[Android Repo的manifest XML文件格式](http://blog.csdn.net/hansel/article/details/9798189)

2013-08-06 21:38 19272人阅读 [评论](http://blog.csdn.net/hansel/article/details/9798189#comments)(5) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/hansel/article/details/9798189#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

Android（23） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

[**Android**](http://lib.csdn.net/base/android)使用repo来管理多个**[Git](http://lib.csdn.net/base/git" \o "Git知识库" \t "_blank)**项目。它需要一个manifest  XML文件来指示这些git项目的属性。

repo manifest XML可以包含下面的元素。

* manifest: 最顶层的XML元素。
* remote元素: 设置远程git服务器的属性，包括下面的属性
  + name: 远程git服务器的名字，直接用于git fetch, git remote 等操作
  + alias: 远程git服务器的别名，如果指定了，则会覆盖name的设定。在一个manifest中，  
    name不能重名，但alias可以重名。
  + fetch: 所有projects的git URL 前缀
  + review: 指定Gerrit的服务器名，用于repo upload操作。如果没有指定，则repo upload没有效果。

Example:

<remote fetch="ssh://git.example.com" name="test"review="gerrit.example.com"/>

* default元素：设定所有projects的默认属性值，如果在project元素里没有指定一个属性，则使用default元素的属性值。
  + remote: 之前定义的某一个remote元素中name属性值，用于指定使用哪一个远程git服务器。
  + revision: git分支的名字，例如master或者refs/heads/master
  + sync\_j: 在repo sync中默认并行的数目。
  + sync\_c: 如果设置为true，则只同步指定的分支(revision 属性指定)，而不是所有的ref内容。
  + sync\_s: 如果设置为true，则会同步git的子项目

Example:

<default remote="main" revision="platform/main"/>

* manifest-server元素: 只能有一个该元素。它的url属性用于指定manifest服务的URL，通常是一个XML RPC 服务。  
  它要支持一下RPC方法：
  + GetApprovedManifest(branch, target): 返回一个manifest用于指示所有projects的分支和编译目标。  
        target参数来自环境变量TARGET\_PRODUCT和TARGET\_BUILD\_VARIANT，组成$TARGET\_PRODUCT-$TARGET\_BUILD\_VARIANT。
  + GetManifest(tag):  返回指定tag的manifest

* project元素：指定一个需要clone的git仓库。
  + name: 唯一的名字标识project，同时也用于生成git仓库的URL。格式如下：  
          ${remote\_fetch}/${project\_name}.git
  + path: 可选的路径。指定git clone出来的代码存放在本地的子目录。如果没有指定，则以name作为子目录名。
  + remote: 指定之前在某个remote元素中的name。
  + revision: 指定需要获取的git提交点，可以是master, refs/heads/master, tag或者SHA-1值。
  + groups: 列出project所属的组，以空格或者逗号分隔多个组名。所有的project都自动属于"all"组。每一个project自动属于  
    name:'name' 和path:'path'组。例如<project name="monkeys" path="barrel-of"/>，它自动属于default, name:monkeys, and path:barrel-of组。如果一个project属于notdefault组，则，repo sync时不会下载。
  + sync\_c: 如果设置为true，则只同步指定的分支(revision 属性指定)，而不是所有的ref内容。
  + sync\_s: 如果设置为true，则会同步git的子项目。
  + upstream: 在哪个git分支可以找到一个SHA1。用于同步revision锁定的manifest(-c 模式)。该模式可以避免同步整个ref空间。
  + annotation: 可以有多个annotation，格式为name-value pair。在repo forall 命令中这些值会导入到环境变量中。
  + remove-project: 从内部的manifest表中删除指定的project。经常用于本地的manifest文件，用户可以替换一个project的定义。

Example:

             <project groups="aosp" path="device/driver/armv7" revision="600aab270ce712b62b268055737cabcded59bf04"/>

* include: 通过name属性可以引入另外一个manifest文件(路径相对与manifest repository's root)。

本地manifest

本地的manifest文件存放在$(TOP\_DIR)/.repo/local\_manifest/\*.xml。$TOP\_DIR/.repo/local\_manifests/local\_manifest.xml  
如果存在，会被最先装入。然后是以字母顺序的$TOP\_DIR/.repo/local\_manifests/\*.xml文件。这些manifest会在repo sync  
之前被处理。以便repo下载和管理额外的project。

Example：

  $ ls .repo/local\_manifests

  local\_manifest.xml

  another\_local\_manifest.xml

  $ cat.repo/local\_manifests/local\_manifest.xml

  <?xml version="1.0"encoding="UTF-8"?>

  <manifest>

    <project path="manifest"

             name="tools/manifest"/>

    <projectpath="platform-manifest"

             name="platform/manifest"/>

  </manifest>