

runtest_howto

Last edited by **caoquanli** 1 month ago

Android runtest用法

runtest简介

runtest是谷歌内置在Android项目源码下的一个自动化测试脚本，其在项目中的位置是：

```
development/testrunner/runtest.py
```

一般我们在对自己的Java模块进行单体测试时，需要的步骤都比较繁琐，往往包括了编写测试apk，编译测试apk，安装测试apk，命令行启动测试apk进行测试，取出测试结果，手动生成覆盖率报告等步骤。而runtest脚本就是集成了其中的许多步骤，可以大大节省开发者测试的时间。此外，runtest还支持native测试，是Android Framework开发者的好伙伴。

在源码环境下配置好环境（加载环境变量，lunch）并通过adb连接实机或模拟器后，我们就可以直接在源码下输入指令来查看runtest的具体用法：

```
runtest -help
```

如何使用runtest进行自动化测试？

Java测试

1.编写测试apk

测试apk的具体写法这里不再详细介绍，大家可以参考frameworks/base/media/tests/MediaFrameworkTest，来编写自己的apk。

举个例子，如果我们要在MediaFrameworkTest中增加一个TestRunner，那么在编写完相关测试代码后，需要在MediaFrameworkTest的清单文件中把这个runner给添加上去：

```
<instrumentation android:name=".MediaUnitTestRunner"

    android:targetPackage="com.android.mediaframeworktest"

    android:label="Media Unit tests InstrumentationRunner">

</instrumentation>
```

这样，在编译完成后我们就可以使用我们熟悉的单体测试的方法来测试这个apk了，但是想要用runtest脚本来更方便地测试的话，还需要做一些事情。

2.将测试项目加入清单中

```
runtest -l
```

上面这个命令可以列出所有被记录在案的测试项目的名称，代码位置以及介绍。那么它们是在哪里被记录的，我们又该如何将自己的测试项目加入这个列表中呢？

```
development/testrunner/test_defs.xml
```

就是在这个文件里记录的。我们可以参照它的格式，把我们刚才新写好的runner给添加进去：

```
<test name="mediaut"

    build_path="frameworks/base/media/tests/MediaFrameworkTest"

    package="com.android.mediaframeworktest"
```

```
runner=".MediaUnitTestRunner"

coverage_target="framework"

description="mediaut." />
```

现在我们再运行runtest -l，就可以看到我们刚才添加的runner了：

name	build path	description

mediaut	frameworks/base/media/tests/MediaFrameworkTest	mediaut.

3.运行测试

1) 完整测试

```
runtest mediaut
```

我们可以直接使用这条命令来完整测试我们刚才添加的新的runner。脚本会为我们将测试apk编译然后安装到测试环境中，再逐项运行我们编写的测试项目。然后终端会输出运行结果，比如：

```
Test results for MediaUnitTestRunner=..F.F.F..F.F.F.....

Time: 19.168

FAILURES!!!

Tests run: 17,  Failures: 6,  Errors: 0
```

此外还会输出发生错误的测试项的详细错误信息，方便我们进一步调查。

我们并不总是需要进行完整测试，尤其是在修改验证阶段，我们可以使用如下的方法来缩小测试范围，提高测试效率。

2) 测试单个包

我们可以使用-p参数来指定测试某一个package：

```
runtest mediaut -p com.android.mediaframeworktest.unit
```

3) 测试单个类

我们可以使用-c参数来测试单个类：

```
runtest mediaut -c com.android.mediaframeworktest.unit.MediaScannerUnitTest
```

4) 测试单个方法

我们可以使用-m参数来测试单个方法（需要与-c配合使用）：

```
runtest mediaut -c com.android.mediaframeworktest.unit.MediaScannerUnitTest -m testScanExternal
```

4.调试技巧

1) 跳过编译步骤

如果我们并没有修改测试代码，那么我们可以使用-b参数来跳过编译环节，来节省测试时间：

```
runtest mediaut -b
```

2) 仅打印命令，不执行

因为runtest的参数选择众多，有时候我们可能会因为写错而导致程序跑不过，这时候我们可以使用-n参数来打印脚本将会执行的命令，但不真正执行。这样我们可以检查一下命令是否有错误，方便我们修改指令。例如：

```
runtest mediaut -n
```

终端会输出要执行的命令：

```
adb root

adb root

ONE_SHOT_MAKEFILE="frameworks/base/media/tests/MediaFrameworkTest/Android.mk" make -j16 -C

adb shell am instrument -w 'com.android.mediaframeworktest/.MediaUnitTestRunner'
```