

AndroidManifest

- Fichier xml qui contient :
- Permissions
- Liste des activités
- Services

Assets

Autres ressources (polices...)

Classes.dex

Code binaire de l'application, limité à 65 000 méthodes par fichier, mais il faut éviter de le dépasser, parce que le support au-delà n'est

pas assuré

META-INF

Informations autour de l'application :

- La classe à lancer
- Version du package
- Numéro de version

res

Ressources non compilées, mais standards

ex : Layout, Drawables,

resources.arsc

Ressources compilées, en format binaires xml

Les applications Android : Composition

• Les applications
Android

AndroidManifest

- Fichier xml qui contient :
- Permissions
- Liste des activités
- Services

Assets

Autres ressources (polices...)

Classes.dex

Code binaire de l'application, limité à 65 000 méthodes par fichier, mais il faut éviter de le dépasser, parce que le support au-delà n'est

pas assuré

META-INF

Informations autour de l'application :

- La classe à lancer
- Version du package
- Numéro de version

res

Ressources non compilées, mais standards

ex : Layout, Drawables,

resources.arsc

Ressources compilées, en format binaires xml

Les applications Android : Composition

• Les applications Android

- AndroidManifest.xml
- ▶ assets
- classes.dex
- ▶ lib/
- ▶ META-INF/
- ▶ res/
- resources.arsc

AndroidManifest

- Fichier xml qui contient :
- Permissions
- Liste des activités
- Services

Assets

Autres ressources (polices...)

Classes.dex

Code binaire de l'application, limité à 65 000 méthodes par fichier, mais il faut éviter de le dépasser, parce que le support au-delà n'est

pas assuré

META-INF

Informations autour de l'application :

- La classe à lancer
- Version du package
- Numéro de version

res

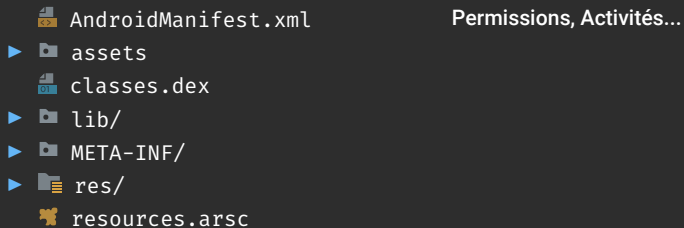
Ressources non compilées, mais standards
ex : Layout, Drawables,

resources.arsc

Ressources compilées, en format binaires xml

Les applications Android : Composition

• Les applications Android



AndroidManifest

- Fichier xml qui contient :
- Permissions
- Liste des activités
- Services

Assets

Autres ressources (polices...)

Classes.dex

Code binaire de l'application, limité à 65 000 méthodes par fichier, mais il faut éviter de le dépasser, parce que le support au-delà n'est

pas assuré

META-INF

Informations autour de l'application :

- La classe à lancer
- Version du package
- Numéro de version

res

Ressources non compilées, mais standards
ex : Layout, Drawables,

resources.arsc

Ressources compilées, en format binaires xml

Les applications Android : Composition

• Les applications Android

- AndroidManifest.xml
- ▶ assets
- classes.dex
- ▶ lib/
- ▶ META-INF/
- ▶ res/
- resources.arsc

Permissions, Activités...

Ressources non compilées, non standards

AndroidManifest

- Fichier xml qui contient :
- Permissions
- Liste des activités
- Services

Assets

Autres ressources (polices...)

Classes.dex

Code binaire de l'application, limité à 65 000 méthodes par fichier, mais il faut éviter de le dépasser, parce que le support au-delà n'est

pas assuré

META-INF

Informations autour de l'application :

- La classe à lancer
- Version du package
- Numéro de version

res

Ressources non compilées, mais standards
ex : Layout, Drawables,

resources.arsc

Ressources compilées, en format binaires xml

Les applications Android : Composition

• Les applications Android

- AndroidManifest.xml
- ▶ assets
- classes.dex
- ▶ lib/
- ▶ META-INF/
- ▶ res/
- resources.arsc

Permissions, Activités...

Ressources non compilées, non standards

Code binaire de l'application

AndroidManifest

- Fichier xml qui contient :
- Permissions
- Liste des activités
- Services

Assets

Autres ressources (polices...)

Classes.dex

Code binaire de l'application, limité à 65 000 méthodes par fichier, mais il faut éviter de le dépasser, parce que le support au-delà n'est

pas assuré

META-INF

Informations autour de l'application :

- La classe à lancer
- Version du package
- Numéro de version

res

Ressources non compilées, mais standards
ex : Layout, Drawables,

resources.arsc

Ressources compilées, en format binaires xml

Les applications Android : Composition

• Les applications Android

- AndroidManifest.xml
- ▶ assets
- classes.dex
- ▶ lib/
- ▶ META-INF/
- ▶ res/
- resources.arsc

Permissions, Activités...

Ressources non compilées, non standards

Code binaire de l'application

Librairies externes

AndroidManifest

- Fichier xml qui contient :
- Permissions
- Liste des activités
- Services

Assets

Autres ressources (polices...)

Classes.dex

Code binaire de l'application, limité à 65 000 méthodes par fichier, mais il faut éviter de le dépasser, parce que le support au-delà n'est

pas assuré

META-INF

Informations autour de l'application :

- La classe à lancer
- Version du package
- Numéro de version

res

Ressources non compilées, mais standards
ex : Layout, Drawables,

resources.arsc

Ressources compilées, en format binaires xml

Les applications Android : Composition

• Les applications Android

- AndroidManifest.xml
- ▶ assets
- classes.dex
- ▶ lib/
- ▶ META-INF/
- ▶ res/
- resources.arsc

Permissions, Activités...

Ressources non compilées, non standards

Code binaire de l'application

Librairies externes

Informations autour de l'application

AndroidManifest

- Fichier xml qui contient :
- Permissions
- Liste des activités
- Services

Assets

Autres ressources (polices...)

Classes.dex

Code binaire de l'application, limité à 65 000 méthodes par fichier, mais il faut éviter de le dépasser, parce que le support au-delà n'est

pas assuré

META-INF

Informations autour de l'application :

- La classe à lancer
- Version du package
- Numéro de version

res

Ressources non compilées, mais standards
ex : Layout, Drawables,

resources.arsc

Ressources compilées, en format binaires xml

Les applications Android : Composition

• Les applications Android

- AndroidManifest.xml
- ▶ assets
- classes.dex
- ▶ lib/
- ▶ META-INF/
- ▶ res/
- resources.arsc

Permissions, Activités...

Ressources non compilées, non standards

Code binaire de l'application

Librairies externes

Informations autour de l'application

Ressources standards, non compilées

AndroidManifest

- Fichier xml qui contient :
- Permissions
- Liste des activités
- Services

Assets

Autres ressources (polices...)

Classes.dex

Code binaire de l'application, limité à 65 000 méthodes par fichier, mais il faut éviter de le dépasser, parce que le support au-delà n'est

pas assuré

META-INF

Informations autour de l'application :

- La classe à lancer
- Version du package
- Numéro de version

res

Ressources non compilées, mais standards
ex : Layout, Drawables,

resources.arsc

Ressources compilées, en format binaires xml

Les applications Android : Composition

• Les applications Android

- AndroidManifest.xml
- ▶ assets
- classes.dex
- ▶ lib/
- ▶ META-INF/
- ▶ res/
- resources.arsc

Permissions, Activités...

Ressources non compilées, non standards

Code binaire de l'application

Librairies externes

Informations autour de l'application

Ressources standards, non compilées

Ressources compilées

Le code java est compilé en byte-code Java, puis rassemblé, compacté et optimisé via le programme dx en un exécutable dalvik, byte-code spécifique, prêt à être exploité par la machine virtuelle dalvik ou ART.

Ce fichier n'est pas compressé, ce qui permet à la machine virtuelle de mapper le code en mémoire et de le partager très facilement

Les applications Android : Compilation

• Les applications
Android

Le code java est compilé en byte-code Java, puis rassemblé, compacté et optimisé via le programme dx en un exécutable dalvik, byte-code spécifique, prêt à être exploité par la machine virtuelle dalvik ou ART.

Ce fichier n'est pas compressé, ce qui permet à la machine virtuelle de mapper le code en mémoire et de le partager très facilement

JIT :

- Just In time :

L'application est compilée en langage machine uniquement au moment où le code est nécessaire. Occupe moins d'espace mémoire

Plus lents

Disponible pour les versions inférieures à Kitkat

AOT :

- Ahead of time :

L'application est compilée en langage machine au moment de son installation. Bien plus rapide

Occupe plus d'espace mémoire

Disponible pour les versions supérieures à Kitkat

A partir de nougat, ART a été modifié pour utiliser à la fois un comportement JIT et AOT, permettant d'allier le meilleur des deux méthodes

Les applications Android : Compilation

• Les applications Android

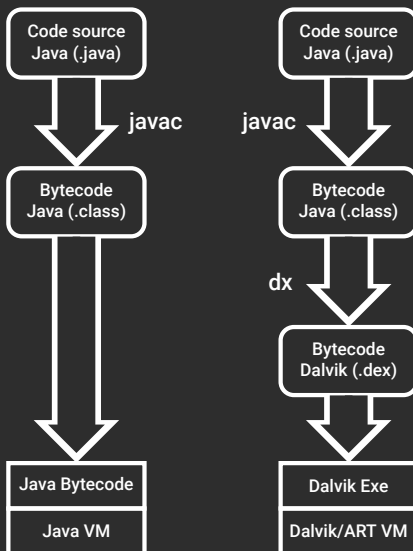


FIGURE – Compilation Java & Android

Le code java est compilé en byte-code Java, puis rassemblé, compacté et optimisé via le programme dx en un exécutable dalvik, byte-code spécifique, prêt à être exploité par la machine virtuelle dalvik ou ART.

Ce fichier n'est pas compressé, ce qui permet à la machine virtuelle de mapper le code en mémoire et de le partager très facilement

JIT :

- Just In time :

L'application est compilée en langage machine uniquement au moment où le code est nécessaire. Occupe moins d'espace mémoire

Plus lents

Disponible pour les versions inférieures à Kitkat

AOT :

- Ahead of time :

L'application est compilée en langage machine au moment de son installation. Bien plus rapide

Occupe plus d'espace mémoire

Disponible pour les versions supérieures à Kitkat

A partir de nougat, ART a été modifié pour utiliser à la fois un comportement JIT et AOT, permettant d'allier le meilleur des deux méthodes

Les applications Android : Compilation

• Les applications Android

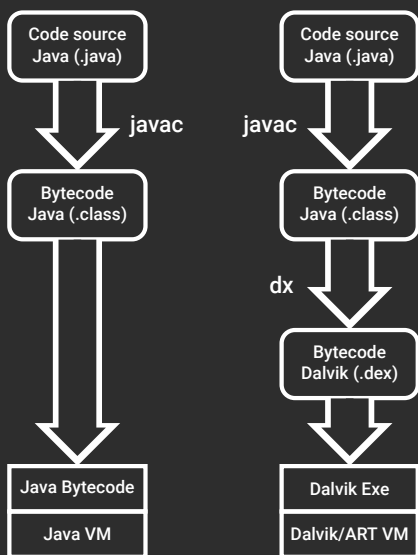


FIGURE – Compilation Java & Android

ART	vs	Dalvik

Le code java est compilé en byte-code Java, puis rassemblé, compacté et optimisé via le programme dx en un exécutable dalvik, byte-code spécifique, prêt à être exploité par la machine virtuelle dalvik ou ART.

Ce fichier n'est pas compressé, ce qui permet à la machine virtuelle de mapper le code en mémoire et de le partager très facilement

JIT :

- Just In time :

L'application est compilée en langage machine uniquement au moment où le code est nécessaire. Occupe moins d'espace mémoire

Plus lents

Disponible pour les versions inférieures à Kitkat

AOT :

- Ahead of time :

L'application est compilée en langage machine au moment de son installation. Bien plus rapide

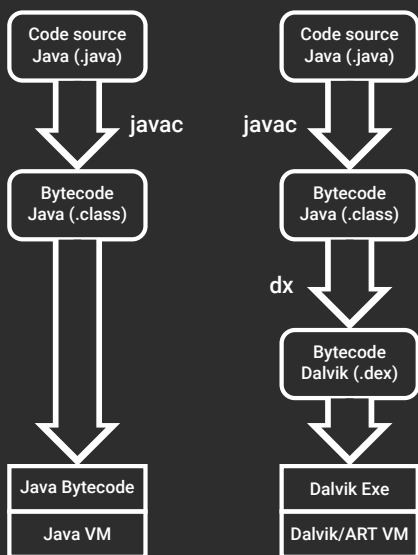
Occupe plus d'espace mémoire

Disponible pour les versions supérieures à Kitkat

A partir de nougat, ART a été modifié pour utiliser à la fois un comportement JIT et AOT, permettant d'allier le meilleur des deux méthodes

Les applications Android : Compilation

• Les applications Android



ART	vs	Dalvik
JIT ≤ 4.4		AOT ≥ 4.4

FIGURE – Compilation Java & Android

Le code java est compilé en byte-code Java, puis rassemblé, compacté et optimisé via le programme dx en un exécutable dalvik, byte-code spécifique, prêt à être exploité par la machine virtuelle dalvik ou ART.

Ce fichier n'est pas compressé, ce qui permet à la machine virtuelle de mapper le code en mémoire et de le partager très facilement

JIT :

- Just In time :

L'application est compilée en langage machine uniquement au moment où le code est nécessaire. Occupe moins d'espace mémoire

Plus lents

Disponible pour les versions inférieures à Kitkat

AOT :

- Ahead of time :

L'application est compilée en langage machine au moment de son installation. Bien plus rapide

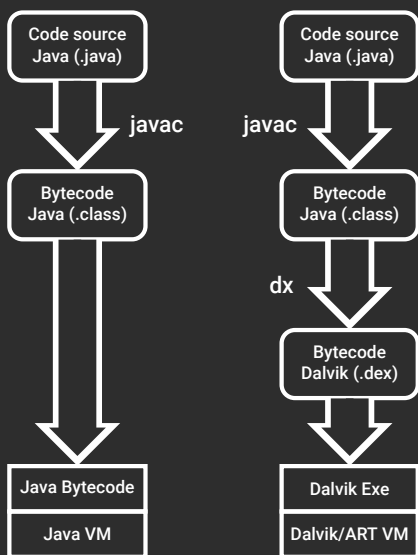
Occupe plus d'espace mémoire

Disponible pour les versions supérieures à Kitkat

A partir de nougat, ART a été modifié pour utiliser à la fois un comportement JIT et AOT, permettant d'allier le meilleur des deux méthodes

Les applications Android : Compilation

• Les applications Android



ART	vs	Dalvik
JIT		AOT
≤ 4.4		≥ 4.4
≥ 7.0 : AOT & JIT		

FIGURE – Compilation Java & Android