THINGMAGIC ELARA USER GUIDE

COPYRIGHT INFORMATION

JADAK
A Novanta Company

TABLE OF CONTENTS

LIST OF TABLES

JADAK
A Novanta Company

REVISION HISTORY

Date	Version	Description
3/20/2019	TM_ELARA-UG Rev 03202019	• Initial Release.

Introduction

The Elara is a ThingMagic finished UHF RFID reader. It is an enclosed reader with the EL6e smart module mated to an internal antenna.

The Elara is built around the ThingMagic Nano module. A processor alongside the Nano module will provide a simplified interface with the ability to mimic other readers, such as keyboard-emulating barcode readers. It has a flexible platform and is capable of large set of tasks. Its modular design allows for extension without redesigning the entire system.

This document is for hardware designers and software developers. The ThingMagic Configuration Tool (TCT) application and code samples to control the Elara reader from an external host are written using the Java programming language.

Release Notes

This document applies to Elara reader with firmware version 1.D.1 or later. It explains how to set up Elara readers, how to configure them through RAIN interface and ThingMagic Configuration Tool (TCT), and how to configure them for keyboard operation. See the corresponding Elara Firmware Release Notes for operational differences from what is in this User Guide specific to a firmware version. Any new features or known issues that are not reflected in this User Guide will be found in the Release Notes for the version of firmware you are using. Release notes can be downloaded from the same web site where you obtained this document.

Hardware Overview

The Elara UHF RFID reader is a UHF Gen2V2 RFID reader that is controlled, responds, and is powered via USB 2.0. An integrated antenna provides the air interface to 900 MHz UHF RFID tags. Specific SKUs/part numbers provide operation at the FCC 900 MHz ISM band or at the 865 MHz European self-certified band.

The USB 2.0 input connection from a USB host source powers the Elara with a nominal 5 volts and up to 500mA average current. The 5 volt input is regulated to 3.3 volts via a switching regulator at a nominal fixed 1.5 MHz switching frequency.

The USB 2.0 connects to a microprocessor in the Elara that handles USB inbound and outbound data communication. The microprocessor is powered by the 3.3 volt switching regulator. The microprocessor connects via 3.3 volt logic to a modularly certified UHF RFID reader.

The UHF RFID module used in the Elara is a JADAK ThingMagic Nano reader with FCC certification QV5MERCURY6N. It is controlled by the microprocessor. It is powered by the 5 volt USB, conditioned and filtered to maintain USB power compatibility. The Nano performs all the Gen2V2 RFID read operations. To collect RFID tag information, the Nano reader transmits 900 MHz signals to an antenna integrated in the Elara. The connection to the antenna is via a short microstrip and short coaxial cable.

The antenna is a ceramic patch antenna mounted on its own ground plane. Its output is circularly polarized. It radiates 900 MHz with a bore site approximately vertical to the upper flat side of the Elara enclosure. Antenna signals go through the enclosure to RFID tags up to two meters away. Those tags back scatter signals according to the Gen2V2 protocol. The antenna then captures those backscattered signals and returns them to the Nano on the same cable used for the transmit signal.

The Nano receives the tag back scatter signals and decodes tag data present in them. The Nano buffers and forwards the tag data back to the microprocessor.

The microprocessor accepts the decoded tag data, reformats it for USB, and sends the data via USB to the USB host. The Elara primary mode of data transfer is "streamed tag data" where any tag in the antenna field is read and the data is sent immediately to the host. The Elara also conforms to the RFID RAIN protocol for additional RFID operations.

Hardware Options

Antennas

The Elara RFID reader has an internal EU (Europe) or NA (North America) range antenna inside its enclosure. The reader is aware of whether it was manufactured for the EU or NA range and will only present region options that fall within the appropriate range.

Protocols

The Elara reader supports only Gen2 protocol tags.

Supported Regions

The Elara reader supports all regions of operation supported by the Nano module, but is configured in the factory to operate in either the NA or EU range. The EU range supports regions within the 865 to 869 MHz frequency band and the NA range supports regions within the 915 to 928 MHz frequency band.

JADAK
A Novanta Company

The following is the list of regions supported by each range and their abbreviation in the RAIN protocol setting:

NA Range

- FCC (NA2, NA3)
- Argentina, Brazil (AR)
- Australia (AU)
- Bangladesh (BD)
- China (PRC)
- Indonesia (ID)
- Japan (JP, JP2, JP3)
- Korea (KR2)
- Macao (MO)
- Malaysia (MY)
- New Zealand (NZ)
- Philippines (PH)
- Singapore (SG)
- Taiwan (TW)
- Thailand (TH)

EU Range

- ETSI (EU3)
- Hong Kong (HK)
- India (IN)
- Russia (RU)
- Vietnam (VN)

Regions Range

Frequency Range	Frequency Band (MHz)	Default Region	Supported Regions Within the Range
NA	915 to 928	NA2 (TMR_REGION_NA2)	FCC (NA, SA), Argentina, Australia, Bangladesh, China, Indonesia, Japan, Korea, Macao, Malaysia, New Zealand, Philippines, Singapore, Taiwan, Thailand
EU	865 to 869	EU3 (TMR_REGION_EU3)	ETSI (EU), Hong Kong, India, Russia, Vietnam

The Elara reader allows you to configure only the regional settings that fall within the configured range (NA or EU). The host must configure a region of operation before tags can be read.

Countries not listed either have adopted the FCC or EU standards as-is or they have adopted these standards with an abbreviated frequency range. If the latter, JADAK provides a way to redefine the channels within a regional setting without changing any other characteristics of that region.

Different regions can be configured within the NA/EU region range on an Elara reader using the ThingMagic Configuration Tool (TCT) or by sending ""FreqReg" RAIN command through the RAIN Command Interface (RCI).

TCT Command to Configure Region
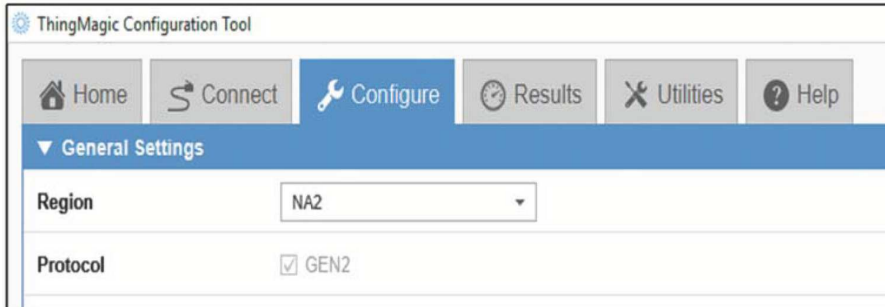
RAIN Commands to Configure Region

RAIN Command	Purpose
{"Cmd":"SetCfg", "FreqReg":"NA2"} (or) {"Cmd":"SetCfg", "FreqReg":"NA3"} (or) {"Cmd":"SetCfg", "FreqReg":"AR"} (or) {"Cmd":"SetCfg", "FreqReg":"AU"} (or) {"Cmd":"SetCfg", "FreqReg":"BD"} (or) {"Cmd":"SetCfg", "FreqReg":"JP"} (or) {"Cmd":"SetCfg", "FreqReg":"JP2"} (or) {"Cmd":"SetCfg", "FreqReg":"JP3"} (or) {"Cmd":"SetCfg", "FreqReg":"ID"} (or) {"Cmd":"SetCfg", "FreqReg":"PRC"} (or) {"Cmd":"SetCfg", "FreqReg":"KR2"} (or) {"Cmd":"SetCfg", "FreqReg":"MO"} (or) {"Cmd":"SetCfg", "FreqReg":"MY"} (or) {"Cmd":"SetCfg", "FreqReg":"NZ"} (or) {"Cmd":"SetCfg", "FreqReg":"PH"} (or) {"Cmd":"SetCfg", "FreqReg":"TW"} (or) {"Cmd":"SetCfg", "FreqReg":"SG"} (or) {"Cmd":"SetCfg", "FreqReg":"TH"}	Configures a region in NA region range: FCC (NA, SA), Argentina, Australia, Bangladesh, China, Indonesia, Japan, Korea, Macao, Malaysia, New Zealand, Philippines, Singapore, Taiwan, Thailand
{"Cmd":"SetCfg", "FreqReg":"EU3"} (or) {"Cmd":"SetCfg", "FreqReg":"IN"} (or) {"Cmd":"SetCfg", "FreqReg":"RU"} (or) {"Cmd":"SetCfg", "FreqReg":"VN"} {"Cmd":"SetCfg", "FreqReg":"HK"}	Configures a region in the EU region range: ETSI (EU), Hong Kong, India, Russia, Vietnam

![JADAK - A Novanta Company]

Interfaces

The Elara reader only supports a USB interface.

USB Reader

The Elara USB reader doesn't need an external power supply. The power transmitted through the USB cable from a host PC is sufficient to power on the reader. In instances where a single USB port is unable to provide sufficient power, a USB Y-cable adapter can be used to connect the Elara to two USB ports: one for power and one for communication.

HID (keyboard) and serial modes and are enabled by default. When you connect the Elara USB cable to a host PC, HID will be detected under the **Human Interface Devices** section and the CDC port will be detected under the **Ports** section of the Device Manager. The Elara can be configured to either HID mode or serial mode.

The Elara reader supports a keyboard emulation interface (sometimes called a keyboard wedge) and displays data when it is configured to HID mode. The keyboard emulator is used to transfer gathered data to PC hosts without need of a special driver. Management of the reader is not supported through the keyboard interface.

When the reader is connected via a USB serial port, you can connect, configure settings, and read tags using the ThingMagic Configuration Tool (TCT) or by sending RAIN commands through the RAIN Command Interface (RCI).

Keyboard (HID) Interface

The Elara reader connects on the HID port by default and activates a keyboard emulation interface and reports tag records. You must stop the read to issue RAIN commands through the RAIN Command Interface (RCI).

Use one of the following methods to suspend the read:

- Two quick presses of the Caps Lock key on the keyboard.
  Note: Two quick presses of the Caps Lock key a second time causes the reader to begin reading again.
- Connect the reader using the USB COM port using the ThingMagic Configuration Tool (TCT). Turn off the HID interface, and **Set** the configuration.
- Enter the RAIN command that suspends reading.

NOTE: **The green LED stops flashing when reading is suspended. The output from the HID port may continue if tags have been read, but the data has not yet transferred via the interface.**

USB Serial Interface

The Elara reader supports the RAIN Command Interface (RCI) in serial mode. The command-response interface protocol allows you to configure different settings on the reader.

The RCI interface provides the ability to receive data from the reader and issue commands to control it. The Elara reader is accessed using a COM port emulation connection.

The HID/Serial mode can be configured using the ThingMagic Configuration Tool (TCT) or by using RAIN commands.

RAIN Commands to Enable/Disable HID Mode

RAIN Command	Purpose
`{"Cmd":"SetCfg","_USBKBEnable":True}`	Enables HID (Keyboard Wedge) mode
`{"Cmd":"SetCfg","_USBKBEnable":false}`	Disables HID (Keyboard Wedge) mode

RAIN Commands to Start and Stop Read

RAIN Command	Purpose
`{"Cmd":"StartRZ"}`	Start reading tags
`{"Cmd":"StopRZ"}`	Stop reading tags

Warning: **Keyboard output must be suspended in one of the suggested ways before a connection to the RAIN interface is attempted. Otherwise, the PC will interpret the tag data coming from the keyboard interface as commands that are to be entered into the RAIN interface, resulting in many error messages being returned.**

SKU Ordering

SKUs

SKU	Packaging	Interface	Frequency Range
PLT-RFID-EL6-UHB-4-USB	Enclosed USB Reader	USB	NA
PLT-RFID-EL6-ULB-4-USB	Enclosed USB Reader	USB	EU

Power Requirements

When Transmitting (Duty Cycle and RF Power Level)

USB Power consumption: 3.5 Watts peak, less than 2.5 Watts average with duty cycle 50%.

4.5 to 10 Volt power in power consumption: 4.7 Watts

10 to 26 Volt power in power consumption: 4.3 Watts.

When Idle (Settings)

USB power idle consumption: 0.7 Watts.

USB Characteristics

USB Start up current: 275 mA after 120 second power down.

Tag reading input power: Less than 2.5 Watts average with duty cycle 50%.

Microsoft® Windows® 64-bit Operating System

The ThingMagic Configuration Tool (TCT) only supports a Windows 64-bit operating system.

NOTE: In Windows 7 and lower operating systems, the Elara COM port driver (elara.inf) needs to be installed manually. Windows 10 automatically installs and detects using a generic driver.

JRE 1.8 or later

Java 1.8 or later needs to be installed on the host PC to work with the ThingMagic Configuration Tool (TCT) and to run Java samples.

User Indicators

LEDs

The Elara operational mode/status is indicated by a bi-color LED. The Elara reader has 2 bi-color LEDs:

Status LED (red/green)

Green, Green Flashing: Normal operations like ON, READY and TAG READING state.

Red, Red Flashing: No valid application image, invalid hardware information, or while booting.

Status LED Indicators

LED	Solid	Flashing
Red	Booting	Booted, but not ready to read (i.e., invalid application image)
Green	Ready, but not reading	Reading tags

NOTE: Red and green LEDs are never on simultaneously.

User LED (yellow/blue)

Yellow, Yellow Flashing: Blinks whenever a transient error occurs (i.e., for any event that produces an error or warning message on the command interface). When the LED blinks, it stays on for 250ms. The yellow user LED glows solid when a fatal error occurs (i.e., hardware failure makes it impossible to read), indicating that user intervention is required.

Blue Flashing: Blinks for every successful tag operation. If the purpose of the workflow is to read tags, the blue LED flashes for each successful read. If the purpose of the workflow it to write to tags, the blue LED flashes when the write is successful and the yellow LED will flash if the write is unsuccessful. No tags detected is a normal occurrence and not worthy of indication.

User LED Indicators

LED	Solid	Flashing
Yellow	Fatal error (i.e., region not set)	Transient error (i.e., tag write failed)
Blue		Tag operation successful (i.e., tag write succeeded)

NOTE: Yellow and blue LEDs are never on simultaneously.

Error Indicators

Errors are indicated to the end user via the user LED. The user LED blinking or flashing indicates soft faults or recoverable errors. The user LED glowing solid indicates hard faults or non-recoverable errors.

Error Indicators

Error Name	Error Type	LED
Tag write fail	Soft fault	Blink
Tag read fail	Soft fault	Blink
Lock operation fail	Soft fault	Blink
Permalock operation fail	Soft fault	Blink
Over temperature	Soft fault	Blink
Watchdog fault	Soft fault	Blink
Invalid command	Soft fault	Blink
No/Invalid application image	Hard fault	Solid
Insufficient DC power	Hard fault	Solid
Nano module interface fail	Hard fault	Solid
High return loss	Hard fault	Solid
Over current	Hard fault	Solid
Processor exceptions	Hard fault	Solid

Audio Indicators

The beeper indicates successful tag reads as well as successful and unsuccessful tag operations. Beepers are configurable to the four following levels:

• Mute

• Low (Default)

• Medium

• High

By default, the beeper is configured to low volume. Configure the beeper to other levels using the ThingMagic Configuration Tool (TCT) or by sending RAIN commands through the RAIN Command Interface (RCI) channel.

Audio Indicator RAIN Commands

RAIN Command	Purpose
{"Cmd":"SetCfg","_AudioVolume":"MUTE"}	Mute the beeper
{"Cmd":"SetCfg","_AudioVolume":"LOW"}	Set beeper to low
{"Cmd":"SetCfg","_AudioVolume":"MED"}	Set beeper to medium
{"Cmd":"SetCfg","_AudioVolume":"HIGH"}	Set beeper to high

Software Overview

Host Software

No software installation is needed for basic or default operation. But the following software is required if you wish to configure the Elara.

ThingMagic Configuration Tool (TCT)

The ThingMagic Configuration Tool is a desktop application developed using Java language and built in javafx technology. TCT is used to connect, configure, and read tags. TCT is only supported on the Windows 64-bit operating system.

Refer to ThingMagic Configuration Tool, for more information.

Java Code Samples

You may also perform a few operations on the Elara reader using Java. The following Java code samples can be executed using an IDE (NetBeans, Eclipse) through a terminal. You must install JRE 1, 8 or later to work with the following code samples.

SingleTagRead.java

This demonstrates read functionality against an HDR workflow. It reports one closest tag per second.

BulkTagRead.java

This demonstrates read functionality against a MONITOR workflow. It reports tag records for every second.

FirmwareUpdate.java

This allows you to update firmware on the Elara reader. This code sample takes approximately 30 seconds to update the firmware on the reader. You must provide the firmware path.

TagCommissioning.java

This performs a write operation on the closest tag in the field.

TagUpdate.java

This performs a write operation on all available tags in the field.

Serial Emulator (Optional)

You can also control the Elara reader by sending RAIN commands manually either through a third party serial emulator such as Putty or Dock light, or through a customized application. Refer to RAIN Command Interface, beginning on page 25, for more information.

Firmware Update

You can update the firmware on all variants of the Elara reader using the ThingMagic Configuration Tool (TCT) or Java sample code.

JADAK
A Novanta Company

Update Firmware Using TCT

1. Download and install TCT using the following steps.

    •   Use a web browser to navigate to https://www.jadaktech.com/documentation/.
    •   Download the SDK (elara_sdk-ZOLA-01.0D.01.zip), available under the Elara folder.
    •   Once downloaded, extract the zip archive to the desired location.
    •   Find the ThingMagic Configuration Tool executable and double click on it to install to the host PC.
    •   Follow the steps for your OS.

2. Open the ThingMagic Configuration Tool (TCT) and navigate to the **Connect** page.

3. Select the proper COM port under **Available Devices** and click on **Connect**.

4. Navigate to the **Utilities** tab.

5. Click on **Browse** and go to the SDK extracted file location. Select the **elara_app-01.0D.00.xx.bin** file.

6. Click on **Update**. The firmware update takes approximately 35-40 seconds and redirects you to the **Connect** page when complete.

NOTE: Once the firmware update process is initiated, the existing firmware is erased. To revert to the old firmware, repeat the firmware update process with an old firmware image file.

Communication Protocol

Workflow Overview

The RAIN interface specification introduces the concept of Modes, where a single setting can modify a range of reader protocol settings. JADAK has extended this concept to define Autonomous Workflows where a wider range of settings, and reader behavior, can be changed with a single command.

By default, the Elara board is connected to the PC using a USB cable and is detected by the PC as a keyboard device. For basic or default operation, no driver or software installation is needed. You can set the reader into different modes using Java code samples or using the ThingMagic Configuration Tool.

There are 4 following autonomous workflows you can configure on the Elara reader:

- HDR (single tag read)
- Monitor (bulk tag read, default mode)
- Tag Commissioning
- Tag Update

HDR

This mode is similar to a barcode scanner and is used for applications such as inventory or equipment logging and attendee registration at events.

HDR mode is also known as single tag read. With HDR mode, The Elara reader may read many tags but reports the closest tag every second. The reader reports the tag again if the tag is moved out of range for a specified length of time and then presented to the reader again. The reader reads the next tag if the first tag is moved away from the reader and a different tag has the highest field strength.

By default, HDR mode reports EPC of tag reads irrespective of the configured data format. But you can configure the reader to report other tag read metadata information along with tag EPC.

Tag data to be read:
- EPC and time (default)
- Metadata

Output data format:
- Scan engine format (default); keyboard is QUERTY data presented in ASCII
- Verbose format
- RAIN format

User indicators:
- Application shows status LED in blinking green for tag reads.
- Application beeps for every tag report.

JADAK
A Novanta Company

Monitor

Monitor mode is the default workflow profile for all variants of the Elara reader. It is a simplified way of logging a small batch of items in close range. In this mode, the reader automatically reads all tags in range, stores all the tags read in a set time (scan delay should be configurable, default delayis 1second), reads again, and reports tag records every second. Data can be recorded in any application that accepts keyboard input.

Tag data to be read:

- EPC and time (default)
- Metadata

Output data format:

- Scan engine format (default); keyboard is QUERTY data presented in ASCII
- Verbose format
- RAIN format

User indicators:

- Application shows status LED in blinking green for tag reads.
- Application beeps for every unique tag report.

Tag Commissioning

This main use case of this mode is to perform Read Data, Write Data, Tag Lock, and Permalock tag operations on the closest tag in the reader field. This mode uses similar logic to the HDR mode to select the closest tag and then write user-specified data at a user-specified bank address location. Tag Commissioning mode allows you to write data to the different banks of a Gen2 tag.

Tag Commissioning on various banks:

- Reserved
- EPC
- TID (Supports only Read Data operation)
- User Memory

User indicators:

- Application shows the user LED in blue for approximately 250msec on a successful write operation.
- Application shows the user LED in yellow for approximately 250msec on a write failure.
- Application beeps for every report.

Read Data

This operation reads the specified bank data of the closest tag. Use the following sequence to read data in Tag Commission mode.

1. Set the mode:

```
{"Cmd":"SetCfg","Mode":"TagCommission"}
```

JADAK
A Novanta Company

2. Set the Tag op:

```
{"Cmd":"SetProf","Read":[[MemBank, StartWordAddress,
NumberOfWordsToRead, RetryLimit]]}
```

3. Execute the Tag op

```
{"Cmd":"ThisTag", "Prof":[1]}
```

Example: Read full EPC (6 words) including PC bits (1 word).

1. Set the mode:

```
{"Cmd":"SetCfg","Workflow":"TagCommission"}
```

```
{"Report":"SetCfg","ErrID":0}
```

2. Set the Tag op:

```
{"Cmd":"SetProf", "Read":[[1,1,7,0]]}
```

```
{"Report":"SetProf","ErrID":0}
```

3. Execute the Tag op:

```
{"Cmd":"ThisTag","Prof":[1]}
```

```
{"Report":"ThisTag","ErrID":0}
```

```
{"Report":"TagEvent","ErrID":0,"MB":[[1,1,":3000:aaaa:bbbb:cccc:dddd:
eeee:ffff"]]}
```

Write Data

This operation writes the data of the closest tag. Use the following sequence to write data to a memory bank in Tag Commission mode.

1. Set the mode:

```
{"Cmd":"SetCfg","Mode":"TagCommission"}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Write":[[MemBank, StartWordAddress,
NumberOfWordsToWrite, RetryLimit,["VAL","WriteData"], true]}
```

3. Execute the Tag op

```
{"Cmd":"ThisTag", "Prof":[1]}
```

Example: Write a full EPC (including PC bits).

1. Set the mode:

```
{"Cmd":"SetCfg","Workflow":"TagCommission"}
```

```
{"Report":"SetCfg","ErrID":0}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Write":[[1,1,7,0,["VAL",":3000:1111:2222:3333:4444:
5555:6666"],true]]}
```

```
{"Report":"SetProf","ErrID":0}
```

3. Execute the Tag op:

```
{"Cmd":"ThisTag","Prof":[1]}
```

```
{"Report":"ThisTag","ErrID":0}
```

Tag Lock

Tag Lock is available as an option to Tag Write, an extension of write tuple. The 7th argument of the write tuple specifies the lock method. The main use case of this method is to lock the specified bank temporarily or permanently after writing data into the bank.

Gen2 tags require an access password for lock operation. By default, both the tag and the reader have 0x0000 access password.

Tag Lock Values

"" (Empty string) or **"NOLOCK"**	Take no action. Leave memory bank unlocked. The Lock Method argument may also be omitted entirely (i.e., provide only 6 arguments instead of 7). This also means "no action."
"SECURE"	Locks the specified memory bank if it was successfully written.
"PERMALOCK"	Permanently locks the specified memory bank if it was successfully written.

Use the following sequence to add data to a memory bank in Tag Commission mode.

1. Set the mode:

```
{"Cmd":"SetCfg","Mode":"TagCommission"}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Write":[[MemBank, StartWordAddress,
NumberOfWordsToWrite, RetryLimit,["VAL","WriteData"], true, "SECURE/
PERMALOCK]}
```

3. Execute the Tag op:

```
{"Cmd":"ThisTag", "Prof":[1]}
```

Example: Write a full EPC (including PC bits), then lock it.

1. Set the mode:

```
{"Cmd":"SetCfg","Workflow":"TagCommission"}
```

```
{"Report":"SetCfg","ErrID":0}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Write":[[1,1,7,0,["VAL",":3000:AAAA:BBBB:CCCC:DDDD:
EEEE:FFFF"],true,"SECURE"]]}
```

```
{"Report":"SetProf","ErrID":0}
```

3. Execute the Tag op:

```
{"Cmd":"ThisTag","Prof":[1]}
```

```
{"Report":"ThisTag","ErrID":0}
```

Example: Write a full EPC (including PC bits), then permanently lock it.

1. Set the mode:

```
{"Cmd":"SetCfg","Workflow":"TagCommission"}
```

```
{"Report":"SetCfg","ErrID":0}
```

2. Set the Tag op:

```
Set Tag op(lock).
```

```
{"Cmd":"SetProf","Write":[[1,1,7,0,["VAL",":3000:AAAA:BBBB:CCCC:DDDD:
EEEE:FFFF"],true,"PERMALOCK"]]}
```

```
{"Report":"SetProf","ErrID":0}
```

3. Set the password:

```
{"Cmd":"SetRZ","Password":":11223344"}
```

```
{"Report":"SetRZ","ErrID":0}
```

4. Execute the Tag op:

```
{"Cmd":"ThisTag","Prof":[1]}
```

```
{"Report":"ThisTag","ErrID":0}
```

NOTE: You can't unlock any memory bank of a tag in Tag Commission mode. Here, the NOLOCK option just works as Write Tag operation.

Tag Update

Tag Update is like tag commissioning, but the use case is for managing item information by updating the tag user memory, if connected to a database, then also updating the host. This is used for issuing and tracking event access passes, monitoring use of consumable items, and logging maintenance events. The encoded data is system-generated and does not require manual data input. Tag information is locked after the operation is successful. (You may configure this locking configuration.)

Tag Update operation on various banks:

- Reserved
- EPC
- User Memory

User indicators:

* Application shows the user LED in blue for approximately 250msec on a successful write operation.
* Application shows the user LED in yellow for approximately 250msec on a write failure.
* Application beeps for every report.

Read Data

This operation reads the specified bank data of all available tags in the field. Use the following sequence to read data in Tag Update mode.

1. Set the mode:

```
{"Cmd":"SetCfg","Mode":"TagUpdate"}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Read":[[MemBank, StartWordAddress, NumberOfWordsToRead, RetryLimit]]}
```

3. Execute the Tag op

```
{"Cmd":"ThisTag", "Prof":[1]}
```

Example: Read full EPC (6 words) including PC bits (1 word).

1. Set the mode:

```
{"Cmd":"SetCfg","Workflow":"TagUpdate"}

{"Report":"SetCfg","ErrID":0}
```

2. Set the Tag op:

```
{"Cmd":"SetProf", "Read":[[1,1,7,0]]}

{"Report":"SetProf","ErrID":0}
```

3. Start the read:

```
{"Cmd":"StartRZ"}

{"Report":"StartRZ","ErrID":0}

{"Report":"TagEvent","ErrID":0,"EPC":"aaaabbbbccccddddeeeeffff","TID"
:"0133f1000f19625d03030013070044ebbffffdc50","MB":[[1,1,":3000:aaaa:bb
bb:cccc:dddd:eeee:ffff"]],"DT":"2010-1-1T00:06:13.000Z"}
```

Write Data

This operation writes data to the specified bank whenever a tag is available in the read field. Use the following sequence to write data to a memory bank in Tag Update mode.

1. Set the mode:

```
{"Cmd":"SetCfg","Mode":"TagUpdate"}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Write":[[MemBank, StartWordAddress,
NumberOfWordsToWrite, RetryLimit,["VAL","WriteData"], true]]}
```

3. Execute the Tag op

```
{"Cmd":"ThisTag", "Prof":[1]}
```

Example: Write a full EPC (including PC bits).

1. Set the mode:

```
{"Cmd":"SetCfg","Workflow":"TagUpdate"}
```

```
{"Report":"SetCfg","ErrID":0}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Write":[[1,1,7,0,["VAL",":3000:AAAA:BBBB:CCCC:DDDD:
EEEE:FFFF"], true,""]]}
```

```
{"Report":"SetProf","ErrID":0}
```

3. Start the read:

```
{"Cmd":"StartRZ"}
```

```
{"Report":"StartRZ","ErrID":0}
```

```
{"Report":"TagEvent","ErrID":0,"EPC":"111122223333444455556666","TID"
:"0133f1000f19625d0303013130044ebbffffdc50","MB":[[1,1,7,["VAL",":300
0:aaaa:bbbb:cccc:dddd:eeee:ffff"]]],"DT":"2010-1-1T00:43:03.000Z"}
```

Tag Lock

Tag Lock is available as an option to Tag Write, an extension of write tuple. The 7th argument of the write tuple specifies the lock method. The main use case of this method is to unlock, lock, or permanently lock the specified bank.

In Tag Update mode, if the tag is already locked, the reader validates the access password in every stage. Otherwise, it validates only for an actual operation.

1. Unlocks the tag.

2. Writes the data to the specified bank.

3. Performs the specified NOLOCK/SECURE/PERMALOCK operation.

Tag Lock Values

"" (Empty string) or **"NOLOCK"**	Take no action. Writes the data to the specified memory bank. The Lock Method argument may also be omitted entirely (i.e., provide only 6 arguments instead of 7). This also means "no action."
"SECURE"	Locks the specified memory bank if it was successfully written.
"PERMALOCK"	Permanently locks the specified memory bank if it was successfully written.

Use the following sequence to write data to a memory bank in Tag Update mode.

1. Set the mode:

```
{"Cmd":"SetCfg","Mode":"TagUpdate"}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Write":[[MemBank, StartWordAddress,
NumberOfWordsToWrite, RetryLimit,["VAL","WriteData"], true, "NOLOCK/
SECURE/PERMALOCK]}
```

3. Execute the Tag op:

```
{"Cmd":"ThisTag", "Prof":[1]}
```

Example: Write a full EPC (including PC bits), then lock it.

1. Set the mode:

```
{"Cmd":"SetCfg","Mode":"TagUpdate"}
```

```
{"Report":"SetCfg","ErrID":0}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Write":[[1,1,7,0, ["VAL",
":3000:1111:2222:3333:4444:5555:6666"], true,"SECURE"]]}
```

```
{"Report":"SetProf","ErrID":0}
```

3. Set the password:

```
{"Cmd":"SetRZ","Password":":11223344"}
```

```
{"Report":"SetRZ","ErrID":0}
```

4. Start the read:

```
{"Cmd":"StartRZ"}
```

```
{"Report":"StartRZ","ErrID":0}
```

```
{"Report":"TagEvent","ErrID":0,"EPC":"aaaabbbbccccddddeeeeffff","TID"
:"0139f1000f1961c1031e013730045ffbffffdc50","MB":[[1,1,7,["VAL",":300
0:1111:2222:3333:4444:5555:6666"]]],"DT":"2010-1-1T00:33:02.000Z"}
```

Example: Write a full EPC (including PC bits), then unlock it.

1. Set the mode:

```
{"Cmd":"SetCfg","Mode":"TagUpdate"}
```

```
{"Report":"SetCfg","ErrID":0}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Write":[1,1,7,0, ["VAL",
":3000:AAAA:BBBB:CCCC:DDDD:EEEE:FFFF"], true,"NOLOCK"]]}
```

```
{"Report":"SetProf","ErrID":0}
```

3. Set the password:

```
{"Cmd":"SetRZ","Password":":11223344"}
```

```
{"Report":"SetRZ","ErrID":0}
```

4. Start the read:

```
{"Cmd":"StartRZ"}
```

```
{"Report":"StartRZ","ErrID":0}
```

```
{"Report":"TagEvent","ErrID":0,"EPC":"111122223333444455556666","TID"
:"0139f1000f1961c1031e013670045bfbffffdc50","MB":[[1,1,7,["VAL",":300
0:aaaa:bbbb:cccc:dddd:eeee:ffff"]]],"DT":"2010-1-1T00:36:59.000Z"}
```

Example: Write a full EPC (including PC bits), then permanently lock it.

1. Set the mode:

```
{"Cmd":"SetCfg","Workflow":"TagUpdate"}
```

```
{"Report":"SetCfg","ErrID":0}
```

2. Set the Tag op:

```
{"Cmd":"SetProf","Write":[[1,1,7,0,["VAL",":3000:AAAA:BBBB:CCCC:DDDD:
EEEE:FFFF"],true,"PERMALOCK"]]}
```

```
{"Report":"SetProf","ErrID":0}
```

3. Set the password:

```
{"Cmd":"SetRZ","Password":":11223344"}
```

```
{"Report":"SetRZ","ErrID":0}
```

4. Start the read:

```
{"Cmd":"StartRZ"}

{"Report":"StartRZ","ErrID":0}

{"Report":"TagEvent","ErrID":0,"EPC":"111122223333444455556666","TID"
:"0139f1000f1961c1031e013670045bfbffffdc50","MB":[[1,1,7,["VAL",":300
0:aaaa:bbbb:cccc:dddd:eeee:ffff"]]],"DT":"2010-1-1T00:36:59.000Z"}
```

Autonomous Workflow Specifications

Item	Bulk Read	Single Read	Tag Commission	Tag Update
Output Report Rate	1 report/second	1 tag/second	N/A: Reports result	N/A: Reports result
Detection Filter	All reads	Closest reads	Closest reads	Closest reads
RSSI Threshold	Off	-40 dBm	-40 dBm	-40 dBm
Tag Priority	N/A	Least recently read	Least recently read	Least recently read
Report Time	1/second	Tag first seen	(Action results)	(Action results)
TX Read Power	20 dBm	20 dBm	20 dBm	20 dBm
TX Write Power	23 dBm	23 dBm	23 dBm	23 dBm
Gen2 Session	S1	S0	S0	S0
Gen2 Target	A	A	A	A
Gen2 Q	Dynamic	Dynamic	Dynamic	Dynamic
Gen2 Init Q	4	2	2	2
Gen2 Encoding	M4	M2	M2	M2
Duty Cycle On Time	250	1000	1000	1000
Duty Cycle Off Time	0	0	0	0

Workflow RAIN Commands

RAIN Command	Purpose
`{"Cmd":"SetCfg","Mode":"HDR"}`	To set HDR workflow
`{"Cmd":"SetCfg","Mode":"MONITOR"}`	To set MONITOR workflow
`{"Cmd":"SetCfg","Mode":"TagCommission"}`	To set Tag Commission workflow
`{"Cmd":"SetCfg","Mode":"TagUpdate"}`	To set Tag Update workflow

Apart from RAIN commands, all the above mentioned operations (read, write, lock, etc.) can also be performed in the ThingMagic Configuration Tool (TCT) under the **Results** section.

Save and Restore Configuration

You may need to use the same reader configuration multiple times. You can save the current existing reader configuration to flash memory, then restore the saved configuration on the Elara reader. This can be done using the ThingMagic Configuration Tool (TCT) and by issuing RAIN Command Interface commands.

Save Configuration

You can store the current reader configuration in permanent memory (flash). Send a **SaveFields** RAIN command through the Rain Command Interface (RCI), or click the **Save** button on the TCT Configuration page.

Restore Configuration

You can discard current active settings and apply the settings retrieved from the reader's permanent memory. Send a **ReadFields** RAIN command through the Rain Command Interface (RCI), or click the **Revert** button on the TCT Configuration page.

Load Default Configuration

You can discard current active settings and replace them with factory provided settings. Send a **DefaultFields** RAIN command through the Rain Command Interface (RCI), or click the **Load Defaults** button on the TCT Configuration page.

RAIN Commands to Save and Restore Configurations

RAIN Command	Purpose
`{"Cmd":"SaveFields"}`	Save current reader configuration in nonvolatile memory
`{"Cmd":"ReadFields"}`	Load configuration on reader from nonvolatile memory
`{"Cmd":"DefaultFields"}`	Load factory provided configuration on reader

Data Reporting Formats and Metadata

When the Elara reader is connected on the HID port, you can configure various data formats for the output string. By default, the keyboard interface is configured to EPC format, and returns only the EPC of tags read.

Data formats you can configure when the reader is connected via a HID port are:

* EPC only
* Plain metadata
* JSON metadata

EPC Only

The Elara reader returns only the tag read EPCs when data is transmitted on a HID port. You cannot apply this data format for USB CDC interface.

Plain Metadata

By default, the reader reports tag EPCs along with a timestamp on a HID port when you configure the Metadata format. But the reader has no concept of time zone. You should set the time to your local time zone before starting a read. You cannot apply this data format for USB CDC interface.

JSON Metadata

By default, the reader reports tag EPCs along with timestamp on a HID port when you configure the RAIN format. This is a simplified version of the RAIN format that is in machine-parseable format.

Keyboard Wedge Data Format RAIN Commands

RAIN Command	Purpose
`{"Cmd":"SetCfg","_KBDataFormat":"EPC"}`	Set keyboard wedge output format to EPC
`{"Cmd":"SetCfg","_KBDataFormat":"Metadata"}`	Set keyboard wedge output format to Metadata
`{"Cmd":"SetCfg","_KBDataFormat":"RAIN"}`	Set keyboard wedge output format to RAIN

When using Plain Metadata and JSON Metadata, you can enable following metadata using the ThingMagic Configuration Tool (TCT) or by sending RAIN commands. The reader reports the selected metadata information as part of tag reads.

Metadata RAIN Commands

RAIN Command	Purpose
`{"Cmd":"SetCfg","SpotAnt":true,"SpotDT":true,"SpotInvCnt":true,"SpotPhase":trtue,"SpotProf":true,"SpotRSSI":true,"SpotRZ":true}`	Enable/disable tag read metadata (standard RAIN fields)
`{"Cmd":"SetCfg","SpotFreq":true,"SpotGen2_BI":true,"SpotGen2_Q":true,"SpotGen2_LF":true,"SpotGen2_Target":true,"SpotGPIO":true,"SpotProt":true,"SpotSensor":true}`	Enable/disable tag read metadata (ThingMagic custom fields)

NOTE: Only the JSON (RAIN) data format is supported when the reader is connected on a USB COM port.

Autonomous Operation

Using automomous mode, you can enable auto reads on boot regardless of the workflow settings on the reader. To enable this mode, send a **RdrStart** RAIN command with the **ACTIVE** option set and then save it to non-volatile memory.

To disable auto reads on boot or autonomous operation, send a **RdrStart** RAIN command with the **NOACTIVE** option set and then save it to non-volatile memory.

Autonomous Read RAIN Commands

RAIN Command	Purpose
`{"Cmd":"SetCfg","RdrStart":"ACTIVE"}` `{"Cmd":"SaveFields"}`	Enables autonomous operation
`{"Cmd":"SetCfg","RdrStart":"NOTACTIVE"}` `{"Cmd":"SaveFields"}`	Disables autonomous operation

You can also can enable or disable autonomous mode using the ThingMagic Configuration Tool (TCT), in the **Configure** page under the **General settings** section.

The Elara reader has a real-time clock for reporting time stamps along with tag records, but it does not have a battery backup. By default, the reader reports the date 01 Jan 2010 and time after power on. You can change the reader date and time by sending a RAIN command or using the ThingMagic Configuration Tool (TCT). The reader then keeps track of the date and time until powered down.

Date and Time RAIN Commands

RAIN Command	Purpose
`{"Cmd":"SetCfg", "DateTime":"2018-12-18T19:30:00.000Z"}`	Change the reader date and time
`{"Cmd":"GetCfg","Fields":["DateTime"]}`	Get the reader's current date and time

By default, the Elara reports the date and time along with tag reads in Plain Metadata and JSON Metadata formats. When just Plain Metadata is configured on the reader, the Elara reports a time stamp. When just JSON Metadata is configured on the reader, the Elara reports the calendar date and time.

The Elara reader supports a heartbeat feature, which indicates the presence of the reader. When heartbeat is enabled on the reader, the reader name with the serial number is reported in configured time intervals. A RAIN command must be sent to the reader to enable or disable the heartbeat option.

Heartbeat RAIN Commands

RAIN Command	Purpose
`{"Cmd":"SetCfg","HBPeriod":1}`	Enable heartbeat for every 1 second
`{"Cmd":"SetCfg","HBPeriod":0}`	Disable heartbeats
`{"Cmd":"SetCfg","HBPeriod":10}`	Enable heartbeat for every 10 seconds

NOTE: The ThingMagic Configuration Tool (TCT) does not have an option to enable or disable heartbeat.

You can reboot the Elara reader by sending the following RAIN command to the reader through the Rain Command Interface (RCI). After the reader reboots, it applies the configuration stored in non-volatile memory on the reader.

Reboot RAIN Command

RAIN Command	Purpose
`{"Cmd":"Reboot"}`	Reboot the reader

NOTE: The ThingMagic Configuration Tool (TCT) does not have an option to reboot the reader.

Set Read Power Support

Default read and write powers are allocated for each workflow. You can change the read and write powers using a RAIN command or via the ThingMagic Configuration Tool (TCT).

Set Read and Write Power RAIN Commands

RAIN Command	Purpose
`{"Cmd":"SetRZ","ID":1,"ReadPwr":20.0}`	Set read power
`{"Cmd":"SetRZ","ID":1,"WritePwr":22.0}`	Set write power

JADAK
A Novanta Company

RAIN Command Interface

The RAIN Command Interface provides a human-readable format for controlling the reader and obtaining its output. It provides a structure for commands as well as a structure for organizing and grouping reader configuration settings. Status and error message formats and content are also defined. The RAIN Command Interface Specification allows customization at all levels beyond basic functionality (see RAIN Command Reference on page 29).

RAIN command capabilities:

- Enable using Caps Lock to toggle reading on and off
- Set the Hop table definition within a region
- Show available regions
- Turn on and off keyboard interface output
- Turn on and off RAIN Interface tag output (retains command/response capability)
- Turn off reporting of Heartbeat messages
- Set autonomous Reading
- Set Save, Default, and Restore settings
- Set entering SGTIN and GID information
- Set Tag Updates
- Set Memory Bank and location
- Set length of the Date/Time Format
- Set the Counter length and maximum allowed value
- Select the interpreted output format for data fields
- Set time for when a tag is re-reported for single tag reads

Autonomous Workflow

Each workflow has a name by which it can be selected. The Elara terminology differs from the RAIN RCI terminology.

Autonomous Workflows in Elara and RAIN terms

RAIN Term	Elara Term
"Mode"	Workflow
"Mode":"AUTO"	Default settings, i.e., no workflow-specific tuning
"Mode":"MONITOR"	Workflow: Bulk Read
"Mode":"HDR"	Workflow: Single Read

```
{"Cmd":"SetCfg", "Mode":" MONITOR"}

{"Cmd":"SetCfg", "Mode":" HDR"}
```

Each workflow may have additional, optional parameters. If a parameter's value is not specified, it takes on the workflow's default value. If the workflow does not define a default value, then it retains its currently set value.

Autonomous Workflow Breakdown

The Elara utilizes the RAIN Communication Interface (RCI). The basic JSON format and command structure of RAIN is general enough to accommodate more commands, but there may be some cases where custom extensions to the protocol are required to accommodate additional user and product scenarios.

The actual "work" of a workflow is divided between various computing resources and levels of abstraction:

- **Host**: The host computer. We will deliver code samples for each workflow.

  In some cases, these code samples will be trivial, if most of the functionality is present in the Elara firmware. In other cases, significant logic may be housed in the host program. Initially, the code samples will be implemented in Java, to maximize reuse opportunities in the TCT.

- **RAIN**: The host communicates to the Elara reader via RCI protocol.
- **Elara**: The Elara main processor (SAM4).

Autonomous Workflow: Single Read

Single Read is appropriate for focusing on reading one tag at a time.

Single Read Workflow

Abstraction Level	Commands
Host	N/A – Not required unless data needs postprocessing

RAIN	`{"Cmd":"SetCfg", "Mode":"HDR"}`
	`{"Cmd":"StartRZ"}`
	
	`{"Report":"TagEvent", …}`

Elara	Filter incoming tag reads from Mercury API, choosing only one to report per read cycle
Mercury API	TMR_startReading() or TMR_Read(), depending on use case

Autonomous Workflow: Bulk Read

Bulk Read is appropriate for reading a large number of tags.

Bulk Read Workflow

Abstraction Level	Commands
Host	N/A – Not required unless data needs postprocessing

RAIN	`{"Cmd":"SetCfg", "Mode":"MONITOR"}`
	`{"Cmd":"StartRZ"}`
	
	`{"Report":"TagEvent", …}`

Elara	N/A – Just translate between RAIN and Mercury API
Mercury API	TMR_startReading() or TMR_Read(), depending on use case

Autonomous Workflow: Tag Commission

Tag Commission is used to initialize tag memory.

Tag Commission Workflow

Abstraction Level	Commands
Host	Convert from Data Standard format (e.g., SGTIN, UDI) to a plain, binary EPC

RAIN	`{"Cmd":"SetCfg", "Mode":"TagCommission"}`
	`{"Cmd":"SetProf",`
	`"Write":[[1,1,7,0,["VAL",":3000:1111:2222:3333:4444:5555:`
	`6666"],true]]}`
	
	`{"Cmd":"SetProf", "Write":[`
	`[1,1,7,0,["VAL",":3000:1111:2222:3333:4444:5555:6666"],tr`
	`ue], [3,0,2,0,["VAL",":1234:5678"],true]`
	`]}`
	
	`{"Cmd":"ThisTag", "Prof":[1]}`

Elara	N/A – Just translate between RAIN and Mercury API
Mercury API	Iterate over write tuples: run `TMR_executeTagOp() with TMR_TagOp_GEN2_WriteData` for each one

Autonomous Workflow: Tag Update

Tag Update is used for incremental writes to tag memory. Some scenarios include:

- Write a fixed value to tag, e.g., Mark a tag as having been present at a certain location by writing a Reader ID.
- Write the current time to a tag, e.g., Timestamp a tag's presence at a certain point

Tag Update Workflow - Write a Fixed Value

Abstraction Level	Commands
Host	N/A – Just set things up

RAIN	`{"Cmd":"SetCfg", "Mode":"TagUpdate"}`
	`{"Cmd":"SetProf",`
	`"Write":[[3,0,2,0,["VAL",":1234:5678"],true]]}`
	`{"Cmd":"StartRZ"}`

Elara	N/A – Just translate between RAIN and Mercury API

Mercury API	Embedded Tag Write
	Create a read plan with an embedded tag write TagOp
	`TMR_RP_init_simple()`
	`TMR_RP_set_tagop() with TMR_TagOp_GEN2_WriteData`
	`TMR_paramSet(TMR_PARAM_READ_PLAN)`
	`TMR_startReading() or TMR_Read(), depending on use case`

Tag Update Workflow - Write a Timestamp

Abstraction Level	Commands
Host	N/A – Just set things up

RAIN	`{"Cmd":"SetCfg", "DateTime":"2018-09-25T21:04:05.123Z"}`
	
	`{"Cmd":"SetCfg", "Mode":"TagUpdate"}`
	`{"Cmd":"SetProf", "Write":[[3,0,2,0,["DT"],true]]}`
	`{"Cmd":"StartRZ"}`

Elara	`TMR_RP_init_simple()`
	
	`while (RZ still activated)`
	`{`
	`  now = tmr_gettime() / 1000`
	`  TMR_RP_set_tagop()`
	`with TMR_TagOp_GEN2_WriteData(Bank=USER, Addr=0,`
	`Data=[(now>>16)&0xFFFF, now&0xFFFF])`
	`  TMR_paramSet(TMR_PARAM_READ_PLAN)`
	`  TMR_startReading()`
	`  <wait until time to get a new timestamp value>`
	`  TMR_stopReading();`
	`}`

Mercury API	Embedded Tag Write Create a read plan with an embedded tag write TagOp Periodically update the embedded tag write value See Elara pseudocode, above

Start Reading Automatically

The procedure for making the reader run autonomously is the same, regardless of workflow. Set the RdrStart setting and save it to non-volatile configuration.

Enable Autonomous Operation

Abstraction Level	Commands
Host	N/A

RAIN	`{"Cmd":"SetCfg", "RdrStart":"ACTIVE"}`
	`{"Cmd":"SaveFields"}`
	
	Setting RdrStart=ACTIVE makes the reader automatically restore config (ReadFields) and start reading (StartRZ(ALL)) at boot.
	
	SaveFields commits the current configuration to non-volatile memory.

Elara	On boot, handle the RdrStart=ACTIVE case
Mercury API	N/A

Disable Autonomous Operation

Abstraction Level	Commands
Host	N/A
RAIN	`{"Cmd":"SetCfg", "RdrStart":"NOTACTIVE"}` `{"Cmd":"SaveFields"}`
Elara	N/A
Mercury API	N/A

RAIN Command Reference

Command Summary

The RCI can be carried over any serial channel. In a typical Elara this is a USB-Serial port (USB CDC ACM). Even though there is only one physical USB connection, multiple logical devices can be presented via USB Composite Device functionality. Both USB HID Keyboard and USB Serial interfaces will be available simultaneously, but you can ignore the USB Serial.

The default configuration is optimized for an easy-to-use experience with no drivers required. The defaults are appropriate to the USB HID keyboard wedge interface.

Command Reference Format

Elara RAIN commands follows the RAIN Communication Interface (RCI) specification, with proprietary extensions.

Message Format

RAIN RCI messages are JSON objects. They always begin with an open curly brace **{** and end with a matching close curly brace **}**. Arbitrary whitespace is allowed between message elements for readability purposes (within JSON syntax constraints).

Protocol messages are indented.

`{<message>}`

Command Format

Commands are JSON objects which contain a "Cmd" field. RAIN recommends that "Cmd" be the first field, but the implementation should be lenient about its placement.

Commands (messages sent host-to-reader) are boldfaced.

`{"Cmd":"Reboot"}`

Response Format

Reponses are JSON objects that contain a "Response" and "ErrID" field. RAIN recommends that these fields come first, but parser implementations should be lenient about their placement.

Reports (messages sent reader-to-host) are italicized.

`{"Report":"Reboot", "ErrID":0}`

RAIN Core Command Examples

RAIN Command Examples	Description

`{"Cmd":"SetCfg","Mode":"AUTO"}` *(default)*
`{"Cmd":"SetCfg","Mode":"HDR"}`
`{"Cmd":"SetCfg","Mode":"MONITOR"}`
`{"Cmd":"SetCfg","Mode":"TagCommission"}`
`{"Cmd":"SetCfg","Mode":"TagUpdate`[1]`"}`
	Set tag reading profile.
	
	A "Mode" is a bundle of parameter settings that is recommended for a particular use case.

`{"Cmd":"SetCfg","_KBDataFormat":"EPC "}` *(default)*
 	
`{"Cmd":"SetCfg","KBDataFormat":"Metadata"}`
 	
`{"Cmd":"SetCfg","_KBDataFormat":"RAIN"}`
	Set keyboard wedge output format .

`{"Cmd":"SetCfg","_USBKBEnable":true}` *(default)* `{"Cmd":"SetCfg","_USBKBEnable":false}`	Turn USB HID Keyboard interface on or off.
`{"Cmd":"GetCfg", "Fields":["ALL"]}`	Retrieve user configuration list.

`{"Cmd":"SetCfg", "FreqReg":"NA2"}`
`{"Cmd":"SetCfg", "FreqReg":"EU3"}`
`{"Cmd":"SetCfg", "FreqReg":"OPEN"}`
	Set region.

`{"Cmd":"SetRZ","ReadPwr":20.0}` *(default)*	Set read power.
`{"Cmd":"SetRZ","DutyCycle":[0,250,750]}`	Set duty cycle. *[start_delay_ms, on_ms, off_ms]*
`{"Cmd":"StartRZ"}`	Start reading.
`{"Cmd":"StopRZ"}`	Stop reading.
`{"Cmd":"ActivateUpdateMode"}`	Go to firmware update mode.
`{"Cmd":"SaveFields"}`	Save configuration from RAM to NVM.
`{"Cmd":"ReadFields"}`	Load configuration from NVM to RAM.
`{"Cmd":"DefaultFields"}`	Clear in-RAM configuration, i.e., load configuration from hard-coded defaults to RAM.
`{"Cmd":"GetInfo","Fields":["ALL"]}`	Get reader information.
`{"Cmd":"Reboot"}`	Reboot the reader.
`{"Cmd":"GetGPIOs"}`	Get GPIO states.
`{"Cmd":"SetGPIOs"}`	Set GPIO states.
`{"Report":"HB"}`	Reader heartbeat.

RAIN Advanced Command Examples

RAIN Command Examples	Description

`{"Cmd":"SetCfg","_AudioVolume":"MUTE"}`
`{"Cmd":"SetCfg","_AudioVolume":"LOW"}`
(default)
`{"Cmd":"SetCfg","_AudioVolume":"MED"}`
`{"Cmd":"SetCfg","_AudioVolume":"HIGH"}`
	Set beeper volume.

`{`
`"Cmd":"SetCfg",`
`"SpotAnt":false,`
`"SpotDT":false,`
`"SpotInvCnt":false,`
`"SpotPhase":false,`
`"SpotProf":false,`
`"SpotRSSI":false,`
`"SpotRZ":false`
`}`
(default)
	Enable/Disable tag read metadata (standard RAIN fields).

`{`
`"Cmd":"SetCfg",`
`"SpotFreq":false,`
`"SpotGen2_BI":false,`
`"SpotGen2_Q":false,`
`"SpotGen2_LF":false,`
`"SpotGen2_Target":false,`
`"SpotGPIO":false,`
`"SpotProt":false,`
`"SpotSensor":false`
`}`
(default)
	Enable/Disable tag read metadata (ThingMagic custom fields).

`{`
`"Cmd":"SetProf",`
`"Read":`
`[[3,0,4,0]]`
`}`
 	
Note: The "read tuple" arguments are [MemBank, StartWord, NumWords, RetryLimit]
	Enable embedded data read on tag reads.

`{"Cmd":"SetProf",`
`"Write":[[1,1,7,0, ["VAL", ":3000:1111:2222:3333:4444:5555:6666"], true]]}`
 	
Note: The "write tuple" arguments are [MemBank, StartWord, NumWords, RetryLimit, WriteMethod, Check]
	Embedded data write.
	
	See Command Reference for supported Write types.

`{"Cmd":"ThisTag", "Prof":1}`	Immediate data operation.

RAIN Advanced Command Examples

RAIN Command Examples (Continued)	Description
`{"Cmd":"SetCfg", "DateTime":"2018-09-25T21:04:05.123Z"}`	Set current time.

`{"Cmd":"GetCfg", "Fields":["DateTime"]}`
`{"Report":"GetCfg", "ErrID":0,`
`"DateTime":"2018-09-25T21:04:05.123Z"}`
	Get current time.

Bootloader Commands

If the application firmware becomes corrupted, the reader runs a permanently installed application called the bootloader. The bootloader is used to update the application firmware image in Flash memory.

The application cannot update its own image because it executes out of Flash. Modifying that image in-place risks corrupting the program as it runs, so a separate bootloader must be executed in order to guarantee that no access is made to the app image while it is being rewritten.

RAIN Bootloader Command Examples

RAIN Command Examples	Description

`{"Cmd":"GetInfo","Fields":["ALL"]}`
 	
`{"Report":"GetInfo","ErrID":0,"_BootloaderVersion":"YYMMDDRR","BootloaderLockBits":"0x0000003F","BootModeTrigger":"0x2X","PartNumber":"540-XXXX-XX" ,"SerialNumber":"WWYYcccccENNNN","ProductRevision":"0A","Interface":"ALL","Sensor":"ALL","RegionRange":"NA"}`
 	
`{"Cmd":"GetInfo","Fields":"_BootloaderVersion","_BootloaderLockBits"]}`
`{"Report":"GetInfo","ErrID":0,"_BootloaderVersion":"0x00000000","_BootloaderLockBits":"0x0000003F"}`
	Retrieve Bootloader info that includes version and locked area in Flash.

`{"Cmd":"_SetFlashLock", "Section":"BootLoader", "State":"Lock"}`
 	
` {"Cmd":"_SetFlashLock", "Section":"HWInfo", "State":"Lock"}`
 	
`{"Cmd":"_SetFlashLock", "Section":"App","State":"Lock"}`
`{"Report":"_GetFlashLock","ErrID":0,"Section":"App","State":"Unlock"}`
	Set lock on or off to section in Flash. State is "Lock" or "Unlock."

![JADAK - A Novanta Company]

RAIN Bootloader Command Examples

RAIN Command Examples (Continued)	Description

`{"Cmd":"_GetFlashLock", "Section":"Bootloader"}`
`{"Report":"_GetFlashLock","ErrID":0,"Section":"B`
`ootloader","State":"Lock"}`
 	
`{"Cmd":"_GetFlashLock", "Section":"HWInfo"}`
`{"Report":"_GetFlashLock","ErrID":0,"Section":"H`
`WInfo","State":"Lock"}`
 	
`{"Cmd":"_GetFlashLock", "Section":"App"}`
`{"Report":"_GetFlashLock","ErrID":0,"Section":"A`
`pp","State":"Unlock"}`
	Retrieve current lock state for section in Flash. State is "Lock" or "Unlock."

`{"Cmd":"StartUpdate", "Section":"HWInfo",`
`"Password":"0x07584172"}`
 	
`{"Cmd":"StartUpdate", "Section":"App",`
`"Password":"0x02254410"}`
	Initiate firmware update for section with password required if protected. Password is an unsigned long value in HEX.

`{"Cmd":"SendData","Data":"MDEwMjAzMDQwNTA2MDcwOD` `A5MEE="}`	Send data to be updated in firmware. Data is binary data encoded with base64.
`{"Cmd":"EndUpdate"}`	Terminate firmware update.

ThingMagic Configuration Tool

Introduction

The ThingMagic Configuration Tool is a desktop application developed using Java language and built in javafx technology. TCT is used to connect, configure, and read tags.

- TCT is only supported on the Windows 64-bit operating system.
- TCT requires JRE version 1.8 or later.

Install the USB Driver

For Windows 10 operating systems, no additional driver is required. USB is automatically detected after plugging in the USB cable. If the automatic detection fails, follow these steps to recognize the Elara module:

1. Plug the Elara module USB cable into the PC.

2. Windows should report it has **Found New Hardware** and open the Hardware Installation Wizard.

3. Select **Install from a list or specific location (Advanced)**. Click **Next**.

4. Select **Don't search…** Click **Next**, then **Next** again.

5. Click **Have Disk** and navigate to where the SDK zip is extracted. Select **ELARA.inf** under the drivers folder. Click **Open**, then **OK**.

6. A COM port should now be assigned to the Elara. If you aren't sure what COM port is assigned you can find it using the Windows Device Manager.

In order to use the USB interface with a Windows 7 operating system, you must first install the ELARA.inf file, available in the SDK download package.

1. Right click **Computer** on the Start Menu and select **Manage.**

2. Select **Device Manager**.

3. Select **Elara reader** and click on **Update Drivers**.

4. Click on **Browse my computer for driver software**.

5. Click on **Let me pick from a list of available drivers on my computer**.

6. Click on **Have Disk**.

7. Provide the .inf file path.

8. Proceed with the driver installation.

Install TCT

Download and install TCT using the following steps.

JADAK
A Novanta Company

1. Use a web browser to navigate to https://www.jadaktech.com/documentation/.

2. Download the SDK available under the Elara folder.

3. Once downloaded, extract the zip archive to the desired location.

4. Find the ThingMagic Configuration Tool executable and double click on it to install to the host PC.

5. Follow the steps for your OS.

Home

The Home page displays version details and main functionalities of TCT.
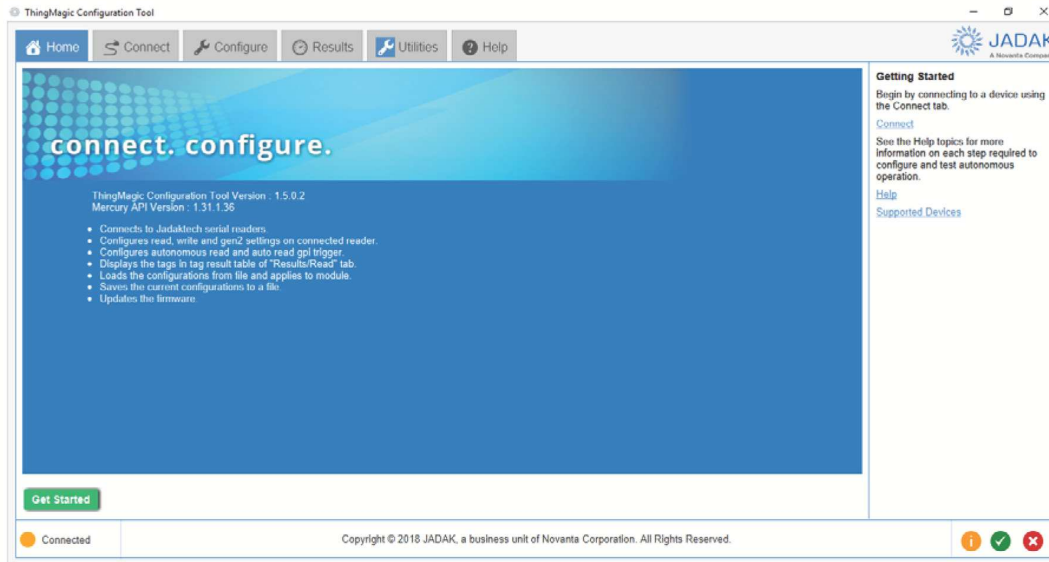
This screen introduces the set of available features of TCT. Across the top of the window are the following tabs:

Connect: Used to discover and connect to available readers

Configure: Presents all the configuration settings that can be saved.

Results: Provides a quick way to view streaming tag reads

Utilities: Loads firmware on the Elara reader

Help: Displays information for using TCT.

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

• The right section of each screen contains a short summary of contextual help along with links to more detailed help.

- The bottom left shows the reader connection status.
- Warning, Status and Error messages are displayed from the corresponding bottom right message icons.

Connect

The Connect tab provides connection options to find and connect to a reader. Refer to the product's Hardware Guide for details on how to establish a connection to the reader and how to identify the correct port to identify it.

When Elara reader is connected to a host, it is shown as:

Elara (COMxx): USB serial Device (where xx is the COM port number)

Elara HID: USB Input Device

Clicking on the refresh icon ⟳ next to **Available Devices** displays all the devices connected to the host. You must click the refresh icon whenever a new device is connected to the host or an existing device is disconnected.

HID Mode

When you select **Elara HID**, you are sent to the **Results** tab. If tags are present in field, the Elara reader starts reading tags that are then displayed on the screen.

CDC Mode

When you select **Elara (COMxx)**, you can choose the baud rate for serial communication and turn on or off the Transport Logging feature.

Transport Logging

Transport Logging lets you see the RAIN communication messages between TCT and the Elara reader for debugging purposes.

Once the reader has been connected, you can configure your device.

Configure

The Configure tab provides all the available reading settings and allows you to enable metadata and configure the beeper volume.

General: This tab allows you to change the **Region**, **Date/Time**, **RF Power**, and **Workflow** settings.

Advanced: This tab allows you to change **Gen2 Session**, **Target**, **Encoding**, **BLF**, **Q**, **Tari**, and **InitQ** settings.

User Interface: This tab allows you to enable/disable **Metadata** and change **Beeper Volume** settings.

Once entered, you can Set, Save, or Revert your changes. You can also Load Defaults on the reader.

Set: Applies the modified settings temporarily in the reader's volatile memory (RAM) during the current session.

Save: Applies and stores the modified settings in the reader's permanent memory.

Revert: Discards the current active settings and applies the settings retrieved from reader's permanent memory.

Load Defaults: Discards the current active settings and replaces them with the factory-provided settings. You must click **Save** to store the default settings in the reader's permanent memory.

NOTE: In order to change the configuration on a HID port, you need to connect the reader on a COM port.

Results

The Results tab displays tag reads like a text editor. Double click the **Caps Lock** key on the keyboard to start reading tags. Double clicking **Caps Lock** again stops the tag reads.

Utilities

The Utilities tab is used to upgrade the Elara firmware. Select the **.bin** file from the host PC.

Help

The Help tab contains a short summary of contextual help.

Compliance and IP Notices

EMC FCC 47 CFR, Part 15

Industrie Canada RSS-210

Federal Communication Commission (FCC) Interference Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

• Reorient or relocate the receiving antenna.

• Increase the separation between the equipment and receiver.

• Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

• Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

To comply with FCC RF exposure limits for general population/uncontrolled exposure, the Elara may not be used with a separation distance of less than 20 cm from all persons, and must not be collocated or operating in conjunction with any other antenna or transmitter.

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

Industry Canada

Under Industry Canada (IC) regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the Equivalent Isotropically Radiated Power (EIRP) is not more than that necessary for successful communication.

This radio transmitter (identify the device by certification number, or model number if Category II) has been approved by Industry Canada to operate with the antenna types listed below with the maximum permissible gain and required antenna impedance for each antenna type indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the Equivalent Isotropically Radiated Power (EIRP) is not more than that permitted for successful communication.

This device has been designed to operate with the antennas listed in the Authorized Antennas table. Antennas not included in these lists are strictly prohibited for use with this device.

To comply with IC RF exposure limits for general population/uncontrolled exposure, the antenna(s) used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be colocated or operating in conjunction with any other antenna or transmitter.

Industrie Canada (French Canadian)

Conformément à la réglementation d'Industrie Canada, le présent émetteur radio peut fonctionner avec une antenne d'un type et d'un gain maximal (ou inférieur) approuvé pour l'émetteur par Industrie Canada. Dans le but de réduire les risques de brouillage radioélectrique à l'intention des autres utilisateurs, il faut choisir le type d'antenne et son gain de sorte que la puissance isotrope rayonnée équivalente (p.i.r.e.) ne dépasse pas l'intensité nécessaire à l'établissement d'une communication satisfaisante.

Le présent émetteur radio (identifier le dispositif par son numéro de certification ou son numéro de modèle s'il fait partie du matériel de catégorie I) a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés ci-dessous et ayant un gain admissible maximal et l'impédance requise pour chaque type d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur

Le fonctionnement de l' appareil est soumis aux deux conditions suivantes:

1. Cet appareil ne doit pas perturber les communications radio, et

2. cet appareil doit supporter toute perturbation, y compris les perturbations qui pourraient provoquer son dysfonctionnement.

Pour réduire le risque d'interférence aux autres utilisateurs, le type d'antenne et son gain doivent être choisis de façon que la puissance isotrope rayonnée équivalente (PIRE) ne dépasse pas celle nécessaire pour une communication réussie.

Au but de conformer aux limites d'exposition RF pour la population générale (exposition non-contrôlée), les antennes utilisés doivent être installés à une distance d'au moins 20 cm de toute personne et ne doivent pas être installé en proximité ou utilisé en conjonction avec un autre antenne ou transmetteur.

Appendix A: RAIN Error Messages

"Report" messages occur in various contexts:

• In response to a command. The report value matches the command.

• Reports also occur asynchronously in response to non-command events.

Bootloader Faults and Application Faults

Number (ErrID)	Description (ErrDesc)	Optional information (ERRInfo)	Notes
0	No error(s)		No error on the command when in response to a command. Error condition cleared when reported as an event.
1	Bad message	JSON string with the bad message	The JSON is not correct or the message is missing parts.
2	CRC error	JSON string with actual CRC calculated	
3	Buffer full	JSON number with the receive buffer size	
4	Response too big	JSON number with the transmit buffer size	This may happen when a reader uses a fixed size transmit buffer or runs out of memory.
5	Memory overrun	JSON string: < which memory>	
6	Reader too cold	JSON string: < which component>	This may result in inaccurate calibration or settings.
7	Reader hot	JSON string: <which component>	This does NOT result in a functional termination or malfunction.
8	Reader too hot	JSON string: <which component>	This results in a functional termination or malfunction.
20	Command not supported	JSON string showing which command is not supported	
21	Field not supported	Array of strings of not supported fields	
22	Field value not supported	Array of strings of fields of which the value is not supported	
23	Field value changed	Array of strings of fields of which the value is not supported	The reader may change requested field values to a more appropriate supported value.

JADAK
A Novanta Company

Bootloader Faults and Application Faults (Continued)

Number (ErrID)	Description (ErrDesc)	Optional information (ERRInfo)	Notes
30	GPIO toggle value the same	Array of numbers identifying the GPIO IDs with the problem	A toggle could not be performed.
31	GPIO not settable	Array of numbers identifying the GPIO IDs with the problem	The GPIO is not an output, D2A, or register.
25	Trigger not an input switch	Array of number identifying the offending GPIO	
30	SpotProfiles full		
31	SpotProfile error	Array with: JSON number: SpotProfile number, JSON string: <more info>	A spot profile resulted in an air protocol configuration error.
32	Illegal SpotProfile	Number array listing the illegal SpotProfiles	
33	Thig Tag timeout	String stating one of the following: "No tags inventoried." "No SpotProfile triggered."	No tags were spotted during the ThisTag duration.
34	Spot error	String describing the error	The spot event could not be completed.
40	ReadZones full		
41	ReadZone start error	Array of which the first element is a string describing the start error, followed by numbers indicating the ReadZones with a start error	
42	ReadZone definition error	An array listing the offending fields	
≥1000	<Proprietary errors>	<Vendor specific>	Refer to Nano User Guide for vendor-specific error codes.

JADAK
A Novanta Company

Appendix B: Environmental Considerations

Electrostatic Discharge (ESD) Considerations

Contact discharge to +/- 6 kV

Air discharge to +/- 8 kV

JADAK internal lab test reports available on request for EN60601, 4th edition compatibility for electrostatic discharge immunity +/-15kV air discharge and +/-8kV contact discharge.

Operating Temperature

0°C to +50°C

Installation Considerations

The Elara enclosure is equipped with 4 mounting inserts, accepting M3 x 0.5 mm threaded screws to a 3.5 mm depth.
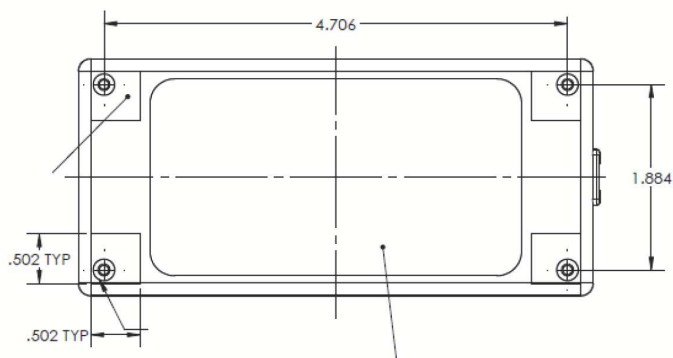
Alternatively, rubber feet can be mounted over the inserts in the detents provided.