Développement Android

Module 02 - Premiers pas

WARNING

Le contenu de cette présentation est basé sur la documentation anglophone officielle d'Android, diffusée sous licence *Creative Commons Attribution 2.5* :

developer.android.com

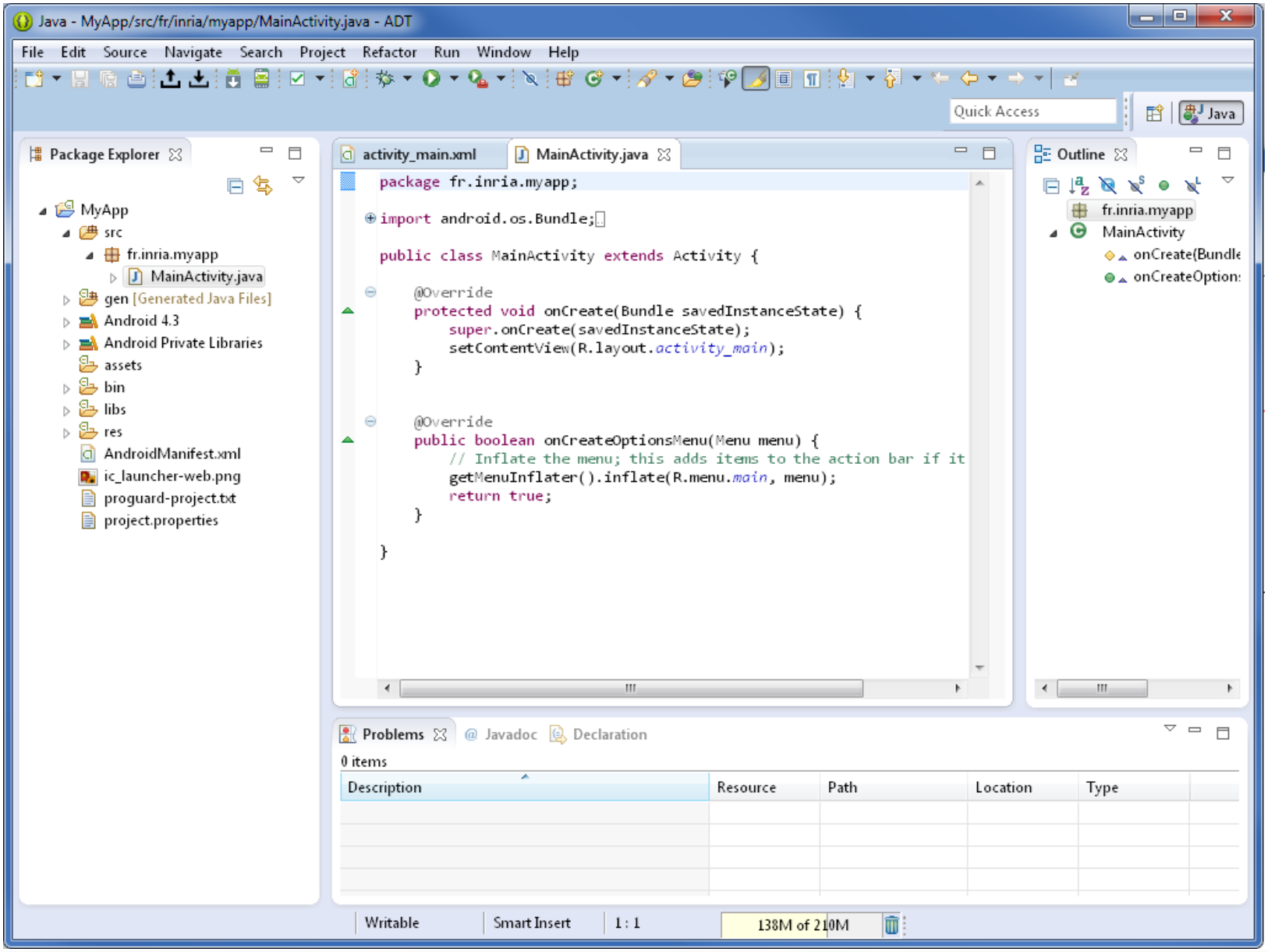
La plupart des schémas qui composent ce cours proviennent de cette documentation et sont, par conséquent, soumis à cette même licence.

http://creativecommons.org/licenses/by/2.5/

Android Developer Tools

- Eclipse + SDK + Emulateur + Outils
- Disponible sur developer.android.com
- Une simple archive à décompresser.

UN PREMIER PROJET



Un premier projet

- Par défaut, l'IDE a généré :
  - Une activité (MainActivity.java)
  - Un manifest (AndroidManifest.xml)
  - Un dossier de ressources (/res) diverses contenant notamment une vue, des icônes et divers fichiers (gestion des tailles, du texte, etc.).
- Lors de la compilation, /gen est généré.
- /gen/<package>/R.java permet de faire le lien entre les ressources XML et la logique de l'application.

Le manifest

- Décrit l'ensemble des composants qui constituent l'application.
- Décrit les permissions requises par l'application (accès internet, envoi de SMS, etc.), ainsi que les fonctionnalités matérielles et logicielles nécessaires (ex : appareil photo).
- Autres (SDK préféré, résolution supportées, etc.)

Le manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.inria.myapp"
    android:versionCode="1"
    android:versionName="1.0" >

<uses-sdk android:minSdkVersion="8" android:targetSdkVersion="18" />

<application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="fr.inria.myapp.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Le manifest

```
<uses-permission android:name="android.permission.CALL_EMERGENCY_NUMBERS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.WRITE_PROFILE" />
...
```

```
<supports-screens android:resizeable=["true"| "false"]
                  android:smallScreens=["true" | "false"]
                  android:normalScreens=["true" | "false"]
                  android:largeScreens=["true" | "false"]
                  android:xlargeScreens=["true" | "false"]
                  android:anyDensity=["true" | "false"]
                  android:requiresSmallestWidthDp="integer"
                  android:compatibleWidthLimitDp="integer"
                  android:largestWidthLimitDp="integer"/>
```

```
<uses-feature android:name="android.hardware.bluetooth" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.software.app_widgets" />
...
```

Les resources

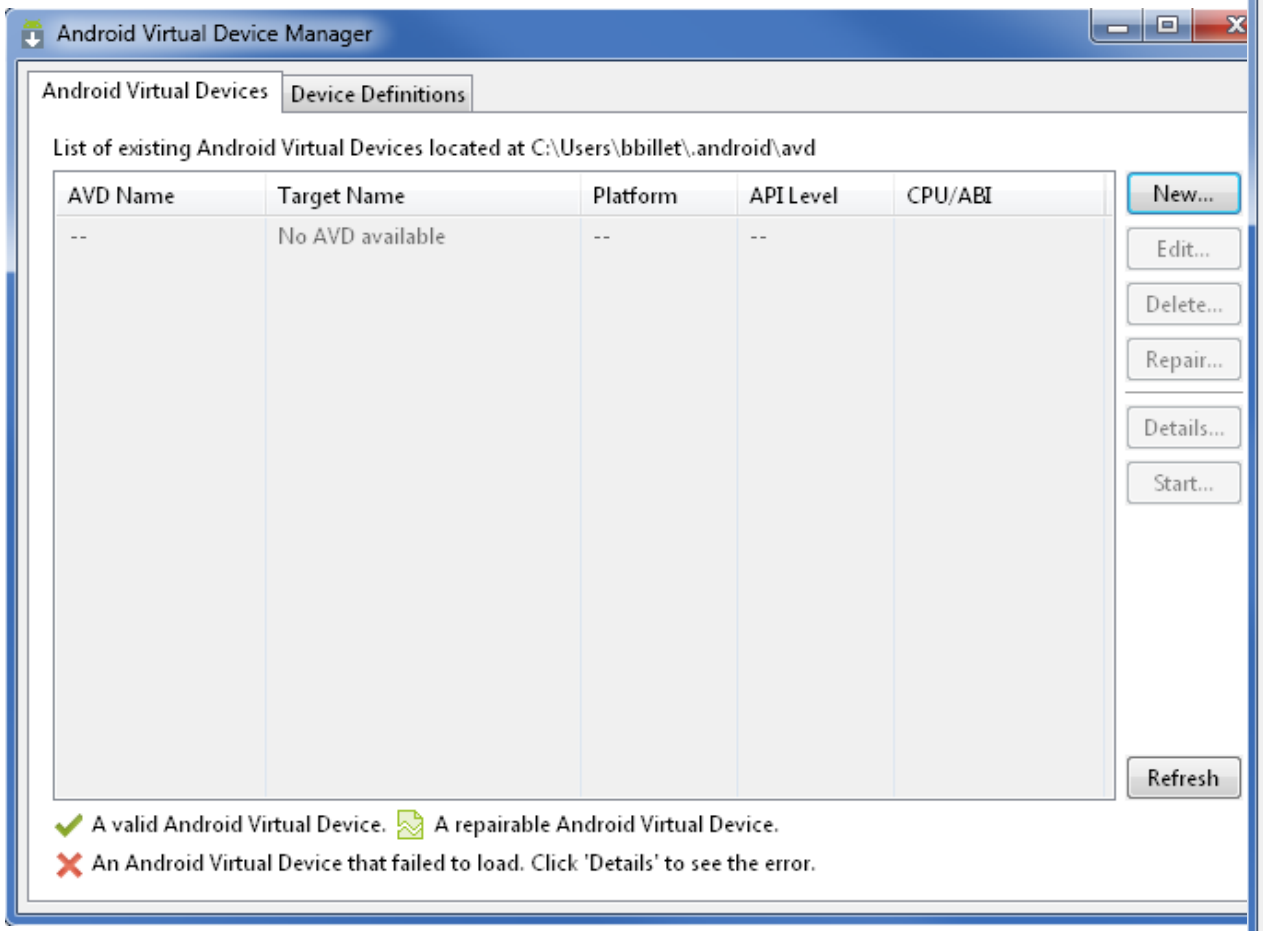
/res		
	/layout	Layout descriptions
	/menu	Menu descriptions
	/values	
		/strings.xml Default strings
		/dimens.xml Default dimensions
		/styles.xml Default styles
	/values-*	Specific values
	/drawable-*	Graphical resources (images)

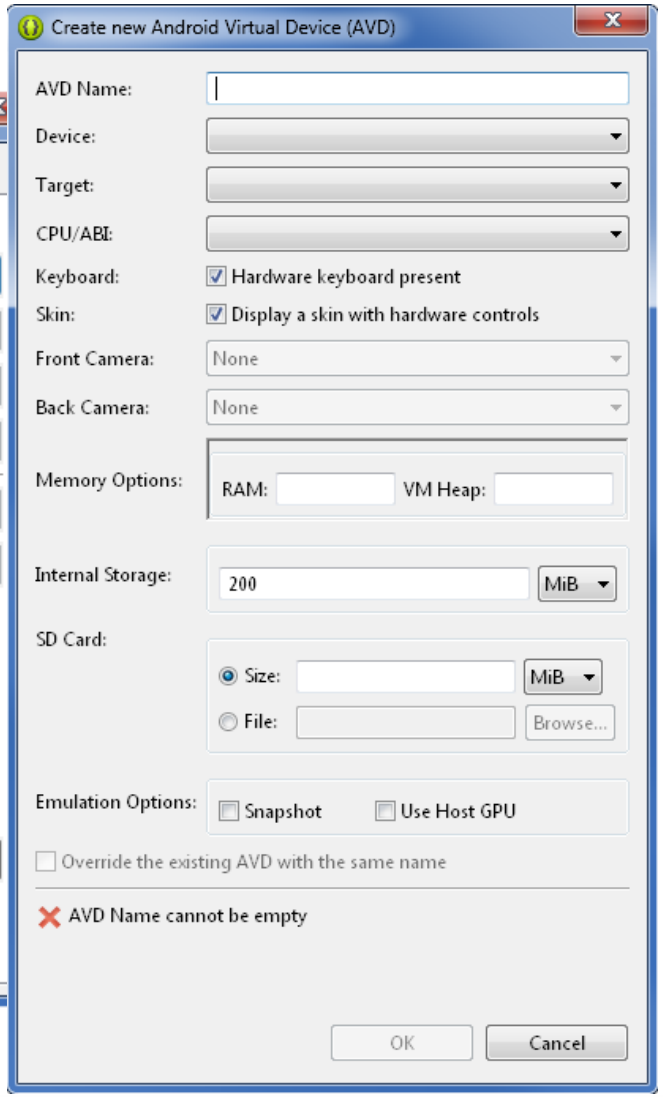
/res/values-en/strings.xml : traduction anglaise.
/res/drawable-ja : images optimisés pour le public japonais.
/res/values-v14/styles.xml : style spécifique à l'API 14
/res/drawable-hdpi : images optimisées pour écran haute densité (~240 dpi).

developer.android.com/guide/topics/resources/providing-resources.html

VIRTUAL DEVICE

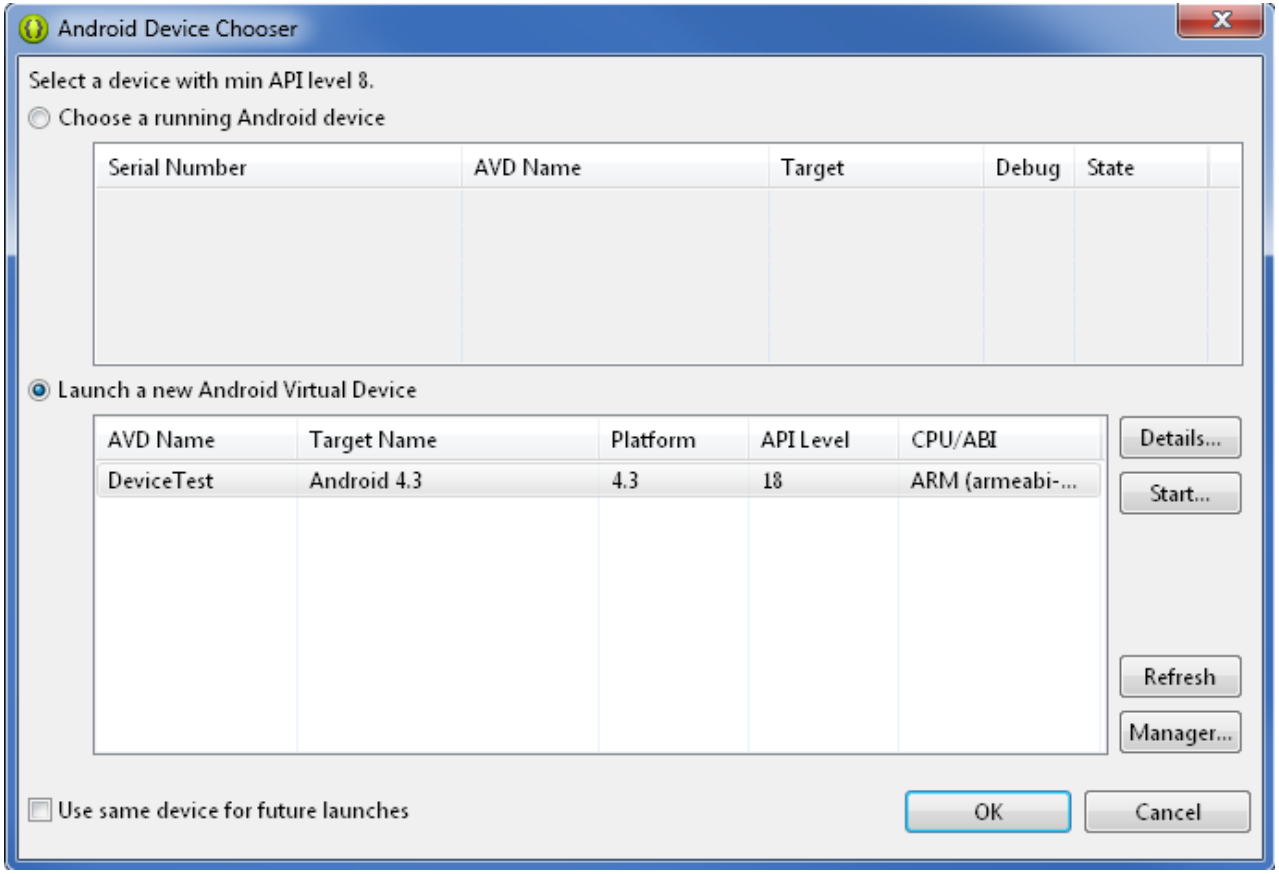
Window -> Virtual Device Manager





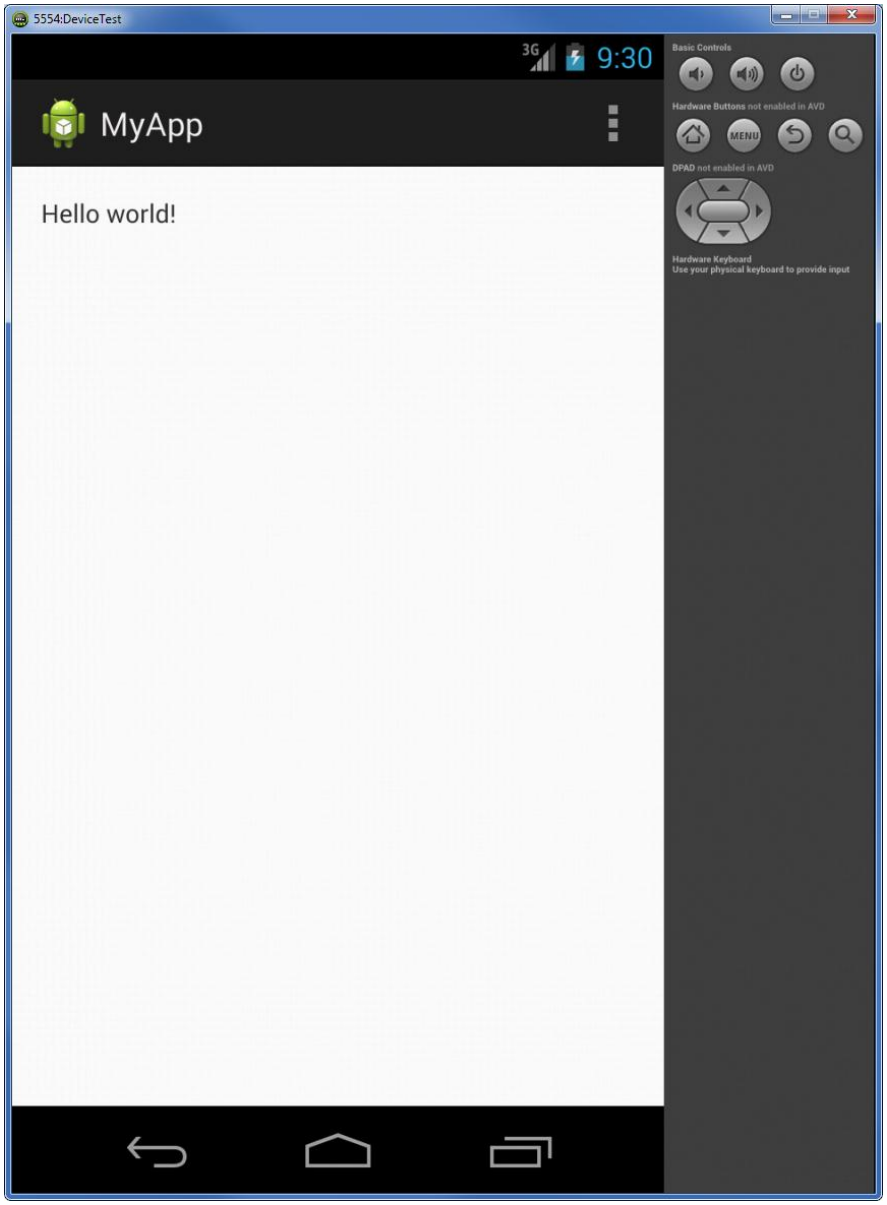
DEPLOIEMENT

- Run As -> Android Application
- Par défaut, utilise le virtual device



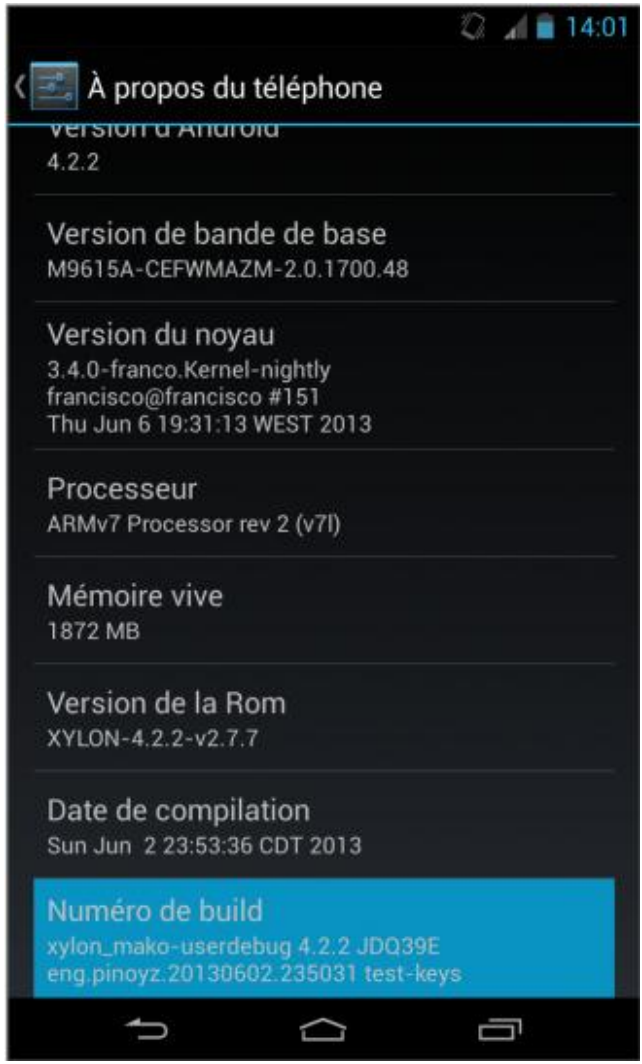
DEPLOIEMENT

L'émulateur ne doit être démarré qu'une seule fois. Il suffit ensuite de le laisser en arrière plan ; il sera automatiquement réutilisé pour les futures exécutions.

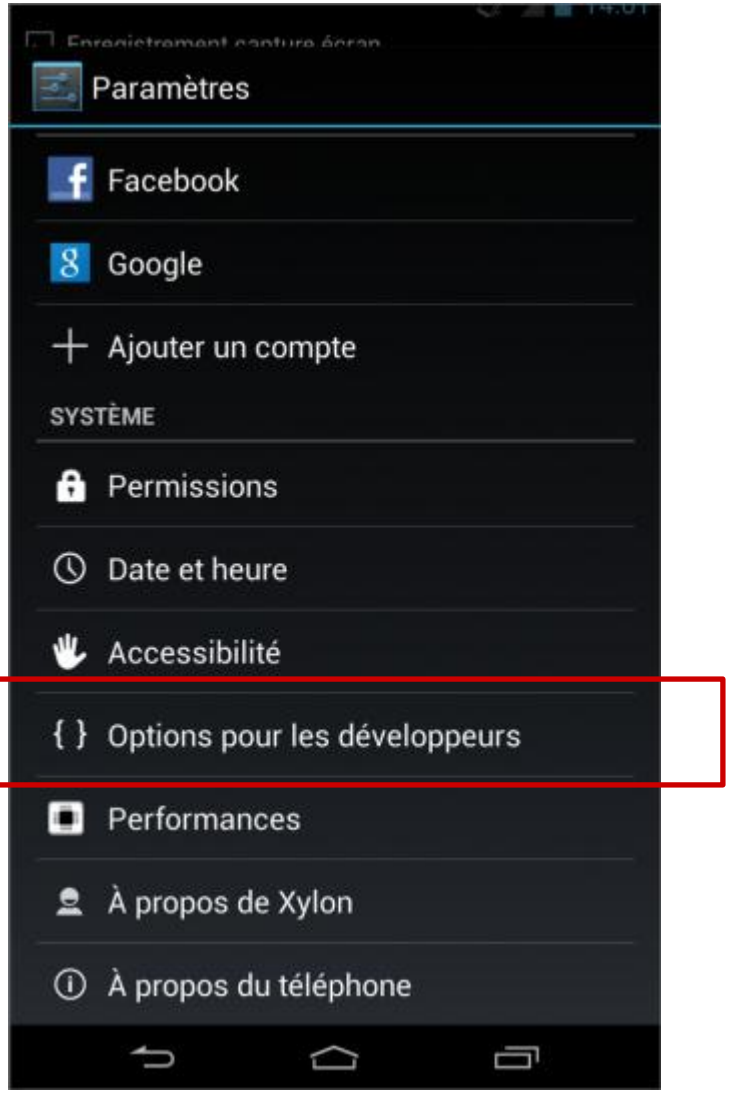


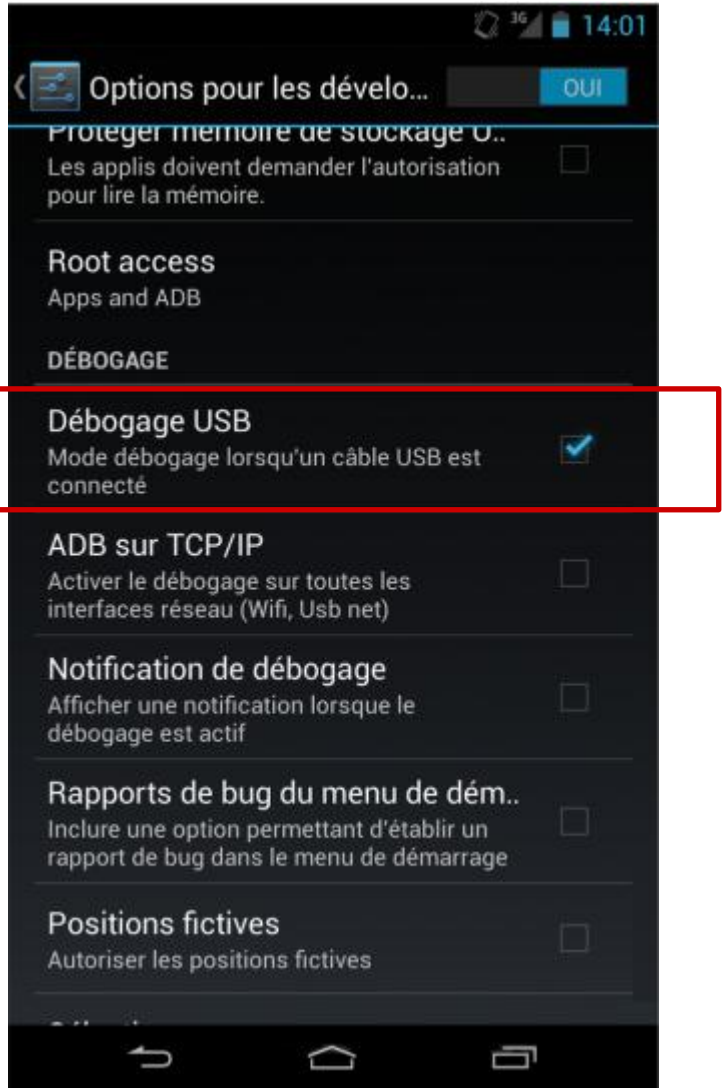
UTILISER UN VRAI DEVICE

- Passer le device en mode debug USB.
- Depuis Android 4.2, les options pour développeur doivent être réactivées en premier lieu :
  1. Aller dans le menu Paramètres/A propos.
  2. Taper sept fois sur le numéro de build.
  3. ~~Tourner huit fois sur vous-même en brandissant un crucifix, puis~~…

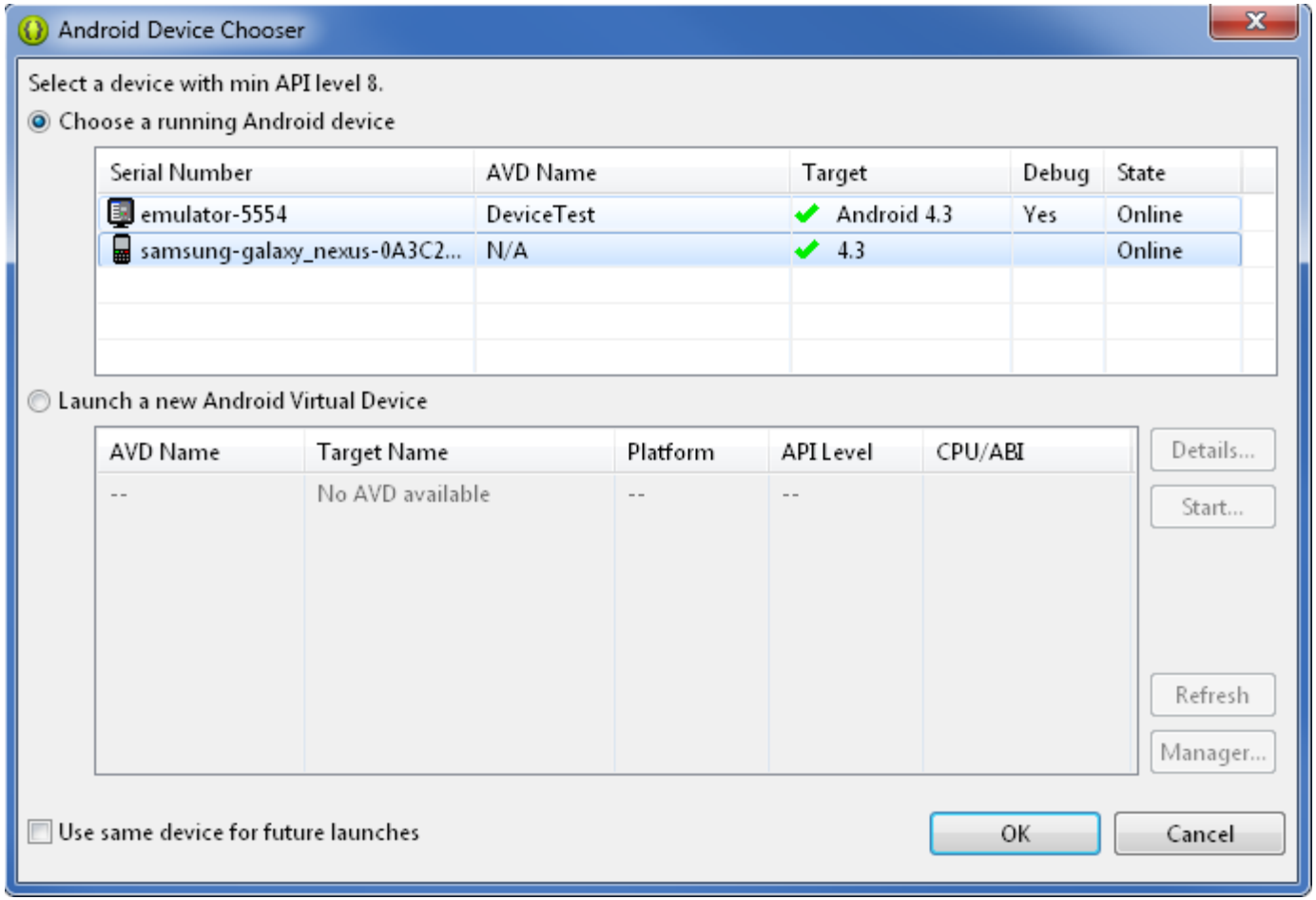


UTILISER UN VRAI DEVICE





UTILISER UN VRAI DEVICE



Android Debug Bridge

- ADB est un outil de déboggage en trois parties :
  - Sur la machine du développeur :
    - Un client, utilisable en ligne de commande.
    - Un serveur (port 5037), qui gère la communication avec les différents appareils.
  - Un démon qui tourne sur chaque appareil (ports 5554 – 55585).
- L'exécutable adb se situe dans <sdk>/platform-tools.

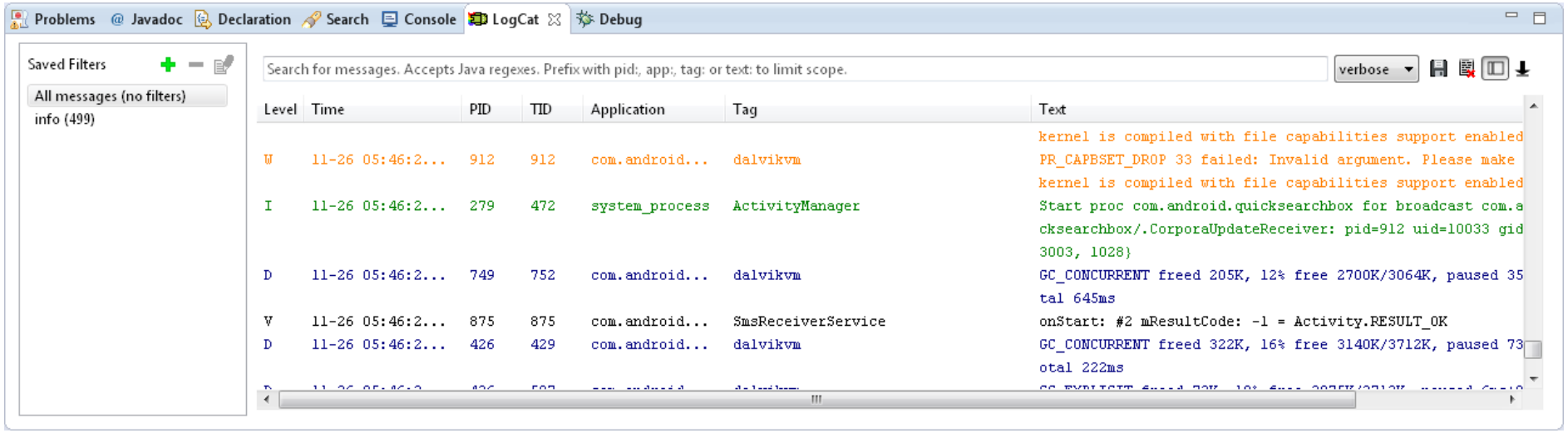
Android Debug Bridge

```
> adb devices
List of devices attached
0A3C23C719015019        device
emulator-5554           offline
emulator-5556           device


> adb -s emulator-5556 install helloWorld.apk


> adb -s emulator-5556 push foo.txt /sdcard/foo.txt
> adb -s emulator-5556 pull /sdcard/foo.txt foo.txt


> adb -s emulator-5554 shell
shell@maguro:/ $ sqlite3 /data/data/fr.inria.example/databases/somedb.db


> adb -s emulator-5556 shell screenrecord /sdcard/demo.mp4
```

LOGCAT

- Logcat permet d'accéder aux différents log enregistrés par les appareils.

```
> adb –s emulator-5554 logcat
11-26 05:46:27.260: D/AlertService(649): No fired or scheduled alerts
11-26 05:46:27.300: D/AlertService(649): Scheduling next alarm with AlarmScheduler.
sEventReminderReceived: null
11-26 05:46:27.330: D/AlarmScheduler(649): No events found starting within 1 week.
11-26 05:46:28.450: D/dalvikvm(279): GC_EXPLICIT freed 383K, 61% free 5334K/13492K, paused
5ms+14ms, total 155ms
...
```



Console de l'émulateur

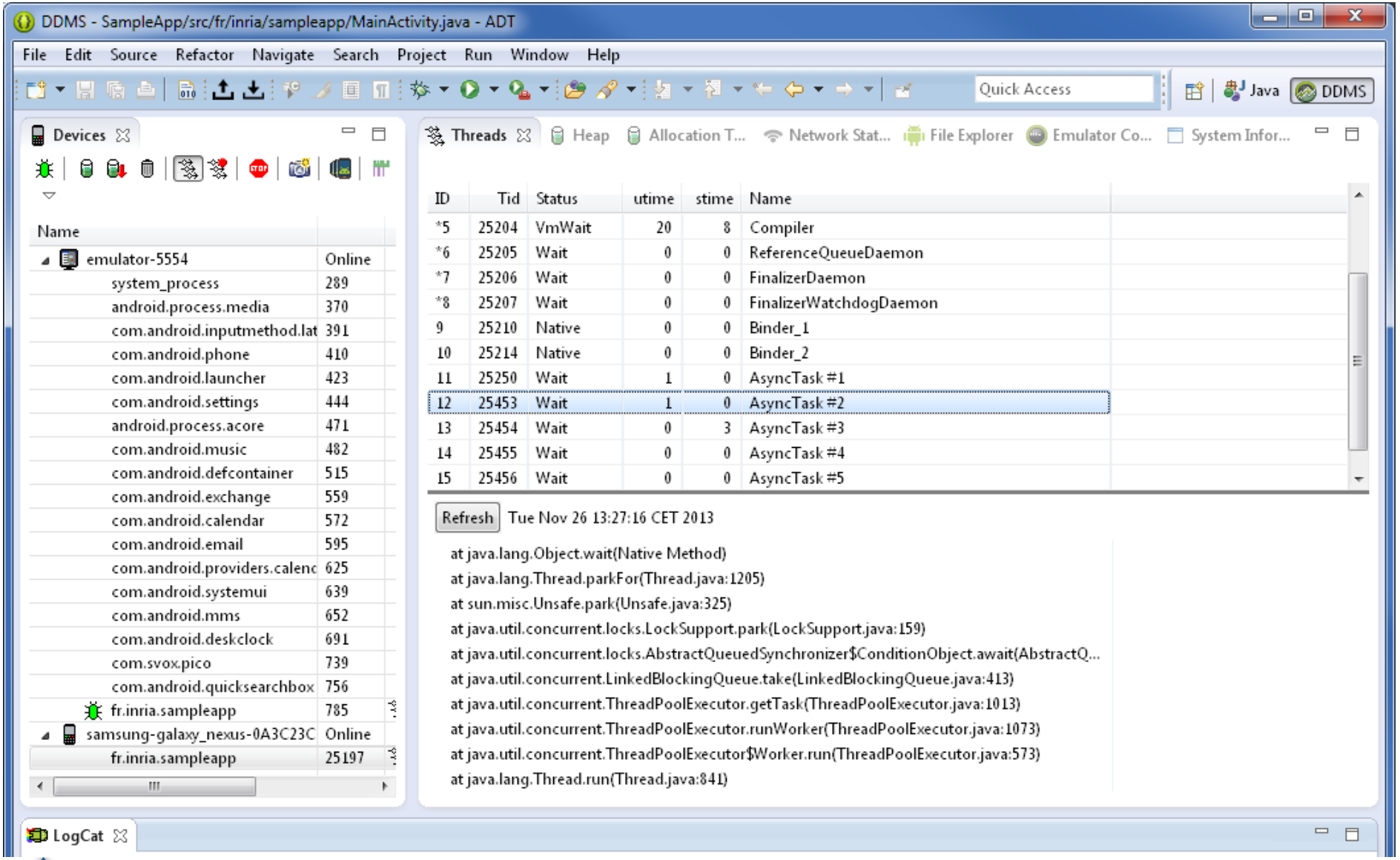
- L'émulateur peut être manipulé avec une console très polyvalente, directement accessible depuis telnet.

```
> telnet localhost 5554
Android Console: type 'help' for a list of commands
OK

power capacity 75
power status charging

gsm call 012041293123
sms send 12345 Will be home soon

# longitude latitude <altitude>
geo fix 2.0983248486704418 48.836726551297495 120

network delay gprs
network delay 50 100
network speed gprs
```

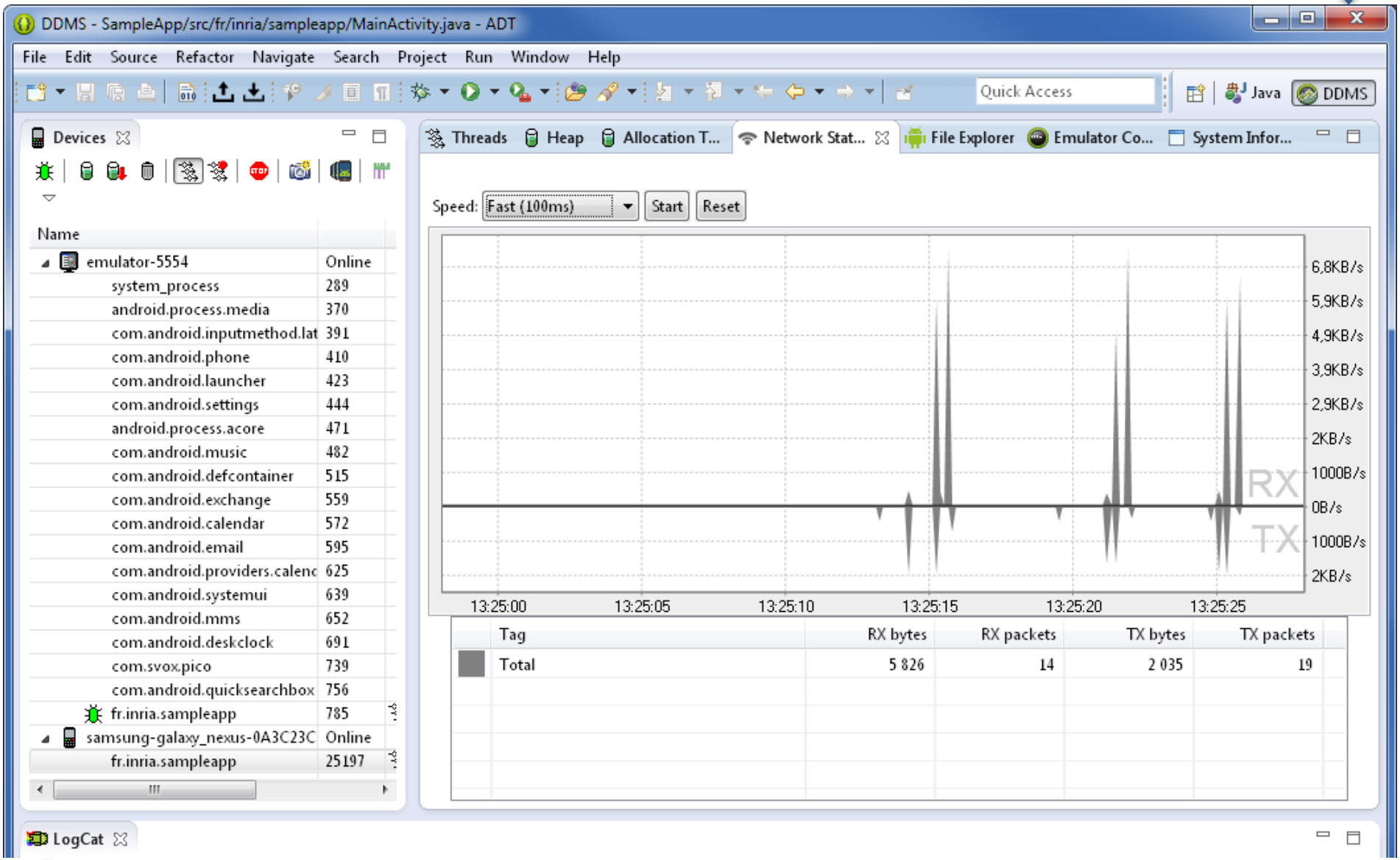
Dalvik Debug Monitor Server

- DDMS est un outil de déboggage (graphique), basé sur ADB, Logcat, etc.
  - Mesurer l'utilisation de la mémoire (allocation, heap).
  - Monitorer les threads et les processus.
  - Monitorer le trafic réseau.
  - Simuler le GPS, les sms/appels et les différents changements d'état dans l'émulateur.
  - Gérer les redirections de port dans l'émulateur.
  - Analyser les logs.
  - Envoyer/récupérer des fichiers.
  - …

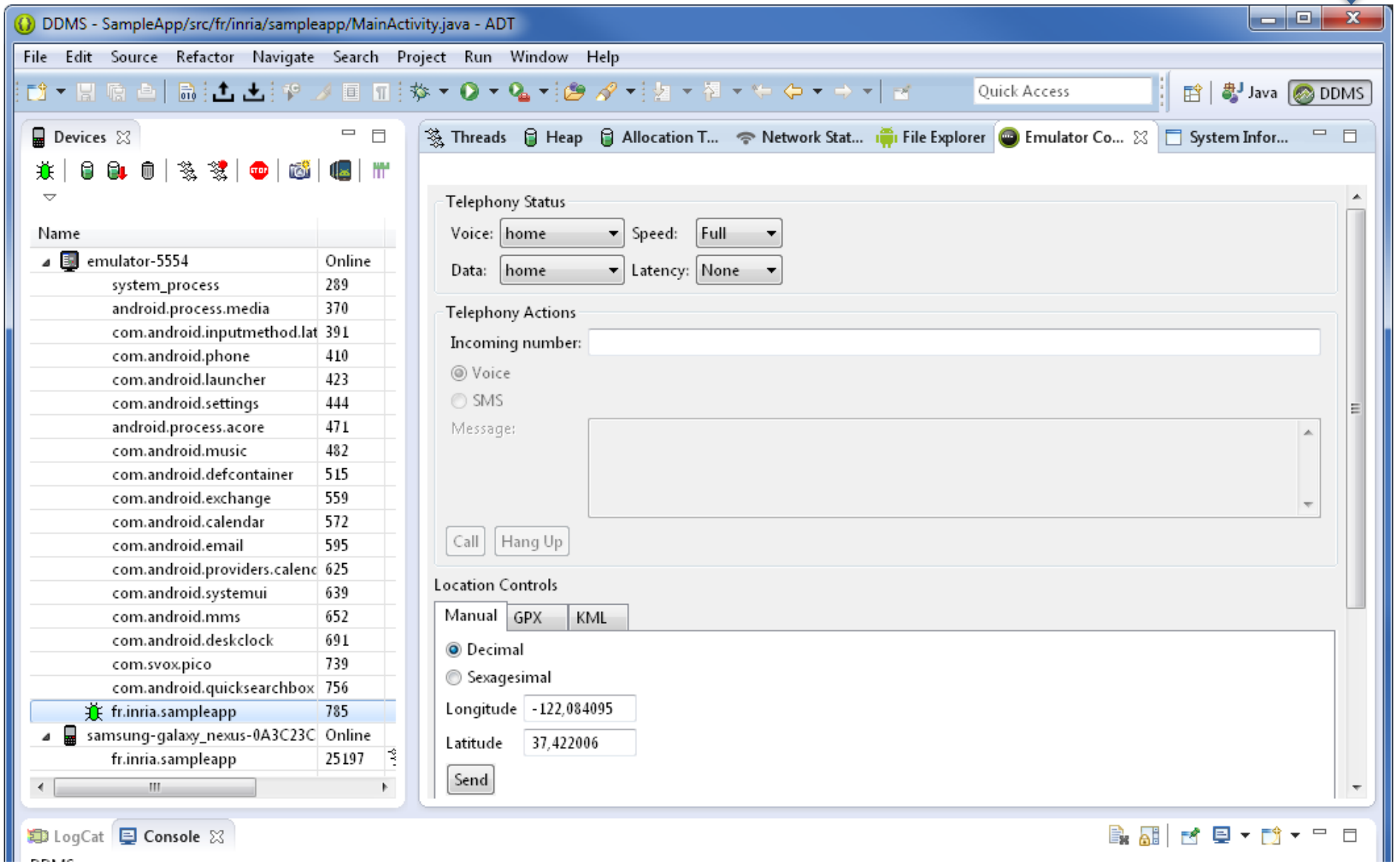
DALVIK DEBUG MONITOR SERVER



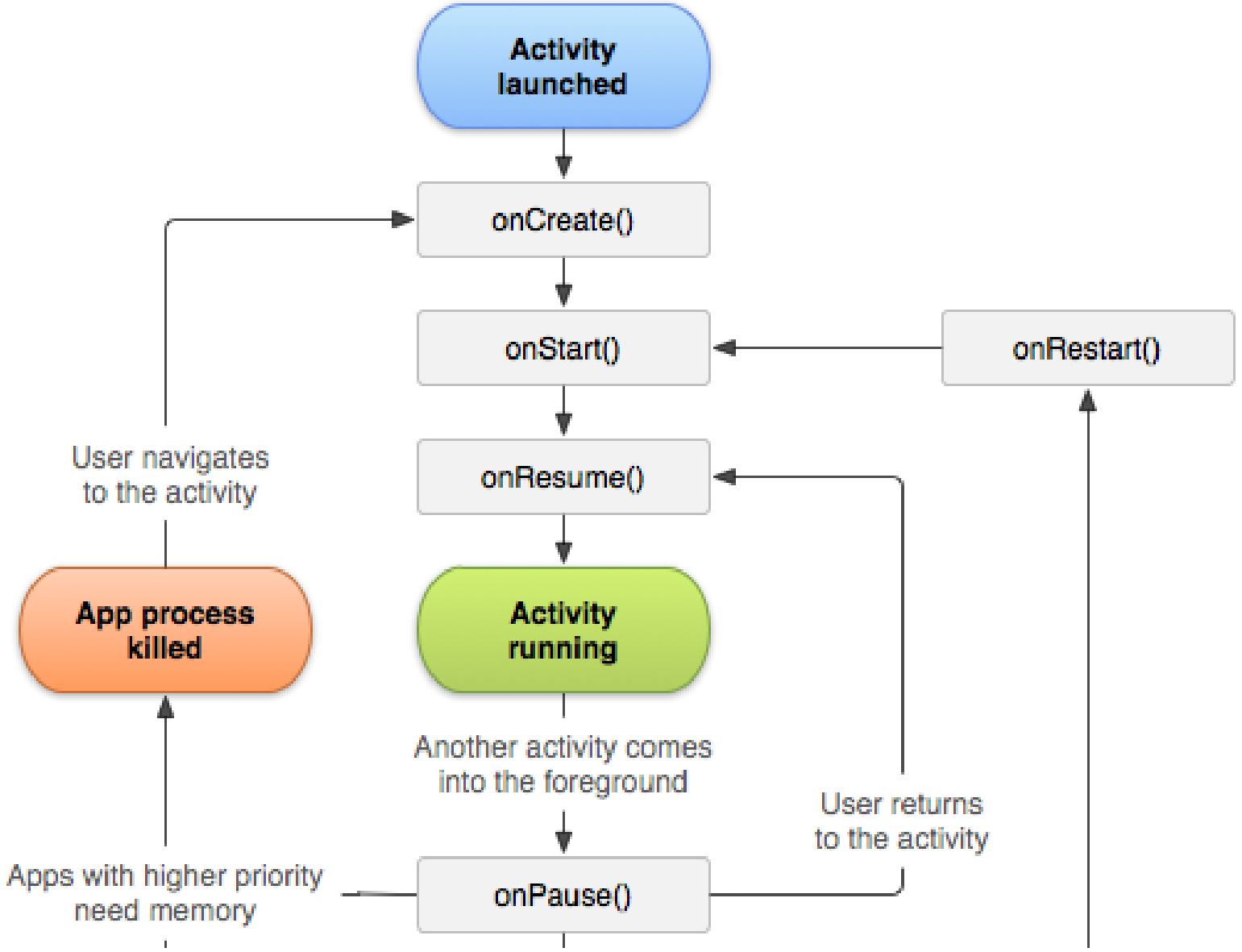
Dalvik Debug Monitor Server



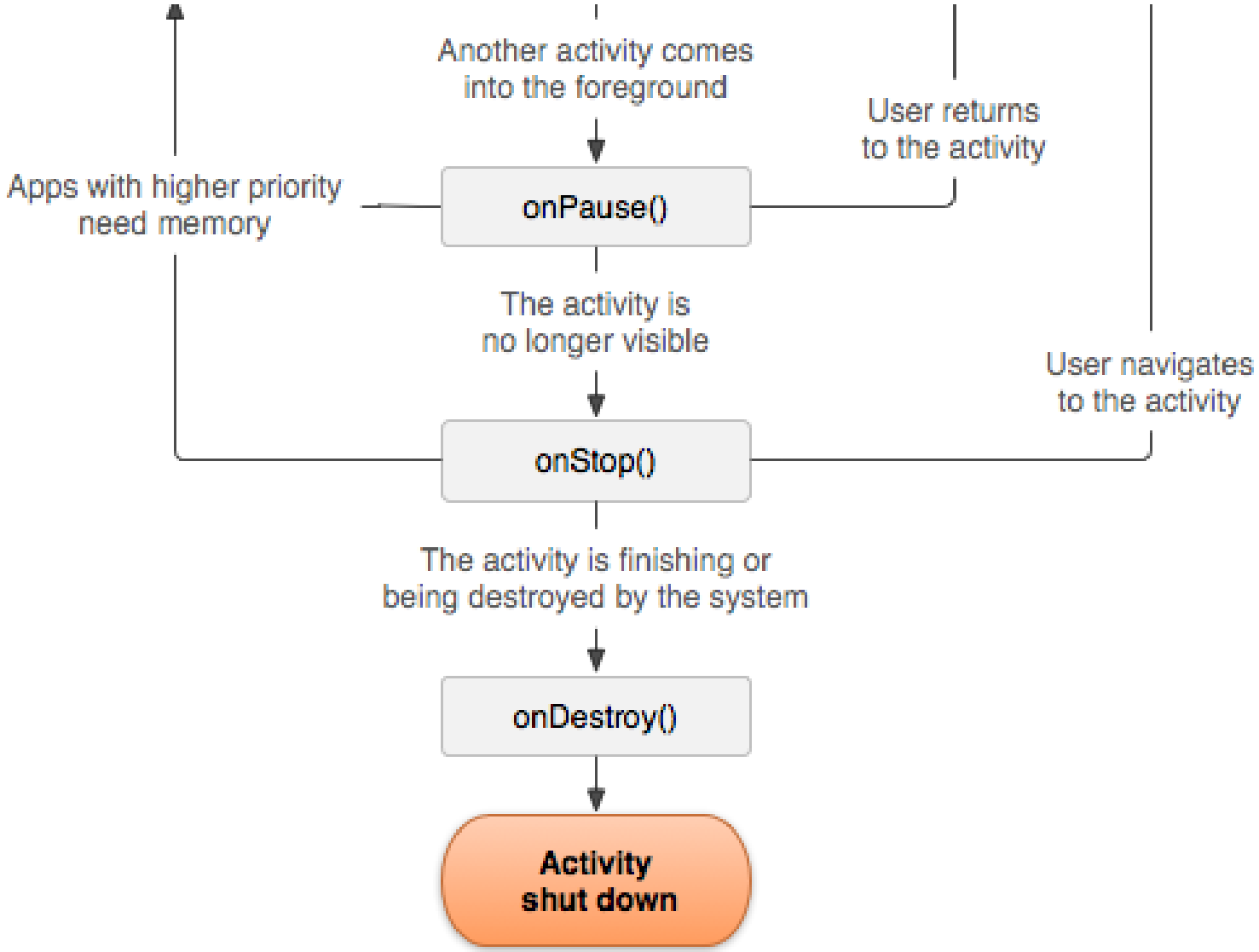
Dalvik Debug Monitor Server



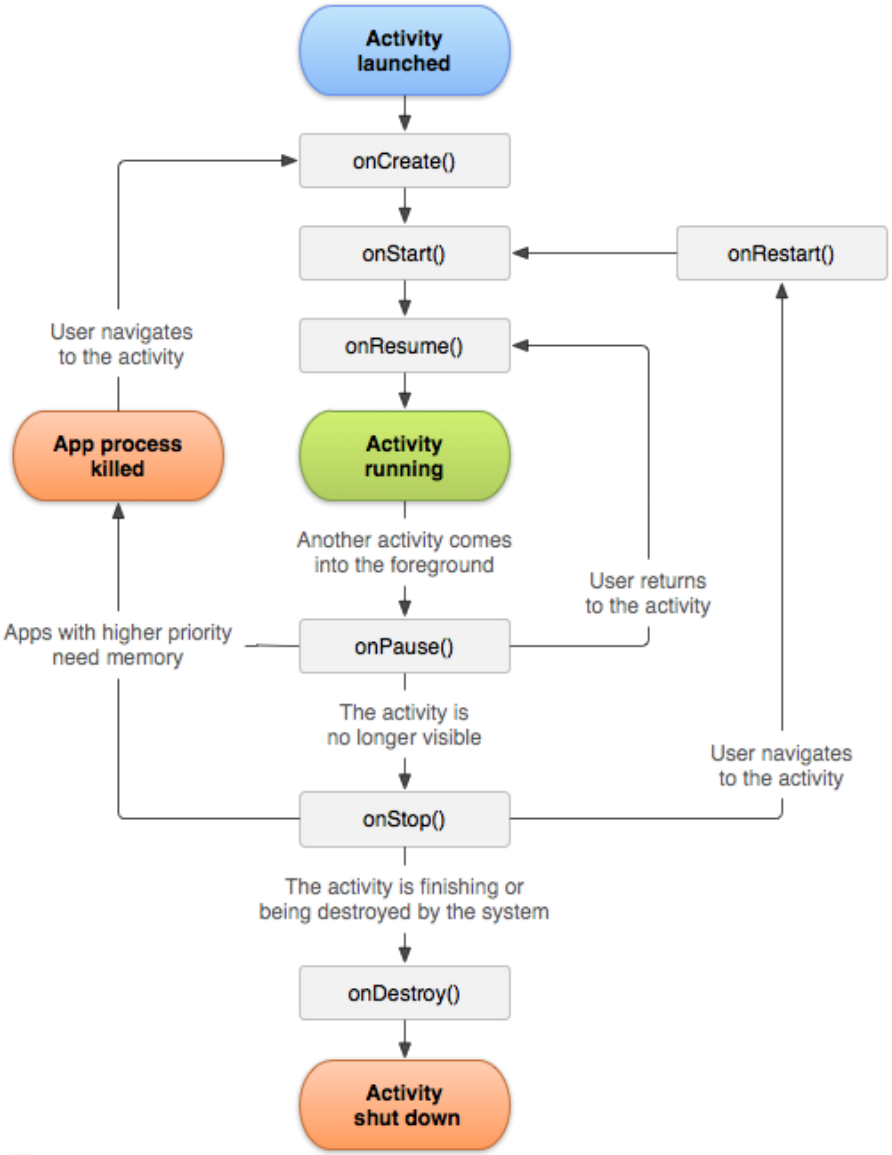
LE COMPOSANT ACTIVITY



LE COMPOSANT ACTIVITY



LE COMPOSANT ACTIVITY



- Durée de vie totale : de onCreate() à onDestroy()
- Arrière plan : de onStart() à onStop(), l'application s'exécute en fond et n'est plus visible par l'utilisateur.
- Premier plan : de onResume() à onPause(), l'application est au premier-plan et l'utilisateur intéragit avec celle-ci.

Intent Messaging

- L'échange de message entre les composants applicatifs se fait au travers de messages, appelés Intent.
- Un Intent est une structure de données qui décrit soit une opération à effectuer, soit un évènement.

ComponentName	La classe et le package d'un composant.
Action	Une action qui doit être, ou a été, réalisée.
Data	Une URI (tel:, http:, geo:, …) vers des données à traiter, et un type MIME.
Category	Une catégorie de composant.
Extras	Des paramètres (clé/valeur).
Flags	Options diverses.

Intent Messaging

- Android route les Intent d'un composant à un autre en exploitant les informations fournies dans le manifest.
- Un Intent qui définit un ComponentName est directement transmis (intent explicite).
- Les autres Intent (implicites) nécessitent de trouver les composants qui matchent les différents champs, avec trois issues possibles :
  - Un seul composant a été trouvé, le message est transmis.
  - Plusieurs composants ont été trouvé, une boîte de dialogue demande alors à l'utilisateur de faire un choix.
  - Aucun composant n'a été trouvé, une exception est levée.

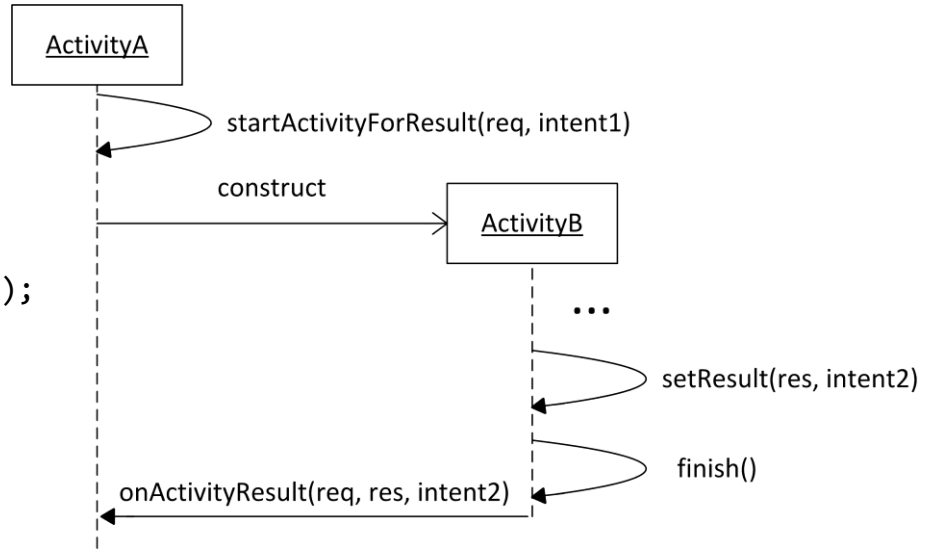
Un simple Intent explicite

ActivityA

```
Intent i = new Intent(this, ActivityB.class);
i.putExtra("something", 10);
startActivityForResult(i, 42);

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode == 42 && resultCode == Activity.RESULT_OK)
        int val = data.getExtras().getInt("somethingPlusOne");
}
```

ActivityB

```
Intent request = getIntent();
if(request.hasExtra("something"))
{
    int val = request.getIntExtra("something");
    Intent response = new Intent();
    response.putExtra("somethingPlusOne", val + 1);
    setResult(Activity.RESULT_OK, response);
}
else
    setResult(1); // error

finish();
```



Intent Resolution

- Seuls les champs action, data (URI, type MIME) et category sont utilisés pour le matching.
- Lorsqu'un composant est évalué, celui-ci doit passer trois tests pour être considéré comme un destinataire potentiel :
  - Action test.
  - Category test.
  - Data test.

Action test

Component

Intent

	No action	Set of actions A
No action	Failure	Success
Set of actions B	Failure	Success if A and B shares at least one element : $\lvert A \cap B \rvert \geq 1$

Category test

Component

Intent	No category	Set of categories A
No category	Success	Success
Set of categories B	Failure	Success if every category of A is defined in B : $A \subseteq B$

- Android considère que tous les Intents implicites passés à startActivity() appartiennent à une catégorie “default”…
- … il faut donc l’ajouter dans le manifest si l’on désire que l’activité puisse recevoir les intents…
- … sauf pour les activités qui définissent l’action “main” ou la catégorie “launcher”.

Les intent-filters

```
<intent-filter ... >
    <action android:name="com.example.project.SHOW_CURRENT" />
    <action android:name="com.example.project.SHOW_RECENT" />
    <action android:name="com.example.project.SHOW_PENDING" />
    ...
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    ...
    <data android:mimeType="video/mpeg" android:scheme="http" ... />
    <data android:mimeType="audio/mpeg" android:scheme="http" ... />
    ...
</intent-filter>
<intent-filter ... >
    ...
</intent-filter>
<intent-filter ... >
    ...
</intent-filter>
```

- Un Intent doit matcher au moins un filtre.
- Les étiquettes de catégorie/action prédéfinies incluent le nom de package complet :

Intent.CATEGORY_DEFAULT = android.intent.category.DEFAULT

Quelques Intent implicites

```
// exemple 1
Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("http://inria.fr"));
startActivity(i);

// exemple 2
Intent i = new Intent();
i.setAction(Intent.ACTION_CALL);
i.setData(Uri.parse("tel:0102030405"));
startActivity(i);

// exemple 3
Intent i = new Intent(Intent.ACTION_PICK, ContactsContract.Contacts.CONTENT_URI);
startActivityForResult(i, 42);

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 42 && resultCode == Activity.RESULT_OK)
    {
        Uri contactData = data.getData();
        ...
    }
}
```