

# Freeways

# UNE HISTOIRE DE LIBRE . . .



## "Stop looking the world through your Windows"



# Contents

|   |                                       |   |
|---|---------------------------------------|---|
| 1 | Freeways                              | 1 |
| 2 | Android Debug Bridge                  | 2 |
| 3 | Nanotechnologies peut sauver des vies | 6 |





!  
hum!

# Android Debug Bridge

## Généralités

ADB est l'acronyme de Android Debug Bridge, est un outil de communication entre l'appareil **Android** et l'**ordinateur** auquel ce dernier est connecté. Pour ceux qui ne le savent pas encore, Android est un système d'exploitation de la famille Linux (Multilingue), basé sur un noyau

**Monolithique**, (noyau Linux modifié), sous licence **Apache 2.0** et **GNU GPL 2**, dont la console est un moyen rapide et puissant pour accéder au système. Ainsi, l'ADB permet d'accéder à la console de l'appareil Android via l'invite de commande.

## Utilités

ADB est très puissant et permet d'administrer le système en profondeur. De ce fait il nous est possible d'installer des applications, déplacer des fichiers, redémarrer le Smartphone et possède aussi

beaucoup de commandes possibles. Ce tutoriel recensera donc les commandes les plus couramment utilisées et détaillera les actions qui en découlent.

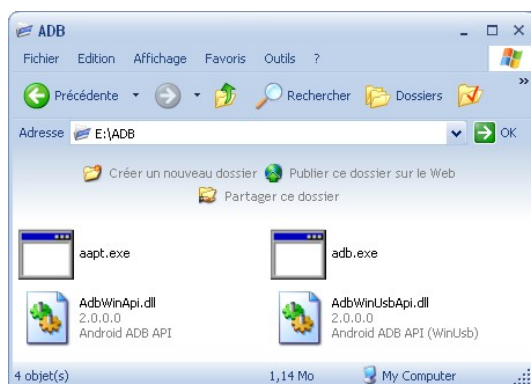
## Installation et conseils

✿ **Windows:** Installer en toute simplicité les outils Android adb, aapt  
ADB fait partie des outils du SDK Android (Software Development Kit).  
Pour installer le SDK Android, rien de plus simple, extraire le .zip dans un répertoire de votre choix. Pour faciliter l'utilisation d'ADB, je vous conseille de créer un répertoire à la racine de votre disque dur. Dans ce dossier, copiez les fichiers suiv-

ants, que vous trouverez dans le dossier **android-sdk/platform-tools** :

- adb.exe
- aapt.exe
- AdbWinApi.dll
- AdbWinUsbApi.dll

Ce sera votre répertoire de travail, dans lequel vous mettrez et récupérerez les fichiers en transit entre votre ordinateur et votre Smartphone. La plus simple encore est d'ajouter (pas remplacer, ni écraser) "android-sdk/platform-tools" au PATH variable d'environnement, afin de ne pas devoir spéci-





fier le chemin d'accès d'un outil du SDK chaque fois que nous en utiliserons un.

Pour ce faire, copiez le chemin de stockage du dossier platform-tools dans le SDK Android (E:\android-sdk-windows\platform-tools). Ensuite menu démarrer ou sur le bureau, Poste de Travail et faites un clic avec le bouton droit Propriétés et allez à l'onglet Avancé>Variable d'environnement. Recherchez alors la variable Path au sein du menu déroulant et, une fois trouvée, cliquez sur Modifier. À la fin de la valeur, rajoutez le séparateur ; (point virgule) puis collez le chemin d'accès de android-sdk/platform-tools soit E:\android-sdk-windows\platform-tools que vous avez copié auparavant. Terminez l'opération en cliquant sur OK afin d'enregistrer les modifications effectuées.

Exemple l'ancien Path contient E:\Windows, il sera

❁ **Linux:** Installer en toute simplicité les outils Android adb, aapt, apktool Vous serez content d'apprendre qu'un développeur du site XDA, Tahl, a écrit un script permettant d'installer en toute simplicité ADB.

Ce script, qui se nomme andadb, va vous permettre:

- de contrôler si votre configuration possède tous les prérequis (dépendances)
- de télécharger et d'installer les SDK et NDK d'Android (Native Development Kit)
- d'installer les commandes adb et fastboot

Pour les installer sur Ubuntu, Linux Mint et dérivées, il faut dans un premier temps installer le package git-core :

```
sudo apt-get install git-core
```

Récupérez le script depuis le site du projet :

```
git clone git://gist.github.com/1658401.git gist-1658401
```

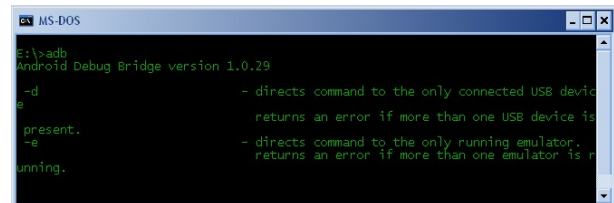
Lui donnez les droits d'exécution :

```
chmod +x andadb.sh
```

```
./andadb.sh #Exécuter le script
```

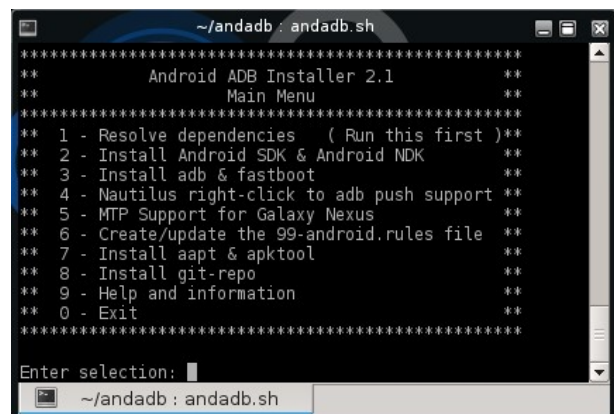
E:\Windows; E:\android-sdk-windows\platform-tools

Notez : Pour rendre effectifs les changements, vous n'avez pas besoin de redémarrer l'ordinateur. Vous pouvez dès à présent tester l'efficacité de la manœuvre effectuée, ouvrez l'invite de commande et tapez par exemple : adb.



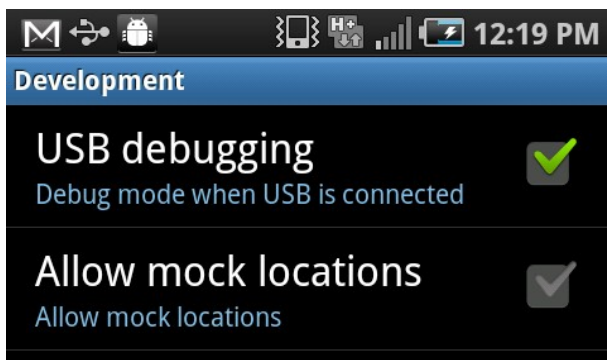
Cette technique que j'utilise permet de simplifier au maximum les commandes, on n'a plus besoin de taper les chemins des fichiers qui peuvent être très longs.

- d'installer une extension nautilus pour déployer des fichiers depuis le gestionnaire de fichiers
- de créer les règles nécessaires pour que vos équipements sous Android soit reconnues rapidement
- d'installer des outils complémentaires comme git-repo (nécessaire pour récupérer les sources du projet Cyanogen), aapt et apktool (des outils de reverse engineering)
- d'installer le support MTP pour le Galaxy Nexus



Un menu qui ressemble à l'image ci-dessus devrait apparaître. A partir de là vous devriez vous en sortir car le script est assez explicite. Le premier step, téléchargement et installation des dépendances prend à lui seul 30 minutes (bien sur cela dépend de votre ligne ADSL et des capacités de votre ordinateur).

## Configuration du Smartphone



### Liste des commandes ADB

Pour les autres systèmes (Mac, Linux) il vous suffira de remplacer `adb` par `./adb` dans toutes les commandes.

Voici une liste partiel des commandes ADB pour modifier son appareil ou effectuer des actions diverses:

```
adb help
```

Avoir la liste des commandes disponibles via l'ADB, cette dernière est très utile lorsque nous oublions les paramètres à passer aux commandes.

```
adb logcat
```

Visualiser le log spécifique à l'appareil ou à l'émulateur.

```
adb devices
```

Afficher la liste des émulateurs (lancés via le SDK Android) et des appareils branchés en USB à l'ordinateur et pouvant être administrés avec l'ADB.

```
adb kill-server
```

Cette commande a pour but d'éteindre le serveur ADB. Une fois cette commande tapée, il vous faudra relancer la commande **adb devices** pour remettre en ligne le serveur adb.

```
adb start-server
```

S'assurer qu'un serveur est toujours en cours d'exécution.

```
adb wait-for-device
```

Bloquer toute exécution jusqu'à ce qu'un appareil ou un émulateur soit connecté.

```
adb get-state
```

Connaître en temps souhaité l'état de l'appareil ou de l'émulateur connecté, hors-ligne (offline), bootloader, normal (device).

```
adb get-serialno
```

Tout d'abord, il faut activer la fonction Débogage USB sur votre Smartphone Android :

**2.0-2.3.x** Menu -> Paramètres -> Applications -> Développement -> Débogage USB (coché)

**3.0-4.1.x** Menu -> Paramètres -> Options développeur -> Débogage USB (coché)

**4.2-5.0.x** Menu -> Paramètres -> Système -> A propos du téléphone -> Numéro de build, Tapoter 3 à 5 fois sur Numéro de build pour activer les options développeurs! (procéder désormais comme l'étape '3.0-4.1.x' pour activer le Débogage USB)

En savoir plus: <http://bit.ly/1Ctw0kf>

Obtenir le numéro de série de l'appareil connecté.

```
adb -s "device" "commande"
```

Appliquer une commande spécifique sur un des terminaux listés.

```
adb pull "cheminaccès-fichier-appareil" "cheminaccès-placer-fichier-ordinateur"
```

Déplacer un fichier se trouvant sur un appareil ou un émulateur vers l'ordinateur directement sans passer par la gestion de la carte mémoire.

```
adb push "cheminaccès-placer-fichier-ordinateur" "cheminaccès-fichier-appareil"
```

Déplacer un fichier se trouvant sur l'ordinateur vers l'appareil ou l'émulateur connecté sans passer par la gestion de la carte mémoire.

```
adb -s "device" shell #adb -s emulator-5554 shell
```

Ouvrir une saisie de commande shell, comme si nous étions sur le terminal Linux. De cette façon nous pouvons utiliser les commandes shell basiques de Linux.

```
adb -s -d "device" shell "commande shell" #adb -s emulator-5554 shell ls
```

Cette commande permet d'utiliser les commandes shell basiques de Linux directement sur le terminal sélectionné, sans passer par l'application Terminal Emulator.

```
adb -s emulator-5554 install nom-application.apk
adb install "cheminaccès-fichier-apk"
```

Cette commande, en indiquant le chemin d'accès entier vers le fichier .apk à installer permet d'installer l'application souhaitée sur le Smartphone connecté à l'ordinateur ou bien sur l'émulateur directement. Cela peut être pratique pour tester son application avant déploiement final sur le Play Store.

```
adb uninstall nom-application.apk
```



Permet de supprimer de l'appareil ou de l'émulateur un package sans passer par l'interface graphique. Il est également possible d'utiliser le paramètre "-k" afin de garder la mémoire cache de l'application si à l'avenir vous souhaitez réutiliser cette dernière.

```
adb bugreport
```

Reporter toutes les données de l'appareil ou de l'émulateur pouvant avoir été incluses dans le bug

report.

```
adb reboot "mode souhaité"#adb reboot recovery
```

Rebooter (NDLR : redémarrez) votre appareil ou l'émulateur dans le mode souhaité. Cela est souvent utilisé pour flasher une nouvelle ROM ou un nouveau packlage .zip.

Pour ceux qui veulent plus de détail sur les commandes ADB, consulter <http://developer.android.com/tools/help/adb.html>

## Références

- 🐞 Londatiga.net
- 🐞 Developer.android.com

**M. M<sup>ed</sup> Ali Mortadha DAHMANI**

- ❖ Directeur & Fondateur, TechnoDom Laboratory
- ❖ Président, Freeways Association



# Nanotechnologies peut sauver des vies

Une équipe d'enquêteurs à l'Université Nord-Ouest a développé un dispositif promoteur basés sur des nanomatériaux biomédicaux qui pourrait être utilisé pour délivrer des médicaments de chimiothérapie localement à des sites où les tumeurs cancéreuses ont été enlevées chirurgicalement pour éviter tout risque de réapparition du tumeurs).

## Parylène Biostable

Dans leur étude, le Dr Ho et ses collègues ont intégré des millions de minuscules nanodiamants de drogue. Actuellement utilisé comme revêtement pour les implants, le parylène biostable est un matériau souple et polyvalent qui ressemble à une pellicule de plastique. Une quantité importante de médicament peut être chargée sur des grappes de nanodiamants, qui ont une grande surface.

## NanoDimands

Dans les expériences de contrôle, où le médicament était présent mais sans les nanodiamants, la quasi-totalité de la drogue a été libéré dans le 1 jour. En ajoutant les nanodiamants de drogue chargé à l'appareil, le médicament a été libéré tout au long d'un mois ce qui met en ?uvre la capacité de ces matériaux à prolonger les effets du médicament pour une guérison plus efficace.

En plus de leur grande surface, nanodiamonds ont beaucoup d'autres avantages que l'on peut utiliser dans la délivrance de médicaments. Ils peuvent être fonctionnalisés avec presque n'importe quel type de thérapie. Ils peuvent être facilement mis en suspension dans l'eau, ce qui est important pour des applications biomédicales. Les nanodiamonds, chacun étant de 4 à 6 nanomètres de diamètre, sont minimalement invasive pour les cellules, sont biocompatibles, et ne provoquent pas d'inflammation. En outre, nanodiamants sont relativement simples à réaliser en grandes quantités.

## Références

Ces travaux sont détaillés dans le document «Nanodiamond-Embedded Microfilm Devices for Localized Chemotherapeutic Elution." Un enquêteur de l'Université Shinshu à Nagano, au Japon, a également participé à cette étude.

**Marouen CHAOUACHI**

❖ Membre, Freeways ISI

## Notes

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.