

android系统 service单体测试手順

Last edited by **caoquanli** 1 month ago

Android系统 Service单体测试手順

第一步：

在frameworks/base/Android.mk 添加如下两行代码：

```
LOCAL_MODULE := framework
LOCAL_EMMA_INSTRUMENT := true
EMMA_INSTRUMENT := true
```

第二步：

在build/core/tasks/check_boot_jars/package_whitelist.txt 配置中添加：

```
com\google\vr\platform.*
net\suntec\merbok.*
com\google\protobuf.*
org\.objectweb\.asm.*
org\.jacoco\.agent\.rt.*
org\.jacoco\.agent.*
org\.jacoco\.core\.analysis.*
org\.jacoco\.core\.instr\.Instrumenter.*
org\.jacoco\.core\.instr.*
org\.jacoco\.core\.tools.*
org\.jacoco\.core\.runtime.*
org\.jacoco\.core.*
com\.vladium\.emma\.rt.*
com\.vladium\.emma.*
```

第三步：

make -j16编译整个系统, 编译完成后会在:out/target/common/obj/JAVA_LIBRARIES/framework_intermediates目录下面生成一个coverage.em的文件,并烧写整个系统。

第四步： 编写测试APK：

测试APK的Android.mk 如下：

```
LOCAL_PATH:= $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE_TAGS := tests
LOCAL_STATIC_JAVA_LIBRARIES := android.test.runner
LOCAL_SRC_FILES := $(call all-java-files-under, src)
#修改APk的名字
LOCAL_PACKAGE_NAME:=KeyManagerServiceTests
LOCAL_STATIC_JAVA_LIBRARIES := \
    legacy-android-test \
    junit \
    android-support-test \
    mockito-target \
include $(BUILD_PACKAGE)
```

测试APK的AndroidManifest的文件：

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.keymanagerservice.tests" > <!-- 修改包的名字-->
    <application>
        <uses-library android:name="android.test.runner" />
    </application>
    <instrumentation android:name="android.support.test.runner.AndroidJUnitRunner"
        android:label="keymanagerservice test cases">
    </instrumentation>
</manifest>
```

第五步：

编写测试APK，用mm命令编译完成后，安装APK后，后执行: adb shell am instrument -w -e coverage true com.android.keymanagerservice.tests/android.support.test.runner.AndroidJUnitRunner 执行完成后，会生成一个coverage.ec文件。

第六步：

执行java -jar jack-jacoco-reporter.jar --metadata-file <metadata_file> -coverage-file <execution_file> --report-dir <report_directory> 将会生成一个覆盖率报告文件

jack-jacoco-reporter.jar： 存放目录prebuilts/sdk/tools下
metadata_file： 第三步生成的coverage.em文件
execution_file： 第五步生成的coverage.ec 文件
report_directory： 将要生成覆盖率报告存放目录.
完成以上步骤后将会在report_directory生成一个index.html文件.打开后就能查看覆盖率情况。

※在默认的情况下生成的报告，会覆盖framework中的所有类因此我们可以在framewoke/base/Android.mk中添加如下属性过滤掉我们不需要测试的类

LOCAL_JACK_COVERAGE_INCLUDE_FILTER: a comma-separated list of class names to include
LOCAL_JACK_COVERAGE_EXCLUDE_FILTER: a comma-separated list of class names to exclude These filters will be passed on the Jack command-line (see below) only when code coverage is enabled.

使用例子：LOCAL_JACK_COVERAGE_INCLUDE_FILTER := \
com.android.server. LocalServices, com.android.server. NetworkManagementSocketTagger, com.android.server.*
类名之间必须要用逗号， “\”无效 “+”无效。

参考链接：<https://android.googlesource.com/platform/prebuilts/sdk/+/master/tools/README-jack-code-coverage.md>