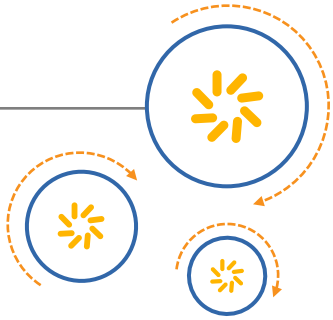
Qualcomm Technologies, Inc.



QTI Logkit.LA.3.0

User Guide

80-P9168-1 Rev. C

December 8, 2017

For additional information or to submit technical questions, go to: https://createpoint.qti.qualcomm.com

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision history

Revision	Date	Description
A	November 2016	Initial release
B	May 2017	Updated title to QTI Logkit.LA.3.0
C	December 2017	Updated with Android O related changes

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document provides information on how to use the QTI Logkit, an on-device logging solution for precommercial and commercial devices manufactured by Qualcomm Technologies, Inc. (QTI) MSM™ licensees.

The QTI Logkit collects modem and HLOS logs, reports of crashes, failures, and device statistics obtained while running on Android devices.

1.2 Conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, `#include`.

Code variables appear in angle brackets, for example, `<number>`.

Button and key names appear in bold font, for example, click **Save** or tap **Enter**.

1.3 Technical assistance

For assistance or clarification on information in this document, submit a case at: https://createpoint.qti.qualcomm.com/

If you do not have access to the CDMATech Support website, register for access or send email to support.cdmatech@qti.qualcomm.com

2 Basic QTI Logkit Information

2.1 Installing QTI Logkit

QTI Logkit is an Android APK that functions on Qualcomm® distributions of Android only.

APK installation is via standard Android procedures (see Appendix A).

2.2 Launching the application

QTI Logkit does not start by default on bootup.

Depending on whether the device image is a USER or USERDEBUG variant, there is more than one way to launch QTI Logkit. For details on determining the build variant, see Appendix B.

Launching with the dialer is supported on all variants, while launching via the native Android application menu or the app icon is available only for USERDEBUG builds.

On subsequent device resets, the application automatically launches unless the user manually exits the application (see Chapter 5) or autolaunch is disabled in the qti_logkit_config.xml (see Section 7.6).

2.2.1 Dialer

1. Use the native dialer.
2. Dial **1*553** (i.e., 1*LK3).

NOTE: The key sequence can be customized (see Section 7.8).

2.2.2 Application menu

The native Android application menu contains an entry for launching QTI Logkit for USERDEBUG builds only (see Figure 2-1).

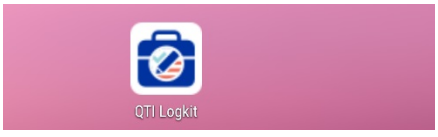


Figure 2-1 QTI Logkit application icon (on home screen)

2.2.3 Relaunching application from background

When the QTI Logkit application is in the background, it can be brought to the foreground again via the app icon, dial sequence, notifications in the notification tray, or from the Recent Apps menu.

NOTE: All recent applications are listed in the Recent Apps menu, so even if QTI Logkit is exited completely (and no process is running) it still appears in this menu.

2.2.4 Application status

When QTI Logkit is running, the user sees the QTI Logkit notification icon in the upper-left notification area (see Figure 2-2). See Section 8.1 for more information on the QTI Logkit session status notification.

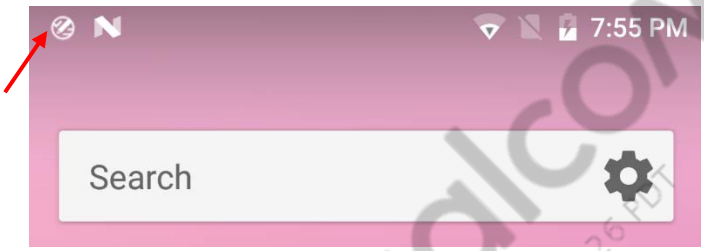


Figure 2-2 QTI Logkit notification icon in notification bar

2.3 Managing the APK

2.3.1 Application version

The QTI Logkit version information can be displayed through the UI. This document describes version 3.00.012.

1. On the **Home** screen, tap the **Settings** gear icon in the upper right.
2. On the **Settings** screen, scroll to **About QTI Logkit** as shown in Figure 2-3 to see versions for all QTI Logkit components.

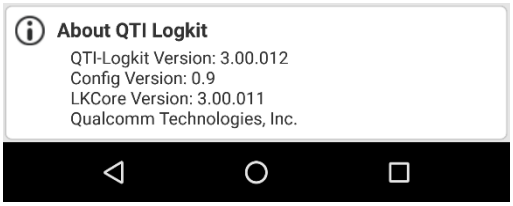


Figure 2-3 QTI Logkit version information

2.3.2 Upgrading application

QTI Logkit APK is distributed in embedded software via Qualcomm CreatePoint. To locate the APK on CreatePoint:

1. Visit https://createpoint.qti.qualcomm.com and enter log-on credentials.
2. Select **Tools** from the upper menu.
3. Search Logkit.LA.3.0 Files and select Result.
4. Select the most recent version from the dropdown menu and then click **Download** to download the zip file.
5. Unzip the file. The zip includes the qti-logkit.apk and two configuration files (qti_logkit_config_O.xml for Android O and qti_logkit_config.xml for Android N and prior).

See Appendix A for installation instructions.

NOTE: QTI Logkit contains configuration files and caches some data during use. You may have modified some configuration items, and upon an APK upgrade, these modifications will generally be kept intact. However, QTI Logkit may not be able to preserve certain configuration items if the APK upgrade materially modifies the feature to which the configuration item pertains.

3 Managing Sessions

A *session* comprises all log files and diagnostic data, and events occurring between starting and stopping log collection.

Tapping the **Start/Stop** button to start log collection initiates a unique session; tapping **Start/Stop** again to stop log collection completes the session. A new session is initiated and completed each time you start and stop log collection.

QTI Logkit keeps all data collected between the starting and stopping actions in one session for easy management.

After launching the application, the **Home** screen displays the **Storage Status Card** to access the **Location Picker** in the top left, and the **Session Status Card** with the **Start/Stop** button it the top right, as shown in Figure 3-1.

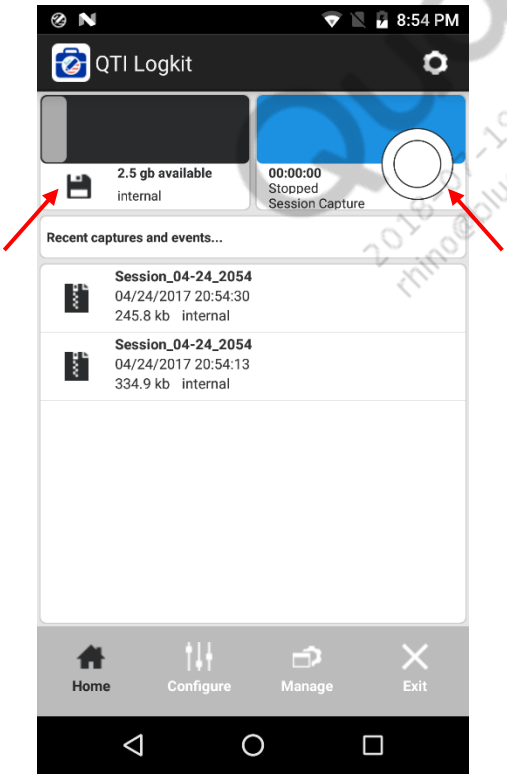


Figure 3-1 Home screen

3.1 Choosing a storage location

The **Storage Status Card** is in the upper-left corner of the home screen (see Figure 3-1) and displays details about the currently-selected storage location, including the available space.

Clicking within this area opens the **Storage Selection** screen that shows how much space is available at the chosen location, the full location path, and a dropdown menu to select other locations (see Figure 3-2).

Once the user has chosen the appropriate storage location, tap the < icon in the upper-left corner (or press the phone's **Back** button) to return to the home screen.

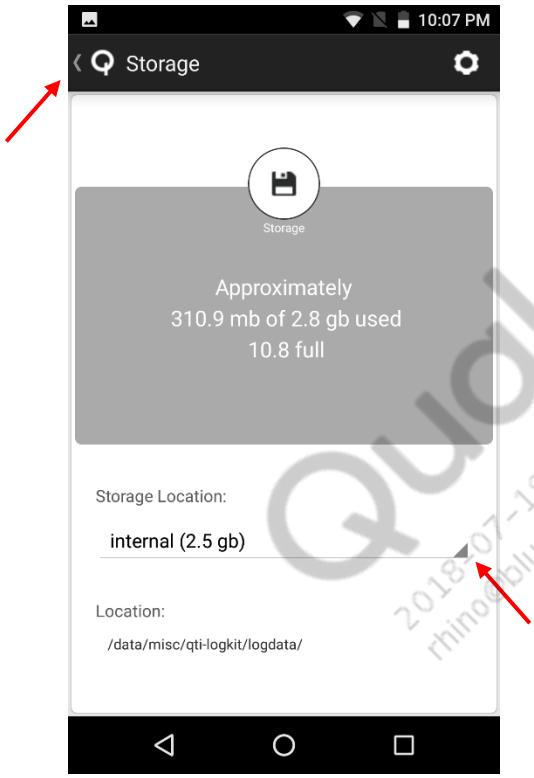


Figure 3-2 Storage selection screen

3.2 Starting a session

Tap the **Start/Stop** button in the Session Status Card area to start collecting log data.

Once the session displays as **Running**, the screen shows which logs are being collected and periodically updates to reflect the current size of the individual logs (see Figure 3-3).

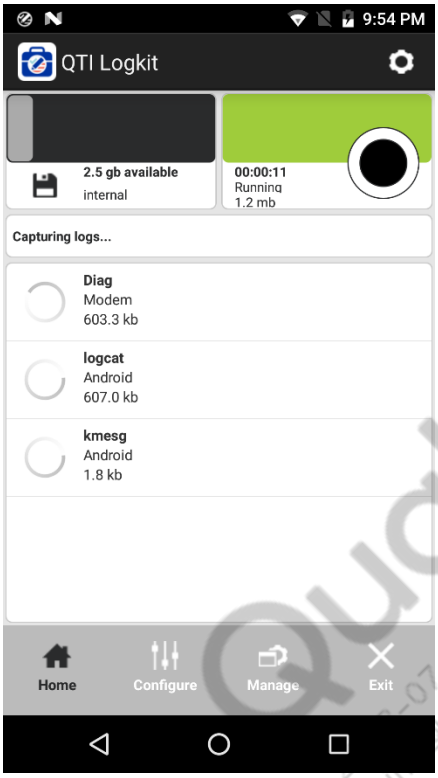


Figure 3-3 Active logs: Diag (Modem), logcat and kmesg (Android)

NOTE: On first launch, the **Session Status Card** shows as **Stopped**. On subsequent launches, the previous status (Running or Stopped) persists.

3.3 Stopping a session

1. When a session is Active or Running, tap the **Start/Stop** button to stop collecting log data.
2. The **Session Capture** screen displays (see Figure 3-4). This screen allows users to:
   - Use the default session name or enter a custom name.
   - Change the storage location where the session will be archived.
   - Add a description to be saved to a Description.txt file in the session folder.
   - Combine all activity and data for a session into one package by pressing **Package**.
   - Delete all session logs by pressing **Discard**.

NOTE: Discarded sessions are not recoverable.

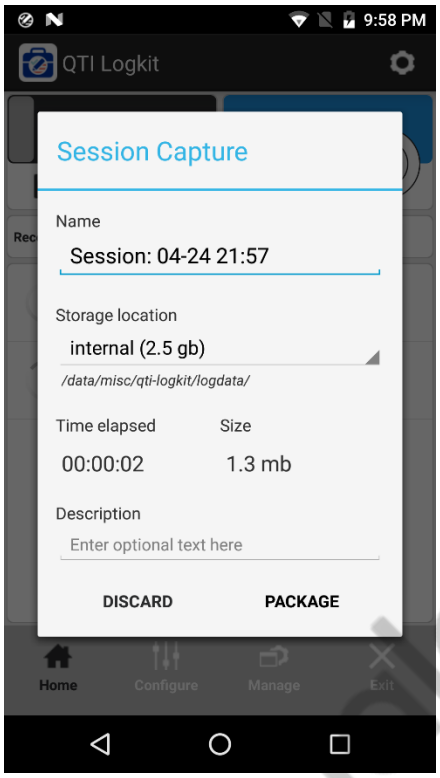


Figure 3-4 Session Capture screen

3.4 Recent session and event history

The **Home** screen displays newly collected and packaged sessions in sequential order. The name, date and time, size, and storage location is shown for each session, as shown in Figure 3-5. Clicking any session opens the session details page for that session.

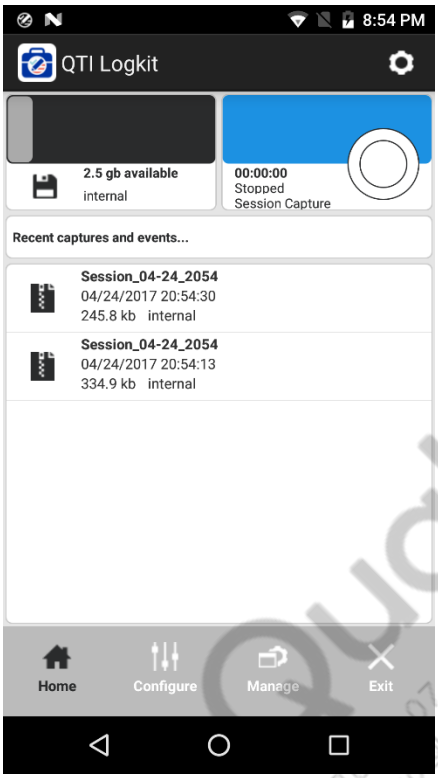


Figure 3-5 Home screen with recent sessions and events

3.4.1 Sessions stored on a removable storage card

If a session is saved to a removable storage card, i.e., an SD card, the session history is only displayed if the removable storage card is inserted and has mounted (upon device startup or reboot, an SD card can take several minutes to mount).

If the removable storage card mounts after QTI Logkit launches, the saved session is shown once QTI Logkit refreshes the UI.

3.4.2 Deleted or extracted sessions

The **Home** screen and **Manage** screen also include placeholders for deleted or extracted sessions. These sessions are depicted with a red icon, as shown in Figure 3-6.

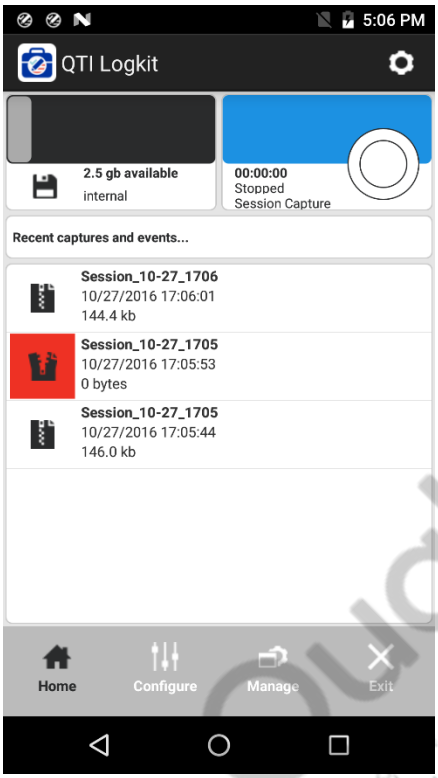


Figure 3-6 Deleted/extracted sessions shown in red

No action can be taken on deleted sessions. Deleted sessions are shown to maintain a complete collection history as sessions can be modified or removed from a storage location using third party tools (e.g., Media Transfer Protocol (MTP)) without the knowledge of QTI Logkit.

Deleted sessions can be hidden from view via a setting in the **Settings** menu (see Section 4.2.3).

3.5 Managing sessions

View session details and delete sessions from the **Manage** screen shown in Figure 3-7.

1. Tap **Manage** at the bottom of the screen.
2. To view session details, tap the appropriate session and it opens the **Package Details** page (see Figure 3-7). When finished, press the < icon in the upper-left corner (or the phone's **Back** button) to return to the **Manage** page.
3. To select all sessions, tap the  icon and all check boxes will be selected.
4. To delete one or more sessions, select the check box next to the sessions to be deleted and tap the **Trash** icon.

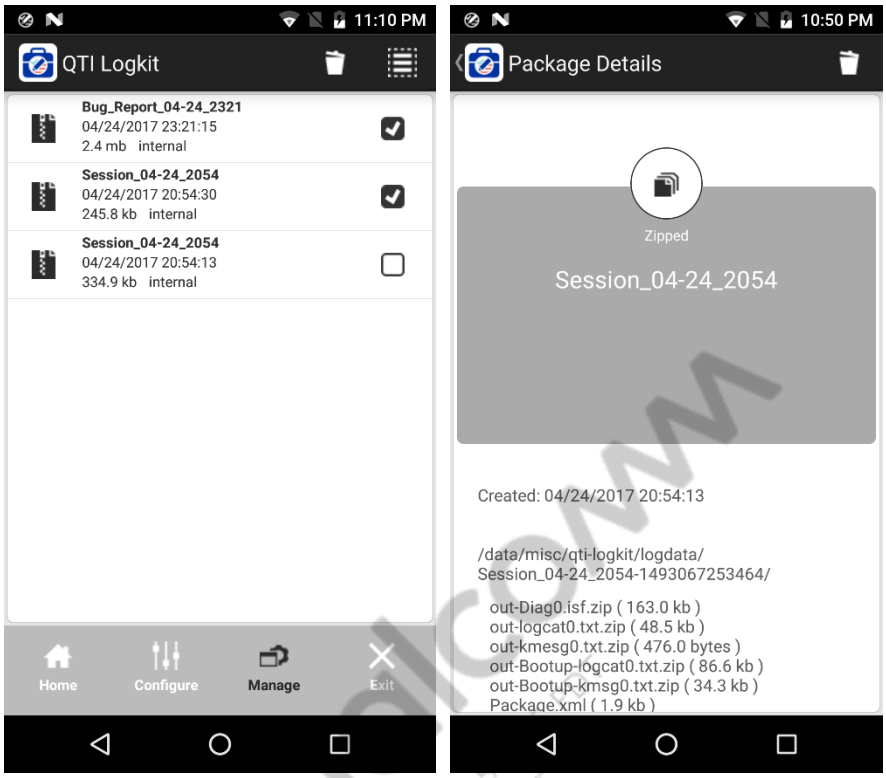


Figure 3-7 Manage screen and Package Details page

4 Changing Configurations and Settings

4.1 Selecting configuration options

The QTI Logkit application includes a large set of features and diagnostic logs that are configurable for each session. Table 4-1 lists QTI Logkit feature and diagnostic log options.

Table 4-1 Feature and diagnostic log options

Option	Description
Modem	
Golden	The Golden DMC is a Qualcomm-defined set of modem packets that should be sufficient for most standard diagnostic situations.
Audio	The Audio DMC is a Qualcomm-defined set of modem packets that are specific to the voice path.
Custom	The Custom DMC allows users to push a custom DMC to a location specified in the qti_logkit_config.xml to enable directly from the UI without modifying the config. (See Chapter 7).
Android	
Logcat radio	Enables collection of radio-specific logcat logs.
Logcat event	Enables collection of events-related messages.
TCPDump	Enables the collection of all TCPDump data.

1. Tap **Configure** at the bottom of the screen.
2. Toggle the appropriate switches to **Green** (on) or **Gray** (off).
   - Sessions include the set of features/logs selected at the time the session is started. Switches are disabled during sessions so changes cannot be made.

Figure 4-1 shows the Golden DMC modem configuration enabled with the path to the DMC below the switch.

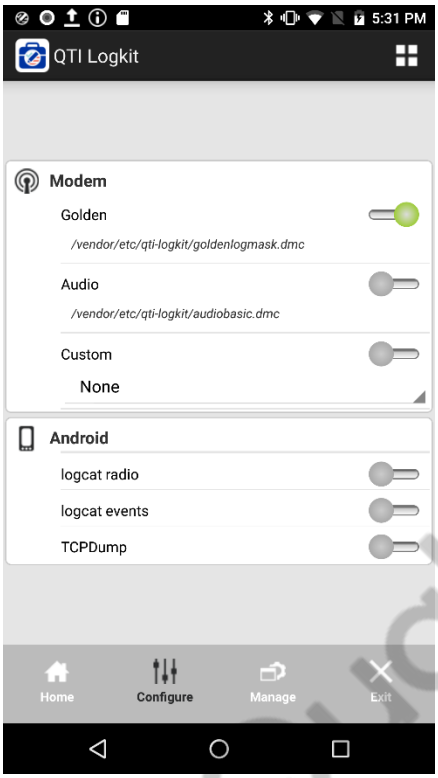


Figure 4-1 Golden DMC modem configuration enabled; all other logs are disabled

4.2 Changing settings

QTI Logkit also includes settings that control asynchronous events outside the session and settings related to the application and its performance.

4.2.1 Enabling/disabling event monitoring

Event monitoring refers to Android events that QTI Logkit can monitor. If enabled when the event occurs QTI Logkit notifies the user, collects relevant logs and opens the Session Capture screen to package the event. Table 4-2 lists event monitoring options.

Table 4-2 Event monitoring

Setting	Description
Tombstone	A tombstone file is a log file collected when a process crashes. It contains information related to the process at the time of crash, similar to a crash dump.
ANR-System	An application not responding (ANR) event occurs when a system application is sluggish or temporarily nonresponsive. It is not a crash or even an error, but is indicative of a poorly-constructed application.

Setting	Description
ANR-Data	An ANR event occurs when a non-system application is sluggish or temporarily nonresponsive. It is not a crash or even an error, but is indicative of a poorly constructed application.
Framework Reboot	This event occurs if the Android framework has an unscheduled reset, generally the result of a critical Android app or service failing.
Subsystem Reset	A mini-dump if one of the attached peripheral subsystems crashes. Prevents a full system crash if possible.
Crash Collection	If the device or subcomponent of the device has crashed, QTI Logkit can detect that this has occurred and respond accordingly. NOTE: see the Crash Mode option (Table 4-4) for Crash Collection options.

1. On the **Home** screen, tap the **Settings** gear icon in the upper right.
2. As shown in Figure 4-2, toggle the appropriate switches to on or off for the **Event Monitoring** options.

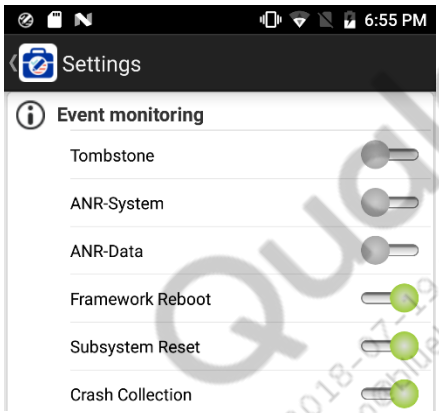


Figure 4-2 Event monitoring settings

NOTE: If a session is active when the Event is triggered, the session stops and all logs from that session are packaged together with the report files.

4.2.2 Triggering reports

There are several report types that can be manually triggered by the user. Report options are described in Table 4-3.

Table 4-3 Reports

Setting	Description
Bug Report	This is a native Google log collection feature in which data specific to Android is collected.
Surface Flinger	Surface flinger is a UI/graphics toolkit that generates its own log files.
Dumpsys	Synonymous with “adb shell dumpsys”, this provides details on all system services.
Meminfo	Synonymous with “cat /proc/meminfo”, this provides details of current memory usage by the system and applications.

Setting	Description
Procrank	Displays a summary of process memory utilization.
Dumpsys media.camera	Displays details on just a single system service as opposed to all services.

1. On the **Home** screen, tap the **Settings** gear icon in the upper right.
2. As shown in Figure 4-3, tap the intended **Run Command** button to generate an on-demand report.
   - While the report is generating, the user will see a spinning progress wheel. Some reports take several minutes to complete.

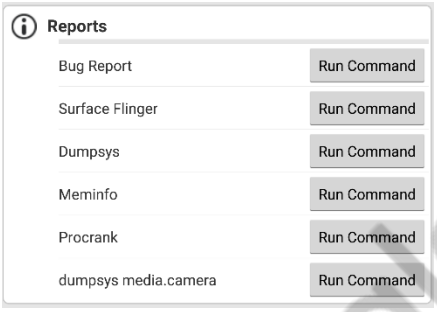


Figure 4-3 Reports settings

3. When the report is finished, the **Session Capture** screen displays (see Figure 3-4).

NOTE: If a session is active when the Report is triggered, the session stops and all logs from that session are packaged together with the report files.

4.2.3 Using additional settings

Devices come with different internal and external memory sizes. To allow efficient use of these resources, the log size, format, and Qshrink database settings can be customized. The user can also alter settings to control how the application behaves.

Table 4-4 describes all additional settings

Table 4-4 Additional settings

Setting	Description
Storage	Select a different storage location from the dropdown. NOTE: This is the same as selecting a storage location from the **Home** screen.
Crash Mode	Select the type of crash to be collected. ▪ **Sdcard collection** – Collect and package full sdcard crash. ▪ **Sdcard notify only** – Create event indicating crash occurred without collecting the full crash. ▪ **Internal mini collection** – Collect internal EMMC mini-dump.
Log chunk size	Select the default maximum chunk size and the maximum number of chunks. NOTE: During a log session, all log chunks are saved. The number of chunks only applies to continuous logs collected outside of a session. The size of individual log types can be overridden in the qti_logkig_config.xml.
Modem log format	Select QMDL or ISF format.
Deleted files	Select if deleted files are Visible or Hidden.
Qshrink 4 database	Selected if the Qshrink 4 database is “Included in every package” or “Excluded from packages”. NOTE: Duplicating the database in every package can use substantial space.
QSH circular buffer	Select if QSH circular buffering is set to Enable or Disable. If QSH circular buffering is enabled a “Flush” button appears in the **Active Logs** section of the **Home** page for the log, allowing the user to flush the circular buffer on demand.
Tool warning	Select if QTI Logkit warnings are set to Enable or Disable.
Continuous Collection Mode	When this mode is enabled, the default session behavior is changed to always collecting the logs which are enabled. 1. Rather than a Start and Stop button, there is only a flush button. This button packages all current logs into a session. 2. If the log chunk count exceeds the specified (see <LogCount>), the oldest log file will be deleted, resulting in curcular buffering.

1. On the **Home** screen, tap the **Settings** gear icon in the upper right.
2. As shown in Figure 4-4, tap the dropdown to change any setting.

NOTE: Some settings cannot be modified while a session is running and appear grayed out.

Figure 4-4 Additional settings

4.3 Understanding device software permission limitations

Device software variants, such as USER vs. USERDEBUG builds, come with varying permission levels. Depending on a particular device's software variant, some features may or may not be supported.

Examples include collection of modem logs, ANRs, and TCPDump. The list may vary depending on ongoing changes to device software permission models, and may further vary based on OEM preferences.

To account for these limitations, QTI Logkit grays out features unavailable due to permission constraints, as shown in Figure 4-5.

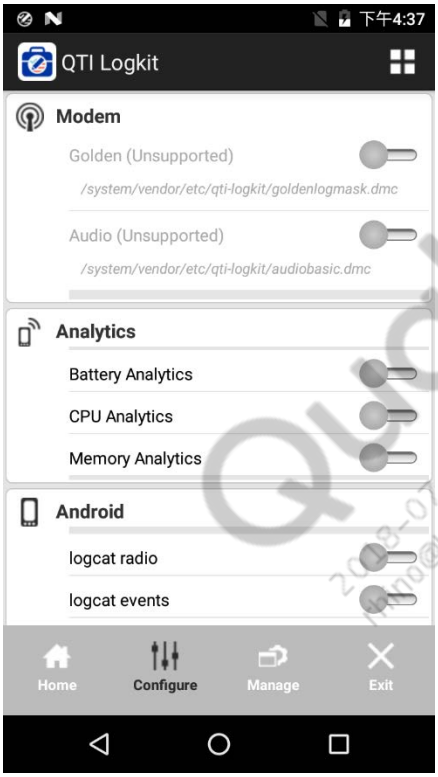


Figure 4-5 Golden DMC and Audio DMC are grayed out (unsupported)

5 Exiting the Application

To exit the application, perform either action:

- Continue logging on the background and hide the UI
- Stop all services and log collection

When a user navigates away from QTI Logkit or opens another application, QTI Logkit remains active in the background collecting any logs it was when it was still in the foreground. QTI Logkit can be monitored using the notification area (see Section 8.1) without bringing it to the foreground.

To stop all services and log collection:

1. On the **Home** screen, tap **Exit** at the bottom of the screen.
   - The **Exit** dialog box shown in Figure 5-1 displays.

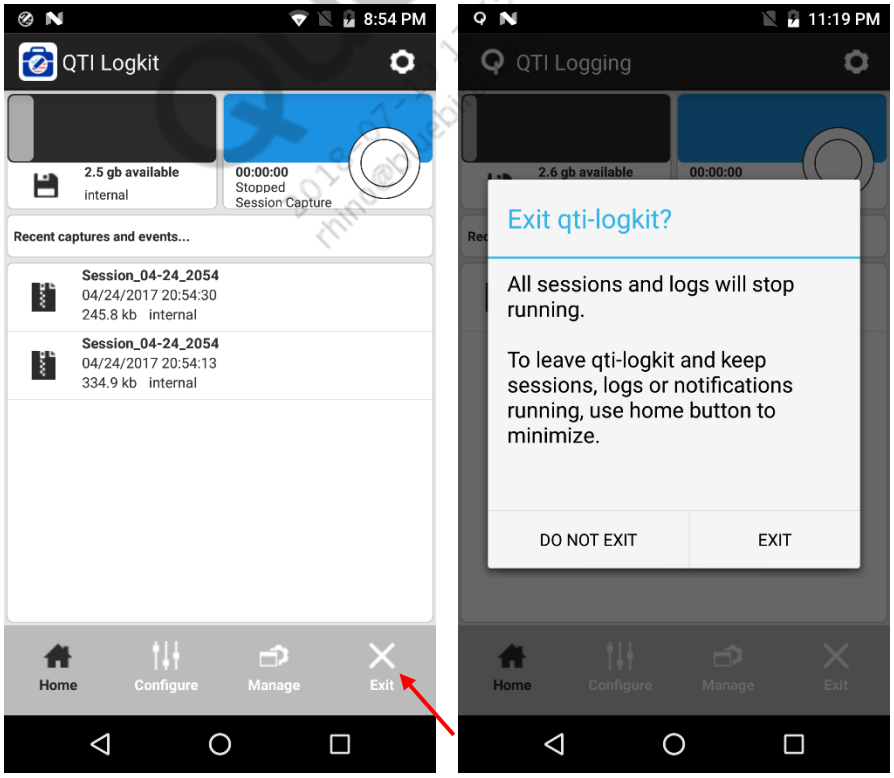


Figure 5-1 Exit dialog box

2. To stop all services and log collection, tap **Exit**.

6 Using DMCs

6.1 About diagnostic monitor configuration files

Diagnostic monitor configuration (DMC) files contain a list of all modem data that should be collected. The amount of modem data enabled in a DMC file can impact the size of log files and in some cases even stress the device's ability to collect all modem log data (i.e., some log packet drop may occur).

Qualcomm provides a default DMC file suitable for most situations. However, specific use cases or technology areas may require creating and using a custom DMC.

More than one DMC file can be used simultaneously. By selecting more than one DMC in the configuration screen, the modem data enabled by each DMC is muxed together, leading to the collection of modem data for both DMCs. However, only a single ISF file is created and the modem data is not separated per DMC.

6.1.1 Creating a custom DMC file

QTI Logkit does not provide a mechanism to create a DMC file. The QXDM Professional™ tool, supplied by Qualcomm, should be used to create a DMC. Refer to *QXDM Users Guide* (80-V1241-1) for instructions.

6.1.2 Selecting a custom DMC file added to the qti_logkit_config.xml

To make a custom DMC file selectable, the QTI Logkit application uses a custom configuration that includes the DMC location. This custom configuration involves modifying the qti_logkit_config.xml file (see Section 7.1).

After the DMC file location has been added to the qti_logkit_config.xml file, the custom DMC is visible under the **Modem** group on the **Configure** screen**.**

Toggle the **Custom** switch to enable the custom DMC, as shown in Figure 6-1.

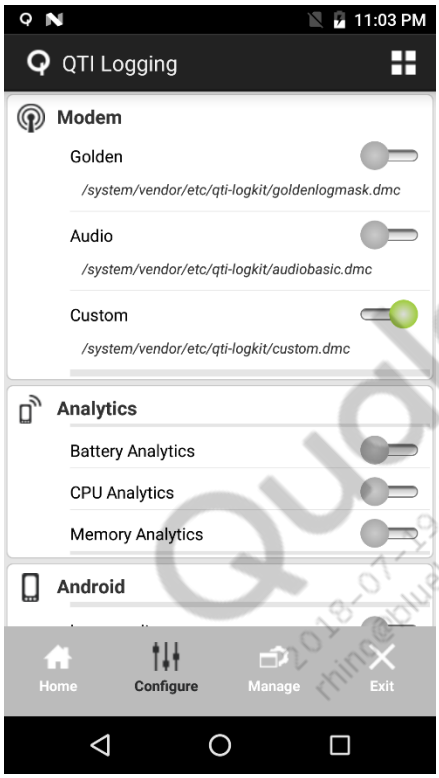


Figure 6-1 Custom DMC enabled

6.1.3 Runtime selection of DMC files from default qti_logkit_config.xml

To add custom DMCs without editing the qti_logkit_config.xml file, the DMCs files must be pushed to a specific location on the device. By default, the location is:

- /data/misc/qti-logkit/shared-public/ (Android N and prior)
- /data/vendor/qti-logkit/shared-public/ (Android O)

This location can be altered in the qti_logkit_config.xml, (see Section 7.2 for details).

After the DMC files have been pushed to the appropriate location on the device, they are visible in the QTI Logkit UI.

Tap the dropdown to select the custom DMC (see Figure 6-2).

NOTE: Only one custom DMC can be selected at a time. To disable the custom DMC, either select **None** or toggle the switch to disable custom DMCs.

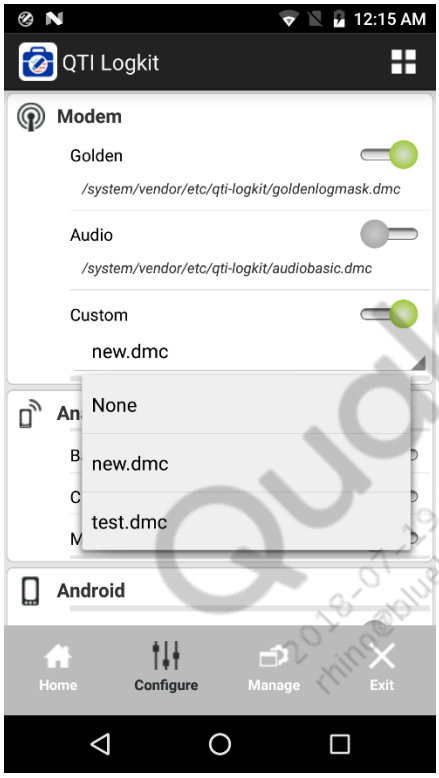


Figure 6-2 Select custom DMC without changing config file

7 Customizations

QTI Logkit has a default set of assumed capabilities and configurations. However, many of these features and capabilities can be modified.

QTI Logkit stores the configuration in the qtilogkit-config.xml file. The path to this file is:

`/system/vendor/etc/qti-logkit`

In USER builds, any changes to these files must be compiled into the device image. For USERDEBUG builds, the qti_logkit_config.xml file can be updated via `adb push`.

NOTE: The device must be restarted after pushing a new .xml to USERDEBUG builds. Changes to .xml files in USER builds must be compiled into the device image.

7.1 Adding a custom DMC

1. To add a custom DMC to QTI Logkit (as described in Section 6.1.2), add the text highlighted in yellow to the `<ConfigureLogs>` section of the qti_logkit_config.xml file.
2. Add the corresponding DMC file (generated by QXDM to the device at the location specified in the .xml file.

```
<ConfigureLogs>
   <Submenu Name="Modem" Enable="true">
      <UIElementDMC
         Name="Golden"
         Path="/system/vendor/etc/qti-logkit/goldenlogmask.dmc"
         Visible="true"
         Enable="true"/>
      <UIElementDMC
         Name="Audio"
         Path="/system/vendor/etc/qti-logkit/audiobasic.dmc"
         Visible="true"
         Enable="false"/>

      <UIElementDMC
         Name="Custom"
         Path="/system/vendor/etc/qti-logkit/custom.dmc"
         Visible="true"
         Enable="false"/>

   </Submenu>
```

7.2 Modifying the custom DMC path

To modify the custom DMC path (as described in Section 6.1.3), change the text highlighted in yellow in the qti_logkit_config.xml file to the new path.

```
<UIElementCustomDMCPath
   Name="Custom"
   Path="/data/misc/qti-logkit/shared-public/"
   Visible="true"
   Enable="false"/>
```

NOTE: In Android O, the default path is /data/vendor/qti-logkit/shared-public/

7.3 Hiding deleted sessions

1. To hide deleted sessions, or to limit the number of deleted sessions shown, change the options under `<DeletedPackages>` (highlighted in yellow) in the qti_logkit_config.xml file.
2. Change `bVisible` to `"false"` to prevent the display of sessions.
3. Change `MaxNum` to the preferred number to limit the number of deleted sessions shown.

```
<!--                        DeletedPackages

         Show deleted packages in the UI

         bVisible - Are deleted packages visible
         MaxNum - Max number of deleted packages to keep across reboot
-->
<DeletedPackages bVisible="true" MaxNum="100"/>
```

7.4 Changing AutoPackage setting

Default application behavior can be altered in the qti_logkit_config.xml file. The user can choose from the following options if sessions or events are automatically packaged without user interaction:

- Always automatically packaged – Set to “true”
- Never automatically packaged – Set to “false”
- Only events are automatically packaged (sessions still prompt user before they are packaged) – Set to “Events”

```
<!--                        AutoPackage
         Auto package (accept default name and location) when dialog would
            have been shown
         Only one entry should exist for this
         bEnable - Enable auto package or not
                   "false"    - nothing will be auto-packaged
```

```
                    "true"     - everything will be auto-packaged
                    "Events"   - only events will be auto-packaged
                                 sessions will prompt with the package
dialog
-->
<AutoPackage bEnable="Events"/>
```

7.5 Changing AutoZip setting

Default application behavior can be altered in the qti_logkit_config.xml file. The user can set if packages are automatically zipped during packaging.

```
<!--                        AutoZip

          Auto Zip when package completes
          Only one entry should exist for this

          bEnable - Enable auto zip or not (boolean)
-->
<AutoZip bEnable="true"/>
```

7.6 Changing AutoLaunch setting

Default application behavior can be altered in the qti_logkit_config.xml file. The user can set if QTI Logkit is automatically launched on reboot. AutoLaunch takes place only if QTI Logkit was running at the time the phone was reset.

```
<!--                        AutoLaunch

          Auto launch the UI after reboot
          Only one entry should exist for this

          bEnable - Enable auto launch or not (boolean)
-->
<AutoLaunch bEnable="true"/>
```

7.7 Check modem version

Default application behavior can be altered in the qti_logkit_config.xml file. The user can set if QTI Logkit queries diag to get the MPSS version when the client launches.

```
<!--                        CheckMPSSVer

          On client launch, attempt to query diag to get the MPSS version.
          This may cause a brief interruption if QXDM or another tool is
logging
          over USB
```

```
        Only one entry should exist for this
        If not present the default is disabled

        bEnable - Enable checking MPSS version (boolean)
-->
<CheckMPSSVer bEnable="false"/>
```

7.8 Customizing the launch dial sequence

The launch dial sequence can be customized by modifying (or adding) the `/system/vendor/etc/qti-logkit/startup.txt` file to include the preferred dial sequence (on a single line).

8 Notifications and Error Handling

Error handling and notifications are processed differently by QTI Logkit depending if the application is in the background or foreground.

8.1 Session status notifications

When QTI Logkit is running, the user sees the QTI Logkit notification icon in the upper-left notification area. By dragging down the notification tray, the user can determine if QTI Logkit is actively collecting logs in the background.

Stopping log collection from the notification screen

1. When the notification screen displays, tap **Stop** to stop log collection.
2. Tap anywhere in the notification area to bring QTI Logkit back into the foreground.

Figure 8-1 shows the notification displayed when QTI Logkit has log collection running while in the background.

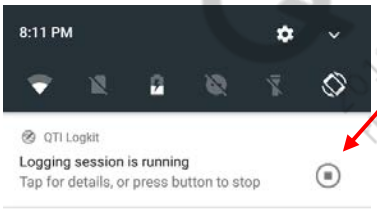


Figure 8-1 Background collection notification while session is running

Starting log collection from the notification screen

1. When the notification screen displays, tap **Start** to start log collection.
2. Tap anywhere in the notification area to bring QTI Logkit into the foreground.

Figure 8-2 shows the notification displayed when QTI Logkit has log collection stopped while in the background.

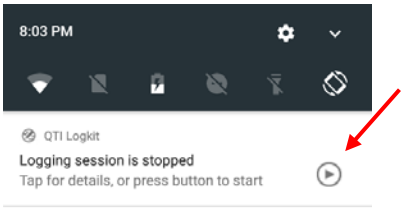


Figure 8-2 Background collection notification while session is Stopped

NOTE: See Chapter 3 for starting and stopping log collections.

8.2 Error handling with QTI Logkit in foreground

If QTI Logkit is running in the foreground when an error occurs (e.g., a write error), an error message displays to the user (see Figure 8-3).

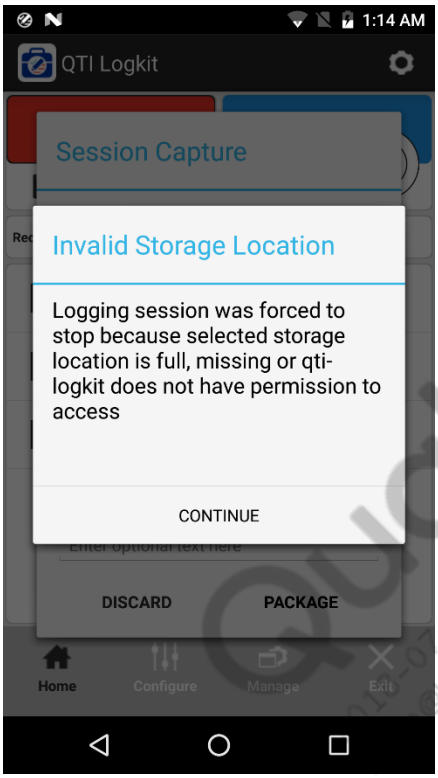


Figure 8-3 An error occurred at the selected storage location during session

8.3 Error notifications with QTI Logkit in background

If an error occurs when QTI Logkit is in the background, it triggers a notification. As shown in Figure 8-4, there are two QTI Logkit notifications visible; one is the standard session status notification and the other is the error notification.

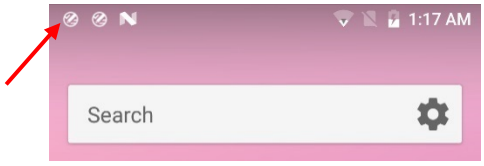


Figure 8-4 Extra QTI Logkit notification

The notification tray shows users details about the error or warning that occurred (see Figure 8-5). Tapping anywhere in the notification area brings QTI Logkit to the foreground. Swiping the notification to the left dismisses it.

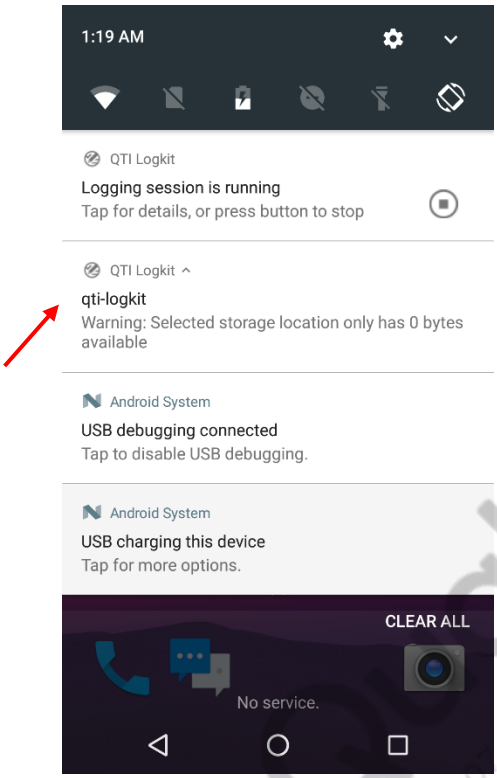


Figure 8-5 Error/warning notification

A Installation/Upgrade of APK

In most product lines, QTI Logkit comes pre-integrated and no installation is required, however, newer versions may become available. This chapter describes how to install or upgrade the APK. There are many ways to install an Android APK, however, this document only covers the adb method.

The adb commands to run to install the APK are:

```
adb install -r qti-logkit.apk
```

On USERDEBUG builds, the config file can be updated using the following commands.

Android O

```
adb root (Note: this will fail on USER builds)
adb shell rm -rf /data/vendor/qti-logkit/shared-privileged/*.xml
adb push qti_logkit_config_O.xml /data/vendor/qti-logkit/shared-privileged/
adb reboot
```

Android N and prior

```
adb root (Note: this will fail on USER builds)
adb shell rm -rf /data/misc/qti-logkit/shared-privileged/*.xml
adb push qti_logkit_config.xml /data/misc/qti-logkit/shared-privileged/
adb reboot
```

NOTE: QTI Logkit has dependencies in the system image. Check with your Product Manager to see if your base Android image can support QTI LogKit.

B Determining USER vs. USERDEBUG Build Variants

If unknown, there are multiple methods to determine if a build is a USER or USERDEBUG build.

- If the device is rootable via ADB, it is a USERDEBUG build. If not, it is a USER build.
- Inspect the system software version name.
  - Using the device's native UI, navigate to **Settings > About Phone > build number**. The filename should conform to the convention *MSMXXXX-user-ABC* or *MSMXXXX-userdebug-ABC*.