

tf框架单体测试手顺

Last edited by **caoquanli** 1 month ago

Android Trade Federation(TF)框架简介

开发环境的设置

- 创建TradeFed
 - 在源代码树的根目录处执行如下命令：
 - \$ source build/envsetup.sh
 - \$ lunch [device-target]，注意这个device-target根据自己的编译环境选择合适的选项。
 - \$ make tradefed-all -j8
 - \$ tradefed.sh
- TradeFed框架要求在环境变量中添加adb路径
 - export PATH=\$PATH:<path/to/adb>
- 备注：开发环境的设置参考https://source.android.com/devices/tech/test_infra/tradefed/fundamentals/machine_setup

Android test生命周期

- 使用 Trade Federation 执行的测试，其测试生命周期包括Build Provider、Target Preparer、Target Preparer、Test Invocation Listener,TF框架测试的实体则是通过XML文件来声明测试组件的生命周期.如下将对测试组件简单的说明。
 - 版本提供者(Build Provider):这项在构建XML文件进行集成测试中不常见，主要是提供一个测试版本信息，必要时可下载适当的文件的功能.
 - 目标准备器(Target Preparer):这一项是进行集成测试必须的，提供了测试所需要的环境，主要包括软件安装和设备配置等.
 - 测试（Test):这一项也是必须的，主要声明了测试的类型，例如JUnit等，并且执行测试和收集测试结果.
 - 测试调用监听器（Test Invocation Listener）： 监听测试结果，通常用于将测试结果转发到存储区或将其显示给测试运行者.这一项主要实在测试代码中体现的，在测试类中的run()方法中传入ITestInvocationListener listener来体现的.
- 备注:Android test生命周期参考https://source.android.com/devices/tech/test_infra/tradefed/fundamentals/lifecycle

AndroidTest.xml

- 构建AndroidTest.xml是使用Android Trade Federation(TF)进行进行测试的重要环节.将以Android提供的AndroidTest.xml文件展开简述：

```
<?xml version="1.0" encoding="utf-8"?>
<configuration description="Runs Frameworks Utility Tests.">
  <target_preparer class="com.android.tradefed.targetprep.TestAppInstallSetup">
    <option name="test-file-name" value="FrameworksUtilTests.apk" />
  </target_preparer>

  <option name="test-suite-tag" value="apct" />
  <test class="com.android.tradefed.testtype.AndroidJUnitTest" >
    <option name="package" value="com.android.frameworks.utiltests" />
    <option name="runner" value="android.support.test.runner.AndroidJUnitRunner" />
  </test>
</configuration>
```

- 对上述的AndroidTest.xml组成进行介绍：
 - onfiguration description:对该XML文件实现的功能进行简单的描述.
 - <target_preparer class="com.android.tradefed.targetprep.TestAppInstallSetup">对应的是Android test生命周期中的target preparer.本例子target_preparer提供了测试APK从host(主机)安装到devices功能.详细功能请参考TestAppInstallSetup.java .
 - option name="test-file-name" 声明了测试APK.本例子中的测试APK为FrameworksUtilTests.apk .

- 4. option name="test-suite-tag" value="apct" 这个配置项是Android 定义的，用来运行Android平台的核心测试程序.在(android-path)tools/tradefederation/core/res/config/suite/apct.xml 进行了声明.在AndroidTest.xml中声明test-suite-tag的value="apct"的测试实例，执行apct.xml，这些测试都将会执行.基于此，在集成测试过程中可以根据项目的需要，构建自己的XML文件，并且设置自己需要的value值,来满足需求.
- 5. test class="com.android.tradefed.testtype.AndroidJUnitTest" 声明了测试的类型.Android TF框架提供了 instrumentation, uiautomator, native/gtest, host-based JUnit等测试.本例子中的测试类型为Juit测试.
- 6. option name="package" value="com.android.frameworks.utiltests" 声明了测试APK的包名.
- 7. option name="runner" value="android.support.test.runner.AndroidJUnitRunner 声明了测试的执行者.

备注：上述的AndroidTest.xml最重要的两个组成者是target_preparer class 和test class.本例子中的target_preparer 提供了指定的测试APK从host（主机）到devices（测试设备）的安装的功能.test class 则声明了测试的类型,通过配置不同的option选项实现测试.Android TF框架提供了不同类型的target_preparer类和test class类,关于详细的构建AndroidTest.xml所需要的类请参考<https://source.android.com/reference/tradefed/classes>