



Packet Ship *Streamline* Media Server

Release Note 3.1 “Antigua” Update 5

Package	Version	Revision
ps-streamd	3.1.5	1
ps-index-mpeg2ts	3.1.5	1
ps-analyse-mpeg2ts	3.1.2	1



Release Note

The 3.1 “Antigua” Update 5 release of the Packet Ship Streamline media server contains the following changes:

- Handle HLS session timeout and restart properly
- Near-live buffering fixes
- Position and duration returned by live streams
- New **<ts-pat-pmt-adjust>** output filter
- Fix occasional misses in **<ts-cc-adjust>** and **<ts-timing-adjust>** output filters
- SOAP `stream-control` message with attributes acts as keepalive
- Fix to position reported when running off the end of an asset in trick mode
- Fix selection of end position in captured programme in RTSP
- Withdrawal of 'proportion' attribute of the **<limit>** rate profile
- Removal of unused filters and rate-profiles from standard configuration
- Improve handling of unusual cases in indexer

HLS session timeout

If an HLS client started a stream and then stopped fetching chunks for a while – for example while paused, or during a network failure – then restarted, the previous version would restart the stream but not recreate the session in the HLS controller. The effect was that the second chunk after restart would cause a second stream to be started with the same ID, which would fail. Clients which refetched the playlist on a failure would recover, others would not.

The HLS controller now re-establishes a session if one is quoted and it does not previously exist, which fixes this issue.

Near-live buffering

When a stream being recorded by the Timeline IPTV Recorder was being played out very close to the live position (within a few seconds) the buffering algorithm could introduce duplicate data as it went back to load small chunks of additional data as it was written, causing errors and subsequent non-alignment of the output stream. This has now been fixed.

Position and duration for live streams

When a live captured stream is played with RTSP, the previous version returned positions (in both Range: headers and GET_PARAMETER(position) results) which were based on the position within the playlist returned by the Timeline recorder. This is irrelevant to the client and caused some confusion. The position returned is now relative to the start point in the playlist as expected.



Also, live capture streams previously returned a duration (in Range: headers and GET_PARAMETER(duration) results) which was the length of the playlist available from Timeline. Again this is not relevant to the user, and the new version now follows the RFC in leaving out the duration in a Range header and reporting it as -1.0 if explicitly requested with GET_PARAMETER.

One effect of these two changes is that VLC now suppresses the progress bar for near-live streams, as would be expected.

New PAT/PMT adjustment output filter

If a number of assets are concatenated in a playlist the PMT may change as each new asset is cued, but unless the version number is incremented some clients will not be able to spot this. A new **<ts-pat-pmt-adjust>** output filter provides an option to rectify this. This filter observes the PAT and PMT data being output, and if the contents of either changes without a version increment, it forces one.

The PAT/PMT adjustment filter is configured just like the other TS filters within the **<filter>** section of an **<output>**, with the option to enable it for normal play and trick play.

```
<filters>
...
  <ts-pat-pmt-adjust>
    <normal play="yes"/>
    <trick mode="yes"/>
  </ts-pat-pmt-adjust>
...
</filters>
```

Fix misses in other adjustment output filters

The **<ts-cc-adjust>** and **<ts-timing-adjust>** output filters could occasionally miss out updating a packet if the relevant part of it was split across two disk buffers. The result was very occasional jumps back to the original CC or timing values in the stream. This has now been fixed.

SOAP play message as keepalive

The stream-control SOAP message (bound by default to /control in HTTP and packetship.stream.control in XMLMesh) can be used in two ways; with no attributes except the stream ID it just acts as the necessary keepalive for a server-controlled stream. With attributes such as 'speed' it can be used to control the stream playback speed, seek it and so on.

Previously if attributes were provided then the message did not act as a keepalive, and an attribute-less one was also required. This was obviously unexpected and inefficient so the version with attributes now also acts as a keepalive as would be expected.



Fix to position reported in trick mode

If the client fast-forwarded or rewound past the end or beginning of an asset, the position reported could be inconsistent. This has been fixed and the position reported will now be that of the last I-frame in either case.

Capture end position in RTSP

A bug was introduced in v.3.1.4 whereby an end position specified on a captured program through integration with Timeline (e.g. by 'for' or 'to' on the URL) would not be honoured in RTSP. This has now been fixed, and the earlier of the end point specified by the URL and the end point specified by any other mechanism (e.g. URL parameters or PLAY request) is used.

Withdrawal of proportional rate limits

The **proportion** parameter of the **<limit>** rate profile used to allow minimum and maximum limits to be set as a proportion of the current asset rate. However, this did not work properly at asset boundaries within a playlist or at the end of a stream, because the 'current asset' changes, or becomes invalid, while rate-pacing is still continuing from buffered data.

There is no way to safely fix this so this feature has been withdrawn. The **<limit>** rate profile still provides absolute minimum and maximum **rate** limits as before, the **proportion** parameter of each will simply be ignored.

Cleanup of unused filters and rate profiles

The standard configuration contained examples of all the filters and rate profiles available but with parameters which effectively disabled them. This was slightly inefficient because they were still being called as part of the output process even though they did nothing.

The filters and rate profiles have now been moved to **<disabled-filters>** and **<disabled-profiles>** elements respectively, within the “udp” output. To enable them they simply need to be copied or moved to the (now empty) **<filters>** and **<rate-profiles>** elements. Since they are not actually used they have now been given sensible example parameters.

Unusual cases in ps-index-mpeg2ts

Two problems handling unusual input have been fixed in ps-index-mpeg2ts:

Firstly a PAT with no program streams would cause an indexer crash. Of course such a stream is useless, but the indexer now fails gracefully.

Secondly a large negative delta in a single PCR value would 'stick' even if then corrected in subsequent packets. This would propagate errors to the end of the file and throw out rate calculations. This has now been fixed and the out-of-line value is ignored.