

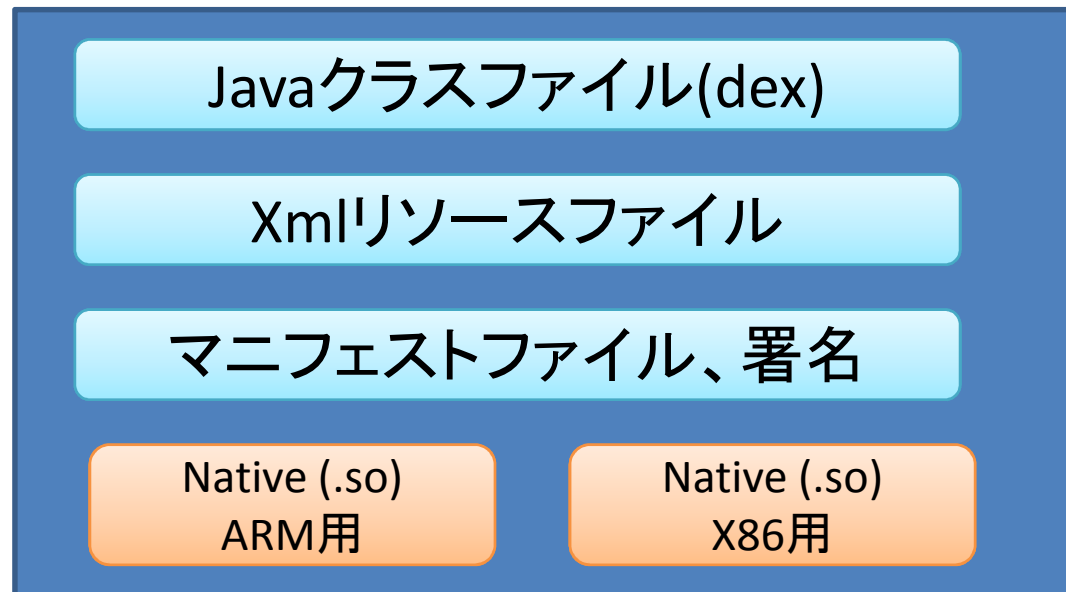
# Android Native Development Toolkit

日本Androidの会

木南英夫

# NDKとは

- Android Native Development Tools(NDK)
  - アプリケーションパッケージ内にCPUネイティブで実行可能なモジュールを組み込む仕組み  
アプリケーションパッケージ (apk)

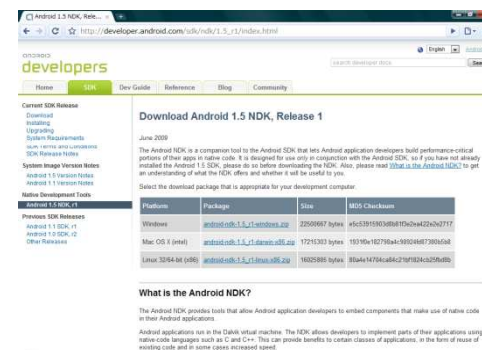


# NDKの用途と注意事項

- 用途
  - 既存のC言語のライブラリの活用
  - パフォーマンスの改善
    - NativeのCPUで実行して改善できる部分のみ
- 注意事項
  - Java言語で開発するアプリの一部
    - すべてがC言語で開発できるわけではない。
  - JNIのオーバーヘッドの考慮が必要
  - 実行される端末のCPUごとに作成する必要がある
  - プリミティブなライブラリだけが使用できる

# NDKに含まれるもの

- SDKのページから入手
  - <http://developer.android.com/sdk/>
- NDKに含まれるもの
  - C/C++言語のビルドツール
  - apkに配置するためのツール
  - ドキュメントとサンプルコード
  - C言語のヘッダーファイルと、ライブラリ



# 添付されているドキュメント

- INSTALL.TXT
  - インストール方法。
- OVERVIEW.TXT
  - 概要
- ANDROID-MK.TXT
  - C,C++用のMakefileの元
- APPLICATION-MK.TXT
  - Java用のアプリケーションのMakefileの元
- HOWTO.TXT NDK
  - 開発用のHow To
- - SYSTEM-ISSUES.TXT
  - 既知の問題
- STABLE-APIS.TXT
  - 使用できるライブラリの一覧

# 使用できるライブラリ

- libc (C library)
  - docs/system/libc/OVERVIEW.TXTに解説あり
- libm (math library)
- JNI interface headers
- libz (Zlib compression)
- liblog (Android logging)
- 最小限のC++
  - `<cstdint>` `<new>` `<utility>` `<stl_pair.h>`
- 将来的には、後方互換性を確保できたものから追加されていく予定。

# NDKが動く環境

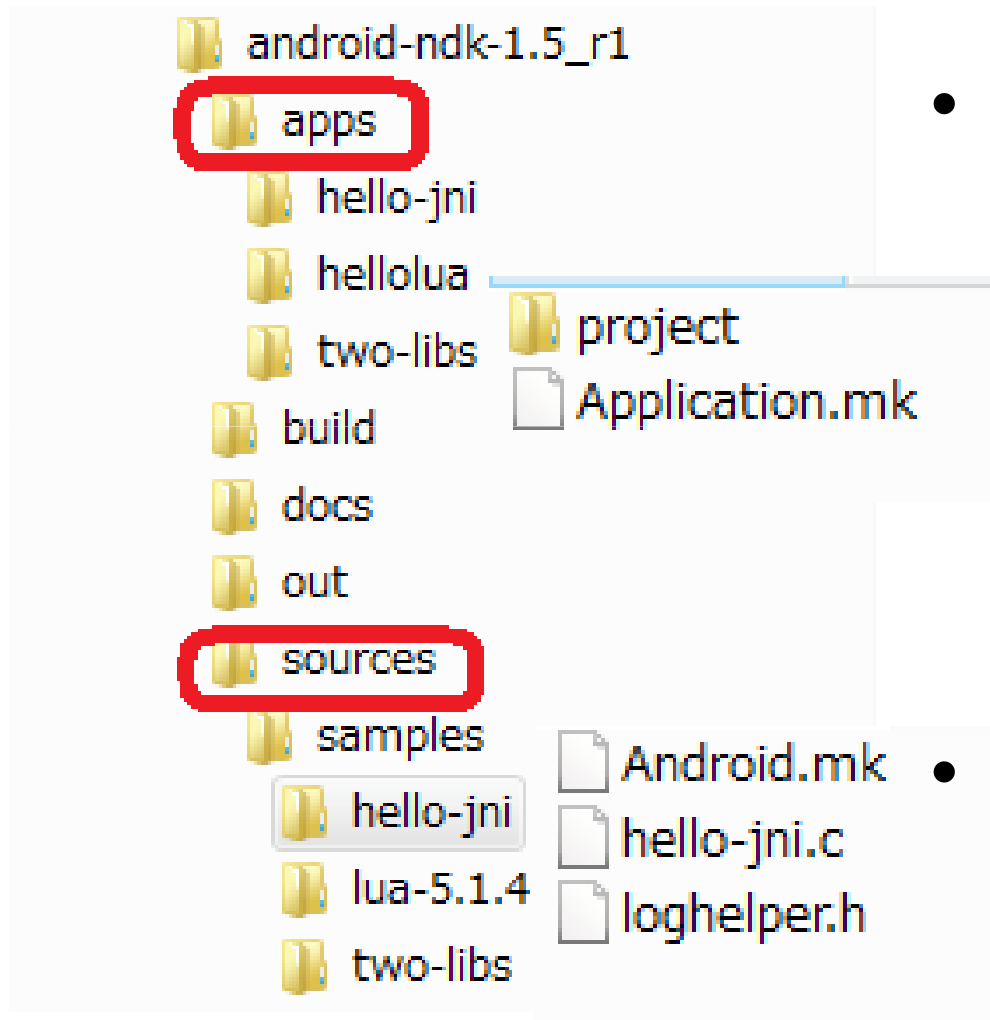
- Windows XP (32-bit) or Vista (32- or 64-bit)
  - Windowsの場合には、Cygwinのgmake, gccが必要
- Mac OS X 10.4.8 or later (x86 only)
- Linux (32- or 64-bit, tested on Linux Ubuntu Dapper Drake)

# インストール方法

- ダウンロードしたNDKを展開する
  - Windowsの場合、空白の入っていないディレクトリの使用がお勧め
- build/host-setup.shで、ツール類のセットアップを行う
  - Windowsの場合、sh build/host-setup.shで起動
  - out/host/config.mkが作成される



# 開発用のディレクトリ



- appsの下にJavaプロジェクトへの参照

– Application.mkの中でProjectを指定

- Sourceの下にC言語

– Android.mkの中でソースを指定

# Makeコマンドの実行

- $\{\text{ndk}\}$ のトップで、「make APP=<yourapp>」
  - Appsディレクトリの中の<yourapp>ディレクト参照
  - Application.mkのAPP\_MODULESが参照するモジュールをAPP\_PROJECT\_PATHに配置
    - apps/<yourapp>/project/libs/armeabi/lib<module1>.so
  - モジュール名は、Android.mkで定義されています。



# HOW TOに書かれていること

- コンパイルコマンドを表示：
  - `make APP=<yourapp> V=1`
- すべてのソースをリビルドする：
  - `make APP=<yourapp> -B`
- ソースをNDKディレクトリ以外におく：
  - `ln -s`でシンボリックリンクする
- 複数のモジュール間でインクルードする：
  - `-Isources/foo`、または、`$(LOCAL_PATH)/../foo`

# ロギングライブラリを使用する

- Android.mkに、ライブラリへのリンクを追加
  - LOCAL\_LDLIBS := -L\${SYSROOT}/usr/lib -llog
  - ドキュメントには、-L\${SYSROOT}/usr/libの記述はないが、必要。(Win, Macの場合)
- ソースコードでは、マクロを自分で定義して使用すると便利
  - #include <android/log.h>
  - #define LOGV(...)\_\_android\_log\_print(ANDROID\_LOG\_VERBOSE, LOG\_TAG, \_\_VA\_ARGS\_\_)