Leepi 二进制环境单体测试以及code coverage生成手顺

Last edited by **caoquanli** 1 month ago

1. 在build/core/tasks/check_boot_jars/package_whitelist.txt中添加如下jar包：

```
 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.*
```

   ivi/framework下生成instrumention code的时候会依赖这些包.

2. 运行 EMMA_INSTRUMENT_STATIC=true make -j8 全编译系统,然后烧卡，这样板子上运行的系统包含的iauto.framework就已经包含instrumention相关信息了。我们可以在out目录下面运行如下命令 find ./ -name "*.em",可以找到有多少代码是被instrument过。编译的过程中可能会报packages/apps/DocumentsUI的错误，可以修改其Android.mk文件，注释掉下面的两行代码：

```
 #LOCAL_JACK_FLAGS := \
#  -D jack.optimization.inner-class.accessors=true
```

ivi/frameworks测试代码的编写

以ivi/frameworks/service/radio单体测试为例，在radio目录下创建test目录，test目录结构要求如下：

- test/com/iauto/radio和java/com/iauto/radio这两个目录结构要保持一致。
- 在tests目录下的Android.mk 文件构建如下：

```
LOCAL_PATH:= $(call my-dir)
include $(CLEAR_VARS)

# We only want this apk build for tests.
LOCAL_MODULE_TAGS := tests

LOCAL_JAVA_LIBRARIES := android.test.runner

# Each update should be signed by OEMs
LOCAL_CERTIFICATE := platform
LOCAL_PROGUARD_ENABLED:= disabled

LOCAL_STATIC_JAVA_LIBRARIES := \
    android-support-test \
    legacy-android-test \
    mockito-target \
    iauto.av.radio_aidl \
    iauto.hardware.ucom.tuner-V1.0-java \

LOCAL_JAVA_LIBRARIES := \
         iauto.av.radio \
         iauto.av.media

# Include all test java files.
LOCAL_SRC_FILES := $(call all-java-files-under, com)

LOCAL_PACKAGE_NAME := radio_unit_test

LOCAL_INSTRUMENTATION_FOR := CarMediaAppServer <!--被测试的APK-->

include $(BUILD_PACKAGE)
```

- 在ivi/frameworks/service/radio目录下构建AndroidTest.xml文件，此文件主要用于TF框架,具体的信息可以参考: Android Trade Federation(TF)框架单体测试.ivi/frameworks/service/radio目录下的AndroidTest.xml构建如下：

```
<configuration description="Runs Tests for iauto framework">
    <target_preparer class="com.android.tradefed.targetprep.InstallBuildEnvApkSetup">
        <option name="apk-name" value="radio_unit_test" /> <!--测试APK-->
    </target_preparer>
     <option name="test-suite-tag" value="iautotest" />
    <test class="com.android.tradefed.testtype.CodeCoverageTest" >
        <option name="package" value="com.iauto.radio" /> <!--测试APK的包名-->
        <option name="runner" value="android.support.test.runner.AndroidJUnitRunner" />
        <option name="metadata-file"
         value="out/target/common/obj/APPS/CarMediaAppServer_intermediates/coverage.em" /
>
        <option name="source-dir" value="ivi/frameworks/server/mediaserver" /> <!--被测试A
PK代码路径-->
    </test>
</configuration>
```

- 目录下构建AndroidManifest.xml文件，构建如下：

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.iauto.radio"> <!--测试APK的包名-->
    <application>
        <uses-library android:name="android.test.runner" />
    </application>
    <instrumentation android:name="android.support.test.runner.AndroidJUnitRunner"
        android:targetPackage="com.iauto.mediaserver" <!--被测试APK的包名-->
        android:label="Test Cases">
    </instrumentation>
</manifest>
```

编译生成测试APK

在源码目录下运行make radio_unit_test,在out/target/product/mt2712/data/app/radio_unit_test目录下生成测试radio_unit_test.apk.

使用TF框架运行单体测试并生成单体测试报告

在源码路径下面运行 tradefed.sh，然后输入 run ivi/frameworks/api/tests/iautoframeworkstests/AndroidTest.xml 命令:

- host log（PC端屏幕显示的log）如下:

```
 tf >run ivi/frameworks/service/radio/test/AndroidTest.xml
01-03 15:28:45 I/TestInvocation: Invocation was started with cmd: ivi/frameworks/service/
radio/test/AndroidTest.xml
01-03 15:28:45 W/TestInvocation: Using the test-tag from the build_provider. Consider upd
ating your config to have no alias/namespace in front of test-tag.
01-03 15:28:45 I/TestInvocation: Starting invocation for 'stub' with '[ BuildInfo{bid=0,
target=stub, serial=GFEDCBA0987654321} on device 'GFEDCBA0987654321']
01-03 15:28:45 W/FileUtil: Chmod is not supported by this OS.
01-03 15:28:45 I/InstallBuildEnvApkSetup: Installing radio_unit_test.apk on GFEDCBA098765
4321
01-03 15:28:47 W/FileUtil: Chmod is not supported by this OS.
01-03 15:28:47 I/TextResultReporter: Saved device_logcat_setup log to /tmp/0/stub/inv_691
3749277805987989/device_logcat_setup_4949831287736433209.zip
01-03 15:28:47 I/RemoteAndroidTest: Running am instrument -w -r   -e timeout_msec 300000
-e coverage true com.iauto.radio/android.support.test.runner.AndroidJUnitRunner on unknow
n-car_on_mt2712-GFEDCBA0987654321
01-03 15:28:50 I/InvocationToJUnitResultForwarder: run com.iauto.radio started: 136 tests
01-03 15:28:53 I/InvocationToJUnitResultForwarder: run ended 3545 ms
tf >
01-03 15:28:53 I/CodeCoverageTest: coverage file from device: /tmp/coverage_6600400057907
663421.ec
01-03 15:28:53 W/FileUtil: Chmod is not supported by this OS.
01-03 15:28:53 I/TextResultReporter: Saved com.iauto.radio_runtime_coverage log to /tmp/
0/stub/inv_6913749277805987989/com.iauto.radio_runtime_coverage_932866009480428416.zip
01-03 15:28:53 I/CodeCoverageTest: create report dir: out/target/testcases/coverage/com.i
auto.radio successfully
01-03 15:28:53 I/CodeCoverageTest: Jacoco code coverage report generation success
01-03 15:28:53 W/FileUtil: Chmod is not supported by this OS.
01-03 15:28:54 I/TextResultReporter: Saved device_logcat_test log to /tmp/0/stub/inv_6913
```

```
749277805987989/device_logcat_test_4461067581343919337.zip
01-03 15:28:54 W/FileUtil: Chmod is not supported by this OS.
01-03 15:28:54 I/TextResultReporter: Saved device_logcat_teardown log to /tmp/0/stub/inv_
6913749277805987989/device_logcat_teardown_5217203794424155016.zip
01-03 15:28:54 W/FileUtil: Chmod is not supported by this OS.
01-03 15:28:54 I/TextResultReporter: Saved host_log log to /tmp/0/stub/inv_69137492778059
87989/host_log_5342641992142817308.zip
```

- 备注： 覆盖率生成的报告在out/target/testcases/coverage目录下，如果测试过程中生成了.ec 文件但是没生成测试报告，请使用eng版本.