

SDK Diagnostics

Note: This feature is applicable for Framework integrated and Network integrated mobile apps only.

The Workspace ONE® SDK Diagnostics feature provides built-in SDK diagnostic capabilities to help developers and administrators troubleshoot SDK-related issues. This feature includes a comprehensive diagnostic UI that displays detailed SDK status, device information, and integration diagnostics, with one-tap report generation for support and troubleshooting.

Table of Contents

- Key Concepts.....2
- Overview.....3
- Launching the Built-in Diagnostic UI.....4
 - Implementation.....4
 - Information Displayed.....5
 - Scan Success.....6
 - Scan Failure.....7
 - Feedback.....8
 - Report Generation and Export.....9
 - Sample Implementation.....9
- Document Information.....10

Key Concepts

- **Diagnostics UI:** A built-in user interface that displays comprehensive SDK diagnostic information.
- **Diagnostic Checks:** A one-tap way to run predefined SDK checks that assess the overall health of the SDK.
- **Report Generation:** Structured diagnostic reports with detailed results are shared for troubleshooting.

Overview

The SDK Diagnostic Module provides visibility into the SDK's operational status and runs comprehensive health checks to quickly identify issues. It enables developers and support teams to:

- View device details, SDK version, enrollment information, and UEM configuration
- Run diagnostic checks for connectivity, secure communication, authentication, and compliance
- Identify configuration mismatches, certificate issues, or network problems
- Generate and share detailed diagnostic reports for troubleshooting

Launching the Built-in Diagnostic UI

The SDK Diagnostics feature includes a ready-to-use diagnostic UI that can be launched directly from your application. The diagnostic module is included as part of the Workspace ONE SDK, so no additional dependencies are required.

Implementation

To launch the diagnostic UI in your Android application:

```
// Launch the diagnostic UI
// context must be an Activity context
DiagnosticLauncher.launch(context)
```

Note: The **context** parameter must be an **Activity context**. Passing an Application context will cause the diagnostic UI to fail to launch.

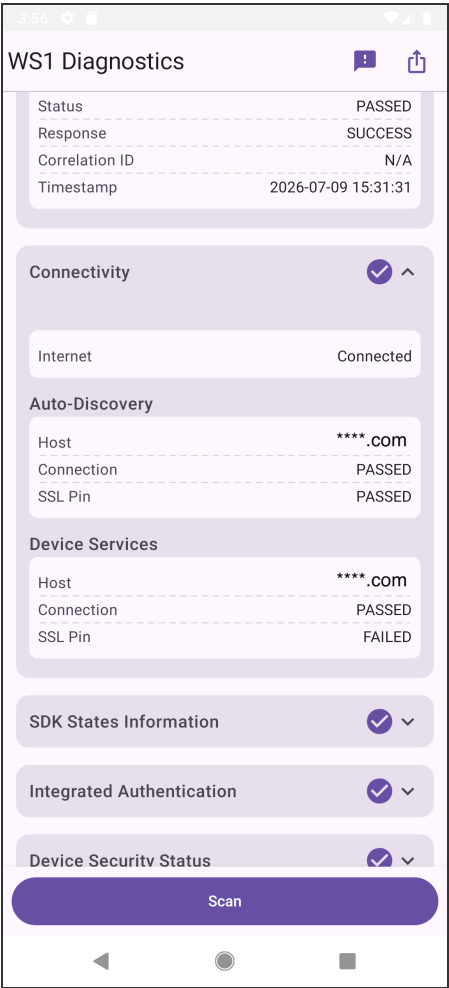
Integrate this as a menu option, button, or developer settings option in your application.

Information Displayed

- Diagnostic checks are user-triggered. If issues are suspected, users can open this screen and tap **Scan** to run a comprehensive diagnostic scan.
- Results are displayed in expandable sections, each showing a pass/fail status indicator.
- Generated reports can be shared with support teams for troubleshooting via options provided in the SDK Diagnostics UI.

Scan Success

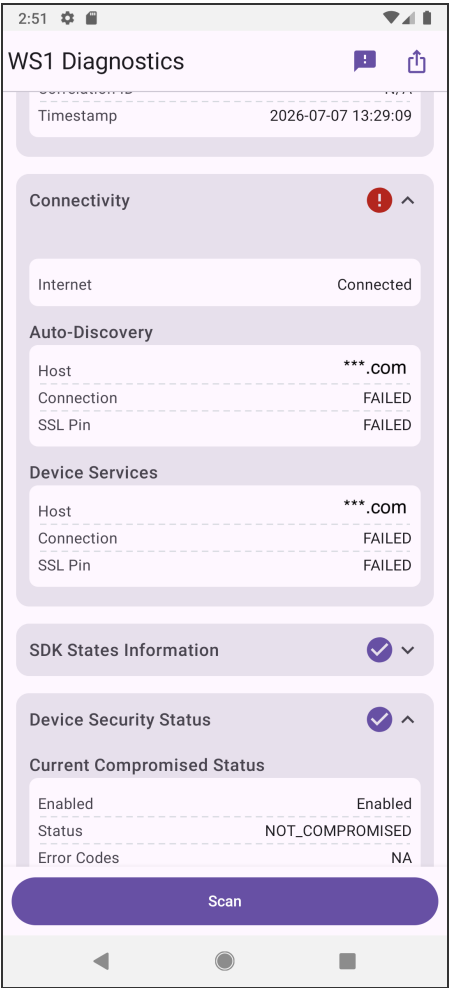
When all checks pass, the section displays a checkmark.



Scan Success

Scan Failure

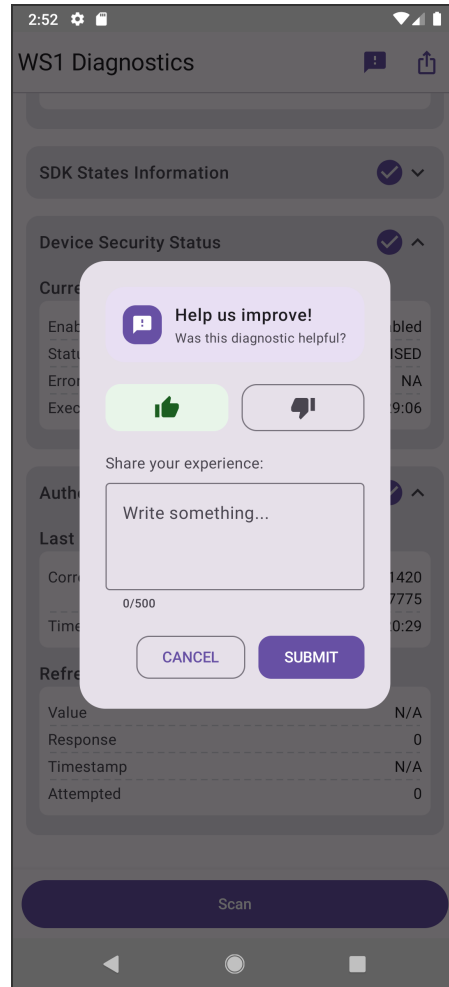
When one or more checks fail, the affected section displays a red alert icon.



Scan Failure

Feedback

A “Help us improve!” dialog allows users to share their feedback with a thumbs-up or thumbs-down and optionally provide written feedback before submitting.



Feedback Dialog

Report Generation and Export

The Diagnostics UI provides one-tap report generation that creates a comprehensive diagnostic report.

Reports can be:

- Exported via email intent for sharing with support teams.

This structured report significantly reduces troubleshooting time by providing all necessary diagnostic data in a single document.

Sample Implementation

A complete working example demonstrating the SDK Diagnostics integration is available in the Workspace ONE SDK Integration Samples repository: In the sample app, go to the main screen, tap the three-dot menu, and select “**WS1 Diagnostics**” to launch the SDK Diagnostics UI.

This sample demonstrates:

- Integrating the diagnostic module into your application
- Launching the diagnostic UI from your application’s menu
- App: [Framework Delegate Sample \(Kotlin\)](#)
- Implementation: [Framework Delegate Sample Implementation \(Kotlin\)](#)

Document Information

Revision History

The following table shows the revision history of this document.

Date	Revision
08 Jul 2026	Initial Publication.

License

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-Party Open Source Software as detailed within the [Omnissa Open Source Notices](#).