webos-tv-media-test application

Summary

This application aims to automatically validate the media framework on the webos tv platform. There are some automated test tools, but there are various limitations in testing in nfs or testing on webos tv platform. This led to the development of native application so that automated testing can be performed nfs or epk without restrictions.

Application Configuration

To use this application, you need the following three.

1. Application ipk It can be installed on TV set on dev mode.
2. Test rule file (rule.xml) Test rule is defined with xml format. You have to use pre-defined cases at this moment but you can change the numbers (position, timeout, duration, etc). How to write a test rule is explained in detail below.
3. Test contents If you specify a single root directory, you can search and test media files in all subdirectories below it.

How to use

Here is how to use the above three.

When all three are ready, copy them to usb.

The test rule file should be created in the root path of usb named webos-tv-media-test and copied under the name rule.xml under it.

```
> rule.xml -> usb root/webos-tv-media-test
> webos-tv-media-test_xxx.ipk -> usb root/anywhere
> contents directory -> usb root/anywhere
```

The rule.xml file consists of xml and the location of the test contents must be specified so that the application can search all the media files under that location and perform the test. The position can be specified by the value of "root" element in the rule element. See examples below. `Note: Omit the root path of usb. Do not add space in front of path or the end.`

```
<root type = "usb">salmon</root>
```

Connect the usb that contains the necessary components to run the application to the tv set and change the tv to dev mode.

See below for how to change it.

```
> touch /var/luna/preferences/devmode_enabled
> touch /var/luna/preferences/debug_system_apps
> reboot
```

Install the ipk file of the copied path to webos tv set using luna command.

The luna command for installing ipk is as follows if Ipk file name is webos-tv-media-test_1.0.0_all.ipk and located in usb device root path.

```
luna-send -i palm://com.webos.appInstallService/dev/install
'{"id":"webos-tv-media-test","ipkUrl":"/tmp/usb/sda/sda1/webos-tv-media-test_1.0.0_
all.ipk","subscribe":true}'
```

Once installed, you can run the installed application through the launcher. However, you must connect usb to tvset before running the application. Otherwise, you will not be able to recognize usb. If you do not recognize usb, reboot the TV with usb connected and run the application again.

Run the application.

Once you launch the application, you will see the load button.

Load

Selecting the load button will parse the rule.xml file and display the parsed rule, and reload, test all, test audio, and test video buttons will be created.

Reload

Reload rule.xml.

Test all

Test all audio / video contents as test rule.

Test audio or test video

Only test audio or video contents.

Performing and terminating the test

Only the stop button is present during the test. The execution termination condition is changed to the UI that shows the test result after inputting the stop button, ending the application, or ending the test. The detailed results of the test are stored in report_Time.csv format in `usb root/webos-tv-media-test/report_xxx.csv` and `the file name is displayed on the screen` at the end of execution.

Exit

Press '0' button of remote controller.

Logging

Detailed report infomation would be save in "usb root/webos-tv-media-test/report_starttime.csv". If signal is detected during testing, call stack would be save in report log file and PmLog and Legacy-Log would be saved under the folder named same as time info. * Report file name : report_0_09h.08m.15s_01-01-1970.csv * PmLog and Legacy-log path : usb root/webos-tv-media-test/0_09h.08m.15s_01-01-1970/

How to write xml test rule

I will explain how to write the xml test rule file here. The test cases that can be tested with the media compatability test tool are as follows. 1. playback 2. seek & eos (25%, 50%, 95%) -> make seek position variable 3. audio track change 4. pause & resume 5. trick play (forward, rewind trick play)

The following additional test cases are added to this project. 1. continuous play : resume play (it can not be selected, need to find the way to test) preload -> load -> play preload -> preseek -> play 2. combination test 3. position can be selected with percentage or time 4. timeout can be selected

Element and Attribute definitions

ELEMENT is used for functions and test cases, ATTRIBUTE is used for setting properties. Below list is the element and attribute lists. Element can have attributes and sub elements, but unit test element and test element can not have sub element. Attribute value defines types of attribute. Below tables are the definition of elements and attribute types in detail. + element : unit test and test case definition

```
root, default, position-update element, stop-after-crash element, functions, wait,
case, test, run, rule
```

- attribute : all properties

```
type, path, file-log, enable, low-performance, stop, timeout, repeat, measure,
position, duration, times, sequence, id, name, case-id
```

ELEMENT

Element to set root path that is located the streams to be tested

element name	possible attribute
root	type, path

Element to set default properties

element name	possible attribute
default element	timeout, measure, repeat, file-log
position-update element	enable
crash element	stop

Function element (unit test case)

All unit test can have timeout attribute than omit timeout and can not have other element.

function	possible attribute
load	timeout
play	-
seek	position, measure
audio-track-change	times
pause-resume	times
playbackrate	rate
stop	position, measure
resume-play	position, measure
preload-play	position, measure
preload-preseek	position, measure
wait	duration, measure

Test case element

Test case that include more than a unit test case, but it only a unit test case is possible at this moment.

element name	possible attributes/elements
case	id, name [mandatory and must unique, both]
	function element, wait [case must have more than a function]
run	times, sequence
	test [must have more than a test element]
test	case-id
	no element
rule	top element
	default, wait, case, run

ATTRIBUTE

Attributes to set property

attribute	type
type	"usb" or "dlna", ...
path	string(uri)
file-log	"enable"(default) or "disable"

attribute	type
enable	"true" or "false"
low-performance	"true" or "false"
stop	"enable" or "disable"
timeout	unsigned int: always ms(milliseconds)
repeat	enable or diable repeat test option
measure	can be "percentage" or "ms"
position	unsigned integer
duration	unsgined integer
times	unsigned integer [0 -> infinite or all]
sequence	"whole" or "each"
rate	integer [-1..16]
id	unsigned integer
name	string(unique name)
case-id	unsigned integer

Example of rule

```
<rule>
  <root type="usb">salmon</root>
  <default timeout="10000" measure="percentage" repeat="disable"> </default>

  <position-update enable="true" low-performance="true"/>
  <crash stop="enable"/>

  <!-- test case definition -->
  <case id="1" name="play">
    <play/>
    <wait duration="2000" measure="ms"/>
  </case>

  <case id="2" name="seek">
    <play/>
    <wait duration="1000" measure="ms"/>
    <seek position="25" measure="percentage"/>
    <wait duration="1000" measure="ms"/>
    <seek position="50" measure="percentage"/>
    <wait duration="1000" measure="ms"/>
    <seek position="97" measure="percentage"/>
    <wait duration="100" measure="percentage"/>
  </case>

  <case id="3" name="audio track change">
    <play/>
    <wait duration="1000" measure="ms"/>
    <audio-track-change times="3"></audio-track-change>
    <wait duration="1000" measure="ms"/>
  </case>


  <case id="4" name="pause/resume">
    <play/>
```

```
    <wait duration="1000" measure="ms"/>
    <pause-resume times="3"></pause-resume>
    <wait duration="1000" measure="ms"/>
  </case>

  <case id="5" name="trick play x2">
    <play/>
    <wait duration="1000" measure="ms"/>
    <playbackrate rate="2"></playbackrate>
    <wait duration="1000" measure="ms"/>
    <seek position="90" measure="percentage"/>
    <playbackrate rate="-2"></playbackrate>
    <wait duration="1000" measure="ms"/>
  </case>
    <case id="7" name="trick play x8">
    <play/>
    <wait duration="1000" measure="ms"/>
    <playbackrate rate="8"></playbackrate>
    <wait duration="1000" measure="ms"/>
  </case>

  <case id="8" name="trick play x16">
    <play/>
    <wait duration="1000" measure="ms"/>
    <playbackrate rate="16"></playbackrate>
    <wait duration="1000" measure="ms"/>
  </case>

  <case id="9" name="resume play">
    <resume-play position="50" measure="percentage"/>
    <wait duration="2000" measure="ms"/>
  </case>

  <case id="10" name="preload play">
    <preload-play position="50" measure="percentage"/>
    <wait duration="2000" measure="ms"/>
  </case>

  <case id="11" name="preload preseek">
    <preload-preseek position="50" measure="percentage"/>
    <wait duration="5000" measure="ms"/>
  </case>

  <case id="12" name="play-eos">
    <play/>
    <wait duration="100" measure="percentage"/>
  </case>

  <!-- run definition of which case need to run -->
  <run times="1" sequence="whole">
    <test case="1"/>
    <test case="2"/>
    <test case="3"/>
    <test case="4"/>
    <test case="5"/>
    <test case="6"/>
    <test case="7"/>
```

```
    <test case="8"/>
    <test case="9"/>
    <test case="10"/>
    <test case="11"/>
    <test case="12"/>
  </run>

</rule>
```

It is strongly recommended to use the test case written by the project owner. Otherwise, if the wrong test case is written in the rule file, it may cause the system to behave abnormally. Please contact heekyoung.seo@lge.com for further questions.