Kostenloses eBook

LERNEN adb

Free unaffiliated eBook created from
Stack Overflow contributors.

#adb

Inhaltsverzeichnis

Über

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: adb

It is an unofficial and free adb ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official adb.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit Adb

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was Adb ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen in der Adb erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für adb neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Installation oder Setup

Speziell für Windows-System und Android Phone:

Bedarf:

1. USB-Kabel
2. Android-Gerät
3. Android-Treibersoftware

Nach dem Anschließen des USB-Kabels erkennt der PC das Android-Gerät und sucht automatisch nach den erforderlichen Treibern für das Android-Gerät. Wenn diese Treiber nicht gefunden werden, müssen Sie sie manuell installieren.

Manuelle Installation:

1. Installieren Sie zuerst das Android SDK auf Ihrem PC (Windows).

2. Nach der Installation in Android SDK-Tools Klicken Sie mit der rechten Maustaste auf den SDK-Manager und wählen Sie "Als Administrator ausführen".

3. Wählen Sie im SDK-Manager "Extras-> Google USB Driver". Aktivieren Sie das Kontrollkästchen und klicken Sie auf "Install 1 Package".

4. Wenn der Google USB-Treiber installiert ist, schließen Sie Ihr Gerät an. Warnung: Der Treiber wird nicht automatisch installiert. Wir werden es in den nächsten Schritten manuell tun.

5. Öffnen Sie den Dialog Systemeigenschaften (drücken Sie auf der Tastatur Win + Break oder suchen Sie im Startmenü nach "Computer"), klicken Sie mit der rechten Maustaste darauf und wählen Sie "Eigenschaften".

6. Klicken Sie auf den Link "Geräte-Manager".

7. Suchen Sie im Geräte-Manager Ihr Android-Gerät. Klicken Sie dann mit der rechten Maustaste darauf und wählen Sie "Treibersoftware aktualisieren".
8. Wählen Sie "Auf dem Computer nach Treibersoftware suchen".
9. Wählen Sie "Ich kann aus einer Liste von Gerätetreibern auf meinem Computer auswählen".
10. Wählen Sie "Alle Geräte anzeigen".
11. Drücken Sie die "Have Disk" -Taste.
12. Geben Sie den Pfad zum Google USB-Treiber ein. Normalerweise befindet es sich im folgenden Verzeichnis: C: \ Programme (x86) \ Android \ android-sdk \ extras \ google \ usb_driver
13. Wählen Sie "Android ADB Interface" aus der Liste der Gerätetypen.
14. Bestätigen Sie die Installation des Treibers mit "Ja".
15. Bestätigen Sie die Installation erneut mit "Installieren".
16. Wenn die Installation abgeschlossen ist, klicken Sie auf "Schließen".
17.

Einführung

`adb` ist ein Befehlszeilenprogramm zum Kommunizieren mit einer Emulatorinstanz oder einem verbundenen Gerät. Es ermöglicht das Installieren und Debuggen von Apps, das Übertragen von Dateien sowie verschiedene andere Interaktionen mit dem angeschlossenen Emulator oder Gerät. Das ADB-System besteht aus einem *Client* , der Befehle vom Hostcomputer sendet, einem *Dämon* , der auf dem verbundenen Gerät ausgeführt wird und vom Client empfangene Befehle ausführt, und einem *Server* , der auf dem Hostcomputer ausgeführt wird und die Kommunikation zwischen dem Client und verwaltet Daemon

Offizielle Dokumentation

https://developer.android.com/studio/command-line/adb.html

Erste Schritte mit Adb online lesen: https://riptutorial.com/de/adb/topic/2633/erste-schritte-mit-adb

Kapitel 2: Anzeigen von Protokollen in ADB

Examples

Anzeigen und Filtern mit Logcat

Das Anzeigen aller Protokolle aus dem Standardpuffer in der Befehlszeile kann wie folgt durchgeführt werden:

```
adb logcat
```

Dieser Befehl zeigt Ihnen alle Protokolle aus dem Hauptpuffer des Geräts. Beachten Sie, dass Sie bei der ersten Verwendung viele Informationen und einen riesigen Datenstrom erhalten. Vielleicht möchten Sie zuerst die Protokolle löschen ...

Reinigung der Protokolle:

```
adb logcat -c
```

Dadurch werden die Protokolle gelöscht und neu gestartet.

Alternative Puffer anzeigen

Neben dem Hauptpuffer gibt es zwei weitere Puffer, die wie folgt angezeigt werden können:

`adb logcat -b` *puffername* ,

Dabei ist *puffername* eine der folgenden:

- `radio` - Zeigen Sie den Puffer an, der Meldungen zu Radio / Telefonie enthält.
- `events` - Zeigt den Puffer an, der ereignisbezogene Nachrichten enthält.
- `main` - `main` anzeigen (Standard)

Protokollausgabe filtern

Logcat-Protokolle erhielten so genannte Stufen:

> **V** - Verbose, **D** - Debug, **I** - Info, **W** - Warnung, **E** - Fehler, **F** - Fatal, **S** - Stumm

Diese Ebenen werden angegeben, wenn die Anwendung diese Protokollfunktion verwendet:

```
Log.v(); // Verbose
Log.d(); // Debug
Log.i(); // Info
Log.w(); // Warning
Log.e(); // Error
```

Wenn Ihr Code-Anruf lautet:

```
Log.i("MainActivtyTag", "Showing the very first fragment");
```

In logcat sehen Sie diese Ausgabe:

```
07-27 11:34:21.027 I MainActivtyTag 66 : Showing the very first fragment
```

Dies ist also die Protokollkonvention:

```
<timestamp> <logLevel> <tag> <line> : <messge>
```

Wenn Sie beispielsweise alle Protokolle anzeigen möchten, die über die Stufe Fatal (F) verfügen:

```
adb logcat *:F
```

* *ist ein sogenannter Platzhalter - steht für alle Paketnamen*

Filtern nach Name des Anwendungspakets

Da die Paketnamen garantiert eindeutig sind, können Sie logcat nach Ihrem Paketnamen filtern. Sie können es natürlich mit dem Level-Filter kombinieren:

```
adb logcat <package name>:<log level>
```

Zum Beenden / Unterbrechen des Vorgangs - drücken Sie `Ctrl + X`

Anzeigen von Protokollen in ADB online lesen: https://riptutorial.com/de/adb/topic/4252/anzeigen-von-protokollen-in-adb

Kapitel 3: App im Debug-Modus starten

Examples

Wie warte ich auf den Debugger, bevor Sie die App starten?

Nehmen wir an, Ihre `MainActivity` heißt `MainActivity` in Ihrer App com.example.myapp. Im Manifest:

```
<activity
        android:name=".MainActivity"
        >
        <intent-filter>
          <action android:name="android.intent.action.MAIN"/>
          <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
</activity>
```

Angenommen, Sie möchten die App starten, damit sie auf den Debugger wartet, bevor die App wirklich startet.

Sie können `adb shell` , um dies zu erreichen.
In unserem Fall einfach ausführen:

```
adb shell am start -D -n com.example.myapp/com.example.myapp.MainActivity
```

Jetzt müssen Sie nur noch Ihren bevorzugten Debugger anhängen. Wenn Sie beispielsweise Intellij oder Android Studio verwenden, gehen Sie zu Ausführen -> Debugger an Android-Prozess anhängen -> und wählen Sie den Namen Ihres App-Pakets

App im Debug-Modus starten online lesen: https://riptutorial.com/de/adb/topic/4009/app-im-debug-modus-starten

Kapitel 4: Protokoll der Erfassung von ADB-Befehlen

Bemerkungen

Stellen Sie sicher, dass Ihre Automation den Befehl `adb kill-server` nicht verwendet.

Examples

in Windows

Öffnen Sie ein *Eingabeaufforderungsfenster* und führen Sie die folgenden Befehle aus:

```
adb kill-server
set ADB_TRACE=sockets
adb nodaemon server 2>&1 | for /f "usebackq tokens=7*" %a in (`findstr /c:"): '"`) do @echo %a
%b >> %USERPROFILE%\Desktop\adb_host_log.txt
```

Jetzt können Sie Ihre Android-Automatisierung ausführen. Wenn Sie fertig sind, führen Sie `adb kill-server` in einem anderen *Eingabeaufforderungsfenster* aus. Nun enthält die Datei *adb_host_log.txt* auf Ihrem *Desktop* das Protokoll aller Befehle, die alle *Adb-Clients* an den *Adb-Host* gesendet haben.

Protokoll der Erfassung von ADB-Befehlen online lesen:
https://riptutorial.com/de/adb/topic/5631/protokoll-der-erfassung-von-adb-befehlen

Kapitel 5: Übertragen von Dateien mit Push und Pull

Syntax

- adb push [-p] LOKALFERNBEDIENUNG
- adb pull [-a] [-p] REMOTE [LOCAL]

Parameter

Parameter	Einzelheiten
LOKAL	Eine Datei oder ein Verzeichnis, das sich auf dem Computer des Benutzers befindet
FERNBEDIENUNG	Eine Datei oder ein Verzeichnis, das sich auf dem Android-Gerät des Benutzers befindet
-ein	Kopieren Sie außerdem die Datei mit dem Zeitstempel und dem Dateimodus der Remote-Datei
-p	Zeigen Sie den Übertragungsfortschritt an, während die Datei oder das Verzeichnis kopiert wird

Bemerkungen

Wenn LOCAL im Adb Pull-Befehl nicht angegeben wird, wird der Dateiname von REMOTE verwendet

LOCAL kann ein relativer Pfad oder ein absoluter Pfad sein, aber REMOTE muss ein absoluter Pfad sein

Examples

Schieben Sie eine Datei auf die SD-Karte

```
adb push file.txt /sdcard/
```

Ziehen Sie eine Datei von der SD-Karte in das aktuelle Arbeitsverzeichnis

```
adb pull /sdcard/file.txt
```

Übertragen von Dateien mit Push und Pull online lesen:
https://riptutorial.com/de/adb/topic/5844/ubertragen-von-dateien-mit-push-und-pull

Kapitel 6: Verbinde mit dem Gerät

Examples

An Ihrem PC angeschlossene Geräte suchen

Aktivieren Sie USB-Debugging auf Ihrem Gerät und von `adb devices` Befehlszeile. Wenn alles in Ordnung ist, sollte die Antwort lauten:

> Liste der angeschlossenen Geräte
> 1234567890 Gerät

Wobei `1234567890` die ID des Geräts ist.
Wenn mehrere Geräte angeschlossen sind, sollten Sie alle sehen:

> Liste der angeschlossenen Geräte
> 1234567890 Gerät
> 2222222222 Gerät
> ...

Wenn Sie ein Gerät zum ersten Mal anschließen, wird ein Popup-Fenster auf Ihrem Gerät angezeigt, in dem Sie zur Bestätigung der Verbindung aufgefordert werden.

Verbinde mit dem Gerät online lesen: https://riptutorial.com/de/adb/topic/3174/verbinde-mit-dem-gerat

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit Adb	Community, Paul Ratazzi, Vesrak r
2	Anzeigen von Protokollen in ADB	Paul Ratazzi, Pavel Durov
3	App im Debug-Modus starten	Ginandi
4	Protokoll der Erfassung von ADB-Befehlen	Alex P.
5	Übertragen von Dateien mit Push und Pull	Carlo B.
6	Verbinde mit dem Gerät	TDG