Mobile Application Logging

Workspace ONE® has a logging module for mobile apps. Use the module's programming interface to

- write messages to a log store on the device.
- send stored logs to the Workspace ONE management console.

The logging module is available in the Workspace ONE software development kit (SDK) for Android and for iOS. Logs will be sent to the Workspace ONE Unified Endpoint Manager (UEM) console with which the app is enrolled. This document covers usage for Android.

Table of Contents

Overview diagram

This diagram represents the logical components of the Workspace ONE app logging system.

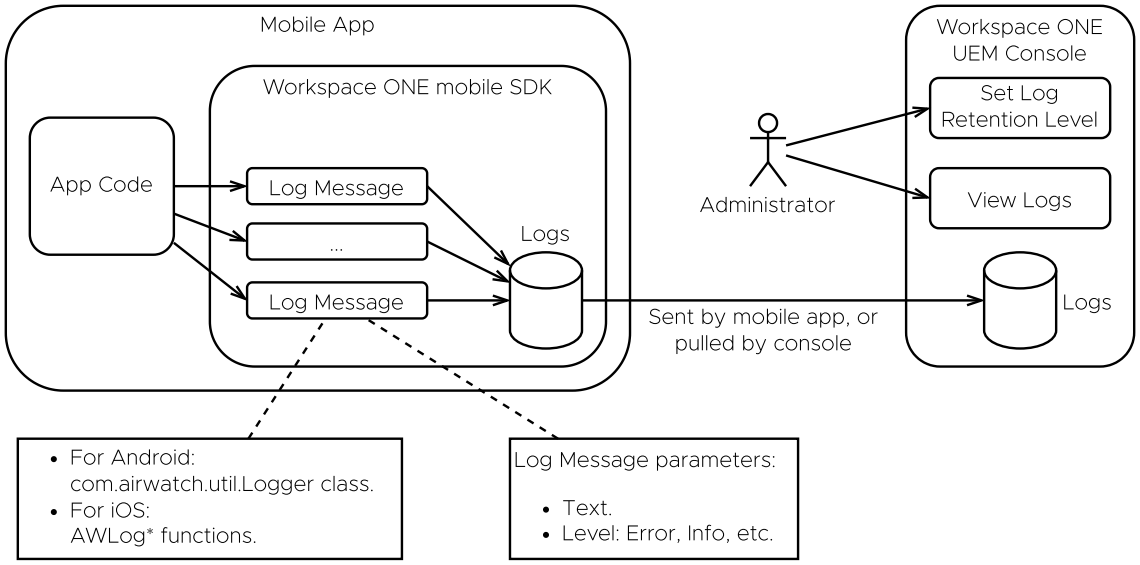


Diagram: Workspace ONE app logging system

The SDK runtime generates log messages itself, and stores them on the device. When the app writes log messages to the SDK those are added to the same store. When the SDK uploads logs to the console it uploads both its own messages and messages from the app.

Programming interface for Android

The Workspace ONE app logging programming interface for Android is in the `com.airwatch.util.Logger` class in the SDK. It is similar to the native `com.util.log` class, which is documented on the Android developer website, for example here developer.android.com/.../util/Log.

- Log messages are String type.
- A tag String and symbolic logging level are specified on each message write.
- Keep tag strings under 20 characters length.

Write log message for Android

This code illustrates the interface for writing log messages.

```
import com.airwatch.util.Logger
// ...
private val TAG = MyClass::class.java.simpleName
// ...
Logger.d(TAG, "Debug message")
// ...
Logger.v(TAG, "Verbose message")
// ...
Logger.i(TAG, "Information message")
// ...
Logger.w(TAG, "Warning message")
// ...
Logger.e(TAG, "Error message")
// ...
try {
    // Code that could throw.
}
catch (throwable:Throwable) {
    Logger.e(TAG, "Error message", throwable)
}
```

Send Logs for Android

The SDK runtime will send logs to the management console when requested to do so by the console. That is sufficient for general requirements. Your app may have a requirement for app and SDK logs to be sent when the user chooses. In that case you can use the `com.airwatch.util.ShareLogUtil sendLogsToConsole()` programming interface as illustrated here.

```
import com.airwatch.util.ShareLogUtil
// ...
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.send_logs-> {
            ShareLogUtil.sendLogsToConsole()
            true
        }
        else -> super.onOptionsItemSelected(item)
    }
}
```

Sending logs is a *snapshot* type of upload. The SDK will upload the content of the log store as it is when the send method is called. The method returns immediately and the SDK manages the upload task asynchronously.

Warning Don't send logs from your app without user intervention. For example, don't send logs when your app launches, nor routinely at regular intervals. Sending logs consumes inbound bandwidth to the console. Requiring user intervention is a form of throttling that limits inbound bandwidth consumption.

Open Source sample code for Android

The Workspace ONE official Open Source sample code for Android has a complete app that illustrates use of the logging programming interface. It is published here.

github.com/.../workspace-ONE-SDK-integration-samples/.../logger

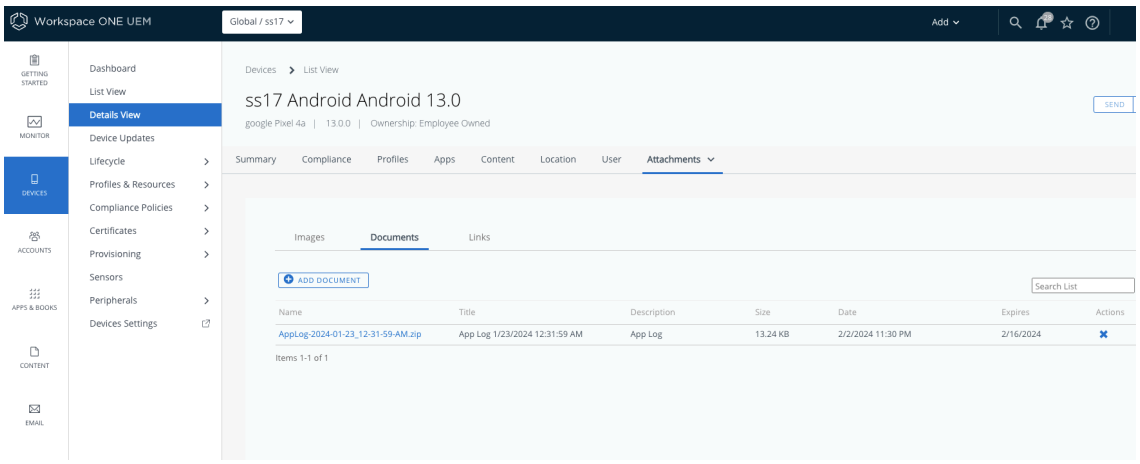
Console user interface

Application and SDK log uploads can be accessed in the UEM console user interface. Proceed as follows.

1. Log in to the enterprise UEM.
2. Navigate to Devices, List View and select the device on which the app is running.
3. Navigate to More, Attachments, Documents.

That will open a list of uploads for the device.

This screen capture shows the list of uploads and the navigation path in the UEM console user interface.



Screen capture: Log uploads in the management console

For more details see the Workspace ONE product documentation, for example here.
https://docs.omnissa.com/bundle/android-device-managementVSaaS/page/AndroidManagementManageAndroid.html

Document Information

Revision History

The following table shows the revision history of this document.

Date	Revision
24 Jan 2024	Initial Publication.
18 Feb 2025	Brand Revision.
04 Aug 2025	Updated the revision history format.

Legal

This software is licensed under the Omnissa Software Development Kit (SDK) License Agreement; you may not use this software except in compliance with the License.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This software may also utilize Third-Pary Open Source Software as detailed within the open_source_licenses.txt file.