

# Android manifest ファイル解説

## ～超ざっくり編～

### 1. manifest ファイルについて

#### 1.1. manifest ファイルとは？

Android および Firefox OS のソースコードは、分散型バージョン管理システムである Git で管理された、数百ものソフトウェアプロジェクトの集合として公開されています。これらのダウンロード等、一括操作は `repo` というツールで行います。本稿では、この `repo` での操作対象ソフトウェアプロジェクトを定義するファイルである、manifest ファイルについて簡単に解説してみます。

サンプルとして、Firefox OS の manifest ファイルを一部抜粋して示します。

*Tbl. 1-1 Firefox OS の manifest ファイル (抜粋)*

	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;manifest&gt;</pre>
1	<pre>&lt;remote name="aosp" fetch="https://android.googlesource.com/" /&gt; &lt;remote name="b2g" fetch="git://github.com/mozilla-b2g/" /&gt; &lt;remote name="mozillaorg" fetch="https://git.mozilla.org/releases" /&gt;</pre>
2	<pre>&lt;default revision="refs/tags/android-4.0.4_r2.1" remote="aosp" sync-j="4" /&gt;</pre>
3	<pre>&lt;!-- Gonk specific things and forks --&gt; &lt;project path="build" name="platform_build" remote="b2g" revision="v1.2"&gt;   &lt;copyfile src="core/root.mk" dest="Makefile" /&gt; &lt;/project&gt; &lt;project path="gaia" name="gaia.git" remote="mozillaorg" revision="v1.2" /&gt; &lt;project path="gecko" name="gecko.git" remote="mozillaorg" revision="v1.2" /&gt; &lt;project path="gonk-misc" name="gonk-misc" remote="b2g" revision="v1.2" /&gt; &lt;!-- Stock Android things --&gt; &lt;project path="abi/cpp" name="platform/abi/cpp" /&gt; &lt;project path="bootable/recovery" name="platform/bootable/recovery" /&gt; &lt;project path="device/common" name="device/common" /&gt; &lt;project path="external/apriori" name="platform_external_apriori"   remote="b2g" revision="v1.2" /&gt; &lt;project path="external/bluetooth/bluez" name="platform/external/bluetooth/bluez" /&gt; &lt;project path="external/gtest" name="platform/external/gtest"   revision="8c212ebe53bb2baab3575f03069016f1fb11e449" /&gt; &lt;!-- Pandaboard specific things --&gt; &lt;project path="device/ti/panda" name="android-device-panda"   remote="b2g" revision="v1.2" /&gt;</pre>
	<pre>&lt;/manifest&gt;</pre>

manifest ファイルの形式は、冒頭の記述から明らかなように xml です。

<manifest>をルート要素として、ソフトウェアプロジェクトのソースコードをダウンロードするのに必要な情報や、ダウンロード時にローカルで行う最低限の操作を記述します。

以下、このサンプルをつかって、最低限知っておきたい manifest ファイルの書式などを説明します。

## 1.2. <remote> 要素

サンプル中の 1 で示された箇所では<remote>要素を用いて、各プロジェクトのソースファイルダウンロード元(一般的には Git の公開レポジトリサーバ) の情報が記載されています。

<remote>要素のもつ主な属性は以下のとおりです。

*Tbl. 1-2 <remote>要素の属性*

属性	意味	備考
name	定義した取得元を表す一意の文字列を指定	
fetch	その remote にある Git レポジトリの URL の共通する部分を指定	

サンプルの最初の例では、" https://android.googlesource.com/"を aosp と名付けています。

## 1.3. <default> 要素

サンプル中の 2 で示された箇所では<default>要素を用いて、ソースコードの取得元(<remote>)やバージョン情報など、共通する情報を一括で定義しています。

<default>要素がもつ主な属性は以下の通りです。

*Tbl. 1-3 <default>要素の属性*

属性	意味	備考
remote	ソースコード取得元を示す<remote>要素の name を指定	
revision	取得するバージョンをブランチ名、タグ名、コミット ID などで指定	
sync-j	repo sync コマンドの実行時に割り当てるジョブ数。	optional

サンプルではソースコード取得元として aosp が、ソースコードのバージョンとしては android-4.0.4\_r2.1 というタグが指定されています。また、各レポジトリからソースコードを取得するコマンド repo sync にジョブを4つ割り当てるような指定もありますが、これが実際に有効かどうかは実行するマシンの性能に左右されます。一般的には、/proc/cpuinfo に出てくる processer 数程度(最近の CPU なら物理コア数の2倍)くらいが最もパフォーマンスが良さそうです。(desync-j の指定はあまり一般的ではないと思われます。必要ならば、コマンドラインで repo sync -j4 など指定すべきです。)

## 1.4. <project> 要素

サンプル中の 3 で示された箇所では<project>要素を用いて、取得するプロジェクトのソースコードおよびその配置先を定義しています。

<project>要素がもつ主な属性は以下の通りです。

Tbl. 1-4 <project>要素の属性

属性	意味	備考
remote	ソースコード取得元を示す<remote>要素の name を指定 指定の無い場合は<default>要素の remote となる	
name	remote での git レポジトリ配置先を指定(ただし、レポジトリ名の最後の ".git" は省略可)。即ち、実際の git レポジトリダウンロード元 URL は以下 のようになる。 \${remote_fetch}/\${project_name}.git	
path	ダウンロードしたソースコードの配置先パス (repo sync コマンドを実行したディレクトリからの相対パス) を指定。 省略された場合は、name が配置先となる	optional
revision	取得するバージョンをブランチ名、タグ名、コミット ID などで指定 指定の無い場合は<default>要素の revision となる	

サンプルの<project path="gaia" name="gaia.git" remote="mozillaorg" revision="v1.2" />では、  
<https://git.mozilla.org/releases/gaia.git> を<作業ディレクトリ>/gaia に配置し、ブランチ v1.2 をチェックアウト  
 することを意味する。

## 1.5. manifest ファイルの書式についてのドキュメント

manifest ファイルの書式については、repo init を実行した際に生成されるファイル  
 .repo/repo/docs/manifest-format.txt  
 が正式なドキュメントとなります。

実際に Firefox OS や Android などのソースコードを取得した際、ここで説明しなかったような属性・要素が使用  
 されている場合もありますので、必要な場合はこちらを参照してください。