

AT&T Entertainment Experience Suite Video Optimizer 1.2

Command Line Interface User Guide

Publication Date: September 2017



Legal Disclaimer

This document and the information contained herein (collectively, the "**Information**") is provided to you (both the individual receiving this document and any legal entity on behalf of which such individual is acting) ("**You**" and "**Your**") by AT&T, on behalf of itself and its affiliates ("**AT&T**") for informational purposes only. AT&T is providing the Information to You because AT&T believes the Information may be useful to You. The Information is provided to You solely on the basis that You will be responsible for making Your own assessments of the Information and are advised to verify all representations, statements and information before using or relying upon any of the Information. Although AT&T has exercised reasonable care in providing the Information to You, AT&T does not warrant the accuracy of the Information and is not responsible for any damages arising from Your use of or reliance upon the Information. You further understand and agree that AT&T in no way represents, and You in no way rely on a belief, that AT&T is providing the Information in accordance with any standard or service (routine, customary or otherwise) related to the consulting, services, hardware or software industries.

AT&T DOES NOT WARRANT THAT THE INFORMATION IS ERROR-FREE. AT&T IS PROVIDING THE INFORMATION TO YOU "AS IS" AND "WITH ALL FAULTS." AT&T DOES NOT WARRANT, BY VIRTUE OF THIS DOCUMENT, OR BY ANY COURSE OF PERFORMANCE, COURSE OF DEALING, USAGE OF TRADE OR ANY COLLATERAL DOCUMENT HEREUNDER OR OTHERWISE, AND HEREBY EXPRESSLY DISCLAIMS, ANY REPRESENTATION OR WARRANTY OF ANY KIND WITH RESPECT TO THE INFORMATION, INCLUDING, WITHOUT LIMITATION, ANY REPRESENTATION OR WARRANTY OF DESIGN, PERFORMANCE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, OR ANY REPRESENTATION OR WARRANTY THAT THE INFORMATION IS APPLICABLE TO OR INTEROPERABLE WITH ANY SYSTEM, DATA, HARDWARE OR SOFTWARE OF ANY KIND. AT&T DISCLAIMS AND IN NO EVENT SHALL BE LIABLE FOR ANY LOSSES OR DAMAGES OF ANY KIND, WHETHER DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, SPECIAL OR EXEMPLARY, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF GOODWILL, COVER, TORTIOUS CONDUCT OR OTHER PECUNIARY LOSS, ARISING OUT OF OR IN ANY WAY RELATED TO THE PROVISION, NON-PROVISION, USE OR NON-USE OF THE INFORMATION, EVEN IF AT&T HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES OR DAMAGES.



Table of Contents

1	Intr	Introduction1		
2	Sys	System Prerequisites1		
3	AT&T Video Optimizer Command Line Interface Package3			
4	Pre	paring a Device for Testing3		
5	Usi	ng AT&T Video Optimizer from the Command Line3		
	5.1	Video Optimizer Command Line Usage Examples5		
	5.2	Analysis Report Output5		
6	Cor	nmand Summary5		
	6.1	startcollector		
	6.2	stop6		
	6.3	analyze6		
	6.4	deviceid7		
	6.5	listcollectors		
	6.6	help7		
7 Error Messages		or Messages7		
	7.1	Core7		
	7.2	Rooted Android Collector8		
	7.3	Console9		
	7.4	VPN Android Collector9		
8	8 Additional Video Optimizer Help10			



Table of Tables

Table 7-1: AT&T Video Optimizer Command Line Tool – Core Error Messa	ages8
Table 7-2: AT&T Video Optimizer Command Line Tool – Rooted Android (Collector Error Messages9
Table 7-3: AT&T Video Optimizer Command Line Tool – Console Error Me	essages9
Table 7-4: AT&T Video Optimizer Command Line Tool – VPN Android Coll	lector Error Messages 10



1 Introduction

The AT&T Video Optimizer Command Line Interface provides access to the main functionality of AT&T Video Optimizer through console commands, allowing you to easily integrate Video Optimizer data collection and analysis into your testing and build tools without having to manipulate the Video Optimizer UI.

Using these commands, you can select from the devices tethered to your computer, start either of the Video Optimizer data collectors, stop the collector, and generate an analysis report in JSON or HTML format – all by using commands from the console.

The AT&T Video Optimizer Command Line Interface is designed primarily for use in the following scenarios:

- As a compliment to other testing tools. Command line access makes Video Optimizer easier to integrate with other tools.
- As a part of automated testing: The ability to start and stop data collection, and output analysis without a UI, makes it much easier to integrate Video Optimizer into your automated functional testing suite.
- As a part of your build cycle: The Video Optimizer Command Line Interface can be used within Jenkins and other build tools to systematically check best practices.

This User Guide lists the system prerequisites for using the AT&T Video Optimizer Command Line Interface, tells you how to prepare your device for testing with Video Optimizer, and shows you how to use Video Optimizer from the command line. This guide also includes a complete summary of all the Video Optimizer Command Line Interface commands and error messages.

2 System Prerequisites

Before using the AT&T Video Optimizer Command Line Interface, ensure that the following have been installed on your computer:

- The <u>Android SDK</u> must be installed so that the Android Debug Bridge (ADB) can be used. Install the latest version with the latest Android API included, that is specific for your device and system OS. Note: When installing the Android SDK, be sure to select the Android SDK Tools, Android SDK Platform-tools, and Android Build-tools for installation as well.
- Java 6 or above.

For computers running a Microsoft Windows OS, the following is also required:

<u>WinPcap or Wireshark</u>

Note: To install WinPcap on Windows 8, set the compatibility mode to Windows 7 in WinPcap Properties.



The AT&T Video Optimizer Command Line Interface requires access to the Android Debug Bridge (ADB). Please add the following to your environmental variables path (depending on your OS):

For Mac OS X:

export ANDROID_HOME=/<installation location>/android-sdk-macosx
export PATH=\${PATH}: \$ANDROID_HOME/tools: \$ANDROID_HOME/platform-tools

For Microsoft Windows:

set ANDROID_HOME=C: <i nstallation location>android-sdk-windows set PATH=%PATH%; %ANDROID_HOME% \tools ; %ANDROID_HOME% $\platform-tools$

For Linux:

.

 $\label{eq:loss} export $$ ANDROID_HOME=/<installation location>/android-sdk-linux $$ export $$ PATH=${PATH}: $$ ANDROID_HOME/tools: $$ ANDROID_HOME/platform-tools $$ and $$ an$



3 AT&T Video Optimizer Command Line Interface Package

The AT&T Video Optimizer Command Line Interface is released as a zip with the following components:

- vo.sh A shell script file to be used on Mac or Linux systems.
- vo.bat A batch file to be used on Windows systems.
- VOCLI.jar An executable jar file with all the required components and executable content needed to utilize the Video Optimizer command line interface.
- ATT_SDK_License.txt AT&T sdk license
- Third_Party_licenses.txt List of third party licenses for open source libraries used to build command line tool.

4 Preparing a Device for Testing

The AT&T Video Optimizer Command Line Interface supports data collection on rooted Android devices, and on un-rooted Android devices by using the Video Optimizer VPN collector. To prepare your device for data collection using either of these methods, do the following:

- Ensure that you have a USB driver installed that matches the Android test device. For example, Samsung devices may require the installation of Samsung Kies, and HTC devices may require the installation of HTC Sync Manager. Consult your device manufacturer's support website for the latest USB driver, and for driver installation instructions that apply to your device.
- 2. Ensure that your device is connected to a computer via USB and that the connection is working properly.
- 3. For cleaner traces, we recommended that you test with a device that has only the test app running.

Note: If you are using the Android rooted collector, ensure that Android version 2.4x or above is installed on your test device.

Note: If you are using the Android VPN collector, ensure that you are testing on a clean, un-rooted Android device running Android 4.4 or above. The Android VPN collector has been tested on the Samsung Galaxy S5 and LG Nexus 5 running Android 4.4 (Kitkat), and Motorola Nexus 6 and LG Nexus 5 running Android 5 (Lollipop).

5 Using AT&T Video Optimizer from the Command Line

To use AT&T Video Optimizer from the command line, do the following:

 Before starting Video Optimizer, make sure that the Android Debug Bridge (ADB) is running and is recognizing your device. Use the following ADB command to list the devices that are connected to your computer:

 $adb \ devi\,ces$



This command will list the serial number of the attached devices and list their status. A status of "device" indicates that the device instance is connected to the adb server (your computer). Note that this state does not imply that the Android system is fully booted and operational, since the instance connects to adb while the system is still booting. However, after boot-up, this is the normal operational state of an emulator/device instance. Commands can then be sent to an individual device instance using the syntax:

adb -s <serialNumber> <command>

For more information and help with ADB commands, see <u>http://developer.android.com/tools/help/adb.html</u>.

- 2. Open a command window and access the Video Optimizer Command Line Interface commands by using the syntax:
- vo [commands] [arguments]

To view the help menu showing a list of all the available commands with syntax and help for each, use the standard console commands for help:

vo -h, vo -?, or - -help

For more help, you can also see the <u>Command Summary</u> section of this document.

- 3. Start the Video Optimizer collector using the startcollector command with parameters that indicate the following:
 - The Video Optimizer collector type: Rooted Android (rooted_andoid) or VPN Android (vpn_android).
 - The location of the trace output.
 - Whether you want Video Optimizer to record a video of the trace (yes|no).

The following example shows the command to start the rooted Android collector, output the trace to a folder called test, and record a video of the trace.

vo --startcollector rooted_android --output /User/documents/test --video slow

- 4. Test the app on your device or use a test script to run automated tests.
- 5. When the test is completed, stop the Video Optimizer collector by using the stop command from the ">" prompt when the collector is running.
- > stop
- 6. Instruct Video Optimizer to analyze the specified trace using the analyze command. Optionally, you can specify where the analysis should be stored, and whether the format of the analysis report should be HTML or JSON.

The following example shows the command to analyze the trace data in a folder called test and output the analysis report in HTML format to a file named report.html.

vo --analyze /User/documents/test --output /documents/report.html --format html



5.1 Video Optimizer Command Line Usage Examples

The following examples demonstrate common Video Optimizer Command Line Tool commands:

1. Run the rooted Android collector and capture a trace file with video to the specified location:

vo --startcollector rooted_android --output /User/documents/test --video slow

2. Analyze the trace that was collected in the previous example and generate an analysis report in HTML format, to the specified location:

vo -- analyze /User/documents/test -- output /documents/report.html -- format html $% \mathcal{A} = \mathcal{A} = \mathcal{A} = \mathcal{A}$

3. Analyze a trace capture file (traffic.cap) and generate an analysis report in JSON format, to the specified location.

vo --analyze /User/documents/test/traffic.cap --output documents/report.json $-format\ json$

5.2 Analysis Report Output

The Video Optimizer Command Line Tool generates analysis reports in JSON or HTML format to support these two common scenarios:

The HTML report is easy to display and read, and is intended for a quick, basic summary of which Video Optimizer Best Practice tests have passed or failed. The format of the report has the same general contents and look as the Best Practices Summary page in the AT&T Video Optimizer.

The JSON report is intended for programmatically extracting information about the analysis results, and could be used as part of an automated build or testing process.

6 Command Summary

The following sections contain a summary of the commands that are available in the Video Optimizer Command Line tool.

For all of the Video Optimizer commands listed here, the usage syntax is:

vo [commands] [arguments]

6.1 startcollector

Starts an instance of the specified Video Optimizer collector and begins capturing the trace.

Syntax:

vo --startcollector [rooted_android|vpn_android] –output [trace location] –video [hd|sd|slow|no]

Parameters:



collector type – Specifies which Video Optimizer collector to start. The acceptable values for this parameter are *rooted_android* to collect data on a rooted Android device and *vpn_android* to collect data on an un-rooted Android device. The default value for this parameter is *vpn_android*.

trace location – Specifies the full path name of a trace folder or a trace file in which to store the captured trace.

video - An optional command that specifies whether Video Optimizer should record a video of the trace while collecting data. The acceptable values for this parameter are *hd* (high definition), *sd* (standard definition) *slow*, and *no* to indicate that a video of the trace should not be recorded. Slow video is 1-2 frames per second. The default value for this parameter is *no*.

Example:

vo --startcollector rooted_android --output /User/documents/test --video $sl\, ow$

6.2 stop

Instructs the Video Optimizer collector to stop. This command is issued from the ">" prompt when the collector is running.

> stop

6.3 analyze

Instructs Video Optimizer to begin analyzing the specified trace folder or trace file.

Syntax:

vo --analyze [trace location] --output [report location] format [json|html]

Parameters:

trace location – Specifies the full path of a trace folder or a trace file to analyze.

report location - Specifies the full path name of the analysis report file.

format – An optional command that specifies the format of the Video Optimizer analysis report. The acceptable values for this parameter are json to specify the JSON format, or html to specify the HTML format. The default value of this parameter is json.

Example:

vo --analyze /User/documents/test --output /User/documents/report.html --format html



6.4 deviceid

An optional command for specifying the device id of an Android device, or the serial number of an iOS device. If more than one device is listed and available when using the *adb devices* command, use the deviceid command to indicate which device Video Optimizer should capture trace data from. The default value of this parameter is the id (or serial number on iOS) of the first device that is connected.

Syntax:

vo --deviceid [device id]

Parameters:

device id – Specifies the device id of an Android device or the serial number of an iOS device.

Example:

vo --deviceid abc1234567

6.5 listcollectors

An optional command that lists the available Video Optimizer data collectors on your computer. The data collectors that are currently available at this time are rooted_android and vpn_android.

Syntax:

vo --listcollectors

6.6 help

Displays a list of the available Video Optimizer commands with syntax and descriptions for each command.

Syntax:

--help,-h,-?

7 Error Messages

The tables in this section list and describe the error messages for Video Optimizer Command Line usage.

7.1 Core

The following are the core error messages.



Code	Error	Description
100	Trace directory not found	Video Optimizer cannot find or access trace directory user specified. Please check if the directory exists.
101	Trace file not found.	Video Optimizer cannot find or access trace file user specified. Please check if the file exists.

Table 7-1: AT&T Video Optimizer Command Line Tool – Core Error
Messages.

7.2 Rooted Android Collector

The following are the error messages for the Rooted Android Collector.

Code	Error	Description
200	Android Debug Bridge failed to start	Video Optimizer Collector tried to start Android Debug Bridge service. The service was not started successfully.
201	Failed to install Android App on device	Video Optimizer tried to install Video Optimizer Collector on device and failed. Try to manually install it, then try again.
202	Found existing trace directory that is not empty	Video Optimizer found an existing directory that contains files and did not want to override it. Some files may be hidden.
203	No Android device found.	Video Optimizer cannot find any Android deviced plugged into the machine.
204	Android device Id or serial number not found.	Video Optimizercannot find any Android deviced plugged into the machine that matched the device Id or serial number you specified.
205	Device has no space	Device does not have any space for saving trace
206	Video Optimizer rooted Android collector already running	There is already a Video Optimizer collector running on this device. Stop it first before running another one.
207	Failed to create local trace directory	Video Optimizer tried to create local directory for saving trace data, but failed.
208	Failed to extract tcpdump	Video Optimizer failed to extract tcpdump from resource bundle and save it to local machine.
209	Failed to install tcpdump	Video Optimizer failed to install tcpdump on Emulator. Tcpdump is required to capture packet
210	Failed to run Video Optimizer Data Collector	Video Optimizer Analyzer tried to run Data Collector on device and received error from device.



Code	Error	Description
211	Failed to connect to device SyncService	Video Optimizer failed to get SyncService() from IDevice which is used for data transfer
212	Failed to set execute permission on Tcpdump on device	Error occurred while trying to set permission on tcpdump file on device. Execute permission is required to run it.
213	Device not rooted	Video Optimizer detected that device is not rooted. A rooted device is required to run this collector

 Table 7-2: AT&T Video Optimizer Command Line Tool – Rooted Android Collector

 Error Messages.

7.3 Console

Code	Error	Description
300	Unsupported collector	Collector name passed in is not supported.
301	Output location missing	Location to save data to was not given.
302	Unsupported report format	Report format entered is not supported. Type - -help to see supported commands and options.
303	Invalid video option	Valid video option is yes or no. Invalid value was entered.
304	File or directory exists	Video Optimizer found existing file or directory
305	Collector not found	Data collector specified cannot be found. Make sure data collector is installed.

The following are the Console error messages.

 Table 7-3: AT&T Video Optimizer Command Line Tool – Console Error Messages.

7.4 VPN Android Collector

The following are the error messages for the VPN Android Collector.

Code	Error	Description
400	Android Debug Bridge failed to start	Video Optimizer Collector tried to start Android Debug Bridge service. The service was not started successfully.
401	Failed to install Android App on device	Video Optimizer tried to install Video Optimizer Collector on device and failed.
402	Found existing trace directory that is not empty	Video Optimizer found an existing directory that contains files and did not want to override it. Some files may be hidden.
403	No Android device found.	Video Optimizer cannot find any Android deviced plugged into the machine.



Code	Error	Description
404	Android device Id or serial number not found.	Video Optimizer cannot find any Android deviced plugged into the machine that matched the device Id or serial number you specified.
405	Failed to run VPN APK	Video Optimizer failed to run Data Collector in device
406	Failed to create local trace directory	Video Optimizer tried to create local directory for saving trace data, but failed.
407	VPN activation timeout	Video Optimizer waited 15 seconds for VPN service to activate.
411	Failed to connect to device SyncService	Video Optimizer failed to get SyncService() from IDevice which is used for data transfer

 Table 7-4: AT&T Video Optimizer Command Line Tool – VPN Android Collector

 Error Messages.

8 Additional Video Optimizer Help

To learn more about Best Practices for optimizing your app, testing your app, and collecting and analyzing data using AT&T Video Optimizer, see <u>AT&T Video</u> <u>Optimizer</u>.