



BTS – Stream

Content

INTRODUCTION.....	2
TEST TOOLS.....	2
KAPANGA	2
SIPp 2	
QUICKTIME PLAYER	2
RTPDUMP	2
TEST ENVIRONMENT	2
TEST EXECUTION.....	3
AUTOMATED TEST CASES	3
Test Documentation	3
Test Location	3
Execute Tests	3
AUTOMATED STREAMING TEST CASES	3
Test Documentation	3
Test Location	4
Execute Tests	4
MANUAL TEST CASES	4
Test Documentation	4
Stream 4	
ccRTPAdapter	5
MediaLibrary	5
TEST CASES	5
VERIFY MAKEFILES	5
VERIFY PLAY	6
VERIFY RECORD	6
REPORTING TEST RESULTS	6
REFERENCES	6
TERMINOLOGY	7

History



Version	Date	Adjustments	Signum
A	2006-06-13	First version.	QMAFL
B	2008-08-05	Updated for automated streaming tests.	EERITEG

Introduction

This document describes basic test information for Stream. It documents the test tools needed, how to set up the environment for various tests, and the procedure for testing. It also specifies the test cases for basic test.

After reading this document, you will know how to set up, execute and document a basic test for this product.

Note: if you choose to renumber the test cases in this document, you will need to renumber also the test cases in the BTR accordingly.

Test Tools

The following tools are used when performing manual testing of the Stream functionality.

Kapanga

Kapanga is used for sending RTP (AMR and DTMF).

SIPp

SIPp is used for setting up, through SIP invite, a call between the test program and Kapanga.

QuickTime Player

QuickTime player is used for playing WAV, MOV and 3GPP files, it can also be used as a RTP listener.

rtpdump

rtpdump is a tool that can be used for recording and analysing streams.

Test Environment

In order to run the basic tests, the following environment variables must be set:

- JAVA_HOME pointing to a Java 1.5 installation
- COBERTURA_HOME pointing to a Cobertura installation
- For running the automated streaming tests, the shared library libccrtpadapter.so must be in LD_LIBRARY_PATH.



Test Execution

This section specifies how to execute the Stream basic tests. The Stream basic tests are divided into automated and manual tests.

Automated Test Cases

Test Documentation

The test cases are documented in the source code using javadoc.

Test Location

The automated test cases can be found here:

`/vobs/ipms/mas/stream/test`

Execute Tests

Run the automated test cases using the ant target *runtest*.

Check test case coverage using the ant target *runcoverage*.

Automated Streaming Test Cases

Test Documentation

Some automated tests involve complete use cases of Stream. These have been separated from the other automated tests that only do unit testing of stream java classes and do not involve the C++ part of stream and the actual streaming over RTP.

Tested use cases

- **Play:** Plays media of type WAV, MOV and 3GP and compares them with a reference RTP streams. In this test the reference files were created by using stream to play the media, so this case is more like a regression test to see that play functionality is not destroyed when making changes in stream.
- **Record:** Records reference RTP streams with payload of types PCMU, PCMU/H.263 and AMR/H.263. The recorded media objects are then played and the resulting RTP streams are compared with the original reference streams.
- **Record with packets out of order:** Same as record but RTP packets in the reference files are first reordered before sending them to stream.
- **Inbound DTMF:** Verifies that inbound DTMF during record generates the correct events.
- **Join:** Joins an inbound and an outbound stream and sends reference RTP streams (PCMU, PCMU/H.263 and AMR/H.263) to the inbound stream and verifies that the same streams are sent by the outbound stream. For now only homogenous joins (audio -> audio and video -> video) are tested,



tests for heterogeneous joins (audio -> video or video -> audio) should be added.

- DTMF over joined streams: Same as join but verifies that DTMF is forwarded to the outbound stream.
- Playing a corrupt MOV file: Verifies that playing a corrupt MOV file generates a Play Failed event.
- Play in progress: Verifies that issuing a play during an ongoing play generates a Play Failed Event.
- Record in progress: Verifies that issuing a record during an ongoing record generates a Record Failed event.

Creating reference RTP streams

Reference RTP streams are stored in rtpdump-format. RTP streams can be recorded with the rtpdump tool. It is also possible to save an RTP stream from a .pcap file in rtpdump format using Wireshark.

Suitable testfiles can be recorded with for instance rtpdump in combination with SIPp and Kapanga.

Reference files for the play test are created with the java program:

```
/vobs/ipms/mas/stream/mtest/com/mobeon/masp/stream/CreatePlayReference.java
```

Test Location

The automated streaming tests are implemented in the java program:

```
/vobs/ipms/mas/stream/mtest/com/mobeon/masp/stream/StreamTest.java
```

Execute Tests

Compile and run the java program.

Manual Test Cases

Since the functionality of Stream is based upon real-time streaming through a protocol stack and a network it has not been simple to apply automatic testing of all use cases. It is not impossible, but for now some of the test cases must be executed manually.

There are test cases for the packages Stream (Java), ccRTPAdapter and MediaLibrary (C++).

Test Documentation

The test cases are documented within this document.

Stream

Test Location

The test programs are located at /vobs/ipms/mas/stream/mtest.



Execute Tests

Use the script runtest.sh with proper class name and arguments (test case).

ccRTPAdapter

Test Location

The test programs are located at /vobs/ipms/mas/stream/ccrtpadapter/mtest.

Execute Tests

Compile and run mtest. Ensure that no errors are reported by the test program.

Clearmake -C gnu basictest

MediaLibrary

Test Location

The test programs are located at /vobs/ipms/mas/stream/medialibrary/mtest.

Execute Tests

Compile and run mtest. Ensure that no errors are reported by the test program.

Clearmake -C gnu basictest

Test Cases

Here are the manual test cases for Stream. All basic tests that are automated are described in javadoc (test source).

The test cases here below should be performed for both voice (audio only) and video (audio and video).

Verify Makefiles

The make files have been shown to be error prone. There have been problems where the clean rules has failed due to updated make structure etc. When clean has failed there has been inconsistencies in the Stream libraries. In order to verify this we need to verify the clean targets of the makefiles.

1. Enter the root directory of Stream (/vobs/ipms/mas/stream/).
2. Ensure that you have a clean view by launching cleartool lsprivate | grep stream. The view is clean if there are no Stream related object and library files.
3. Build the Java parts: ant compile.
4. Build the C++ parts: clearmake -C gnu dist
5. Ensure that there are Stream related object and library files by launching cleartool lsprivate | grep stream.
6. Remove the C++ build: clearmake -C gnu distclean.



7. Ensure the Stream related object and library files are removed by launching cleartool lsprivate | grep stream.

The cleandist target is verified ok if the Stream related object and library files are properly removed.

Verify Play

For verifying play the java program /vobs/ipms/mas/stream/mtest/com/mobeon/masp/stream/MediaPlayer.java is used for streaming media files (WAV, MOV and 3GP). Use Kapanga in combination with SIPp to verify that audio and video is heard/seen ok.

It is also possible to stream directly to the QuickTime player. By specifying the SDP session description in a file and opening it with QuickTime, QuickTime will start playing media received from the ports specified by the SDP description. See the SDP session description example below:

```
v=0
o=test 2890844526 2890842807 IN IP4 150.132.4.148
s=SDPTest
i=SDP file for AMR audio on port 23000
c=IN IP4 150.132.4.148
t=0
m=audio 23000 RTP/AVP 96
a=rtpmap:96 amr/8000
a=fmtp:96 octet-align=1; robust-sorting=0
a=ptime:20
a=maxptime:20
m=video 23002 RTP/AVP 34
a=rtpmap:34 h263/90000
a=fmtp:34 QCIF=4
a=sendrecv
```

Verify Record

In addition to the automated tests of record it is needed to test that the recorded media objects are playable in other media players. To verify this the java program /vobs/ipms/mas/stream/com/mobeon/masp/stream/PlayRecorder.java can be used. PlayRecorder plays media files and records them into another media file. The recorded media file can then be played in for instance the QuickTime Player.

Reporting Test Results

The test results are printed to stdout when executing the test cases.

References

This paragraph is currently left blank.



Approved: Per Berggren

Mobeon Internal

No: 7/BTS-MAS0001 Uen

Copyright Mobeon AB
All rights reserved

Author: Bernard Melsom/Erik Tegeback
Title: BTS – Stream

Version: B
Date: 2008-08-05

7/7

Terminology

Term

Explanation