

thinkRF libtrf API Release Notes v1.9.1 - .2

Oct 16th, 2026

1	Introduction	2
1.1	Supported Devices	2
2	New Features	2
3	Enhancements	2
3.1	Changes Affecting Existing Applications	3
4	Fixed Defects	3
5	Known Issues and Limitations	3
	Contact Us	3

1 Introduction

thinkRF's libtrf API is a multi-platform API library that can be used for interfacing with existing generation thinkRF RTSA devices (R5xx0) and will be maintained into the future to support interfacing with future thinkRF devices. In addition, the API supports and will be evolved to increasingly provide sophisticated signal processing capabilities that can be easily integrated into end-user application code.

The libtrf library is currently released as builds for Microsoft Windows 64-bit (.h/.dll/.lib) and Linux (.h/.so) for both x86_64 (Intel/AMD) and aarch64 (ARM v8+) targets. The interface is presented as undecorated 'C' compatible with the C99 (ISO/IEC 9899:1999) and later standards.

This document describes the release information of libtrf API software version 1.9.1 - .2. Refer to the companion 'libtrf API Reference Manual' document for full libtrf and its usage details.

Important Note: *If your RTSA has firmware v1.7.0/1.7.1 for R55x0 or v1.2.0/1.2.1 for R57x0, it is critical that you should update it to the latest to use libtrf's spectrum capture in v1.5.4 or higher.*

1.1 Supported Devices

The versions of libtrf currently released support all thinkRF's RTSA products:

- R55x0/R57x0 products (408/418/427 models)
- R6000 products – Ethernet and USB connection

2 New Features

The following new features and capabilities were added. Refer to the Reference Manual for full details.

1. Added a **communications-tolerance setting (TRFCommsToleranceMsec)** that controls how long libtrf waits through an interruption on the control or data connection before reporting a fault. It is now *the preferred control for communications timeouts*. Defaults to 5000msec, suitable for USB/LAN; raise to around 15000msec for WAN links.
2. Added a **new device status (trfDeviceNotDeliveringData, -119)** for a device that responds to commands but has sent no capture data since the capture started. This separates a stalled capture from a device that can't be reached at all.

Code examples have been updated to include the new features and provide the best way to learn how to use them.

3 Enhancements

The following other changes were made:

1. **IQ frames** now report the highest signal level seen so far in the stream, in dB relative to full scale, alongside the existing sub-optimal level flag.
2. **Spectrum Masking** usage:
 - a. **Alarms processor frames** now report which passenger trace carries the threshold, so applications no longer need to assume it is always first.
 - b. **Alarms events** now report the first and last frame numbers they cover.
 - c. An **alarm range** can now be marked inactive. It keeps tracking its threshold but stops generating events, and any event still open when it's deactivated is closed.
 - d. An **alarm range's settings** can now be changed while it is in use. Each change is applied in full or not at all, so the range is never left partly updated.
3. **Faster IQ** reconstruction, particularly on ARM-based platforms.
4. Substantially **improved reliability** starting, stopping and recovering captures over slow or remote (WAN) connections.
5. **Loading libtrf** no longer claims **USB** RTSA devices until one is actually enumerated or opened, so another application can no longer be blocked from finding the receiver.

6. **GNSS position** is omitted from frame data entirely when there is no fix, rather than reporting a default location; unknown motion values are now reported as unavailable rather than as invalid data.
A GNSS fix older than 10 seconds, or out of range, is now treated as invalid.

3.1 Changes Affecting Existing Applications

7. The **TRFAdaptFull** and **TRFAdaptStep** parameters have been replaced by a single **TRFMaxAdaptStep** parameter, which controls the largest automatic adjustment applied after an IQ capture. Applications setting the old parameters should be updated.
8. **TRFSCPIQueryTimeout** is still honoured, and an explicit value overrides the one derived from the tolerance *TRFCommsToleranceMsec*.

4 Fixed Defects

The following issues were fixed:

1. Fixed intermittent crashes when a stream or device was closed during an active capture, or when a device was lost while streaming.
2. Fixed a crash on some platforms when a spectrum sweep needed certain FFT sizes. .
3. Fixed sweep captures stopping permanently after a device restarted its own sweep list.
4. Fixed spectrum streams failing to start after an IQ stream on the same device was closed during start-up.
5. Fixed setting changes on a running IQ stream being accepted but not applied to the device
6. Fixed occasional stalls when starting an IQ stream, or when spectrum settings were changed and quickly changed back.
7. Fixed the last data point of some spectra showing an invalid low value.
8. Fixed gaps in spectra where antenna or RF-path ranges overlap or meet.
9. Fixed clearing a min-hold trace briefly showing a flat trace at the bottom of the display. The trace now restarts from the current spectrum.
10. Fixed minimum, maximum and average traces disappearing while an alarm mask was active.
11. Fixed device date and time settings not always taking effect. A value the device rejects is now reported as an error instead of being ignored.
12. Fixed invalid data in device parameters when GNSS motion information was unavailable.
13. Fixed a crash when alarm output was enabled before a threshold was set, and a memory fault when the spectrum size changed while alarms were active.

5 Known Issues and Limitations

The following are the known limitations or other issues present since libtrf version 1.4.0:

1. Rarely, stopping a continuous, undecimated, frequency-stepped spectrum sweep on an R6xx0 device while it is streaming can leave the device unresponsive, requiring a power cycle. This is under investigation.
2. Initial spectrum of narrow-band IQ capture (12.5kHz) on R5x, R6x displays sinc-type distortion (IQ stream -> IQToSpectrumProcessor).

Contact Us

thinkRF Support website provides online documents for resolving technical issues with ThinkRF products at <https://thinkrf.com/documentation/>.

For all customers who hold a valid end-user license, ThinkRF provides technical assistance 9 AM to 5 PM Eastern Time, Monday to Friday. Contact us at <https://support.thinkrf.com/> or by calling **+1.613.271-5451**.