ARL-TR-9005 ● AUG 2020

DEVCOM ARMY RESEARCH LABORATORY

# Radio-Frequency Identification (RFID) Multi-Reader/Integrated Sensor Architecture (ISA) Publisher Software Documentation

by Laurel C Sadler and Jesse Kovach

## NOTICES

### Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-9005 ● AUG 2020


# Radio-Frequency Identification (RFID) Multi-Reader/Integrated Sensor Architecture (ISA) Publisher Software Documentation

**Laurel C Sadler and Jesse Kovach**
*Sensors and Electron Devices Directorate, CCDC Army Research Laboratory*

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED (From: To) |
|---|---|---|
| August 2020 | Technical Report | January 2019–June 2020 |

**4. TITLE AND SUBTITLE**
Radio-Frequency Identification (RFID) Multi-Reader/Integrated Sensor Architecture (ISA) Publisher Software Documentation

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
Laurel C Sadler and Jesse Kovach

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
CCDC Army Research Laboratory
ATTN: FCDD-RLS-SI
2800 Powder Mill Road
Adelphi, MD 20783-1138

**8. PERFORMING ORGANIZATION REPORT NUMBER**
ARL-TR-9005

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**
ORCID ID(s): Laurel C Sadler, 0000-0001-8697-2246; Jesse Kovach, 0000-0002-3624-8257

**14. ABSTRACT**
The RFID Multi-Reader tool is a configurable software package for reading radio-frequency identification (RFID) tags using multiple RFID devices, each with multiple antennae. The detection of an RFID tag is posted as an event message containing the RFID tag data and antenna location and orientation on an Integrated Sensor Architecture (ISA) network. The tool can be configured for simultaneous use with multiple RFID tag readers of varying models/manufacturers at various locations. Each RFID reader can be further configured to have multiple antennae at various locations. This report describes the functions of the RFID Multi-Reader tool and provides installation instructions and configuration examples for the tool. In addition, the RFID Multi-Reader tool can be configured to store the RFID event messages to a file in a human-readable format or, when configured for playback, the software can read the event messages from a text file.

**15. SUBJECT TERMS**
Integrated Sensor Architecture, ISA, sensor networks, radio-frequency identification, RFID

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Laurel Sadler |
| Unclassified | Unclassified | Unclassified | UU | 31 | 19b. TELEPHONE NUMBER (Include area code) (240) 893-6289 |

## Contents

## List of Tables

## Summary

The RFID Multi-Reader tool is an application for interfacing Electronic Product Code radio-frequency identification (RFID) readers with an Integrated Sensor Architecture (ISA) network. The tool can be configured to simultaneously connect to a multiple number of RFID readers as well as multiple types of RFID readers from various manufacturers. Each RFID reader can be configured with multiple antennas at various locations and orientations, each of which will detect RFID tags. The RFID tool will parse the RFID tag information and disseminate the tag information and antenna location and orientation through ISA. This report describes the functions of the RFID Multi-Reader tool and provides installation instructions and configuration examples.

# 1. Introduction

## 1.1 RFID Overview

Radio-frequency identification (RFID) refers to a technology in which digital data are encoded in electronic tags that are read via radio transmissions. These tags consist of an integrated circuit and an antenna. RFID tags can be passive or active. Passive tags are powered by the received radio signal from the reader, while active tags contain batteries or draw power from an external power source. An RFID reader is a device used to gather information from an RFID tag via digital radio transmissions. The RFID tags can be used to track individual objects.

As described by Techopedia, "In concept, RFID technology is similar to that of barcodes. However, unlike the barcode, the RFID tag does not need to be scanned directly and does not require line of sight to the RFID reader."[1] Generally, the RFID tag must be within the range of 3 to 300 ft of the RFID reader antenna, depending on the tag, reader, and antenna being used. RFID technology allows multiple tags to be read simultaneously, enabling rapid identification of a particular object even when it is surrounded by several other objects. RFID tags can be used with vehicles to provide access control, identify a vehicle, retrieve stored information about the vehicle, and track the vehicle.

There are a number of RFID technologies on the market, both standard and proprietary. Each technology is intended for a specific set of applications, and different technologies have widely varying performance characteristics. Generally, readers designed for one system will not read tags designed for another, although some advanced readers support multiple tag technologies. When planning the deployment of an RFID system, it is important to choose a RFID technology that is appropriate for the application and ensure that the tags and readers are compatible.

The software described in this report works with tags and readers supporting the EPC Gen2 (also known as ISO 18000-6C) RFID technology.[2] This is a standardized system originally designed for tracking of inventory in industrial and retail environments, but useful for a wide variety of other applications.

## 1.2 ISA Overview

According to the US Army Night Vision and Electronic Sensors Directorate (NVESD), US Army Combat Capabilities Development Command (CCDC), Control, Computers, Communications, Cyber, Intelligence, Surveillance and Reconnaissance Center (CCDC C5ISR),[3]

> ISA is a U.S. Army Service-Oriented Architecture (SOA) developed by [CCDC C5ISR Center] NVESD. ISA provides capabilities that enable Soldiers to exchange information between their own sensors and those on other platforms in a fully dynamic and shared environment. ISA enables Army sensors and systems to readily integrate into an existing network and dynamically share information and capabilities to improve situational awareness in a battlefield environment.

According to Poltronieri et al.,[4]

> ISA identifies common standards and protocols, which support a net-centric system-of-systems integration. Utilizing a common language, these systems are able to connect, publish their needs and capabilities, and interact dynamically. ISA provides an extensible data model, ISA Data Model Specification[5], with defined capabilities, and provides a scalable approach across multi-echelon deployments, which when coupled with dynamic discovery capabilities, cybersecurity, and sensor management, provides a system which can adjust and adapt to dynamic environments.

## 1.3 RFID Multi-Reader Tool Overview

The RFID Multi-Reader tool has several functions: receiving and interpreting data from various RFID readers on the network; checking if the tag is in the reader's internal list of valid tags; publishing the tag reading event's information in a custom message format to the ISA network; and sending status messages about the availability and readiness of each of the RFID readers on the ISA network. In addition, the RFID Multi-Reader tool can be configured to store the RFID event messages to a file in a human-readable format or, when configured for playback, the software can read the event messages from a text file.

The RFID Multi-Reader tool contains two primary software modules: the RFID reader drivers and the RFID/ISA bridge. The driver module translates between vendor-specific interface protocols and a common Java application programing interface (API). The driver also contains state machine logic for reading various combinations of fields from RFID tags and authenticating tags that support the EPC Gen2V2 Advanced Encryption Standard (AES) extensions. The RFID/ISA bridge module receives information from the RFID reader driver via Java callbacks and translates this information into ISA events and status messages.

The RFID Multi-Reader tool currently supports RFID readers from JADAK (formerly ThingMagic), Alien Technology, and Venture Research.

## 1.4 RFID Multi-Reader Tool Use Cases

The RFID Multi-Reader tool can be used in a number of different applications. Primary uses include the following:

- Determine which vehicles are present in the vicinity of the RFID reader.
- Provide data read from the RFID tag to other ISA-enabled components on the network (e.g., command and control systems or ISA-enabled databases).
- Provide ground-truth data for data and video analytics systems, data fusion applications, and many other scenarios where time and location ground truth is necessary.
- Provide access control of the vehicles entering a secure area.
- Track vehicles along a route monitored with checkpoints equipped with RFID readers.

## 1.5 Usage within the ARL Networked Sensor Research Test Bed

The RFID Multi-Reader tool is used to collect ground-truth data on RFID tagged vehicles within the CCDC Army Research Laboratory (ARL) Networked Sensor Research Test Bed (NSRTB). The NSRTB is an environment designed for the development and testing of new sensing and data/video analytics capabilities for tactical applications. The NSRTB incorporates commercial off-the-shelf (COTS), government off-the-shelf (GOTS), and experimental sensor systems and software. All components of the NSRTB are interconnected using ISA.

In addition to EPC RFID readers, sensors currently employed within the NSRTB include Axis Communications Q1615-E MkII surveillance cameras, Tactical Remote Sensor System (TRSS) seismic/acoustic sensors, and experimental sensor systems under development by the laboratory. Data management and monitoring software within the NSRTB includes the RaptorX mapping system, multiple GOTS video management systems, the ISA Diagnostics package, and custom visualization applications developed by ARL engineers. The NSRTB is being used to support the development of data and video analytics systems for multiple government sponsors.

# 2. EPC RFID Technology Background

The RFID Multi-Reader tool is designed to be useful for a wide range of applications and provides a number of low-level configuration options to control tag reader behavior. The ISA messages that the tool generates also contain a number

of different data items relating to the tags that have been read. To effectively install, configure, and use the tool, some knowledge of EPC RFID technology is useful. This section presents a brief overview of EPC technology as it applies to the RFID Multi-Reader tool.

## 2.1 Tag Memory Banks

EPC RFID tags contain multiple memory banks. All but one of these memory banks are field programmable. The memory banks are as follows:

- *EPC*: The EPC bank is used to store a unique identifier for the item to which the tag is attached. This bank is present on all tags. The EPC value is preprogrammed at the factory and can be reprogrammed in the field using an RFID reader with the proper software. When performing certain tag operations (such as programming tags or reading other tag memory banks), the reader "singles out" individual tags using the tag's EPC value. Therefore, EPC values should be unique, as strange behavior can result when a reader sees multiple tags with the same EPC value.
- *Tag Identification (TID)*: The TID bank contains information about the tag itself. This bank is present on all tags. The value of the TID bank is "burned in" at the factory and is not field programmable. The TID field contains information such as the manufacturer and model of the tag and a factory-assigned tag serial number.
- *User*: This bank stores application-specific data. This bank is not present on all tags. On tags where the user bank is present, the size of the bank can range from a few bytes to a few kilobytes depending on the specific model of tag. This bank is field programmable and is generally not initialized at the factory.
- *Reserved*: This bank is used to store the access and kill passwords for a tag (see Section 2.2).

The EPC consortium has defined a "Tag Data Standard"[6] specifying a common encoding for data stored in the EPC and User banks. This is an application-level standard and its usage is not enforced by tag and reader hardware. Many EPC RFID applications do not use the Tag Data Standard. The RFID Multi-Reader tool does not implement the Tag Data Standard as it simply reports raw tag memory bank contents via ISA. If desired, higher-level applications can process the raw data according to the Tag Data Standard.

When querying tags, readers will always read and return the EPC value. Depending on the particular model of reader in use, reading other banks may require additional operations that slow down the read process. (For example, ThingMagic readers can automatically read the EPC value and either the TID or user data value, but not all three simultaneously. Reading all three values requires the software to use an alternate mode that queries tags individually, significantly slowing down performance.)

The RFID Multi-Reader tool allows the user to specify which banks to read. For maximum performance, the user should configure the tool to only read banks of interest.

## 2.2 Tag Security

All EPC RFID tags contain the following basic security features:

- *Tags can be "locked" against reprogramming*. To lock a tag, an application must program the tag with a 32-bit access password and send a lock command to the tag. Once locked, the tag will reject all programming attempts until an application unlocks the tag by sending an unlock command containing the correct access password.
- *Tags can be "killed"*. To kill a tag, an application must program the tag with a 32-bit kill password and then send the tag a kill command containing the correct kill password. Once killed, a tag will no longer respond to read requests.
- *The TID memory bank includes a factory-set tag serial number that cannot be reprogrammed by any documented means*. By checking this serial number, an application can provide a basic level of protection against tag cloning.

These features are designed to provide basic security in a factory or retail environment (i.e., tag locking prevents an attacker from reprogramming the tag on an expensive item with an identifier for a cheaper item). However, they may not provide an adequate level of security for all applications. Notably, there is no protection against an adversary cloning EPC values onto new tags or spoofing TID values by using a software-defined radio as a tag emulator.

Note that most tags ship with default values for the access and kill passwords. If these values are not reprogrammed by the end user, an attacker can easily lock and disable tags by sending the appropriate commands using the default passwords.

To provide an additional level of security, some newer readers and tags support EPC Gen2V2 encryption extensions.[7] These tags support the following additional security features:

- *Secure storage of two 128-bit AES encryption keys*. Once programmed, the keys cannot be changed or extracted from the tag by any documented means.
- *Secure tag authentication using either of the two encryption keys via a challenge-response protocol*. This feature allows an application to verify that a tag has been programmed by someone with knowledge of a secret key, providing security against tag cloning.
- *Encrypted reads of tag memory using one of the two keys*. This feature allows an application with knowledge of a secret key to read data from a tag without exposing the raw data over the air.
- *An "untraceable" mode where a tag can be programmed to mask part or all of the EPC value from readers that do not know the correct secret key*. This feature prevents tags from being queried by unauthorized readers.

The RFID Multi-Reader tool currently supports basic usage of the EPC Gen2V2 tag authentication feature. Authentication is supported only for NXP's UCODE DNA tags and ThingMagic readers. When authentication is enabled, the tool will check a tag's TID value to determine whether a tag supports authentication prior to attempting any authentication operations on that tag. Enabling authentication slows down reading, so it should only be enabled if needed.

As the techniques used to manage tag keys are specific to individual deployments, the tool does not currently utilize tag keys when performing authentication. Instead, the tool reports raw challenge and response data obtained from the tag and reader during the authentication process. (The challenge data are randomly generated by the reader, and the response data are generated by the tag by encrypting the challenge with one of its programmed secret keys.) An external application with knowledge of tag keys can receive the challenge-response data via ISA and process it to determine whether a tag is authentic or not.

The RFID Multi-Reader tool currently has basic support for encrypted reads of the TID memory bank, reporting the encrypted data through ISA for decryption by an external application. The tool does not currently support the untraceable mode, as this feature cannot be implemented without knowledge of tag keys.

# 3. RFID Multi-Reader Tool Installation Instructions

## 3.1 Assumptions

This report assumes the user has a working knowledge of basic ISA component configuration and ISA certificate usage. See the ISA software development kit (SDK) documentation[8,9] for more information regarding these topics.

## 3.2 Prerequisites and System Requirements

The RFID Multi-Reader has the following system requirements and prerequisites:

- A Java Runtime Environment (JRE) for Java 8 must be installed. The software has been built and tested with OpenJDK 8. It should also function with Oracle JRE 8, but this configuration has not been tested by the authors. Note that, as of the time of this writing, usage of Oracle JRE 8 in a production environment requires a paid license.

## 3.3 Installation Procedure

The following is the installation procedure:

1) Obtain an ISA certificate and private key for this application. The certificate needs to be provided as a `.jks` file. Generating/obtaining certificates is outside the scope of this report—contact the local ISA system administrator for assistance.

2) Extract the distribution zip file to a suitable location (e.g., `c:\isa\database-archiver`). The path to the installation folder cannot contain spaces.

3) Copy the certificate file from step 1 into the folder just created.

4) Change into the folder created in step 2. Copy the `connection_sample.properties` file to `connection.properties`.

5) Open `connection.properties` in a text editor. Change the settings in the file as needed for the application (see Section 3.4). Save the file and exit the editor.

## 3.4 Installation as Windows Service (Optional)

On a Windows platform, the Database Archiver tool can optionally be configured to run as a Windows service. To do so, perform the following steps:

1) Perform all steps of the Installation Procedure in Section 3.3.

2) Open the install_service.bat file in a text editor. Locate the line beginning

   ```
   set LOCAL_JAVA_HOME=
   ```

   Edit this line to specify the path to your JRE installation. If this was a full Java Development Kit (JDK) installation, specify the path to the JRE contained within the JDK, not to the JDK itself. For example, with OpenJDK installed in `c:\openjdk\jdk8u172-b11`, the setting should read

   ```
   set LOCAL_JAVA_HOME=C:\openjdk\jdk8u172-b11\jre\
   ```

   Alternatively, if the `JAVA_HOME` environment variable is properly set on the system, skip this step and the script will use the value of `JAVA_HOME`.

3) Ensure that the `NT AUTHORITY\NETWORK SERVICE` account has "Full Control" permissions to the installation folder. The simplest way to do this is to open an administrator command prompt and run the following command:

   ```
   icacls "<path-to-installation-folder>" /grant "NT AUTHORITY\NETWORK SERVICE":(OI)(CI)F
   ```

4) Install the service by opening an administrator command prompt and running `install_service.bat`.

   The script will attempt to remove any previously installed versions of the service prior to installing. If the service was not previously installed, there will be errors related to this—these errors can be safely ignored.

5) Verify that the service is running by opening the Windows service control panel and verifying that the service is shown as "Running". The name of the service will be the name of the installation folder.

If the service does not start after installation, check the logs in the "ServiceLogs" folder for error messages.

The service may be uninstalled by opening an administrator command prompt and running `install_service.bat /u`.

## 3.5 Starting the Software

On all platforms, the RFID Multi-Reader tool can be started using the following command:

```
java -jar mil.arl.rfid.MultiReaderPub.jar
```

Alternatively, on Windows, the software can be started using the included `run.bat` script.

# 4. RFID Multi-Reader Tool Configuration Instructions

This section discusses each of the configuration settings in the `connection.properties` file, including how they are set and what they mean. This allows the user to configure the software to accomplish exactly what the user requires and no more.

## 4.1 Parameters for the RFID Software Package

The following configuration parameters will be set once for the software package itself, regardless of the number of RFID readers or antennae attached to each reader:

- `isaControllerHost`: Internet Protocol (IP) address or hostname of the ISA controller.
- `isaControllerPort`: Port of the ISA controller.
- `isaControllerUCI`: Universal Component Identifier (UCI) of the ISA controller. If this does not match the UCI configured on the controller, the software will not connect.
- `myUCI:` UCI to use for this instance of the RFID Multi-Reader tool. It must match the name on the certificate being used.
- `isaKeystorePath`: Path/file name of the keystore (`.jks`) file containing the ISA certificates to use. The names embedded in the certificates contained within the file must match the UCI being used. Generally, the name of this file will match the UCI.
- `isaKeystorePassword:` Password for the keystore file.
- `isaKeyPassword:` Password for the private keys in the keystore file. This is usually the same as the keystore password.
- `isaEncoding`: ISA protocol encoding to use. This should always be set to `ipl_3v7` unless special circumstances apply.
- `isaEnableTLS`: If `true`, the software will use transport layer security (TLS) authentication and encryption for the ISA connection. This should always be `true` unless special circumstances apply.

## 4.2 Parameters Defining the RFID Readers and Antennae

The following parameters for each reader and reader antennae must be defined. Readers are numbered starting from 0. Antenna numbering depends on the particular model of reader. At least one reader and reader antenna configuration must be defined in the configuration file. Additional reader and antenna configurations can be defined by incrementing the number (e.g., reader.0.type, reader.1.type, reader.1.antenna.1.name, reader.1.antenna.2.name, so on).

### 4.2.1 RFID Reader Parameters

readers: This is a comma delineated list of enabled RFID readers. For example, if there are two readers (0 and 1) defined in the configuration file, this value should be set to "0,1" to enable both readers.

reader.0.type: RFID Reader Type. Either "alien", "thingmagic", or "venture".

reader.0.name: RFID Reader Name. This sets the name (UCI) of the ISA subcomponent representing the reader.

reader.0.readerAddr: Address of the RFID reader. The precise format depends on the reader type, as follows:

- ThingMagic network readers: the reader's Low Level Reader Protocol (LLRP) URL (e.g., `llrp://192.168.1.1`)
- ThingMagic serial/USB readers: the reader's ThingMagic extensible API (eAPI) URL (e.g., `eapi:///COM6` [Windows] or `eapi:///dev/ttyUSB0` [Linux])
- Venture and Alien network readers: the reader's IP address (e.g., `192.168.1.1`)

reader.0.readerPort: Transmission Control Protocol (TCP) port used to connect to the reader. Not used for ThingMagic readers.

reader.0.enableAntennas: This is a comma-delineated list of the identity of the antennae ports to enable on this RFID reader (e.g., reader.0.forceAntennas=0,8).

reader.0.readPowerStr: This controls the transmit power of the RFID reader. If not specified, the software will not attempt to change the transmit power on the reader.

reader.0.readTid: If set to true, the software will attempt to read and report the contents of tag's TID memory banks.

reader.0.readUserData: If set to true, the software will attempt to read and report the contents of tag user data memory banks. Not all tags have user data banks.

reader.0.tryAuthenticate: If set to true, the software will attempt to authenticate tags if supported by the tag and reader. See Section 2.2 for details regarding the authentication implementation.

reader.0.latitude: Latitude of the RFID reader itself. Used to set the reported position of the ISA subcomponent representing the reader.

reader.0.longitude: Longitude of the RFID reader itself. Used to set the reported position of the ISA subcomponent representing the reader.

reader.0.mil2525Symbol: Character varying: MIL-STD-2525[10] symbol code for the entity described in the message. Used to set the identity property of the ISA subcomponent representing the reader (e.g., SFGPESE--------).

The following are additional connection parameters for Alien readers:

- Alien RFID readers use separate TCP connections for control and data, while also authenticating connections to the reader. The following parameter is needed when connecting to Alien readers (only).
- reader.0.localAddr: Local listen address used to receive data from the Alien reader. If not specified, the software will attempt to automatically pick the correct address to use. For best results, this should be manually specified on systems with multiple network interfaces.

### 4.2.2 Low-Level Tuning of RFID Reader Behavior Parameters

These parameters allow RFID reader behavior to be tuned, either for debugging or for better performance in specific applications. For basic applications, these parameters can be left at default settings (i.e., commented out from the configuration file).

reader.0.debug: If true, the software will print additional messages describing the progress of RFID tag read operations. Useful for troubleshooting.

reader.0.disableOfflineTagOps: Disables certain tag operations when reading. Used for debugging.

reader.0.readRetries: The number of attempts that will be made when reading a tag's TID and user memory areas and when authenticating tags. When attempting to read tag memory or authenticate tags, the reader is not searching for new tags. Therefore, a higher number of retries will increase the likelihood of successfully reading tag memory while also increasing the chance of "missing" tags that rapidly enter and leave the reader's field of view. Defaults to 3.

#### 4.2.2.1 Tuning Parameters for ThingMagic Readers

reader.0.useContinuousRead: If true, the software will use the continuous read mode on ThingMagic readers. If false, the software will poll the reader for tags. The best option to use depends on the particular reader configuration and firmware versions in use, so some experimentation may be required. When in doubt, set to true.

reader.0.readCycleLengthMsec: The length of a polling cycle when continuous reading is not being used.

reader.0.readCycleIntervalMsec: The interval between polling cycles when continuous reading is not being used.

reader.0.useIzarAntennaMap: If true, enables direct control of the internal antenna switch on ThingMagic IZAR RFID readers. Only used for specific configurations.

reader.0.[region, hopTable, hopTime, q, tagEncoding, blf, session, target, tari]: This sets internal tuning parameters in the ThingMagic API. For more information regarding these options, see the ThingMagic API documentation.[11]

#### 4.2.2.2 Tuning Parameters for Alien Readers

reader.0.readMode: Controls the read mode used by the software to connect to an Alien RFID reader. Either "TAG_STREAM or AUTO_NOTIFY. Defaults to TAG_STREAM.

#### 4.2.2.3 Tuning Parameters for Venture Readers

reader.0.useSecurityMode: If true, the software will report whether tags are contained on the Venture reader's internal whitelist.

reader.0.readerStatusPollIntervalMsec: The interval in milliseconds at which the reader temperature will be polled and reported to ISA. Defaults to true.

reader.0.digitalInputsActiveLow: If true, the state of the digital inputs will be inverted before being reported to ISA. Defaults to true**.**

### 4.2.3 RFID Reader Antenna Parameters

This section provides an example configuration and definitions of the required parameters for RFID reader 0, antenna 8.

reader.0.antenna.8.latitude: Latitude of the RFID antenna

reader.0.antenna.8.longitude: Longitude of the RFID antenna

reader.0.antenna.8.altitude: Altitude of the RFID antenna

reader.0.antenna.8.name: User-friendly name of the RFID antenna

reader.0.antenna.8.roll: Roll of the RFID antenna for Orientation

reader.0.antenna.8.pitch: Pitch of the RFID antenna for Orientation

reader.0.antenna.8.yaw: Yaw of the RFID antenna for Orientation

reader.0.antenna.8.hfov: Horizontal field of view of the RFID antenna

reader.0.antenna.8.vfov: Vertical of the field of view of the RFID antenna

reader.0.antenna.8.mil2525Symbol: MIL-STD-2525[10] symbol code for the antenna

reader.0.antenna.8.bsoFormat: Selects which tag information will be placed in the BSO field of generated ISA messages. One of "EPC", "TID", or "TID-EPC".

### 4.2.4 Logging and Playback Parameters

reader.0.writeToLog: If true, RFID tag data will be logged to a file in comma-separated value (CSV) format. This file can be played back at a later date or used for offline analysis. The filename is defined in the next entry.

reader.0.outputFileName: This parameter defines the location and name of the output text files when writing messages to a file is enabled (i.e., writeToLog=true). A date timestamp will be appended the outputFileName.

reader.0.playBackMode: If this parameter is true, the software will read and play back messages from a log previously written using the writeToLog function. This function is useful for development and testing when usage of an actual reader is not possible. When in playback mode, the software will not connect to RFID reader hardware or write new log files.

reader.0.playBackFileName: This parameter defines the location and name of the input text files to be read when in playback mode.

## 5. Example Usage Scenarios

This section provides example configurations for different RFID Multi-Reader tool usage scenarios.

### 5.1 RFID Reader that Publishes to the ISA Only

This configuration receives data from RFID reader antennae and publishes them to the ISA network. It does not write the event messages to a logfile or work in playback mode.

This is accomplished using the following settings:

```
writeToLog=false
```

and

```
outputFileName=can be left blank when not being used
```

and

```
playBackMode=false
```

and

```
playBackFileName=can be left blank when not being used
```

## 5.2 RFID Reader that Publishes to the ISA and Writes to a Log File

This configuration receives data from the RFID reader antennae, publishes them to the ISA network, and writes them to a text file. It does not work in playback mode.

This is accomplished using the following settings:

```
writeToLog=true
```

and

```
outputFileName=RFIDdataFileName
```

and

```
playBackMode=false
```

and

```
playBackFileName=can be left blank when not being used
```

## 5.3 RFID Reader that Reads from a File and Publishes to the ISA (Playback)

This configuration reads the data from the designated file and publishes them to the ISA network. This is known as playback mode. It does not write the files to a log file. This is accomplished using the following settings:

```
writeToLog=false
```

and

```
outputFileName=can be left blank when not being used
```

and

```
playBackMode=true
```

and

```
playBackFileName=RFIDdataFileName_2019-05-02-13-22-51.txt
```

# 6. Custom ISA Messages for EPC RFID Readers

The RFID Multi-Reader tool generates the following ISA custom types and messaging formats. These recommended message types and formats have been submitted to the ISA program office for potential inclusion into the standard ISA data model.

The ISA custom types for EPC RFID tag data (RFID Reading) and EPC RFID tag authentication information (RFID Auth Info) are defined in the following sections. The required and optional fields for ISA Event messages conveying data from an EPC RFID tag read operation are also specified.

## 6.1 Custom ISA Observables

Table 1 shows the custom ISA observables.

**Table 1 Custom ISA observables**

| Observable | ISA Type | Description |
|---|---|---|
| **RFID Reading** | RFID Reading | Provides information about the RFID Tag. |
| **RFID Auth Data** | RFID Auth Info | RFID Tag authentication data |

Note: Fields highlighted in gray are required; the unhighlighted fields (in the subsequent tables) are optional.

## 6.2 Custom RFID Event

The RFID Tag Reading Event is a custom ISA Event of type Detection used to send information describing the detected RFID Tag. It includes information about the RFID tag detected and the location of the RFID Tag reader. Multiple ISA Tag Reading messages can be correlated together at higher echelons to track the objects identified by the RFID tag. This type is designed to accommodate EPC Gen2 (also known as ISO 18000-6C) RFID tags and may not be applicable to other tag technologies. Table 2 details the custom RFID event.

**Table 2 Custom RFID event**

| Category | Name | Custom? | ISA Type | Description |
|---|---|---|---|---|
| Observables | **BSO** | No | BSO ID | Key used to track updates to an object. For RFID readings, this is constructed from a combination of the tag's EPC and TID values. |
| | **RFID Reading** | Yes | RFID Reading | Information about the RFID Tag read. |
| | **RFID Auth Data** | Yes | RFID Auth Info | Raw authentication data from an EPC Gen2V2 RFID Tag. |
| Detector Properties | **Position** | No | Geographic Position | Position of the RFID reader antenna. |
| | **Identity** | No | Standard Identity | Key aspects of the component using MIL-STD-2525[10] nomenclature. |
| | **Field of View** | No | Field of View | Field of view of the RFID reader antenna at the given position. |
| | **Orientation** | No | Rotation | Orientation of the RFID reader antenna at the given position. |

Note: Fields highlighted in gray are required; the unhighlighted fields are optional.

## 6.3 Custom RFID Reading Type

This custom ISA type includes the information from an EPC Gen2 RFID tag reading. It contains the field-programmable EPC value from the tag, the factory-set manufacturer, model, and serial numbers of the tag, and several other parameters relating to the tag read event. The use of the fixed tag serial number in addition to the programmable EPC value provides a minimum level of security against tag cloning. (The use of the programmable EPC value by itself provides no security against tag cloning.) If higher security is required, AES authenticated tags should be employed and the authentication results reported via the Authentication Status field and the RFID Auth Data observable.

The current implementation of the RFID reader tool will not report authentication statuses of COMPLETED_AUTH_FAILED or COMPLETED_AUTH_PASSED. It will report either COMPLETED_NO_KEY or one of the failure statuses. The PASSED/FAILED statuses are defined in Table 3 to allow for future expansion.

**Table 3 Custom RFID reading type**

| Data | ISA type | Description |
|---|---|---|
| **Read Count** | Integer | Number of times the reader has read the RFID Tag since the last report |
| **RSSI** | Integer | RFID Received Signal Strength Indicator for the most recent tag reading |
| **TID** | Byte Stream | Raw contents of the TID memory bank of the RFID Tag. These data uniquely identify the tag hardware. The structure of the TID data is specified by the GS1 EPC Tag Data Standard. Note that on some tags incorporating proprietary TID extensions (notably tags from Alien Technology), certain portions of this value may change every time the tag is read. |
| **EPC** | String | Contents of the EPC memory bank of the RFID Tag, sent as hex. In most applications, these data identify the item the tag is attached to. |
| **Authentication Status** | String | RFID Authentication Status value. One of the following:<br>• NOT_ATTEMPTED: authentication of the tag was not attempted.<br>• NOT_SUPPORTED_BY_TAG: tag does not support authentication.<br>• NOT_SUPPORTED_BY_READER: reader does not support tag authentication.<br>• COMPLETED_AUTH_FAILED: authentication operation completed successfully, tag failed authentication.<br>• COMPLETED_AUTH_PASSED: authentication operation completed successfully, tag passed authentication.<br>• COMPLETED_NO_KEY: the low-level authentication operation completed successfully, but the results could not be verified as no authentication key for this tag was available at the reader. Raw authentication challenge/response data are reported in the RFID Auth Data observable. An external process with access to the needed keys can use the contents of the RFID Auth Data observable to authenticate the tag.<br>• ERROR: authentication was attempted, but did not complete. |
| **RFID TAG Manufacturer** | String | The name of the manufacturer of the RFID Tag. Obtained from the Mask Designer ID (MDID) field of the tag's TID memory bank. The list of valid manufacturer names is maintained by GS1 and is available at https://www.gs1.org/epcglobal/standards/mdid. |
| **RFID Tag Model** | Integer | The model number of the RFID Tag. Obtained from the Tag Model Number (TMN) field of the tag's TID memory bank. |
| **RFID Tag SN** | String | The factory-assigned serial number of the RFID tag. This is obtained by parsing the extended tag ID (XTID) data from the tag's TID memory bank, if available. For tags that do not implement XTID, this is obtained by parsing the tag's manufacturer-proprietary extended TID data format. |

Note: Fields highlighted in gray are required; the unhighlighted fields are optional.

## 6.4 Custom RFID Auth Info Type

This custom ISA type includes the raw authentication data from an EPC Gen2V2 RFID Tag. This information can be used by an application for "offline" authentication of tags when a tag's keys were not available at the reader at the time the tag was read.

Gen2V2 tags can be programmed with two AES encryption keys. Once programmed, the keys cannot be changed or extracted from the tag by any documented means. Readers can use a challenge/response protocol to verify the keys stored on the tag. This allows an application to confirm via cryptographic means that the tag was issued by someone who was in possession of a particular AES key, providing a mechanism for detecting and defeating tag cloning.

Gen2V2 tags support two authentication modes: TAM1 and TAM2. TAM1 mode supports tag authentication using either key 0 or 1. TAM2 mode supports both authentication and encrypted reads of tag memory data but is restricted to using key 1. The mode, key ID, and TAM2 memory bank/length/offset settings are configured on the RFID reader and sent to the tag during the read operation. These values are reported in this message to allow applications to decode and interpret the results of the authentication operation without requiring a priori knowledge of the RFID reader configuration. Table 4 details the custom RFID Auth Info type.

**Table 4 Custom RFID Auth Info type**

| Data | ISA Type | Description |
|---|---|---|
| **Key ID** | Integer | Key ID (0 or 1) used for the authentication operation |
| **Mode** | String | Mode (TAM1 or TAM2) used for the authentication operation |
| **Auth Challenge** | Byte Stream | Challenge value randomly generated by the reader and used to authenticate the tag |
| **Auth Response** | Byte Stream | Response value received from the tag. Generated on the tag by concatenating a random IV with the challenge and then AES encrypting the result with the selected key. |
| **Crypted Data** | Byte Stream | Encrypted tag data from a TAM2 read |
| **Crypted Data Bank** | String | Memory bank identifier for the encrypted data. One of EPC, TID, or USER. |
| **Crypted Data Offset** | Integer | Offset in the selected memory bank for the encrypted data. (Offset applies after the data are decrypted.) |

Note: Fields highlighted in dark gray are required; the unhighlighted fields are optional.

## 7. Conclusion

The RFID Multi-Reader software provides a useful tool for connecting commercial RFID readers to the ISA networks commonly employed within military sensing systems. This package has been deployed to ARL customers and is also being used within the ARL NSRTB. Within the NSRTB environment, the RFID Multi-Reader tool is used to provide ground-truth data and as a surrogate sensor for development of target location prediction algorithms.

The software package is Distribution C (Distribution is authorized to US Government Agencies and their contractors).

The software package is fully US Government-owned and is provided to Government and contractor stakeholders upon request. Interested parties should contact the authors to obtain a copy of the software.

## 8. References


1. Radio frequency identification reader (RFID reader). Technopedia; 2020 [accessed 16 Sep 2019]. https://www.techopedia.com/definition/26992/radio-frequency-identification-reader-rfid-reader.

2. EPC/RFID. Ewing Township (NJ): GS1 US; n.d. [accessed 2020 June 17]. https://www.gs1.org/standards/epc-rfid.

3. Integrated Systems Architecture (ISA). Fort Belvoir (VA): Night Vision and Electronic Sensors Directorate, Army Combat Capabilities Development Command Control, Computers, Communications, Cyber, Intelligence, Surveillance and Reconnaissance Center (US); n.d. [accessed 2019 July 18]. https://confluence.di2e.net/display/ISA/.

4. Poltronieri F, Sadler L, Benincasa G, Gregory T, Harrell JM, Metu S, Moulton C. Enabling efficient and interoperable control of IoBT devices in a multi-force environment. 2018 IEEE Military Communications Conference (MILCOM); 2018 Oct 29–31; Los Angeles, CA. IEEE; c2019. p. 757–762.

5. ISA data model specification, release 6.0, document revision 7. Fort Belvoir (VA): Night Vision and Electronic Sensors Directorate, Army Combat Capabilities Development Command Command, Control, Computers, Communications, Cyber, Intelligence, Surveillance and Reconnaissance Center (US); 2019 Jan 16.

6. EPC Tag Data Standard (TDS). Ewing Township (NJ): GS1 US; 2019 Nov [accessed 2020 June 17]. https://www.gs1.org/standards/epc-rfid/tds.

7. Application note AN11778: how to use the UCODE DNA. Eindhoven (The Netherlands): NXP Semiconductors; 2015 Nov 30.

8. ISA 101 breakout session. Fort Belvoir (VA): Night Vision and Electronic Sensors Directorate, Army Communications Electronics Research, Development, and Engineering Center (US); 2018 Nov 7.

9. ISA certificate usage guide, release 6.0, document revision 1. Fort Belvoir (VA): Night Vision and Electronic Sensors Directorate, Army Combat Capabilities Development Command Command, Control, Computers, Communications, Cyber, Intelligence, Surveillance and Reconnaissance Center (US); 2020 Feb 27.

10. Department of Defense. Department of Defense interface standard: common warfighting symbology. Washington (DC): Department of Defense (US); 2008 Nov 17. Standard No.: MIL-STD-2525C.

11. Mercury API Programmer's Guide for Mercury API v1.27.3 and later. North Syracuse (NY): Jadak; 2020 [accessed 2020 June 17]. https://www.jadaktech.com/resources/rfid-document-library/?rfid_series=software&rfid_product_name=thingmagic-mercury-api.

## List of Symbols, Abbreviations, and Acronyms

| | |
|---|---|
| AES | Advanced Encryption Standard |
| API | application programming interface |
| ARL | Army Research Laboratory |
| Auth Info | authentication information |
| C5ISR | Command, Control, Computers, Communications, Cyber, Intelligence, Surveillance and Reconnaissance |
| CCDC | US Army Combat Capabilities Development Command |
| COTS | commercial off-the-shelf |
| eAPI | extensible API |
| EPC | Electronic Product Code |
| GOTS | government off-the-shelf |
| IP | Internet Protocol |
| IQL | ISA Query Language |
| ISA | Integrated Sensor Architecture |
| JDK | Java Development Kit |
| JRE | Java Runtime Environment |
| LLRP | Low Level Reader Protocol |
| NSRTB | Networked Sensor Research Test Bed |
| NVESD | Night Vision and Electronic Sensors Directorate |
| RFID | radio-frequency identification |
| SDK | software development kit |
| SOA | Service-Oriented Architecture |
| TCP | Transmission Control Protocol |
| TID | tag identifier |
| TLS | transport layer security |
| UCI | Universal Component Identifier |

| | |
|---|---|
| 1 (PDF) | DEFENSE TECHNICAL INFORMATION CTR DTIC OCA |
| 1 (PDF) | CCDC ARL FCDD RLD CL TECH LIB |
| 2 (PDF) | CCDC ARL FCDD RLS SI J KOVACH<br>FCDD RLS SI L SADLER |
| 2 (PDF) | CCDC C5ISR CENTER NVESD C MOULTON<br>M HARRELL |