CLAS
Configuration Location Analytics Software

Server and Software Installation Guide
Copyright

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners. ©2019 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

COPYRIGHTS & TRADEMARKS: For complete copyright and trademark information, go to www.zebra.com/copyright.

WARRANTY: For complete warranty information, go to www.zebra.com/warranty.

END USER LICENSE AGREEMENT: For complete EULA information, go to www.zebra.com/eula.

Terms of Use

• Proprietary Statement
  This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries (“Zebra Technologies”). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

• Product Improvements
  Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

• Liability Disclaimer
  Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

• Limitation of Liability
  In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.
# Revision History

Changes to the original guide are listed below:

<table>
<thead>
<tr>
<th>Change</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-01 Rev A</td>
<td>11/2018</td>
<td>Initial Release</td>
</tr>
</tbody>
</table>
| -02 Rev A | 4/2019  | Updates:  
- Configuring RTLS code.  
- Various URLs.  
- aar_info.csv file name.  
- Guide name change.  
- Chapter name changes to add CLAS.  
- Updated Installing Certificates.  
- Updated code under Configuring RTLS.  
- Replace Table 3 under Advanced Configuration Options.  
- Updated figures/diagrams.  

Additions:  
- Standard About This Guide sections.  
- Local License Server Administrator Guide for Windows/Linux references.  
- Registry and legacy methods RTLS installs.  
- Preparing to run RTLS.  
- Added step to Registry Method. |
# Table of Contents

Copyright ........................................................................................................................................... 2
Terms of Use .................................................................................................................................... 2
Revision History ............................................................................................................................... 3

About This Guide
Introduction ........................................................................................................................................ 6
Chapter Descriptions .......................................................................................................................... 6
Notational Conventions ...................................................................................................................... 6
Related Documents ........................................................................................................................... 7
Service Information ............................................................................................................................ 7
Provide Documentation Feedback .................................................................................................... 7

ZAATS Introduction
Introduction ........................................................................................................................................ 8
Product Overview ............................................................................................................................... 8
  Description and Features .................................................................................................................. 8
  CLAS Software (RTLS Services) ....................................................................................................... 10
  RTLS Configuration and Management Server ............................................................................... 10
  Location Analytics (LA) .................................................................................................................. 11
  Radio Control & Data (CND) .......................................................................................................... 11
  REST Interface ............................................................................................................................... 12
  Location Data Endpoint .................................................................................................................. 12

Getting Started
Introduction ....................................................................................................................................... 13
  Overview ......................................................................................................................................... 13
  Licensing Notes ............................................................................................................................... 13
Hardware Requirements ..................................................................................................................... 13
  Decentralized LA ............................................................................................................................ 13
  Centralized LA ............................................................................................................................... 13
Software Requirements .................................................................................................................... 14
Installation Checklist ....................................................................................................................... 14

Installing CLAS Software
Introduction ........................................................................................................................................ 17
  Installing RTLS Behind a Proxy Server ......................................................................................... 17
Installing RTLS .................................................................................................................................. 19
  Installation Procedure .................................................................................................................... 19
  Installing Certificates .................................................................................................................... 20
    Installing Certificates for Secure REST Interface ..................................................................... 20
    Installing Certificates for Secure Kafka Communication ......................................................... 21
  Changing the Password for the REST Interface .......................................................................... 21
Configuring RTLS ............................................................................................................................ 22
  Basic Configuration Options ......................................................................................................... 24
  Advanced Configuration Options .................................................................................................... 25
Adding ATRs to RTLS ....................................................................................................................... 27
Preparing to Run RTLS ..................................................................................................................... 27
Starting and Stopping RTLS ................................................................. 28
Starting RTLS ..................................................................................... 28
Stopping RTLS ................................................................................... 28

RTLS Simulator
Overview .......................................................................................... 29
Simulator Architecture ....................................................................... 29
Setting Up the Simulator .................................................................. 30
Simulator Profiles ............................................................................. 31
  Adding New Profiles .................................................................... 31
Running the RTLS Simulator ............................................................. 31

Validating a CLAS/RTLS Services Installation
Introduction ....................................................................................... 32
Installing the Kafka Broker ............................................................... 32
Start Simulator .................................................................................. 33
Running RTLS with the RTLS Simulator .......................................... 33
Start RTLS ......................................................................................... 33
Verification ......................................................................................... 34
Stop Simulation .................................................................................. 35

CLAS and RTLS Licensing
Introduction ....................................................................................... 36
RTLS Licensing Model ..................................................................... 36
RTLS Licensing Mechanism ............................................................. 37
  License Acquisition Modes ............................................................. 38
    Cloud-Based License Server ....................................................... 38
    Local License Server ................................................................ 39

Running RTLS with ATR7000s
Introduction ....................................................................................... 40
Bringing the RTLS System Live with ATR7000 Readers .................. 40
Introduction

The Configuration Location Analytics Software (CLAS) Server and Software Installation guide provides information about the requirements and configuration for a server hosting CLAS RTLS software. It also describes the installation of RTLS software onto that server.

Chapter Descriptions

Topics covered in this guide are as follows:

- **ZAATS Introduction** provides an overview of Zebra’s Advanced Asset Tracking System, including the ATR7000 overhead RFID readers and the CLAS/RTLS Services software.
- **Getting Started** provides an overview of the steps required to install and validate a CLAS/RTLS Services software deployment, including hardware, software, and network related requirements.
- **Installing CLAS Software** describes the procedure to install and configure the various software components that comprise RTLS.
- **RTLS Simulator** details the SW emulated ATR7000s used as simulators with RTLS.
- **Validating a CLAS/RTLS Services Installation** provides information on the sequence of operations required to verify an RTLS Services software and server installation with the RTLS Simulator.
- **CLAS and RTLS Licensing** explains the licensing model and the licensing mechanism used in RTLS.
- **Running RTLS with ATR7000s** describes running RTLS with the ATR7000s hardware.

Notational Conventions

The following conventions are used in this document:

- The **Consolas** font is used throughout the guide to represent code.
- **Bold** text is used to highlight the following:
  - Dialog box, window and screen names
  - Drop-down list and list box names
  - Check box and radio button names
  - Icons on a screen
  - Key names on a keypad
  - Button names on a screen.
• Bullets (•) indicate:
  • Action items
  • Lists of alternatives
  • Lists of required steps that are not necessarily sequential.
• Sequential lists (e.g., those that describe step-by-step procedures) appear as numbered lists.

Related Documents

The following documents provide more information about ZAATS.

• ZAATS Tag and Numbering Guide, p/n MN-003199-xx.
• ATR7000 Integration Guide, p/n MN-003191-xx.
• Local License Server Administrator Guide for Windows, p/n MN-003302-xx.
• Local License Server Administrator Guide for Linux, p/n MN-003459-xx.

For the latest version of this guide and all guides, go to: zebra.com/support.

Service Information

If you have a problem with your equipment, contact Zebra Global Customer Support for your region. Contact information is available at: zebra.com/support.

When contacting support, please have the following information available:

• Serial number of the unit
• Model number or product name
• Software type and version number.

Zebra responds to calls by email, telephone or fax within the time limits set forth in support agreements.

If your problem cannot be solved by Zebra Customer Support, you may need to return your equipment for servicing and will be given specific directions. Zebra is not responsible for any damages incurred during shipment if the approved shipping container is not used. Shipping the units improperly can possibly void the warranty.

If you purchased your Zebra business product from a Zebra business partner, contact that business partner for support.

Provide Documentation Feedback

If you have comments, questions, or suggestions about this guide, send an email to EVM-Techdocs@zebra.com.
Introduction

Zebra’s Advanced Asset Tracking System (ZAATS) provides continuous identification, location, and tracking of items tagged with passive UHF RFID tags conforming to the GS1 EPC™ Radio Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface standard. ZAATS is designed to enhance the efficiency and workflows of Zebra’s customers’ operations, which are increasingly focused on cohesive, real-time data.

ZAATS consists of two primary components: the CLAS software (RTLS Services software), which contains the configuration, management, and location analytics components; and the ATR7000 overhead array readers.

Product Overview

Description and Features

ZAATS is a passive UHF RFID based asset tracking solution developed primarily for indoor warehousing, manufacturing, and logistics applications. It is based on the ATR7000 overhead RFID reader containing Zebra’s proprietary advanced array architecture with integral antenna capable of steering beams and estimating the bearing to RFID tagged items with unprecedented accuracy and speed.

A summary of the key system and product features of ZAATS includes:

- A passive UHF RFID RTLS system that provides real-time identification and location data of tagged items for continuous asset monitoring.
- RTLS Software that configures, controls, and manages the system, as well as a high-performance location analytics engine capable of providing up to 10,000 tag location estimates per second with 2-foot typical accuracy.
- APIs to configure, manage, and control the ZAATS system using an HTTP-based RESTful interface.
- Docker container virtualization to simplify integration and deployment into end-user and partner applications.
- Software tools and documentation to facilitate system installation, including site planning, calibration, initial start-up, and deployment validation testing.

Figure 1 illustrates the high-level architecture of the ZAATS system, showing the two main components of the ZAATS system: 1) the AARs (ATR7000 Advanced Readers), and 2) the CLAS (RTLS Services) software.
**Figure 1** Zebra Advanced Asset Tracking System (ZAATS)

The ATR7000 overhead array readers are EPC Gen2 readers with integral phased array antenna capable of steering beams and estimating the bearing (angle of arrival) of EPC Gen2 tags. This is the first product in the RFID portfolio to be based on a Zebra proprietary advanced array architecture that provides unprecedented location accuracy and real time tracking of RFID tags.

A summary of key product features of the ATR7000 is listed below:

- Integral 14 element antenna array.
- Advanced multi-channel radio architecture provides accurate bearing estimations in a single read.
- Host software compatible with Zebra’s family of fixed RFID readers with support for embedded and external applications.
- Integration of Zebra’s proprietary ASIC-based RFID radio.
- GPIO with external power for driving actuators and sensors.

**NOTE:** In Figure 1, the solid lines correspond to configuration, control, and management interfaces, while the dashed lines correspond to data interfaces. Dashed, red lines carry tag ID and bearing information and require high bandwidth and low-latency, and therefore, typically reside on the same segment of a local area network.
• Support for several standard mounting options to simplify installation.
• Two power options; 802.3at Power over Ethernet (PoE+) or external +24 VDC power supply.
• Environmental specifications suitable for industrial and warehouse applications (-20 C to +55 C operation and IP51 sealing).

CLAS Software (RTLS Services)

RTLS Services serves as a data aggregator that executes location analytics to estimate the tag location and reports out unique tag ID, location, and time-stamp in quasi-real-time.

RTLS Services performs the following primary functions:

• Discovers readers on a local network.
• Configures each reader to read tags and report the estimated bearings.
• Estimates the tag location based on the bearings reported by the reader.
• Reports the location estimates to a “location endpoint.”
• Provides software interfaces to middleware applications that enable end-user solutions to associate items with identity, location and movement information and to deliver the business logic to streamline operations or work-flows.
• Provides an interface to the outside world to configure and manage the RTLS system.
• Provides interface to manage and configure the ATR7000 readers in the facility.
• Provides license management functions for RTLS and CLAS software.

The CLAS/RTLS Services software consists of three major components:

• RTLS Configuration and Management Server
• Location Analytics (LA)
• Radio Control & Data (CND)

RTLS Services is deployed as a group of Docker containers. A container is the mechanism that minimizes operating system and hardware dependencies, as well as “isolates” RTLS from the other software components that comprise a solution, allowing them to coexist on the same infrastructure. It is expected that RTLS Services typically resides on the same physical server as the solution.

A description of these and other components important to system operation follow below.

RTLS Configuration and Management Server

As shown in Figure 1, the RTLS configuration and management server is the primary component within RTLS Services responsible for managing and configuring the system, including system start-up and reset, reader discovery, initial and ongoing configuration of LA and Radio CND, firmware and software upgrades, etc. RTLS Configuration and Management is a component of RTLS Services that is always resident on the server.

RTLS also provides the configuration and management interfaces (API) to the solution software through a RESTful interface, a common framework found in enterprise environments.
Location Analytics (LA)

Figure 1 also illustrates that Location Analytics (LA) is the primary component within RTLS Services responsible for aggregating bearing information received from the ATR7000 overhead readers, estimating x-y-z tag location, determining if a tag is moving (dynamic) or not moving (static), applying additional advanced algorithms that enhance static and dynamic location (tracking) accuracy, and reporting a final tag location estimate with meta-data (EPC, timestamp, etc) to a data endpoint using a messaging broker. LA also has the capability of combining raw bearing and location estimates from multiple RFID tags affixed to the same asset (e.g. forklifts) to improve overall location accuracy and/or provide orientation and directionality information.

In addition to supporting a deployment mode with a server-based LA, in most deployments ZAATS utilizes a reader-based, distributed LA to offload the primary server from the intensive processing requirements of the location analytics (LA) or to support cloud-based solutions where RTLS Services software is deployed in an off-premise server. A ZAATS solution with distributed LA is illustrated in Figure 2 on page 12. The figure illustrates three AARs for simplicity, although, operation is designed to scale up to the maximum of 255 readers per site.

In either case, centralized or distributed, the interface between LA and the Radio C&D is optimized to be a high bandwidth, low latency one-way interface that carries only tag ID and bearing information, as indicated by the red arrows in the figure.

Radio Control & Data (CND)

The Radio Control & Data (CND) component is a reader-based application (process) that configures, controls, and maintains a connection to the RFID radio (engine), receives tag bearing reports from the radio and passes them to LA to ensure the timestamps on the bearing reports are synchronized to the system time source.

RTLS is a licensed software that requires a valid license for it to run with ATR7000 readers. CND is the component within RTLS responsible for verifying and enforcing that a valid CLAS license is available. While CND is considered a component of RTLS Services, it is deployed by the RTLS Configuration and Management Server to the readers at system start-up.
The REST Interface is the primary mechanism to configure and manage the ZAATS system. It supports the ability to upgrade software of the readers and RTLS Services, query the version, status, and configuration of the RTLS system; start and stop the system; and reboot the ATR7000 readers. It also supports setting user-defined filters specifying the frequency and format of tag data.

NOTE: In Figure 2, the solid lines correspond to configuration, control, and management interfaces, and the dashed lines correspond to data interfaces. Dashed, red lines carry tag ID and bearing information and require high bandwidth and low-latency, and therefore, typically reside on the same segment of a local area network.

REST Interface

The REST Interface is the primary mechanism to configure and manage the ZAATS system. It supports the ability to upgrade software of the readers and RTLS Services, query the version, status, and configuration of the RTLS system; start and stop the system; and reboot the ATR7000 readers. It also supports setting user-defined filters specifying the frequency and format of tag data.

Location Data Endpoint

The location data endpoint is a broker that receives the location update messages from the Location Analytics block and RTLS. The RTLS end users’ middleware application can consume these location update messages from the broker to transform information about asset location into solutions that enhance efficiency and work-flows of end user operations.
Introduction

This chapter provides an overview of the steps involved in installing and setting up the CLAS/RTLS Services software, the primary software component of Zebra’s Advanced Asset Tracking System (ZAATS). It also covers system requirements and prerequisites, including licensing, hardware, software, and networking.

Overview

The following aspects of an RTLS installation are covered as part of this guide.

• Setting up the system with prerequisites for running RTLS Services software.
• Installing and configuring RTLS Services software.
• Installing the RTLS simulator.
• Validating the RTLS server and software installation using the RTLS simulator.
• Setting up the licensing server and acquiring licenses for running RTLS Services.
• Running RTLS Services with ATR7000 hardware.

Hardware and other deployment aspects are covered in the ZAATS Deployment Guide (p/n MN-003195-xx).

Licensing Notes

CLAS and RTLS Licensing explains the steps required to acquire CLAS licenses and install the license server. CLAS licenses are not required to run RTLS Services with the RTLS simulator. However, CLAS licenses are required when RTLS runs in a system using ATR7000 readers.

Hardware Requirements

The following describes the hardware requirements for a Decentralized LA or Centralized LA configuration.

Decentralized LA

• Quad Core CPU @ 2.4 GHz
• 16 GB RAM
• 16 GB of free hard disk space

Centralized LA

• Intel Xeon CPU E5-2637 v4 @ 3.50GHz
• 64 GB RAM
• 200GB SSD or 5TB HD
Software Requirements

The following describes the operating system and software required to be installed on the server before installing CLAS/RTLS.

- Ubuntu v16.0.4 LTS or higher
- Docker v17.05 or higher
  Refer to the following link to setup Docker:
  https://docs.docker.com/install/linux/docker-ce/ubuntu/
  Be sure to add the 'stable' repository and not the 'nightly' repository.
- Docker Compose v1.8.0 or higher
  Refer to the following link to setup Docker Compose:
  https://docs.docker.com/compose/install/
  Be sure to install the 'stable' release.
- ssh server (if remote access is required)
  The latest version of openssh-server is recommended.
- Local license server (if required)
  See Local License Server on page 39.

Installation Checklist

A CLAS/RTLS software installation involves several steps, all of which should be performed in a precise order. Table 1 provides an outline of the steps. Each step is explained in more detail in their respective sections.

Table 1  CLAS/RTLS Software Installation Steps

<table>
<thead>
<tr>
<th>Step Number</th>
<th>Action Required</th>
<th>Section Reference</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Perform prerequisite steps.</td>
<td>Hardware Requirements on page 13</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software Requirements on page 14</td>
<td></td>
</tr>
<tr>
<td>1a</td>
<td>Additional steps if installing RTLS behind a proxy.</td>
<td>Installing RTLS Behind a Proxy Server on page 17</td>
<td>If RTLS is being installed behind a proxy, then RTLS and Docker must to configured to work behind a proxy.</td>
</tr>
<tr>
<td>2</td>
<td>Install RTLS.</td>
<td>Installing CLAS Software</td>
<td>The next step is to install the RTLS Services software.</td>
</tr>
<tr>
<td>3</td>
<td>Configure RTLS.</td>
<td>Configuring RTLS on page 22</td>
<td>This step involves editing the RTLS configuration files rtls.conf and aar_info.csv using the information obtained in the previous steps.</td>
</tr>
</tbody>
</table>
**Table 1** CLAS/RTLS Software Installation Steps (Continued)

<table>
<thead>
<tr>
<th>Step Number</th>
<th>Action Required</th>
<th>Section Reference</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Validate RTLS installation.</td>
<td>Validating a CLAS/RTLS Services Installation</td>
<td>This step involves installation of the necessary modules like the Kafka broker and RTLS simulator to validate the RTLS installation.</td>
</tr>
<tr>
<td>5</td>
<td>Setup Licensing.</td>
<td>CLAS and RTLS Licensing</td>
<td>This is an essential step before starting RTLS with physical ATR7000 hardware. If the readers don't have Internet access, then a Local License server must be installed. Follow the instructions provided in the relevant Local License Server Administrator Guide listed in the Related Documents section.</td>
</tr>
<tr>
<td>6</td>
<td>Run RTLS with ATR7000s.</td>
<td>Running RTLS with ATR7000s and Post-Installation ZAATS Validation chapter in the ZAATS Deployment Guide.</td>
<td>This is the final step of an RTLS Services software installation. It is performed as part of the deployment process described in ZAATS Deployment Guide.</td>
</tr>
</tbody>
</table>

**Table 2** Required Installation and Configuration Items

<table>
<thead>
<tr>
<th>Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Address of Server</td>
<td>The IP address of the server that is reachable by the ATR7000s. This is required for the deploy.env file. See Table 3 on page 19 for more information.</td>
</tr>
<tr>
<td>Hostname/IP Address of Kafka Broker</td>
<td>The hostname and/or IP address and TCP Port of the Kafka broker are required in the rtls.conf file. See Configuring RTLS on page 22 for more information. Additionally, when using the set up described in setting up a Kafka Broker for validation, the Advertised Host Name should be a host name or IP address that is reachable by the ATR7000s. This is required in the docker-compose-single-broker.yml file. See Installing the Kafka Broker on page 32 for more information.</td>
</tr>
</tbody>
</table>
| TCP Ports | RTLS uses the following set of ports on the host machine:  
  • 123 for NTP service  
  • 20, 21 and the range of ports specified in RTLS_FTP_PASV_PORT_MIN and RTLS_FTP_PASV_PORT_MAX variables for FTP service  
  • 5159, 5160 and the ports specified in REST_PORT while installing RTLS  
See Table 3 on page 19 for more details on the configurable port ranges. |
**Table 2** Required Installation and Configuration Items (Continued)

<table>
<thead>
<tr>
<th>Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy Information</td>
<td>Required if installing RTLS behind a proxy. The configuration of the proxy is needed.</td>
</tr>
<tr>
<td>RTLS License</td>
<td>This information is required before RTLS can be run with physical readers after hardware installation. However, the user may install and validate RTLS with simulators even without an activation ID.</td>
</tr>
<tr>
<td>Activation ID</td>
<td>See <a href="#">CLAS and RTLS Licensing</a> for more information.</td>
</tr>
</tbody>
</table>
Installing CLAS Software

---

**Introduction**

This chapter describes the various steps required to install and configure CLAS/RTLS Services software. It explains different install time and runtime configurations and explains each configuration option in detail, including:

- Installing RTLS using the RTLS release package
- Installing certificates
- Changing the password for the REST Interface
- Configuring RTLS Services

---

**Installing RTLS Behind a Proxy Server**

If RTLS is being installed behind a proxy server, then Docker must be configured appropriately before RTLS is installed. If not installing RTLS behind a proxy, continue to Installing RTLS on page 19.

Since RTLS deployment involves the installation of various packages using the Ubuntu Package Manager, the HTTP/HTTPS proxy must be supplied. However, since RTLS Services also uses http to communicate with simulators and readers, the proxy must be disabled during runtime.

To setup the Docker service to use a proxy during deployment, follow these steps:

1. Create a system drop-in directory for the Docker service:
   
   ```
   sudo mkdir -p /etc/systemd/system/docker.service.d
   ```

2. Create the appropriate configuration file.
   
   a. For an HTTP proxy:
      
      i. Create a file called
         
         `/etc/systemd/system/docker.service.d/http-proxy.conf`
      
         that adds the following HTTP_PROXY environment variable
         
         ```
         [Service]
         Environment = "HTTP_PROXY=http://proxy.example.com:80/"
         ```
   
   b. For an HTTPS proxy:
      
      i. Create a file called
         
         `/etc/systemd/system/docker.service.d/https-proxy.conf`
      
         that adds the following HTTPS_PROXY environment variable
         
         ```
         [Service]
         Environment = "HTTPS_PROXY=https://proxy.example.com:443/"
         ```
3. Flush changes:
   `sudo systemctl daemon-reload`

4. Flush changes:
   `sudo systemctl restart docker`

5. Go to each of the following directories under the rtls directory and perform the steps below.
   Directories: cmm_client, disco, ftpd, httpd, loc_anal, rtls_configuration
   
   a. Create a file named badproxy. Copy and paste into the file the following:
      
      ```
      Acquire::http::Pipeline-Depth 0;
      Acquire::http::No-Cache true;
      Acquire::BrokenProxy true;
      ```
      
   b. Edit the Dockerfile and copy and paste the following line after the first line FROM ...
      
      ```
      COPY ./badproxy /etc/apt/apt.conf.d/99fixbadproxy
      ```

To setup Docker to remove proxy during runtime, follow these steps:

1. In the home directory of the user which starts the containers, create or edit the file `~/.docker/config.json`.
2. Add the following JSON example. Adding this JSON example clears the proxy settings at runtime.
   ```json
   {
     "proxies":
     {
       "default":
       {
         "httpProxy": ""
       }
     }
   }
   ```
Installing RTLS

Installing RTLS Services requires downloading the RTLS Services package from the Zebra Support site, extracting the package, modifying the appropriate parameters, and using the extracted scripts to build and deploy the Docker images locally with the included scripts.

Installation Procedure

To install RTLS follow the steps below:

1. Download the RTLS Services software package (`rtls_services_x.y.z.tar.gz`) from [www.zebra.com/support](http://www.zebra.com/support).
2. Extract the RTLS release package:
   ```
   tar-xzf rtls_services_x.y.z.tar.gz
   ```
   Change to the `rtls` directory:
   ```
   cd rtls
   ```
3. Edit the `deploy.env` file and set the parameters accordingly. The descriptions for the parameters in this file are provided in Table 3 on page 19.

   **NOTE:** Parameters are case sensitive.

   **NOTE:** Under typical circumstances, only the `RTLS_AAR_IFC_IP` in the `deploy.env` file will need to be updated.

4. Deploy RTLS by running:
   ```
   ./rtls.sh deploy
   ```

   **NOTE:** The user running `rtls.sh` must be in the docker group. If the user is not, `sudo` should be used when running `rtls.sh`.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST_PORT</td>
<td>This parameter configures the port on which the RTLS server exposes its REST interface.</td>
<td>80</td>
</tr>
<tr>
<td>RTLS_AAR_IFC_IP</td>
<td>Setting this parameter is mandatory. If the RTLS host machine has a single network interface, this parameter should set to the IP address of the host machine that is running on the RTLS server. In cases where the host machine has multiple network interfaces, this parameter should be set to the IP of the network interface that is in the same subnet as the ATRs.</td>
<td>0.0.0.0</td>
</tr>
</tbody>
</table>
There are two sets of certificates that can be installed in RTLS, both of which must be present in the path mentioned in the CERTIFICATES_PATH parameter with different file names as described later.

- Certificates for secure REST interface
- Certificate for secure Kafka communication

### Installing Certificates for Secure REST Interface

When supplying certificates for the REST interface, the following conditions must be met for certificates to be installed properly.

- The REST interface certificate and key must be supplied in pem format.
- They must have the file name rtls_server.crt.pem and rtls_server_key.pem, respectively.
- The key must not be password protected.

If there is no user-supplied certificates and SSL_ENABLE is set to YES, the CERTIFICATE_PATH should remain the default (shared_vol). This causes RTLS to create self-signed certificates for use on the REST interface.
Installing Certificates for Secure Kafka Communication

RTLS supports publishing location information to a secure Kafka broker. If the Kafka broker is configured for SSL communication, the client certificate must be supplied while deploying RTLS. The client certificate must be present in the CERTIFICATES_PATH as mentioned in deploy.env and it must be named ca-cert.

When RTLS starts, it looks for a certificate that is named ca-cert in CERTIFICATES_PATH and uses it for securing Kafka communication. If a certificate with the name ca-cert is not found at startup, RTLS uses non-secure/plain-text mode of communication with the Kafka broker.

Changing the Password for the REST Interface

By default, the REST interface uses the following credentials:

- Username: rtlsadmin
- Password: Z@@t$R1l$

This can be changed after deploying RTLS Services. The mechanism to do this is described below. Changing the default user password is only possible while RTLS is running.

To change the default user password, follow these steps:

1. Enter the rtls directory
   
   ```
   cd rtls
   ```

2. Start rtls
   
   ```
   sudo ./rtls.sh start
   ```

   **NOTE:** The user running rtls.sh must be in the docker group. If the user is not, sudo should be used when running rtls.sh.

3. Run
   
   ```
   docker exec -it rtls-httpd-container bash
   ```

   **NOTE:** The user running docker must be in the docker group. If the user is not, sudo should be used when running docker.

4. Once you are inside the container, run the following commands:
   
   ```
   /usr/local/apache2/bin/htpasswd -b /usr/local/apache2/conf/password rtlsadmin <NEW_PASSWORD>
   ```

Replace the new password with the desired password. Once changed, the password changes persist even after restarting RTLS.
Configuring RTLS

RTLS Services is configured primarily through two configuration files, rtls.conf and aar_info.csv. The rtls.conf file is a text file that controls the runtime configuration of RTLS and includes both user changeable and auto filled parameters. The aar_info.csv file is described in the next section.

The default rtls.conf file looks like the file shown below. The parameters in the private section of rtls.conf are automatically edited by the RTLS deploy scripts and should not be edited by the user. However, the other values can be edited.

Table 4 includes the parameters available in rtls.conf for basic configuration of RTLS services.

NOTE: When using the simulator, the only parameter that must be changed under typical situations is the kafka_brokers parameter.

NOTE: When using physical ATR7000 hardware, under typical situations, the following parameters should be changed from their defaults:
- license_server_url
- license_activation_id
- kafka_brokers
- radio_c_and_d_config (should be changed from sim to bearing)

Table 5 includes additional parameters available in rtls.conf for advanced configuration of RTLS services.
Installing CLAS Software

[rtls]
This section consists of user changeable parameters.

# This set to centralized for a centralized LA setup and to distributed for a distributed LA setup
# centralized_or_distributed = distributed

# Set this to the license server URL. By default this points to Zebra cloud license server
# license_server_url = https://zebra-licensing.flexnetoperations.com/flexnet deviceservices

# Set this to the activation ID received from Zebra
# license_activation_id = 1234-5678-9abc-def0-1234-5678-9abc-def0

# comma separate multiple brokers
# kafka_brokers = 0.0.0.0:9092

[location_analytics]
# Set this to auto to enable auto start of LA and manual to disable autostart of LA
# location_analytics_start = auto

# This sets the units of distance in LA to feet or meters
# location_analytics_config_units = feet

# This sets the reporting fields in LA
# location_analytics_reporting_fields = report_source, confidence, position, timestamp, epc id

# This sets the time threshold after which the tag report is published
# la_time_filter = 1

# This sets the confidence threshold above after which the tag report is published
# la_confidence_filter = 70

# This sets the velocity threshold in ft/sec or mts/sec depending on the units configured in
# location_analytics_config_units above after which the tag report is published
# la_velocity_filter = 1000

# la_id_filter = 0000000000000000000000000000000000000000000000000000000000000000
# la_id_filter_mask = 0000000000000000000000000000000000000000000000000000000000000000

# la_id_filter_num_bytes = 12
# la_static_or_dynamic_filter = ignore
# la_distance_filter = 5

[radio_c_and_d]
# Set this to auto to enable autostart of CND and manual to disable auto start of CND
# radio_c_and_d_start = auto

# Set this to bearing to get real tag bearings or sim to get run simulation
# radio_c_and_d_config = sim

[private]
# This changes the logging level for rtls config. string should be one of DEBUG, INFO, WARN, ERROR
# log_level = INFO

# location_analytics_address = 0.0.0.0
# location_analytics_version = 1.0.0.7

# radio_c_and_d_app_version = 1.0.2.6

# docker_host_address = 0.0.0.0
## Basic Configuration Options

### Table 4  Basic RTLS Configuration Options

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Allowed Options</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>centralized_or_distributed</td>
<td>This option changes the mode of operation of RTLS. RTLS can operate in either centralized or distributed mode and setting this parameter to one of those values starts the RTLS system in the appropriate mode.</td>
<td>centralized, distributed</td>
<td>distributed</td>
</tr>
</tbody>
</table>
| license_server_url    | The URL of the server that the RTLS uses to obtain a license. In the case of a local license server, this should be the hostname or IP address of a server that can be reached by the ATRs. | For cloud license server: https://zebra-licensing.flexnetoperations.com/flexnet/deviceservices
For local license server: http://<license_server_ip_or_hostname>:<license_server_port>/request | https://zebra-licensing.flexnetoperations.com/flexnet/deviceservices            |
| license_activation_id | The activation ID issued by Zebra for RTLS.                                | A string value containing the activation ID                                       | 32 digit alpha numeric code |
| kafka_brokers         | The IP:PORT or HOSTNAME:PORT combination of the broker to which RTLS publishes location estimates. **Changing this value is mandatory.** | A valid IP:PORT combo or a HOSTNAME:PORT combo that is reachable from LAs.
If running in distributed mode, this must be accessible from the ATRs.
If running in centralized mode, this must be accessible from the server. | 0.0.0.0:9092     |
## Advanced Configuration Options

Table 5  Advanced RTLS Configuration Options

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
<th>Allowed Options</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>location_analytics_start</td>
<td>This option changes the default startup behavior of Location Analytics. If set to manual, RTLS does not start and initialize the LA at startup. The LAs have to be initialized using the REST APIs. Note that this is only required to be set to manual for debugging purposes, the user should never have to change the setting to anything other than auto.</td>
<td>auto, manual</td>
<td>auto</td>
</tr>
<tr>
<td>location_analytics_config_units</td>
<td>The units of distance that should be used in LA location estimates.</td>
<td>feet, meters</td>
<td>feet</td>
</tr>
<tr>
<td>location_analytics_reporting_fields</td>
<td>A comma separated string that specifies what fields must be included in the location estimate report to kafka broker.</td>
<td>message_id, report_source, confidence, position, timestamp, epc_id, readers, velocity, direction, static_or_dynamic</td>
<td></td>
</tr>
<tr>
<td>la_time_filter</td>
<td>This option sets the time threshold after which a tag location must be reported to Kafka.</td>
<td>Integer representing time in seconds</td>
<td>1</td>
</tr>
<tr>
<td>la_confidence_filter</td>
<td>This option sets the minimum confidence a location estimate must be reported to Kafka, even if the tag has not moved by the amount configured in the distance filter (see below).</td>
<td>Integer in percentage</td>
<td>70</td>
</tr>
<tr>
<td>la_velocity_filter</td>
<td>This option will set the velocity threshold in ft/sec or m/sec, depending on the units configured in location_analytics_config_units parameter, above which a location report will be sent to Kafka broker.</td>
<td>Integer value representing distance/sec</td>
<td>1000</td>
</tr>
<tr>
<td>la_id_filter</td>
<td>This option sets the filter that should be applied to epc_id of the tags. All tags matching this tag pattern will be reported to the Kafka broker.</td>
<td>String value for 96 bit EPC</td>
<td></td>
</tr>
<tr>
<td>la_id_filter_mask</td>
<td>The mask to be used to apply the tag ID filter.</td>
<td>String value for 96 bit EPC</td>
<td></td>
</tr>
</tbody>
</table>
### Table 5  Advanced RTLS Configuration Options (Continued)

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
<th>Allowed Options</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>la_id_filter_num_bytes</code></td>
<td>The length of the EPC ID for tag ID filter.</td>
<td>Integer representing the number of bytes in the EPC ID</td>
<td>12</td>
</tr>
<tr>
<td><code>la_static_or_dynamic_filter</code></td>
<td>Filter that specifies whether to publish tag location reports for only static or only dynamic or both.</td>
<td>static_only, dynamic_only, ignore</td>
<td>ignore</td>
</tr>
<tr>
<td><code>la_distance_filter</code></td>
<td>This option specifies the distance a tag needs to move in order for it to be reported to Kafka. The units of distance used is the one specified in <code>location_analytics_config_units</code> parameter.</td>
<td>integer</td>
<td>5</td>
</tr>
<tr>
<td><code>radio_c_and_d_config</code></td>
<td>This option starts RTLS either in sim mode or bearing mode. In sim mode, RTLS simulates a deployment environment with emulated ATRs. In bearing mode, RTLS configures physical ATRs and enables normal system operation.</td>
<td>sim, bearing</td>
<td>sim</td>
</tr>
<tr>
<td><code>radio_c_and_d_start</code></td>
<td>This option changes the default startup behavior of CND. If set to manual, RTLS does not start and initialize the CNDs at startup. The CNDs have to be initialized using the REST APIs. Note that this is only required to be set to manual for debugging purposes, the user should never have to change the setting to anything other than auto.</td>
<td>auto, manual</td>
<td>auto</td>
</tr>
</tbody>
</table>
Adding ATRs to RTLS

In an actual RTLS deployment, the aar_info.csv file must contain reader information about the physical system (as opposed to the simulated readers). This information is usually not available when RTLS software and server are installed as it is generated during deployment. By default, the aar_info.csv file is populated with simulated readers so that the server can work with the simulator with minimal configuration. When adding real ATRs (as opposed to the simulated readers), the user is expected to add the reader information in the file aar_info.csv.

The user is required to enter the following reader information in CSV format, in this order:

1. ATR Host Name
2. IP Address
3. X Coordinate of the reader
4. Y Coordinate of the reader
5. Z Coordinate of the reader
6. Orientation of the reader

The contents of a sample aar_info file is shown below:

```
AAR Host Name, IP Address, x, y, z, Orientation
ATR7000F422C8,192.168.7.201,40.0,100,17.1,0
ATR7000F476E1 192.168.7.202,15.0,87.5,17.1,-5
ATR7000F3F489 192.168.7.203,40.0,75.0,17.1,0
ATR7000F3F316 192.168.7.204,15.0,62.5,17.1,0
ATR7000F3F4A1 192.168.7.205,40.0,50.0,17.1,10
```

NOTE: The process of defining the ATR host names, IP addresses, and determining x-y-z coordinates and orientation is described in the ZAATS Deployment Guide, p/n MN-003195-xx.

NOTE: The units of distance used to supply the coordinates of ATRs should be on the same as mentioned in rtls.conf file in the location_analytics_config_units field.

Prepating to Run RTLS

Before starting RTLS, do the following:

- Stop the NTP service on the host. To do this, run the command:
  ```
sudo service ntp stop
  ```
- Stop any FTP service on the host.
- Ensure the port ranges mentioned in the deploy.env file for FTP (RTLS_FTP_PASV_PORT_MIN and RTLS_FTP_PASV_PORT_MAX) are available and are not used by any other application.
- Ensure the port assigned to RTLS for REST Service in the deploy.env file, 80 by default, is available.
Installing CLAS Software

---

Starting and Stopping RTLS

Starting RTLS

After installing and configuring RTLS software and adding readers, the system is ready to start.

To start RTLS:

1. Enter the rts directory
   ```bash
cd rts
   ```
2. Start rtls
   ```bash
   ./rts.sh start
   ```

   **NOTE:** The user running rts.sh must be in the docker group. If the user is not, sudo should be used when running rts.sh.

This starts RTLS, which in turn connects to all the specified readers and start reading and locating tags. Once RTLS Services is started, tag location estimates are published to the Kafka broker serving as the location endpoint. The end user application can run a Kafka consumer and consume the location data from the Kafka broker to verify the RTLS installation.

   **NOTE:** Running RTLS requires that no NTP daemon runs on the host machine. Stop any NTP daemons running on the host machine before starting RTLS Services.

Stopping RTLS

To stop RTLS:

1. Enter the rts directory:
   ```bash
cd rts
   ```
2. Start rtls
   ```bash
   ./rts.sh start
   ```

   **NOTE:** The user running rts.sh must be in the “docker” group. If the user is not, sudo should be used when running rts.sh.

This stops RTLS and stops the flow of location estimates to the broker.
Overview

The primary function of the RTLS Simulator is to enable testing of RTLS Services in the absence of physical ATR readers. In this manner, the simulator emulates ATR readers and their associated tag and bearing reports, and the LA component of RTLS Services publishes tag data reports to the Kafka broker. This chapter describes the various functionalities provided by the RTLS simulator. It also explains the architecture and different modes of operation of the simulator. Validating a CLAS/RTLS Services Installation provides details on how the simulator can be used to validate an RTLS (ZAATS) installation.

Simulator Architecture

The ATR simulator is an application that runs on a server and can emulate the functionalities of ATR hardware.

The functionalities provided by the RTLS simulator are:

- ATR RM interface emulator to provide reader management functionality (required for seamless interfacing to the RTLS C&M server).
- CND emulator to provide tag bearing reports.
- Distributed LA functionality that provides location estimates.

By default, the RTLS Simulator simulates 28 readers. By default, each of the simulated readers generate tags and the rate of 120 tag reports/second. This setting is currently not configurable by the user.

Figure 3 below illustrates the overall architecture of RTLS when using an RTLS Simulator. The blocks that are named ATR/Simulator 1 to ATR/Simulator n are all simulated by the simulator.
Figure 3  RTLS Architecture and Setup in Simulation Mode

Setting Up the Simulator

Follow the steps below to setup the simulator:

1. Extract the simulator
   
   tar -xzsf simulator_x.y.tar.gz

2. Go to the directory where the simulator was extracted
   
   cd simulator

3. Build the simulator
   
   ./simulator.sh build

NOTE: The user running simulator.sh must be in the docker group. If the user is not, sudo should be used when running simulator.sh.
Simulator Profiles

The RTLS simulator supports loading of various simulator profiles. A simulator profile allows for the simulation of a set of ATR7000s and tags at particular locations. For example, one of the simulator profiles can be named: sample_profile. The simulator contains a default profile that is used at startup unless specified otherwise.

Adding New Profiles

To add a new profile, follow the below steps:

1. Create a new directory under the simulator/profiles directory and name it with the name of the new profile.
2. Add two files: aar_info.csv and tag_info.csv in the newly created directory. The sample file format for each of these files are shown below:

   **Sample aar_info.csv:**

<table>
<thead>
<tr>
<th>AAR Host Name, IP Address, x, y, z, orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AARCD3B65, 10.219.90.12, 39.85, 35.37, 17, 0</td>
</tr>
<tr>
<td>AARFC8131, 10.219.90.13, 22.15, 65.57, 17, 0</td>
</tr>
</tbody>
</table>

   **Sample tag_info.csv:**

<table>
<thead>
<tr>
<th>TagID, x, y, z</th>
</tr>
</thead>
<tbody>
<tr>
<td>3512ebfa1000000015000005, 15, 0, 3</td>
</tr>
<tr>
<td>3512ebfa1000000015005005, 15, 5, 3</td>
</tr>
<tr>
<td>3512ebfa1000000015010005, 15,10, 3</td>
</tr>
</tbody>
</table>

3. Populate the files with appropriate information. The aar_info.csv file contains the host name and location information of all the ATR7000s in the simulation and the tag_info.csv file contains ID and location information of all tags in the simulation.

Running the RTLS Simulator

Details for running the RTLS simulator are described in the following chapter, Validating a CLAS/RTLS Services Installation.
Introduction

The ZAATS Deployment Guide (p/n MN-003195-xx) describes in detail the steps involved in an RTLS installation, although does not cover software installation, which is typically deployed independent of the on-site hardware. This document focuses on the steps involved in deploying RTLS Services software. This chapter specifically focuses on how to validate the RTLS Services installation. The following sections provide information on how to setup external components like the Kafka broker and how to configure RTLS to work with simulators.

1. Install and run the Kafka broker.
2. Start the RTLS Simulator.
3. Configure and start RTLS Services.
4. Verify installation by viewing the location updates on a Kafka consumer.

Installing the Kafka Broker

The Kafka broker is set up as a Docker container. Follow the below steps to setup the Kafka broker.

1. Download the Kafka Docker:
   git clone https://github.com/wurstmeister/kafka-docker.git
2. Go to the directory created:
   cd kafka-docker
3. Modify the KAFKA_ADVERTISED_HOST_NAME field in docker-compose-single-broker.yml file to the IP address or Host name of the host where the Kafka broker is installed. This IP address or Host name must be accessible by the LA component (i.e. by the ATR7000s when running in distributed mode or by the server when running in centralized mode).
4. Run the Kafka broker:
   docker-compose -f docker-compose-single-broker.yml up

**NOTE:** The user running docker-compose must be in the docker group. If the user is not, sudo should be used when running docker-compose.

**NOTE:** Kafka needs to be configured in non-secure mode when used with the simulator.
Validating a CLAS/RTLS Services Installation

Start Simulator

To start simulator:

1. Change to the simulator directory:
   ```
   cd simulator
   ```
2. Start the simulator:
   ```
   ./simulator.sh start
   ```

This uses the default simulator profile. If a user defined profile is desired, add the name of the profile after start (e.g., ```./simulator.sh start myProfile```).

**NOTE:** The user running simulator.sh must be in the docker group. If the user is not, sudo should be used when running simulator.sh.

Running RTLS with the RTLS Simulator

To run RTLS with the RTLS Simulator:

1. Set the following parameter in rtls\config\rtls.conf to tell RTLS Services that it is working with simulators.
   ```
   radio_c_and_d_config = sim
   ```
2. Make sure that RTLS Services is supplied with the same set of ATR details that are being simulated.
   To do this, copy the aar_info.csv file from the profile being used to the rtls/config/aar_info.csv. After initial deployment of the simulator and RTLS, the aar_info.csv for the default profile is same as the one found in rtls/config/.

Start RTLS

To start RTLS:

1. Change to the rtls directory:
   ```
   cd rtls
   ```
2. Start RTLS:
   ```
   ./rtls.sh start
   ```

**NOTE:** The user running rtls.sh must be in the docker group. If the user is not, sudo should be used when running rtls.sh.

**NOTE:** If RTLS is running with ATR simulators, ensure that the simulators are started before starting RTLS.
To verify that RTLS is reporting location estimates on the Kafka broker, follow the steps below:

1. Change to the Kafka directory:
   \[\text{cd kafka-docker}\]
2. Log in to the Kafka shell:
   \[\text{sudo ./start-kafka-shell.sh}\]
3. List the Kafka Topics:
   \[\text{kafka-topics.sh --list --zookeeper=<kafka_broker_ip>:2181}\]
4. If the system is properly configured, a topic with the name rtls.tag_location_update.v2.json should be listed as a response in the previous command.
5. Start a consumer to consume the tag location updates from the topic above:
   \[\text{kafka-console-consumer.sh --topic=rtls.tag_location_update.v2.json --bootstrap-server=<kafka_broker_ip>:9092}\]
6. If the system is properly configured, a steady stream of messages must be seen in the Kafka consumer console as indicated in Figure 4.

**Figure 4**  Kafka Consumer
Stop Simulation

1. Change to the rtls directory:
   
   cd rtls

2. Stop RTLS by running:
   
   ./rtls.sh stop

   **NOTE:** The user running rtls.sh must be in the docker group. If the user is not, sudo should be used when running rtls.sh.

3. Change to the simulator directory:
   
   cd simulator

4. Stop the simulator by running:
   
   simulator.sh stop

   **NOTE:** The user running simulator.sh must be in the docker group. If the user is not, sudo should be used when running rtls.sh.
Introduction

This chapter describes running RTLS and the CLAS software with ATR7000 hardware. It also describes the two different modes of license servers (cloud and local) available for use with RTLS. It does not describe the setup and administration of the license server. Refer to the Local License Server Administration Guide for Windows or Local License Server Administration Guide for Linux, as applicable for detailed explanation of setup and administration aspects of the local license server.

NOTE: Licensing is only needed when running RTLS with ATRs. Licensing is not needed when running with the simulator.

RTLS Licensing Model

RTLS is a licensed software that requires a valid Configuration and Location Analytics Software (CLAS) license to run. The CLAS license(s) purchased from Zebra determine the number of ATR7000 readers that can participate in RTLS location tracking. Only readers that successfully acquire a license from the license server can produce the bearing data and tag reports needed by the location analytics component (LA) to determine x-y-z location of a tagged item.

NOTE: After purchasing the license, Zebra Technologies provides activation IDs for your licenses and a link to access RTLS software downloads.
RTLS Licensing Mechanism

As described in the previous section, the primary role of RTLS licensing is to control the number of readers that can provide the required information to produce location estimates. The reader-based Radio C&D (CND) component is the RTLS component responsible for verifying and enforcing that a valid license is available. Figure 5 shows the process of license acquisition at startup.

**Figure 5  License Acquisition Process**

As shown in Figure 5, the user inputs the licensing information into the rtls.conf file and then starts RTLS Services, as described earlier in this document. RTLS Services then passes that information to the Radio CND that is deployed by the RTLS Services to the ATR7000 readers. CND uses this information and queries the license server (cloud or local) to acquire licenses. If a license is acquired successfully, then Radio CND starts reading tags and location estimates are sent to the Kafka broker upon RTLS startup.

The following fields must be updated in the rtls.conf file for licenses to be acquired successfully.

- **license_server_url**: This is the URL of the licensing server that is used to get licenses. The URL can point to either the cloud license server or the local license server.
- **license_activation_id**: The activation ID that must be used to acquire a license. This is supplied by Zebra after purchase of licenses.
License Acquisition Modes

RTLS supports licenses to be acquired in one of two ways:

- Acquire license from a cloud-based license server.
- Acquire license from a local license server.

Cloud-Based License Server

Figure 6 illustrates the license acquisition process from the cloud.

In this method of license acquisition, the license server is hosted in the cloud and the CND contacts the cloud-based license server to acquire licenses. This implies that the ATR readers must have access to the Internet for them to acquire a license and start operations. This is the default mode of license acquisition in RTLS, as it involves minimal setup and configuration since the license server is hosted in the cloud.

For a cloud-based license server, the value of the license_server_url field in rtls.conf must be set to: https://zebra-licensing.flexnetoperations.com/flexnet/deviceservices.

Figure 6  Cloud-Based Server
Local License Server

Figure 7 illustrates the process of license acquisition from a local license server.

In this method of license acquisition, the license server resides on the same local network as the readers. Information on installing and registering the local license server can be found in the Local License Server Administrator Guide.

Figure 7  Local License Server Setup

This method does not require the readers to have Internet connectivity because they are contacting a local server for license acquisition.

To acquire a license by Radio CND on the ATR7000, the following are required:

1. User installs and configures the local license server.
2. User registers the local license server in Zebra's end user licensing portal.
3. User acquires licenses for the local license server.
4. User configures RTLS to point to local license server and starts RTLS.
5. RTLS Server passes the licensing info to Radio CND on startup.
6. Radio CND contacts the local license server and acquires license.

Steps 1 through 3 are described in greater depth in the Local License Server Administration Guide. Refer to the Local License Server Administration Guide for Windows or Local License Server Administration Guide for Linux, as applicable. This guide includes installation and registration instructions.

Once the local license server is installed, registered, and licenses acquired, RTLS must be setup to acquire licenses from the local license server. This can be done by changing the license_server_url field in the rtls.conf file to the following: http://<license_server_ip_or_hostname>:7070/request. By default, the local license server listens on port 7070, which can be changed in the license server configuration. If a non-default port is configured in the local license server, be sure to update the local_license_server_url field in the rtls.conf file to the same value. The license_activation_id field must be updated to the appropriate value provided by Zebra.
Running RTLS with ATR7000s

Introduction

This chapter describes running RTLS with the ATR7000 hardware.

Bringing the RTLS System Live with ATR7000 Readers

The following list of steps describes how to bring the RTLS system live with ATR7000 readers.

1. Validate RTLS works with Simulators as described in Validating a CLAS/RTLS Services Installation.
2. If RTLS is running, stop RTLS:
   a. Change to the rts directory:
      cd rts
   b. Stop RTLS by running:
      sudo ./rts.sh stop

3. If using local license server, configure local license server as described in CLAS and RTLS Licensing.
4. Ensure the ATR7000s are deployed properly as described in the ZAATS Deployment Guide, p/n MN-003195-xx.
5. Update the aar_info.csv file in the rts directory to match the actual locations and orientations of the deployed ATR7000s.
6. Update the rts.conf file:
   a. If using a local license server, change the license_server_url parameter to the appropriate value as described in CLAS and RTLS Licensing. If using cloud license server, the default value for the license_server_url can be used.
   b. Change the license_activation_id parameter to the activation ID supplied by Zebra at the time of purchase.
   c. Change the radio_c_and_d_config parameter from sim to bearing to configure RTLS to configure the ATRs to report bearings.
   d. Update the kafka_brokers parameter if not using the Kafka broker used in the simulator. If the same Kafka broker is used, the parameter should not be altered.
7. Start RTLS:
   a. Change to the rtls directory:
      cd rtls
   b. Start RTLS by running:
      sudo ./rtls.sh start

**NOTE:** The user running rtls.sh must be in the docker group. If the user is not, sudo should be used when running rtls.sh.