Autonomous Configuration Tool User Guide

Version 1.2 - Supported Hardware:

- Micro (firmware v1.7.3 and later
- Nano (firmware v1.5.0 and later
- M6e (firmware v1.19.0 and later
- USB*Pro* Reader (firmware v1.7.3 and later)

Government Limited Rights Notice: All documentation and manuals were developed at private expense and no part of it was developed using Government funds.

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose the technical data contained herein are restricted by paragraph (b)(3) of the Rights in Technical Data — Noncommercial Items clause (DFARS 252.227-7013(b)(3)), as amended from time-to-time. Any reproduction of technical data or portions thereof marked with this legend must also reproduce the markings. Any person, other than the U.S. Government, who has been provided access to such data must promptly notify ThingMagic.

ThingMagic, A Division of Trimble
1 Merrill St.
Woburn, MA 01801

02 Revision A
March, 2016

Contents

Autonomous Configuration Tool v1.2 User Guide

Introduction

This document explains how to use the Autonomous Configuration Tool (ACT), version 1.2, distributed with MercuryAPI version 1.27.3. It supports the latest releases of Micro, Nano, and M6e module firmware, as well as the USB*Pro* reader. This utility provides the ability to setup the module to read and output tag results without a host MercuryAPI application by enabling hardware read triggers and savable configuration settings in the module.

ACT is written using the high level MercuryAPI in Java. For more information on the MercuryAPI see the *MercuryAPI Programmers Guide* and the *MercuryAPI SDK*, available on the ThingMagic website http://www.thingmagic.com.

This document assume you have read the corresponding Hardware Guide for the Micro module being connected to and understand how the reader is connected.

Note

Autonomous operation can be configured using ACT or via the latest version of MercuryAPI (1.27.3). Sample code is provided that shows how to save the full configuration, including configuration settings, read plan options, and trigger controls, to enable creation of custom auto-config utilities and scripts.

Supported Hardware

- Micro module with firmware v1.7.3 or later.
- Nano module with firmware v1.5.0 or later
- M6e module with firmware v1.19.0 or later
- USB*Pro* reader with firmware v1.7.3 or later

Autonomous Reading Overview

Autonomous operation allows the reader to be configured for two basic behaviors:

Read On Boot

When Read & Write Settings I **Autonomous Read = *enabled*** and **Auto-read on GPI = *disabled*** the reader will start reading within 150ms of powering up. As tags are read it will stream the results in the .Reader Data Packet Format

To stop reading you must shut down the reader (cut power or use the module's SHUTDOWN line).

When using SHUTDOWN line to start/stop reads, the time from setting **SHUTDOWN=HIGH to first tag read is ~150ms**.

Note

This mode is recommended when maximum power savings is a high priority and the small delay of 150ms from power on to read can be tolerated.

Read On GPI Trigger

When Read & Write Settings I **Autonomous Read = *enabled*** and **Auto-read on GPI = *enabled*** the reader will start reading as soon as the specified GPI is set HIGH. As tags are read it will stream the results in the Reader Data Packet Format.

When GPI is low, a Temperature Status Message will be returned approximately every 250ms (this will vary based on the Region setting used as it depends on the internal frequency hopping algorithm which is dependent on the region of operation).

This mode allows for very fast read response to GPI triggers: **less than 15ms from set GPI=High to first tag read**.

This mode will **draw up to 3.2W (at 30dBm power setting) while idle (GPI=LOW).** Idle power consumption decreases with lower RF power settings.

Note

GPI Trigger mode is only recommended when very fast response time to a read trigger event is required due to the high idle power consumption.

USB I/O Specific Notes

- When connected via the USB interface the reader will buffer tag reads if the USB connection is lost or not established. This will eventually cause the buffer to fill and the reader to throttle reading until a USB connection is made and tags can be off-loaded and reading started again.

TTL/UART I/O Specific Notes

- When connected via the TTL/UART interface the reader cannot detect whether a connection exists or not. This means data does not get buffered due to detecting a missing connection as they do with the USB interface. As a result if a connection is lost or none is established the reader will continue reading and streaming results, which will be lost.

Usage Workflow

Once ACT is installed, see Installing ACT section for details, this document will step through the required steps in order to setup and save the desired configuration to the reader and how to process the output. At a high level the steps required, and corresponding document sections, are as follows:

1. Starting ACT - This section provides an overview of the main components of the ACT application displayed immediately upon startup.
2. Connect - In order to configure a reader for autonomous operation the reader must first be connected to. This section provides details on connecting to the reader in ACT.
3. Configure - Once connected the desired settings for autonmous operation must be configured. This section describes all the options available in addition to how to upgrade the reader's firmware to a version that supports autonomous operation.
4. Read - Once the autonomous mode configuration has been selected and applied to a reader ACT can be used to display the tags being streamed from the reader. This section describes using ACT to display results.
5. Processing Data from the Reader - In order to handle tag results from the reader outside of ACT the structure of the messages from the reader must be understood. This section describes the packet format in addition to pointers to reference code in C, C# and Java that can be used to process the results.
6. Troubleshooting - If any of the previously defined steps don't work this section will help diagnose and resolve the issue.

Installing ACT

The Autonomous Configuration Tool is available for 32-bit and 64-bit Linux and Windows. The executables for all are available in the Auto-config zip package available on the ThingMagic website http://rfid.thingmagic.com/devkit.

Required Software

ACT requires Java runtime engine (JRE) version 1.8 or later.

W A R N I N G !

There is a JRE regression (https://bugs.openjdk.java.net/browse/JDK-8055875) introduced in Java 1.8.20 (currently not fixed as of version 1.8.25) which can cause ACT to crash.
Until it is fixed, for stable operation we recommend using Java version 1.8.11.

Download, Install and Run ACT

Follow these steps to download and install ACT:

1. Use a web browser to navigate to http://www.thingmagic.com/manuals-firmware
2. Find the Autonomous Configuration Tool link under Micro l Software and click on it. This will download *autoread-1.27.3.xx.zip*
3. Once downloaded, extract the zip archive to the desired location.
4. Follow the steps for your OS:

Windows

1. Go to the directory created from extracting the zip archive.
2. Double-click the appropriate AutonomousConfigurationToolWin[32l64].exe to invoke the installer.
3. Follow the installer steps.
4. Once complete there will be an entry in the Windows Start menu for *ThingMagic | AutonomousConfigurationTool*

Linux

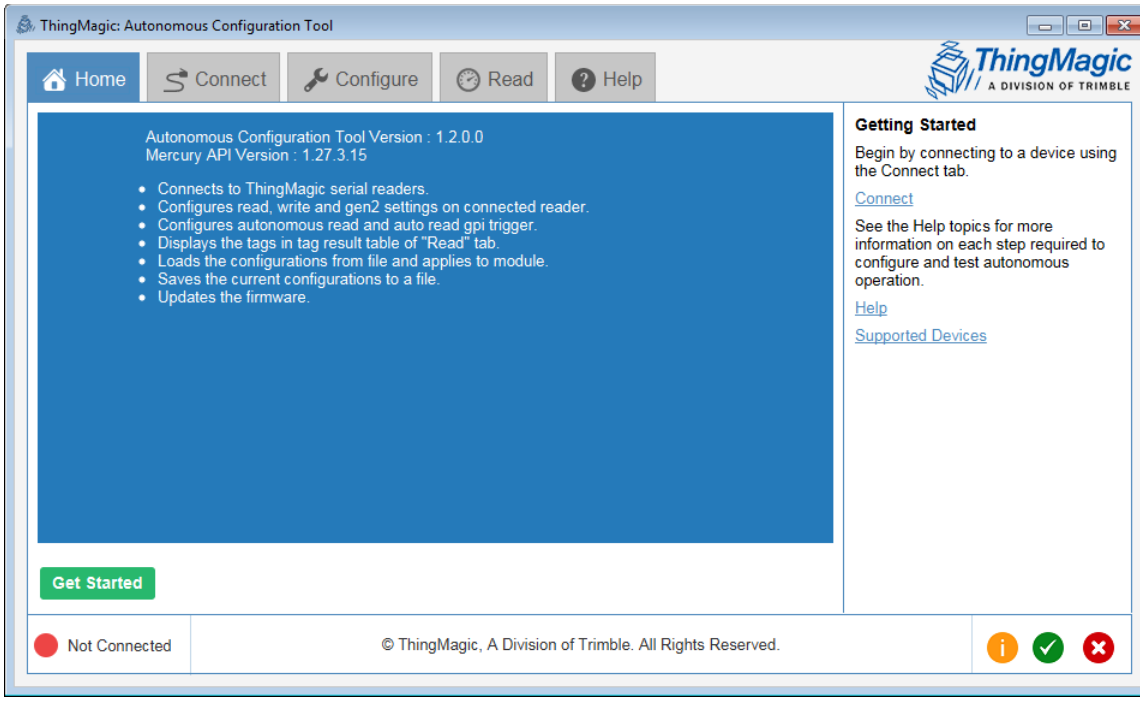
1. Go to the directory created from extracting the zip archive
2. Right-click on the appropriate AutonomousConfigurationToolLinux[32|64].sh
3. Under Permissions check “Allow executing file as a program”. Close
4. Double-click on AutonomousConfigurationToolLinux[32|64].sh and select “Run” to run the tool.

Starting ACT

Main Window

Upon starting up ACT you will see the Home screen as shown in *Figure 1*.

***Figure 1:* Startup Tag Results Screen**



This screen introduces the set of available features of ACT. Across the top of the window are the following tabs:

- Connect (Get Started also brings you to the Connect tab) - The Connect Tab is used to discover and connect to available readers
- Configure - Once connected the Configure tab will present all the saveable configuration settings along with the ability to load firmware.
- Read - After autonomous configuration settings have been applied to the connected module the Read tab will provide a quick way to view streaming tag reads
- **Help** - Contains help information for using ACT.

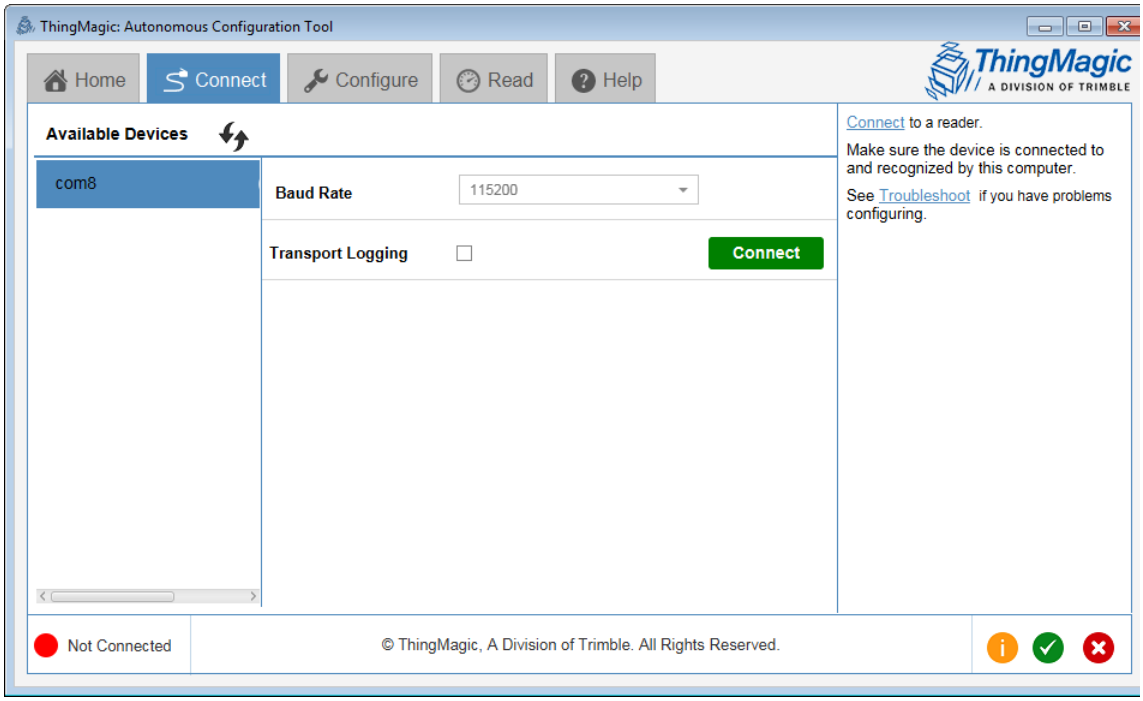
In addition to the main tabs providing access to different functionality, additional information is provided on all screens in the following sections:

- **Help Panel -** The right section of each screen contains a short summary of contextual help along with links to more detailed help.
- **Connected/Not Connected Status** - The bottom left shows the reader's connection status.
- **Message Icons** - Warning, Status and Error messages will pop-up from the corresponding bottom right message icons..

Connect

The Connect tab provides connection options to find and connect to a reader. Please reference the specific product's Hardware Guide for details on how to establish a connection to each reader and how to identify the correct (COM port, etc.) to identify it.

Figure 2: **Connecting to a Reader**



Finding the assigned COM port

In order to connect to a serial reader the correct COM port must be selected from the *Available Devices* list. In order for the associated COM port to be displayed the reader must have been connected to the host prior to starting ACT. If not, you can click *Refresh* to try and re-detect COM ports in use. The specific value depends on what the host system has assigned the new connection, typically it will be the last value in the list.

Note

On Linux the device detection does not work. In order to connect to the reader you must know the device name it has been assigned and enter it manually in the "Enter com port" section of the last entry in the *Available Devices* list. *Typically*, if using the Micro module in the ThingMagic devkit it will be:

/dev/ttyACM0 - When using the Micro's native USB interface (labeled USB on the devkit).
/dev/ttyUSB0 - When using the ThingMagic devkit USB/RS232 port which connects to the Micro's TTL interface via an FTDI USB to Serial converter.

Baud Rate

If the reader is using a TTL/RS232 connection and you know the baud rate of the reader being connected to you can select it prior to connecting. If you don't, ACT will attempt to figure it out, then set the baud rate to the specified value.

Note

The baud rate value set here will be the value saved to the module when the settings are applied on the Configure tab. Make sure to set this to the appropriate values supported by the host hardware when using the TTL interface of the reader.

For native USB connections the baud rate setting does not apply.

Upon successful connect the *Connect* button will change to *Disconnect* and you can move onto the Configure tab.

Transport Log

If you wish to see the hexidecimal communication messages between ACT and the module for debugging purposes, select "Transport Logging" prior to connecting to the module. You will be prompted where to store the file, which will have a date-stamped filename such as this, "ACT_TransportLogs_2016-03-18-18-46-02.txt".

Connecting to a Reader Already in Autonomous Mode

When in autonomous mode the reader is primarily in a reading mode. However, it should be able to handle a Connect operation from ACT, allowing you to then disable or change the autonomous configuration settings.

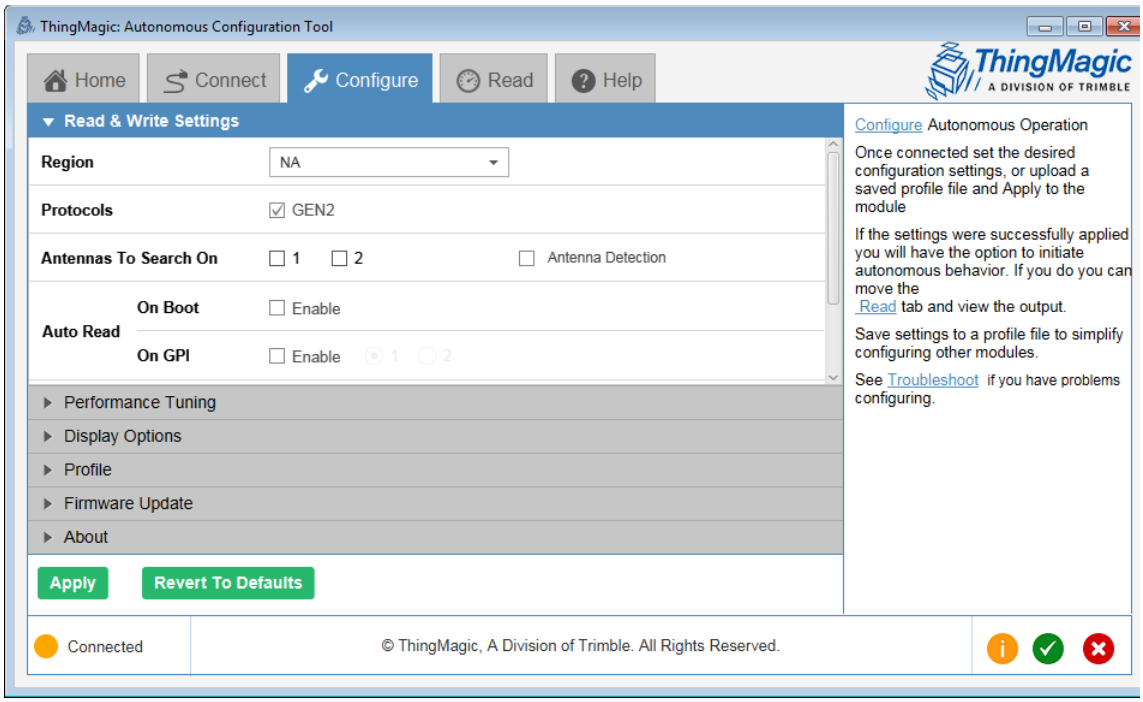
If ACT is not able to connect you may have to use the reader's RESET line to stop it from booting into autonmous operation. The module's RESET line (Micro edge via #17 or Molex pin 20) must be pulled LOW on power up to avoid booting in to autonomous mode. This corresponds to jumping the AUTO_BOOT jumper (J19) on the ThingMagic Devkit to GND.

See the Micro Hardware Guide for more details on the behavior of the RESET input signal.

Configure

After a successful Connect to a reader the Configure tab, as shown in *Figure 3*, provides all the available autonomous reading settings. In addition you can install firmware, configure the Read display options and check the current firmware version of the connected reader.

***Figure 3:* Configure Tab**

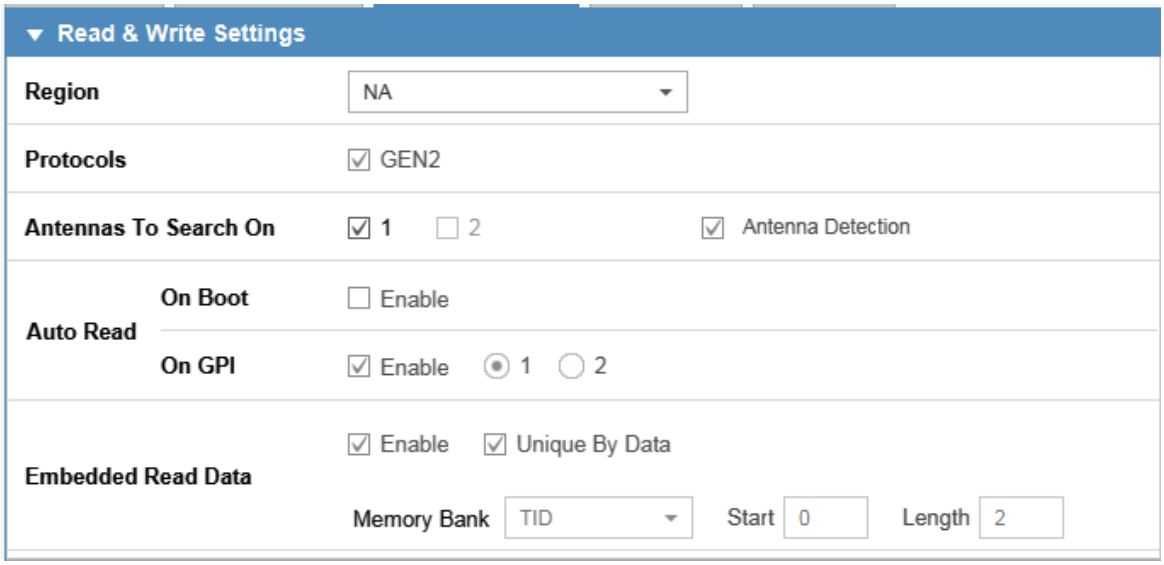


W A R N I N G !

Once *Apply* is clicked all selected settings will be saved to the connected reader. This may cause the reader to start reading at 100% duty cycle.

Read & Write Settings

This sections allows the basic behavior of a autonomous reading to be defined.



The following configuration settings can be defined for autonomous operation:

Table 1: **Read & Write Settings**

Setting	Description
Region	The region of operation for the module.
Protocols	The protocols supported by the connected reader will be displayed. Check those to be queried for. When Read is clicked the selected protocols will be cycled through.
Antennas	• **[Antennas]** - The antennas supported by the connected reader will be displayed. Check those to be used during autonomous reading. • By default reading behavior follows the behavior of a StartReading() MercuryAPI operation with a 1000ms /reader/read/aysncOnTime.setting. • If multiple antennas are selected the read cycles through the antennas, moving to the next antenna when no more tags are found on the current antenna. The cycle resets to the first antenna every *1000ms*. • The “Antenna Detection” button is a convenient way of selecting antennas.

Enable on Boot	• Turns on autonomous operation. Once selected and applied the reader will no longer require a MercuryAPI host application to control its operation. It will start streaming tag read data according to the saved configuration. • **In this mode you must shut down the reader** (cut power or use the module's SHUTDOWN line) **in order to stop reading**. • When using SHUTDOWN line to start/stop reads, the time from setting **SHUTDOWN=HIGH to first tag read is ~150ms**. Note: If Auto-read on GPI is selected it will only transmit RF and output tag reads when GPI = High.
Enable on GPI	• **Enable** - Causes autonomous reads to only start when GPI is set High. • Select the **GPI pin** that will be used as the read trigger • This mode allows for very fast read response to GPI triggers: **< 10ms from set GPI=High to first tag read**. • This mode will **draw up to 3.2W (at 30dBm power setting) while idle (GPI=LOW).** Idle power consumption decreases with lower RF power settings.
Embedded ReadData	For every tag read the specified memory bank location will also be read for that tag and returned. • **Enable** - Displays the embedded read data options to be added to the read operation and adds the Data column. • **Memory Bank** - The memory bank to read from. • **Start** - The starting WORD address to read from. (0 based) • **Length** - The number of WORDs of data to read. Specifying 0 will result in the entire memory bank starting at address *Start* to be read, up to 128 words. • **UniqueByData** - Causes the data read to be used as a unique identifier of the tag. Useful if many tags in the field have the same EPC, read the TID UID and they will be distinguishable. Note: Adding an Embedded Read Data changes the data that is returned with each tag read. See Processing Data from the Reader for details.

Performance Tuning Options

This section provides a set of options, based on high-level usecase characteristics, allowing the reader performance to be optimized.

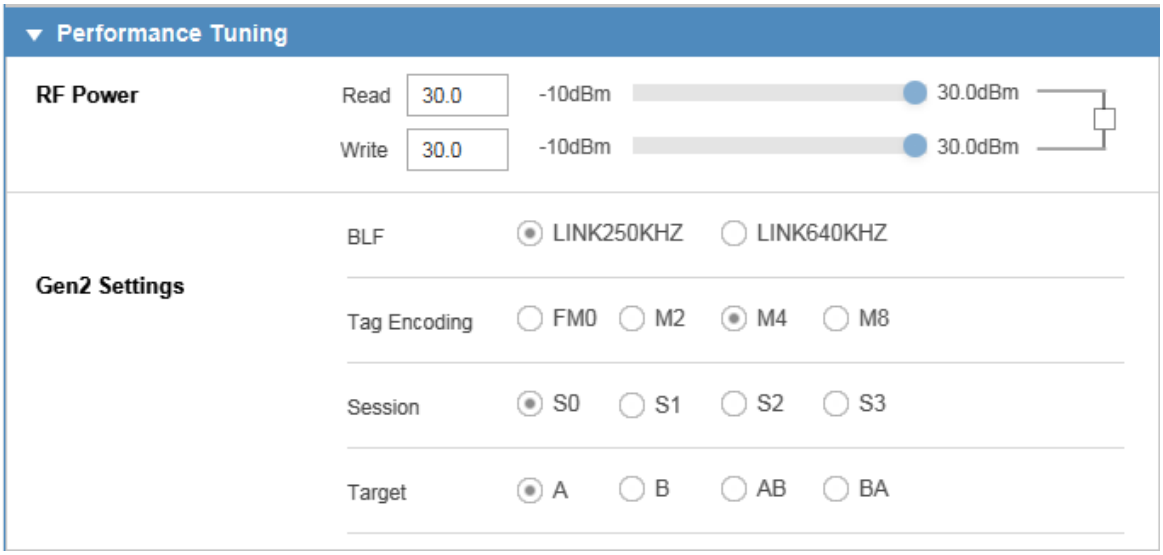


Table 2: **Performance Tuning Options**

Setting	Description
Read and Write Power	Control the amount of RF Power transmitted by the reader, to the antenna, when reading. The RF power directly relates to the range at which the reader can “see” tags, the higher the power, the longer the range. Note that autonomous operation does not yet support tag writing, so this setting will have no effect except to save the value in memory.
Gen2 Settings	See the *MercuryAPI Programmers Guide* | *Performance Tuning* section for more details on these settings.

Display Options

This sections allows the content Read display to be configured. It contains the following options:

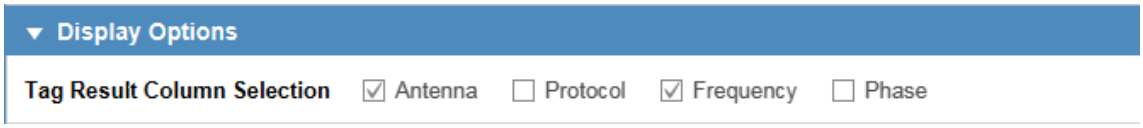
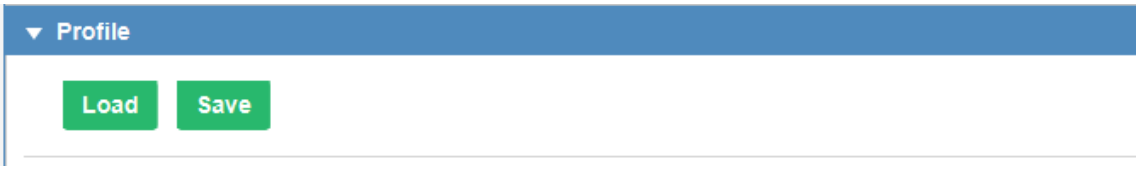


Table 3: **Display Options**

Setting	Description
Tag Results Column	Additional Columns of tag read meta data can be selected (click check-box in drop-down menu) to be displayed: • **Antenna** - The antenna on which the tag was read. If the same tag is read on more than one antenna there will be a tag entry for each antenna on which the tag was read. • **Protocol** - The protocol of the tag. • **Frequency** - The frequency on which the tag was read, in kHz. • **Phase** - Average phase of tag response in degrees (0°-180°) Note: Changing the display options does NOT change the metadata that is returned with each tag read nor the format of the data output from the reader.

Profile

Full settings and display options profiles can be saved to a file, and loaded from a file using the Save and Load buttons of the Profile section. This can simplify loading the same configuration to multiple readers. It uses the same settings format as the Universal Reader Assistant tool and configuration can be shared across tools.



Save Profile

Once connected to a reader the desired autonomous configuration of the reader, including read settings, performance settings and display settings can be saved to a URAC (URA Config) file by following these simple steps:

1. Click “Save”
2. File manager pops up with the default filename field populated with [reader type]_[readername].urac. This maybe changed as desired.
3. Click “Save” in the file manager to save to the specified file.

Load Profile

If you want to load a previously saved profile onto another reader you must first connect to the reader then load the settings into ACT then apply them to the reader:

1. Once connected to the reader click “Load”.
2. Select desired *.urac file and “Open”.
3. The full saved profile and display options are loaded.
4. Click “Apply” to store the loaded settings onto the connected read.

Firmware Update

The section provides a way to upgrade the firmware of the connected reader.

To upgrade:

1. Click *Choose Firmware* and navigate to the filesystem location where the firmware file is located.
2. Select the firmware file and click *Open*.
3. Click *Update*.
4. Upon completion the Status message will change to indicate it has completed.

Note

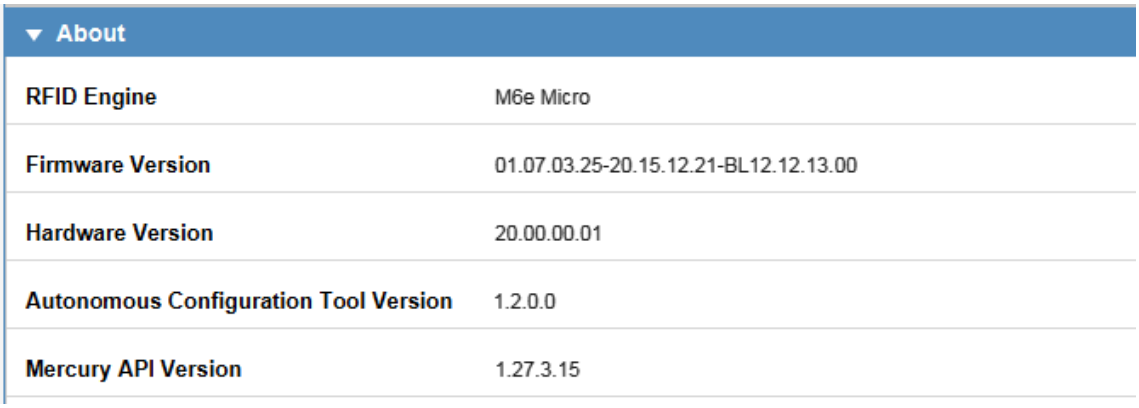
The latest firmware is always available on http://www.thingmagic.com/manuals-firmware. You must accept the software license agreement by typing “Accept” into the password prompt field.

Note

This utility can be used to downgrade firmware also.

About

This section displays firmware version of the connected reader. When contacting ThingMagic support for issues found while using ACT it is recommended this information be provided.



Note

ACT operate best with the firmware versions listed in the introduction. It will not work at all for firmware that was released before autonomous operation was supported.

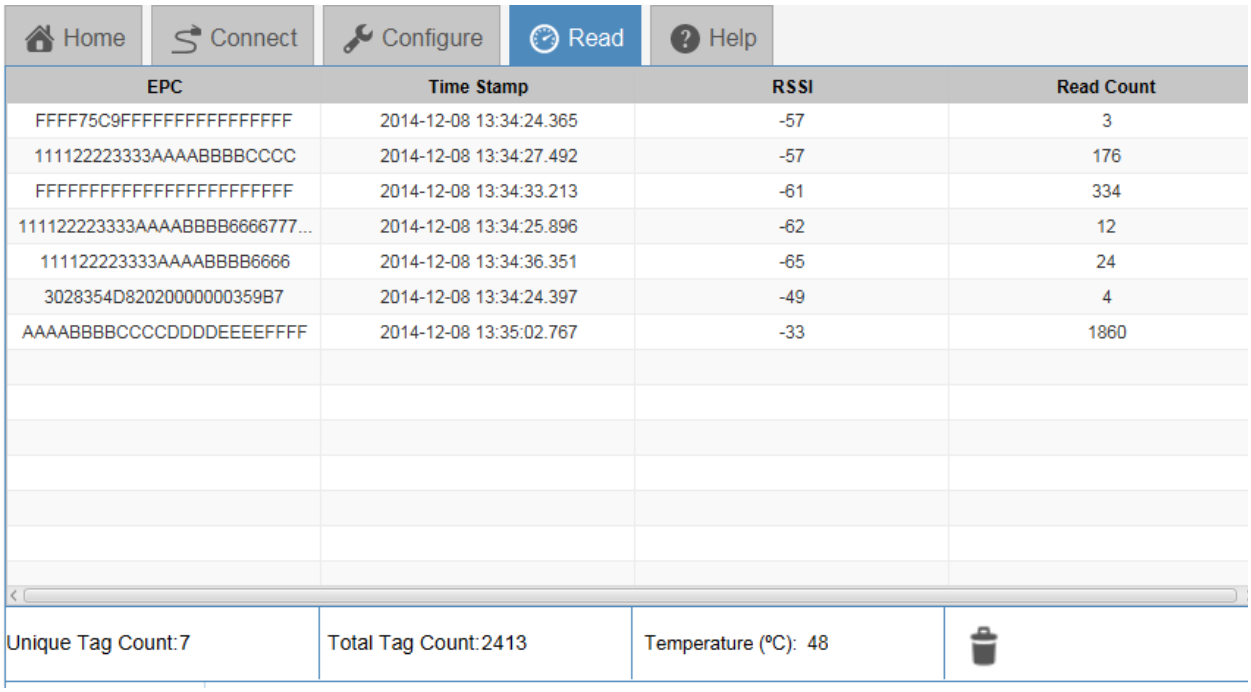
Read

Once an autonomous configuration has been applied and you've chosen to start reading you can use the Read tab to display the output. This function is simply listening on the already established connection to the reader and displaying the results.

The Read tab has no control over the connected reader, it is display only. In order to stop reading you must:

- Turn off the reader (cut power or ground the SHUTDOWN line)
- If a GPI read trigger is enabled, set the GPI to LOW.

Figure 4: **Default Tag Results Display**



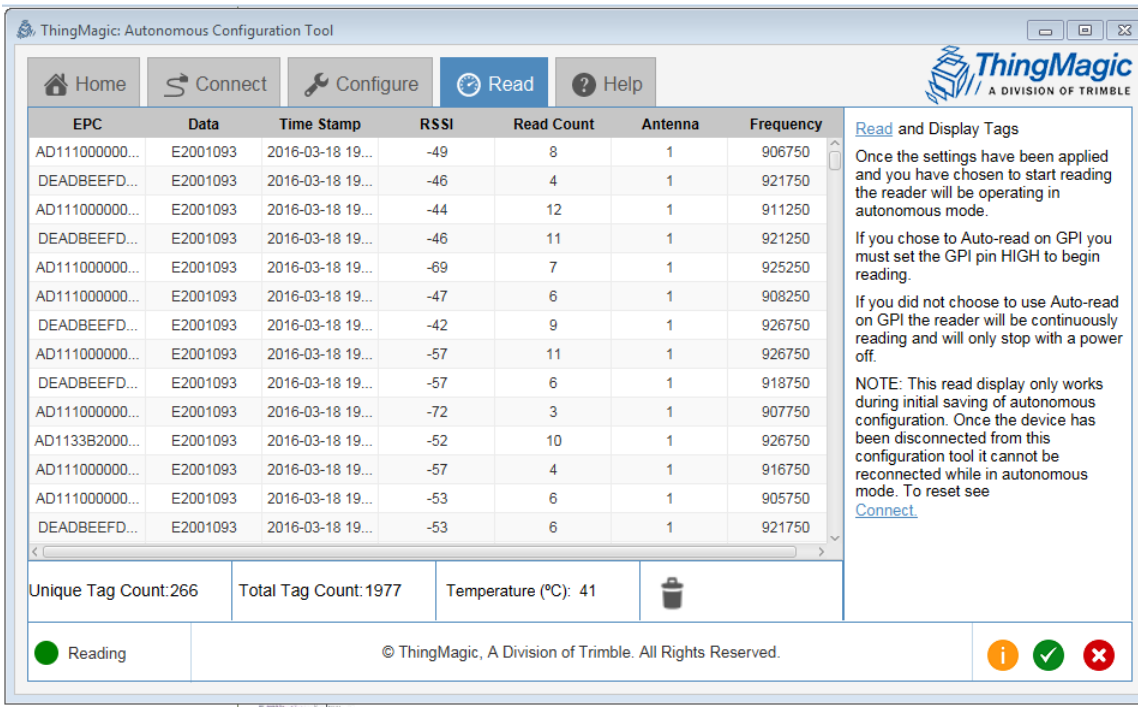
Tag Results Grid

As shown in *Figure 4*, the default displayed tag results show the following columns of data:

- **EPC** - The EPC ID of the tag, typically 96-bits of data shown in Hex.

- **Timestamp** - The timestamp when the tag was last seen during this query.
- **RSSI** - The receive signal strength of the tag response, in dBm.
- **ReadCount** - The number of times the tag was read on [Antenna].

In addition to the above fields, additional fields can be displayed using the Display Options settings. For example, here is a display that adds embedded data, the antenna, and the channel.



When in GPI Triggered mode and GPI is low a keep-alive message is sent periodically containing the module temperature. The temperature is displayed at the bottom of the Read tab.

Note

This read display only works during initial saving of autonomous configuration. Once the device has been disconnected from ACT it cannot be reconnected to the Read display screen while in autonomous mode. If adjustments need to be made to the auto-config behavior you should return to the Connect steps. Otherwise, at this point you should start “catching” the data with your own application, see Processing Data from the Reader for details.

Processing Data from the Reader

Sample Response Processing Code

Strictly speaking, once in autonomous mode the reader no longer requires a host API. However, for practical purposes the data coming from the reader still needs to be parsed. In the current release data is only available in the Reader Data Packet Format which can be difficult to parse, depending on the specific configuration chosen. To facility handling the data, sample parsing code has been provided in C, C# and Java.

See the README.TXT in autoread-1.25.x.x package for steps to run each parsing sample.

Reader Data Packet Format

The following table defines the format of the generic Response Packet sent from the reader to the host.

The fields are summarized in the following table.:

Table 4: **Reader Message Packet Format**

Field	Length	Description
Header (Hdr)	1 byte	Defines the start of the packet. Equal to 0xFF
Data Length (Len)	1 byte	Defines the length, *M*, of the data field contained in the packet. Length can be 0 – 248 bytes
OpCode	1 byte	Indicates the type of message to follow
Status Word	2 bytes	• Specifies the status of the last command, Successful = **0x0000**, else it contains a fault code. • For Autonomous Operation **0x0400** indicates a read-cycle start message...also acts as a heartbeat when the reader is reading but no tags are in the field.
Data	*M* bytes (0 to 248)	Defines the binary data returned by the reader in response to a command. This could, for example, represent data read from a transponder. Data length, *M*, can be a minimum of 0 and a maximum of 248 bytes.
CRC-16 Checksum	2 bytes	CRC-16 checksum (high order byte first). CRC polynomial is CCITT CRC-16, with a preload of 0xFFFF. This does not fully specify the operation of the CRC, see CCITT CRC-16 Calculation.

Message Types Sent By Reader

When operating in autonomous mode there are three different messages that the reader will send: Tag Read Data, Temperature Status (acts as keep-alive in GPI triggered read mode) and Read-cycle reset/Keep-alive. The message types can be differentiated by their *Data Length* field.

These messages contain the following data:

Tag Read Data Message

Table 5: **Tag Read Data Packet Format**

Field	Value	Description
Header (Hdr)	0xFF	Defines the start of the packet.
Data Length (Len)	[1 byte]	Defines the length, *M*, of the data field contained in the packet. **Length value will be 0x10 to 0xFE**
OpCode	0x22	Autonomous reader message
Status Word	0x0000	Since Tag Read Data messages only occur with successful tag reads the status for these should always be **0x0000**
RFU	[7 bytes]	Reserved for future use. Data can be ignored
RSSI	[1 byte]	Return Signal Strength Indicator
Antenna ID	[1 byte]	Antenna ID, 4 MSBs for TX and 4 LSBs for RX
Frequency	[3 bytes]	Frequency in kHz at which the tag was read
Timestamp	[4 bytes]	Millisecond offset from last Read-Cycle Reset/Keep-Alive when tag was read.
Phase	[2 bytes]	Phase of signal tag was read at (0-180)
Protocol ID	0x05	Protocol ID of tag read 0x05 = Gen2/ISO18000/6C
Embedded Tag Data Length	[2 byte]	Length, in bits, of the embbeded tag data read for this tag. This value indicates how many bytes (ceiling[bits/8]), up to 32, will follow. *Example:* if the value is 0x1D (29) then 4 bytes will follow: 29/4 = 3.625 -- ceiling(3.625) = **4**.
Tag Data	[N bytes]	Number of bytes of tag data as specified in *Tag Data Length*

RFU	[1 byte]	Reserved for future use. Data can be ignored
EPC Length	[2 bytes]	Number of bits in EPC including PC and CRC bits
PC Word	[2 bytes]	Tag EPC Protocol Control bits
EPC ID	[N bytes]	Tag EPC.
Tag CRC	[2 bytes]	Tag EPC CRC
CRC-16 Checksum	[2 bytes]	See CCITT CRC-16 Calculation.

Message Example

A sample Tag Read Data message with no embedded read data:

FF	28	22	00 00	10 03 1B 01 FF 01 01	E4	11	0E 1A 5A	00 00 00 10	00 A0	05
SOH	Length	OpCode	Status Word	RFU	RSSI	Tx/Rx Antenna	Frequency	Timestamp	Phase	Protocol

00 00	01	00 80	30 30	AA AA BB BB CC CC DD DD EE EE FF FF	71 87	6E 3F
Embedded Data Length	RFU	EPC Length	PC Word	EPC ID	Tag CRC	Message CRC

A sample Tag Read Data message with embedded read data:

FF	2A	22	00 00	10 03 1F 01 FF 01 01	EB	22	0E 0A BA	00 00 00 11	00 5F	05
SOH	Length	OpCode	Status Word	RFU	RSSI	Tx/Rx Antenna	Frequency	Timestamp	Phase	Protocol

00 60	E2 80 11 04 20 00 22 8D FC 99 00 00	01	00 30	08 30	44 44	F2 11	99 AA
Embedded Data Length	Embedded Tag Read Data	RFU	EPC Length	PC Word	EPC ID	Tag CRC	Message CRC

Temperature Status Message

A Temperature Status Message will be returned approximately every 250ms (this will vary based on the Region setting used as it depends on the internal frequency hopping algorithm which is dependent on the region of operation).

When using Read On GPI Trigger mode the temperature status message is used as a keep-alive when GPI is low (reading is turned off).

Table 6: **Temperature Status Packet Format**

Field	Value	Description
Header (Hdr)	0xFF	Defines the start of the packet.
Data Length (Len)	0x0A	Defines the length, *M*, of the data field contained in the packet. Length can be 0 – 248 bytes
OpCode	0x22	Autonomous reader message
Status Word	0x0000	Should not receive non-0x0000 status.
RFU	[9 bytes]	Reserved for future use. Data can be ignored
Temperature	[1 byte]	Internal temperature of the module in degrees Celcius. Note: If temperature exceeds 85C the module will begin throttling reading and return 0x0504 status in Read-Cycle Reset/Keep-Alive messages.
CRC-16 Checksum	[2 bytes]	See CCITT CRC-16 Calculation.

Message Example

A sample temperature status message indicating a temperature of 29C:

FF	0A	22	00	00	00031b028200820001	1D	B7 71
SOH	Length	OpCode	Status Word		RFU	Temperature	CRC

Read-Cycle Reset/Keep-Alive

The Read-Cycle Reset/Keep-Alive message are sent once per second and always have a non-0x0000 Status Word, with 0x0400 indicating normal.

Table 7: **Read-Cycle Reset Packet Format**

Field	Value	Description
Header (Hdr)	0xFF	Defines the start of the packet.
Data Length (Len)	0x00	Cycle reset contains no additional data after the status code.
OpCode	0x22	Autonomous reader message
Status Word	[2 bytes]	• **0x0400** indicates a read-cycle start message...also acts as a heartbeat when the reader is reading but no tags are in the field. • **0x0504** indicates the maximum temperature of the module (85C as reported by Temperature Status Message) has been exceeded and reading has been temporarily throttled. • Other non-0x0000 fault code. See reader's Hardware Guide for list of fault-codes.
CRC-16 Checksum	[2 bytes]	See CCITT CRC-16 Calculation.

Message Example

A sample Read-cycle Reset/Keep-alive message indicating normal:

FF	00	22	04	00	84 E0
SOH	Length	OpCode	Status Word		CRC

A sample Read-cycle Reset/Keep-alive message indicating over-temperature, RF being throttled temporarily:

FF	00	22	05	04	85 E4
SOH	Length	OpCode	Status Word		CRC

CCITT CRC-16 Calculation

The same CRC calculation is performed on all messages from the reader. The CRC is calculated on the Data Length, Command, Status Word, and Data bytes. The header (SOH, 0xFF) is not included in the CRC.

A sample implementation of the CCITT CRC-16 algorithm can be found in the autonomous reading zip package (autoread-1.25.x.x.zip) in the corresponding subdirectory for Java, C# and C

- C#: ReceiveAutonomousReading.cs | `CalcReturnCRC()`
- Java: SerialTransport.java | `calcCrc()`
- C: receiveAutonomousReading.c | `tm_crc()`

Troubleshooting

Troubleshooting

See the Troubleshooting section of the ACT Help tab.

Collecting Diagnostic Data for ThingMagic Support

When experience problems connecting to readers or operating in autonomous mode it will be necessary to gather the following information to help diagnose the problem. This information will often be the first thing requested when reporting a problem to ThingMagic support.

1. **Reader Serial Number**: See the 2d barcode label.
2. **Firmware version** shown in About
3. Any **output data** from the reader available.
4. **Saved profile** file of settings applied.
5. **Physical Configuration**: any details available about the number and types of antennas connected, cables used, power supply, etc. and tags being used.
6. **Environment**: any details about the physical environment the reader is being used/tested in. Temperature, humidity, vehicle mounted, office, etc.