:** NA

Range of Values: RS232, RADIO/RADIOA/RDD1, RADIOB/RDD2, GPS/RDD3, XPORT/RDD4, EXTB

Example: IODEVICE.MAPSERIALPORT = GPS

IODEVICE.RDD_PWR_ON

Implemented: Yes

Description: Configuration file parameter to turn on the Radio Power for this device

Type of Value: Boolean **Default:** Device dependant

Units: NA **Range of Values:** true/false

Example: IODEVICE.RDD_PWR_ON = true

IODEVICE.M2MXML_ADDRESS

Implemented: Yes

Description: Specified ending for an M2M_IODevice Address

Type of Value: String **Default:** Null

Units: NA **Range of Values:** (User defined String)

Example: IODEVICE.M2MXML_ADDRESS = POWERSENSOR

IODEVICE.NUM_AVG_READINGS

Implemented: Yes

Description: Number of readings to be used for getting an average reading

Type of Value: Integer **Default:** 0

Units: NA **Range of Values:** 0-2147483647

Example: IODEVICE.NUM_AVG_READINGS = 15

M2Mapp.Ini Parameter Definitions

IODEVICE.TIMEZONE

Implemented: No

Description: Set the TimeZone property for which time zone the AVIDdirector is to operate.

Type of Value: String

Default: GMT

Units: NA

Range of Values: *Abbreviations from the time zone table

Example: IODEVICE.TIMEZONE=CST

Time Zone Table

| Offset | Abbrev | Where |
|--------|-------------|----------------------------|
| -11 | MIT | Samoa |
| -10 | HST | Hawaii |
| -9 | AST | Alaska |
| -8 | PST | Pacific Standard Time |
| -7 | MST,PNT | Mountain Standard Time |
| -6 | CST | Central Standard Time |
| -5 | EST,IET | Eastern Standard Time |
| -4 | PRT | Atlantic Standard Time |
| -3.5 | CNT | Newfoundland |
| -3 | AGT,BET | Eastern South America |
| 0 | UTC,GMT,WET | Coordinated Universal Time |
| 1 | ECT | European Central Time |
| 2 | EET,CAT,ART | European Eastern Time |
| 3 | EAT | Saudi Arabia |
| 3.5 | MET | Iran |
| 4 | NET | |
| 5 | PLT | West Asia |
| 5.5 | IST | India |
| 6 | BST | Central Asia |
| 7 | VST | Bangkok |
| 8 | CTT | China |
| 9 | JST | Japan |
| 9.5 | ACT | Central Australia |
| 10 | AET | Eastern Australia |
| 11 | SST | Central Pacific |
| 12 | NST | New Zealand |

IODEVICE.POLLTIME

Implemented: Yes

Description: Specified ending for an IODevice polling time.

Type of Value: Integer

Default: 0

Units: Milliseconds

Range of Values: 0-1200000

Example: IODEVICE.POLLTIME = 300000

M2Mapp.Ini Parameter Definitions

IODEVICE.TIME

Implemented: Yes

Description: Generic definition for the Time property. Used for the reboot command.

Type of Value:

Default:

Units:

Range of Values:

Example:

M2MIODEVICE

Implemented: Yes

Description: Prefix to specify for an M2M_IODevice common parameter for more than one device.

Type of Value: NA

Default: NA

Units: Seconds

Range of Values: To be used as a prefix for other parameters to make the parameter applicable to all M2MIODEVICES

Example: M2MIODEVICE.ABSOLUTE_REPORT_TIME=1800

SETPOINT

Implemented:

Description: The Set point for SetAnalogOutput value

Type of Value:

Default:

Units:

Range of Values:

Example:

DATA

Implemented:

Description: The Set point for SetStringOutput value

Type of Value:

Default:

Units:

Range of Values:

Example:

VALUE

Implemented:

Description: The Value for SetDigitalOutput value

Type of Value:

Default:

Units:

Range of Values:

Example:

IOTYPE

Implemented: Yes

Description: Specified ending for an M2M_IODevice I/O type

Type of Value: String

Default: Null

Units: NA

Range of Values: AIN, AOUT, DIN, DOUT

Example: TTL1.IOTYPE=AIN

M2Mapp.Ini Parameter Definitions

IPOLLING

Implemented: Yes

Description: Specified ending for an M2M_IODevice polling enabled or not

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: TTL3.POLLING=true

READING_PROCESSOR_CLASSNAME

Implemented: Yes

Description: Classname for a IODeviceReadingProcessor instance we are trying to load

Type of Value: String (Class Name) **Default:** Null

Units: NA **Range of Values:** NA

Example: THERMISTOR.READING_PROCESSOR_CLASSNAME=ThermistorReadingProcessor

ABSOLUTE_REPORT_TIME

Implemented: Yes

Description: Set an absolute GMT time to report the current reading value. Time is in GMT seconds for each day. Valid for Analog and Digital input devices.

Type of Value: Integer **Default:** 0

Units: Seconds **Range of Values:** 0-86400

Example: TTL1.ABSOLUTE_REPORT_TIME=1800

PERIODIC_REPORT_TIME

Implemented: Yes

Description: Set an absolute GMT time to report the current reading value. Time is in GMT seconds for each day. Valid for Analog and Digital input devices.

Type of Value: Integer **Default:** 0

Units: Seconds **Range of Values:** 0-86400

Example: TTL1.ABSOLUTE_REPORT_TIME=1800

DIGITAL_TRIGGER_HIGH

Implemented: Yes

Description: Report if the input goes from a low to a high DigitalInputHigh.

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: TTL4.DIGITAL_TRIGGER_HIGH=false

INVERT

Implemented:

Description: If set true this will invert the reported value. This is true for both digital inputs and outputs.

Type of Value: Boolean **Default:**

Units: NA **Range of Values:** true/false

Example:

M2Mapp.Ini Parameter Definitions

PULLUP_INPUT

Implemented: Yes

Description: Set the port to have an active pullup using the PSoC. Mutually exclusive to Pulldown setting.

Type of Value: Boolean

Default: false

Units: NA

Range of Values: true/false

Example: TTL2.PULLUP_INPUT=true

PULLDOWN_INPUT

Implemented: Yes

Description: Set the port to have an active pulldown using the PSoC. Mutually exclusive to Pullup setting.

Type of Value: Boolean

Default: false

Units: NA

Range of Values: true/false

Example: TTL2.PULLDOWN_INPUT=true

DURATION

Implemented:

Description: Set the time interval to set the output. Time in milliseconds. Valid for Digital output devices..

Type of Value: Integer

Default:

Units: Milliseconds

Range of Values:

Example:

PULSEHIGH_TIME

Implemented: Yes

Description: Set the time to pulse the output high. Time in seconds. Valid for Digital output devices.

Type of Value: Integer

Default: 0

Units: NA

Range of Values: 0-2147483647

Example: TTL3.PULSEHIGH_TIME=30

PULSELOW_TIME

Implemented: Yes

Description: Set the time to pulse the output low. Time in seconds. Valid for Digital output devices.

Type of Value: Integer

Default: 0

Units: NA

Range of Values: 0-2147483647

Example: TTL3.PULSELOW_TIME=10

TIMER_HIGHINTERVAL_TIME

Implemented:

Description: Set the time interval to set the output high. Time in seconds. Valid for Digital output devices.

Type of Value: Integer

Default:

Units: Seconds

Range of Values:

Example:

M2Mapp.Ini Parameter Definitions

TIMER_LOWINTERVAL_TIME

Implemented:

Description: Set the time interval to set the output low. Time in seconds. Valid for Digital output devices.

Type of Value: Integer

Default:

Units: Seconds

Range of Values:

Example:

TIMER_HIGHABSOLUTE_TIME

Implemented:

Description: Set an absolute GMT time to set the output high. Time is in GMT seconds. Valid for Digital output devices.

Type of Value: Integer

Default:

Units: Seconds

Range of Values:

Example:

TIMER_LOWABSOLUTE_TIME

Implemented:

Description: Set an absolute GMT time to set the output low. Time is in GMT seconds. Valid for Digital output devices.

Type of Value: Integer

Default:

Units: Seconds

Range of Values:

Example:

DIGITAL_TRANSITION_ALARM_LEVEL

Implemented: Yes

Description: Sets the transition of a digital input to report an alarm value. Argument is either 1=Low to High, 2=High to Low. Valid for Digital input devices.

Type of Value: Integer

Default:

Units: NA

Range of Values:1,2

Example:

ANALOG_OFFSET

Implemented: Yes

Description: Sets offset applied to the analog input or output. For input, this is added to the read value to make the final reading. For output, this is applied to the given value before sent to the analog output. The argument can be either floating or integer format and defaults to 0. Valid for Analog input or output devices.

Type of Value: Double

Default: 0

Units: NA

Range of Values: All double values

Example: TTL4.ANALOG_OFFSET=23.7

M2Mapp.Ini Parameter Definitions

ANALOG_FACTOR

Implemented: Yes

Description: Sets factor (multiplier) applied to the analog input or output. For input, this is multiplied to the read value to make the final reading. For output, this is applied to the given value before sent to the analog output. The argument can be either floating or integer format and defaults to 1.000. Valid for Analog input or output devices.

Type of Value: Double

Default: 1

Units: NA

Range of Values: All double values

Example: TTL4.ANALOG_FACTOR=0.00564

ANALOG_ABSOLUTE_HIGH

Implemented: Yes

Description: Sets absolute thresholds for the analog input to report a reading to the portal. The argument can be either floating or integer format. Valid for Analog input devices.

Type of Value: Double

Default: -1

Units: NA

Range of Values: All double values

Example: TTL5.ANALOG_ABSOLUTE_HIGH = -5

ANALOG_ABSOLUTE_LOW

Implemented: Yes

Description: Sets absolute thresholds for the analog input to report a reading to the portal. The argument can be either floating or integer format. Valid for Analog input devices.

Type of Value: Double

Default: -1

Units: NA

Range of Values: All double values

Example: TTL5.ANALOG_ABSOLUTE_LOW = -25

ANALOG_DEADBAND_HIGH

Implemented: Yes

Description: Sets absolute thresholds for the analog input to report a reading to the portal. The argument can be either floating or integer format. Valid for Analog input devices.

Type of Value: Double

Default: 0

Units: NA

Range of Values: All double values

Example: TTL5.ANALOG_DEADBAND_HIGH=8.5

ANALOG_DEADBAND_LOW

Implemented: Yes

Description: Sets absolute thresholds for the analog input to report a reading to the portal. The argument can be either floating or integer format. Valid for Analog input devices.

Type of Value: Double

Default: 0

Units: NA

Range of Values: All double values

Example: TTL5.ANALOG_DEADBAND_LOW=1.5

M2Mapp.Ini Parameter Definitions

LOAD

Implemented: Yes

Description: Suffix for command to not load or initialize the M2M_IODevice's.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: M2MIODEVICE.LOAD=true

LOADALL

Implemented: Yes

Description: Suffix for command to not load or initialize M2M_IODevice that are not listed in the M2MAll.ini file. This defaults to false, so only load the ones specified with an entry in the M2MApp.ini file.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: M2MIODEVICE.LOADALL=true

USE_SERIAL2

Implemented: Yes

Description: Suffix for command to use Serial2 as a M2M_IODevice.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: M2MIODEVICE.USE_SERIAL2=true

USE_TTL2023

Implemented: Yes

Description: Suffix for command to use Serial2 as a M2M_IODevice for TTL2023.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: M2MIODEVICE.USE_TTL2023L=true

USE_EXT14

Implemented: Yes

Description: Suffix for command to use EXT1-4 as a M2M_IODevice.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: M2MIODEVICE.USE_EXT14=true

GPS.GPS_INIT_STRING

Implemented: Yes

Description: String of characters separated by a ~ for each line. The initialization strings to be used to initialize this GPS module.

Type of Value: String separated by '~' **Default:** Null **Units:** NA

Range of Values: (passed to the GPS module as is. Useful for passing initialization strings for SONY modules. See SONY documentation for more information.)

Example: GPS.GPS_INIT_STRING=@CLR~@CB 4800

M2Mapp.Ini Parameter Definitions

GPS.GPS_INIT_NMEA_STRING

Implemented: Yes

Description: GPS NMEA device initialization string in NMEA format, used by older units. String of characters separated by a ~ for each line.

Type of Value: String separated by '~' **Default:** Null **Units:** NA

Range of Values: (passed to GPS module with checksum, as a NMEA strings. See NMEA user manual for more information)

Example: GPS.GPS_INIT_NMEA_STRING=\$PSRF100,1,57600,8,1,0~\$PSRF103,00,00,01,01

GPS.GPS_START_READING

Implemented: Yes

Description: GPS NMEA command to perform a single reading. Command sent in NMEA format. Null if GPS automatically sends out data.

Type of Value: String separated by '~' **Default:** Null **Units:** NA

Range of Values: (passed to GPS module without checksum. Expects a NMEA string. See NMEA user manual for more information)

Example: GPS.GPS_START_READING=\$PSRF103,01,00,00,01*25

GPS.MAX_BAD_READINGS

Implemented: Yes

Description: GPS NMEA command to perform a single reading. Null if GPS automatically sends out data.

Type of Value: Integer **Default:** 4

Units: NA **Range of Values:** 0-2147483647

Example: GPS.MAX_BAD_READINGS=6

GPS.CONTINUOUS

Implemented: Yes

Description: If the reader reads continuously from the data stream to know its location. Useful for applications that need high accuracy in 5 meter range.

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: GPS.CONTINUOUS=false

GPS.SONY

Implemented: Yes

Description: Set the GPS module type as SONY or not SONY.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: GPS.SONY=true

GPS.USGLOBALSAT

Implemented: Yes

Description: Set the GPS module type as USGLOBALSAT or not USGLOBALSAT.

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: GPS.USGLOBALSAT=false

M2Mapp.Ini Parameter Definitions

GPS. GPS_TYPE

Implemented: Yes

Description: Set the GPS input/output format type.

Type of Value: String

Default: NMEA

Units: NA

Range of Values: NMEA, SIRF

Example: GPS.GPS_TYPE=SIRF

GPS. REQUIRED

Implemented: Yes

Description: The required set of NMEA sentences (separated by comma) to constitute a valid reading.

Type of Value: String separated by ',' **Default:** GPGGA, GPRMC

Units: NA

Range of Values: Any combination of the following

strings: GPGGA, GPRMC, GPGLL, GPVTG

Example: GPS.REQUIRED=GPGGA,GPRMC,GPGLL

GPS. INITIAL_BAUD_RATE

Implemented: Yes

Description: The initial serial port baud rate the GPS module starts up with.

Type of Value: Integer

Default: SERIAL_PORT_BAUDRATE parameter value

Units: NA

Range of Values: 4800(most common), 9600, 38400, 57600

(optimal), 115200

Example: GPS.INITIAL_BAUD_RATE=4800

GPS. ENHANCED_GPS

Implemented: Yes

Description: The configuration parameter to send enhanced GPS readings or not. Enhanced readings include altitude, speed, direction.

Type of Value: Boolean

Default: false

Units: NA

Range of Values: true/false

Example: GPS.ENHANCED_GPS=true

GPS. FASTPOLL_TIME

Implemented: Yes

Description: Set the time interval to send data when performing fast polling.

Type of Value: Integer

Default: 0

Units: Milliseconds

Range of Values: 0-2147483647

Example: GPS.FASTPOLL_TIME=180000

GPS. TRIGGER_FASTPOLL_MPH

Implemented: Yes

Description: Set the change in speed (MPH) trigger to send data when fast polling.

Type of Value: Integer

Default: 0

Units: Miles/Hour

Range of Values: 0-2147483647

Example: GPS.Trigger_FASTPOLL_MPH=10

M2Mapp.Ini Parameter Definitions

GPS. TRIGGER_FASTPOLL_KPH

Implemented: Yes

Description: Set the change in speed (KPH) trigger to send data when fast polling.

Type of Value: Integer **Default:** 0

Units: Kilometers/Hour **Range of Values:** 0-2147483647

Example: GPS.Trigger_FASTPOLL_KPH=15

GPS. TRIGGER_FASTPOLL_METERS

Implemented: Yes

Description: Set the change in speed (Meters) trigger to send data when fast polling.

Type of Value: Integer **Default:** 0

Units: Meters **Range of Values:** 0-2147483647

Example: GPS.Trigger_FASTPOLL_METERS=150

****SERIAL_PORT_BAUDRATE***

Implemented: Yes

Description: Set the serial port baud rate as for all IODEVICES

Type of Value: IODeviceAll **Default:** 4800

Units: Meters **Range of Values:** IODeviceAll

Example: IODeviceAll

SMARTSENSOR. CONTINUOUS

Implemented: Yes

Description: If the reader should read continuously or only send upon request.

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: SMARTSENSOR.CONTINUOUS=false

SMARTSENSOR.SNUM

Implemented: Yes

Description: The unique serial number of the sensor.

Type of Value: String **Default:** Null

Units: NA **Range of Values:** Hex String

Example: SMARTSENSOR.DOOR_SNUM=D0008765

SMARTSENSOR. MIN_THRESHOLD

Implemented: Yes

Description: The minimum value for a sensor - needed for analog.

Type of Value: Double **Default:** NO_VALUE(-99999.99d)

Units: Double **Range of Values:** All Double Values

Example: SMARTSENSOR.DOOR_MIN_THRESHOLD=35

M2Mapp.Ini Parameter Definitions

SMARTSENSOR. MIN_THRESHOLD2

Implemented: Yes

Description: The minimum value for a sensor data 2- needed for analog.

Type of Value: Double **Default:** NO_VALUE(-99999.99d)

Units: Double **Range of Values:** All Double Values

Example: SMARTSENSOR.DOOR_MIN_THRESHOLD2=37

SMARTSENSOR. MAX_THRESHOLD

Implemented: Yes

Description: The max threshold value for a sensor - for data1.

Type of Value: Double **Default:** NO_VALUE(-99999.99d)

Units: Double **Range of Values:** All Double Values

Example: SMARTSENSOR.DOOR_MAX_THRESHOLD=135

SMARTSENSOR. MAX_THRESHOLD2

Implemented: Yes

Description: The max threshold value for a sensor - for data2.

Type of Value: Double **Default:** NO_VALUE(-99999.99d)

Units: Double **Range of Values:** All Double Values

Example: SMARTSENSOR.DOOR_MAX_THRESHOLD2=235

SMARTSENSOR. DISPLAY_ALL_READINGS

Implemented: Yes

Description: If set true then display the Smart sensor readings received even if we don't send them.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: SMARTSENSOR.DISPLAYALLREADINGS=true

SMARTSENSOR. SEND

Implemented: Yes

Description: If the reader should send data on triggers or only on requests. Applicable to all sensors.

Type of Value: Boolean **Default:** false

Units: NA **Range of Values:** true/false

Example: SMARTSENSOR.SEND=true

SMARTSENSOR. SEND (for individual sensor)

Implemented: Yes

Description: If the reader should send data on triggers or only on requests. Applies to individual sensor.

Type of Value: Boolean **Default:** true

Units: NA **Range of Values:** true/false

Example: SMARTSENSOR.DOOR_SEND=true

M2Mapp.Ini Parameter Definitions

SMARTSENSOR. IODEVICE_READING_PROCESSOR_CLASS

Implemented: No-Future

Description: A processor class for the Analog data that could be invoked for different data formatting/calculations.

Type of Value: IODeviceReadingProcessor Class type **Default:** Null

Units: NA **Range of Values:** NA

Example:

SMARTSENSOR. DELTA

Implemented: Yes

Description: The delta change that triggers sending data1 to the server.

Type of Value: Double **Default:** 0

Units: NA **Range of Values:** All Double Values

Example: SMARTSENSOR.DOOR_DETA=30

SMARTSENSOR. DELTA2

Implemented: Yes

Description: The delta change that triggers sending data2 to the server.

Type of Value: Double **Default:** 0

Units: NA **Range of Values:** All Double Values

Example: SMARTSENSOR.DOOR_DETA2=33

SMARTSENSOR. OFFSET

Implemented: Yes

Description: The offset to apply to raw data1 before sending to the server.

Type of Value: Double **Default:** 0

Units: NA **Range of Values:** All Double Values

Example: SMARTSENSOR.DOOR_OFFSET=36.9

SMARTSENSOR. OFFSET2

Implemented: Yes

Description: The offset to apply to raw data2 before sending to the server.

Type of Value: Double **Default:** 0

Units: NA **Range of Values:** All Double Values

Example: SMARTSENSOR.DOOR_OFFSET2=46.9

SMARTSENSOR. FACTOR

Implemented: Yes

Description: The factor to apply to raw data1 before sending to the server.

Type of Value: Double **Default:** -1

Units: NA **Range of Values:** All Double Values

Example: SMARTSENSOR.DOOR_FACTOR=0.00036

M2Mapp.Ini Parameter Definitions

SMARTSENSOR. FACTOR2

Implemented: Yes

Description: The factor to apply to raw data2 before sending to the server.

Type of Value: Double

Default: -1

Units: NA

Range of Values: All Double Values

Example: SMARTSENSOR.DOOR_FACTOR2=0.00056

SMARTSENSOR. NUM_AVG_READINGS2

Implemented: No

Description: The number of readings to average data2 over, use

CONFIG_FILE_NUM_AVG_READINGS from IODevice for number of readings to average data1 over

Type of Value: Integer

Default: 0

Units: NA

Range of Values: 0-2147483647

Example: SMARTSENSOR.DOOR_NUM_AVG_READINGS2=15

SIRIT. TAGTYPE

Implemented: Yes

Description: The tag types we are using. Enter a 0, 1 or 2. Applies to the Gen type ie 2 would mean Gen 2 tags.

Type of Value: Integer

Default: 0

Units: NA

Range of Values: 0,1,2

Example: SIRIT.TAGTYPE=1

SIRIT. READ_REPEAT

Implemented: Yes

Description: Number of times to repeat the read. If more than a couple tags are being read, set this to 20 to 30. Otherwise a value of 4 or so should be sufficient. 4 for single tags, 20-30 for multiple tags.

Type of Value: Integer

Default: 4

Units: NA

Range of Values: 4, 20-30

Example: SIRIT.READ_REPEAT = 25

THERMISTOR. RFIX

Implemented: Yes

Description: Get the type of sensor.

Type of Value: Integer

Default: 10000

Units: NA

Range of Values: 0-2147483647

Example: THERMISTOR.RFIX=5000

THERMISTOR. MAXA2D

Implemented: Yes

Description: The maximum A2D reading value.

Type of Value: Integer

Default: 4096

Units: NA

Range of Values: 0-2147483647

Example: THERMISTOR.MAXA2D=256