

CLAS software version 2.2.23 Release Notes

This document summarizes the following firmware releases:

Software Release Number	Release Date	See page
V2.2.23	25-Mar-2019	Page 1

For support, please visit www.zebra.com/support

CLAS software v2.2.23

RELEASE DATE: 25-Mar-2019

Zebra's CLAS software is a licensed software suite that along with the ATR7000 readers provide real time locations of RFID tags.

Users of CLAS software can consume real time location data streaming on a Kafka interface for further analysis and visualization and take appropriate actions.

Release Notes lists new features, any specific usage instructions, and any known issue.

Note: Please Note that in order to install the CLAS software active internet connection is required.

Contents of the release package:

Component	Version	Date
RTLS CNM	3.5	03/25/2019
Radio CND	1.2.4.28	03/25/2019
Location Analytics	1.1.4.34	03/25/2019
CMM Client	1.1.2	03/25/2019
RTLS Simulator	1.23	03/25/2019

Software Download

CLAS software (rtls_services_2.2.23.tar.gz) is available for download using Legacy method of deployment.

The software install packages can be downloaded from <https://www.zebra.com/us/en/support-downloads/software/developer-tools/clas.html>

NOTE: A container registry based install mechanism of the CLAS software is being setup and will be published at a later point of time. The installation steps will be updated once it is setup to describe the specifics.

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, 5TB HD

Device Compatibility

CLAS software can only be used with the ATR7000 readers and no other fixed readers are supported.

Installation Instructions

This section lists the installation instructions for the legacy method of installation of the CLAS software.

Legacy Method:

To install the CLAS software using legacy method, follow the below steps.

Note :- The current Version (2.2.23) has a known issue with deploying the CMM client. To continue building RTLS without CMM client, edit `rtls.sh` file comment line 702 as below.

`#deploy_cmm`

1. Extract the release package: **`tar -xzf rtls_services_x.y.z.tar.gz`**
2. **`cd rtls`**
3. Edit the file **`deploy.env`** and set the variables accordingly. Below is the description of variables.
REST_PORT: Setting this environment variable will configure the port on which the RTLS server will expose its REST interface. Default is port 80.
RTLS_AAR_IFC_IP: Setting this variable is mandatory. This variable should be set to the IP of the host machine that is running the RTLS server. In cases where the host machine has multiple network interfaces, this variable should be set to the IP of the network interface that is in the same subnet as the ATR7000s. By default, this is set to 0.0.0.0
UPSTREAM_NTP_ADDR: Setting this variable to appropriate value will make sure that the RTLS NTP service is syncing with the proper upstream NTP server. By default, this is set to `time1.google.com`
SSL_ENABLE: setting this variable to YES will deploy RTLS services in HTTPS mode. Setting it to NO will deploy it in HTTP mode. Default value is YES
CERTIFICATES_PATH: Setting this variable to `shared_vol` will start RTLS with self-signed certificates for use with HTTPS mode. To install CA certificates or custom certificates set this variable to the absolute PATH of the folder containing certificates.
RTLS_FTP_PASV_PORT_MIN and **RTLS_FTP_PASV_PORT_MAX:** These variables are used to set the range of ports to be used with FTP passive mode operation for RTLS.
4. To deploy RTLS run **`./rtls.sh deploy`**

NOTE: if the user is not added to the docker group, please use `sudo` for running `rtls.sh` script.

CHANGES in 2.2.23 with respect to 2.1.12

CLAS V2.1.12 was the Beta release for customers

- Support for handling of up to 255 readers.
- PUB/SUB heartbeat monitoring of LA and CND instead of polling.
- Enhanced configuration support of CLAS via rtls.conf file.
- Configuration changes made via the REST interface will be persisted and reapplied upon RTLS restart.
- Multi-tag items added via the REST interface will now be persisted and reapplied upon RTLS restart.
- Support for adding Multi-tag items in GID and GIAI formats.
- Additional REST APIs for log management of Location Analytics and Radio CND.
- Added ability to change the RTLS distance units via the REST interface.
- Support for pure Distributed mode Failover for LA as opposed to using Central LA as failover.
- Updated the REST response for LA status to indicate is a LA is in standby mode.
- Addition of confidence threshold filter in Location Analytics.
- Removed message_id as a default field in tag location report to Kafka broker.
- Support for reporting uptime of all components as status response.
- Support for reporting multitag location when polled.
- Added ability to manually configure olio, inventory, select, packets, hopPerRound, stopconditions, secondarybeamset, taginfofields in Radio CND
- Added ability in LA to report tag locations for tags that would not ordinarily be reported because their bearing was greater than or equal to 60 degrees

KNOWN ISSUES

Following are the list of issues that currently exist in the CLAS software.

- The CND and LA warning and error API response have unicode characters in the end when the REST request fails.
- The location_analytics/filters API response for GET return static as a value for static_or_dynamic filter as opposed to staic_only as mentioned in the document.
- Mutlitag API accepts incorrect data also.
- Posting same data multiple times via the multitag API and then GET of multitag API is showing duplicates.
- Give negative number for velocity threshold filter then velocity threshold filter is not showing in get filter API response.