



NEXUS SPLICE API

Revision History

Revision	Date	Change Description
0.1	5/8/18	First revision

Table of Contents

Overview	1
Use cases	1
1. Starting from MC in playback mode.....	2
2. Starting from RC in playback mode	2
3. Going from RC to MC in playback mode	2
4. Starting from MC in live mode	3
5. Starting from RC in live mode	3
6. Going from RC to MC in live mode	4
7. Playing back to back RC clips.....	4
Playback call sequence diagram	5
Live call sequence diagram	6

Overview

This document provides a high level overview of using the Nexus Splice API to accomplish on-the-fly content replacement in live and recorded streams. In this document the main content (MC) and replacement content (RC) are abbreviated as shown. The points at which replacement content must stop or start are called “splice points”. The splice point at which main content should stop and replacement content should start is called the “stop splice point”. The splice point at which replacement content should stop and main content should resume is called the “start splice point”.

To perform content replacement the following requirements must be met:

1. MC start and stop splice points PTS values must be known in advance
2. RC start and end PTS values must be known in advance
3. RC should be available on the local storage before MC reaches any of the splice points.
4. MC and RC should have the same video and audio codecs
5. MC stop splice point does not need to be an IDR
6. MC start splice point must be an IDR
7. RC must start with an IDR but does not need to end with an IDR
8. RC must fit exactly in between MC start and stop splice points otherwise there will be visible delay or macro blocking at MC start splice point.

For more details please examine files `dvr_ad_replacement.c` and `live_ad_replacement.c` in the `nexus/examples` directory

Use cases

1. Starting from MC in playback mode
2. Starting from RC in playback mode
3. Going from RC to MC in playback mode
4. Starting from MC in live mode
5. Starting from RC in live mode
6. Going from RC to MC in live mode
7. Playing back to back RC clips

This document assumes the user is familiar with the Nexus video and audio playback APIs and so the provided information is concentrated around the splicing API and related Nexus APIs to make transitions possible. Before calling any of the Splice APIs for video and audio decoders, the application must set `spliceEnabled` flag in respective decoder `OpenSettings` structure.

1. Starting from MC in playback mode

When starting from MC in playback mode, the content must transition to RC at the stop splice point. To prevent any delays during RC start it is recommended to setup playpump for RC playback ahead of time and even pre-read the first few blocks of RC if possible. To setup the transition application must execute following call sequence:

1. Call NEXUS_(Audio/Video)Decoder_GetSpliceSettings to obtain initialized settings structure
2. Update settings structure with NEXUS_DecomderSpliceMode_eStopAtPts mode and other parameters
3. Call NEXUS_(Audio/Video)Decoder_SetSpliceSettings to tell corresponding decoder to start looking for splice point PTS
4. Wait until configured callback is invoked or periodically poll using NEXUS_(Audio/Video)Decoder_GetSpliceStatus API. If application elects to poll, it must do so at no less than 100ms intervals as not to miss transition point.
5. When splice point is reached, decoding of MC is stopped and provided callback is invoked.
6. Application then must replace PID values in the PidChannel by calling NEXUS_PidChannel_ChangePid API or if different playpump is used for MC and RC, application must use NEXUS_(Audio/Video)Decoder_SpliceStopFlow and NEXUS_(Audio/Video)Decoder_SpliceStartFlow API to replace PidChannelHandle to the new PidChannelHandle associated with RC playpump.
7. Application then must prepare and send PTS offset BTP packet
8. Application then must prepare and send START marker BTP packet
9. Application then must start to feed RC using playpump API

Note that the splice point is detected sometime before RC should be displayed so the application has time to perform these operations. After RC playpump is started it takes about 1 second for RC to be displayed but the transition will be invisible, provided all is done correctly.

2. Starting from RC in playback mode

When starting from RC in playback mode, the application does not need to transition to RC. It can simply start feeding RC using the playpump API, provided video and audio decoders were configured with the correct PidChannelHandle. If the decoders were configured with MC PidChannelHandle, the application can use NEXUS_(Audio/Video)Decoder_SpliceStopFlow and NEXUS_(Audio/Video)Decoder_SpliceStartFlow API to replace the PidChannelHandle with the new PidChannelHandle associated with RC playpump. The PTS offset BTP packet must be sent and START marker BTP packet can be sent if desired before feeding RC using playpump API.

3. Going from RC to MC in playback mode

RC can transition to MC using start splice point. When feeding RC to playpump API is finished the application must execute the following sequence to transition back to MC:

Nexus Splice API

1. Prepare and send PTS offset BTP packet to turn off PTS adjustment
2. Prepare and send STOP marker BTP packet to stop decoding from this point
3. Replace PID values in the PidChannel to MC PID values by calling NEXUS_PidChannel_ChangePid API or NEXUS_(Audio/Video)Decoder_SpliceStopFlow and NEXUS_(Audio/Video)Decoder_SpliceStartFlow API.
4. Call NEXUS_(Audio/Video)Decoder_GetSpliceSettings to obtain initialized settings structure
5. Update settings structure with NEXUS_DecoderSpliceMode_eStartAtPts mode and other parameters
6. Call NEXUS_(Audio/Video)Decoder_SetSpliceSettings to tell corresponding decoder to start looking for splice point PTS
7. Resume playback of MC using playpump API.
8. Wait until configured callback is invoked, indicating that splice point is reached

Shortly after callback is invoked, video and audio should transition to the MC. The transition should be invisible, provided all is done correctly. This transition is essentially similar to going from MC to RC but done in reverse order.

4. Starting from MC in live mode

This transition is almost the same as transition (1) except the application must use NEXUS_(Audio/Video)Decoder_SpliceStopFlow and NEXUS_(Audio/Video)Decoder_SpliceStartFlow API to replace PidChannelHandle. Instead of PTS offset BTP packets, PCR offset BTP packets should be used in live mode.

5. Starting from RC in live mode

This transition is the most complex of all. In order to ensure a clean transition back to the MC, the application must start from MC and then immediately transition to RC without displaying any frames of MC.

1. Before starting decode, freeze video decoder and mute audio decoder
2. Start decode of MC
3. Call NEXUS_(Audio/Video)Decoder_GetSpliceSettings to obtain initialized settings structure
4. Update settings structure, set pts to 1 and ptsThreshold to 0xFFFFFFFF, capturing any PTS value
5. Call NEXUS_(Audio/Video)Decoder_SetSpliceSettings to tell corresponding decoder to start looking for splice point PTS
6. Wait for the callback which will happen almost immediately
7. Call NEXUS_(Audio/Video)Decoder_SpliceStopFlow() API to stop data flow to the decoders
8. Call NEXUS_StcChannel_GetStc to obtain current STC value for the live stream.
9. Flush decoders
10. Call NEXUS_(Audio/Video)Decoder_SpliceStartFlow() to set new NEXUS_PidChannelHandle that is associated with playback channel.
11. Prepare and send PCR offset BTP
12. Prepare and send START marker BTP

Nexus Splice API

13. Locate starting position in RC which should be first IDR at PTS that is larger than the STC value obtained in step 8
14. Unfreeze video and unmute audio decoder
15. Start RC playback using playpump API.

You will observe a short pause before playback while the STC catches up to the desired PTS of the RC. Then the video and audio should start and play as normal. If everything is done correctly, the transition back to MC should be invisible.

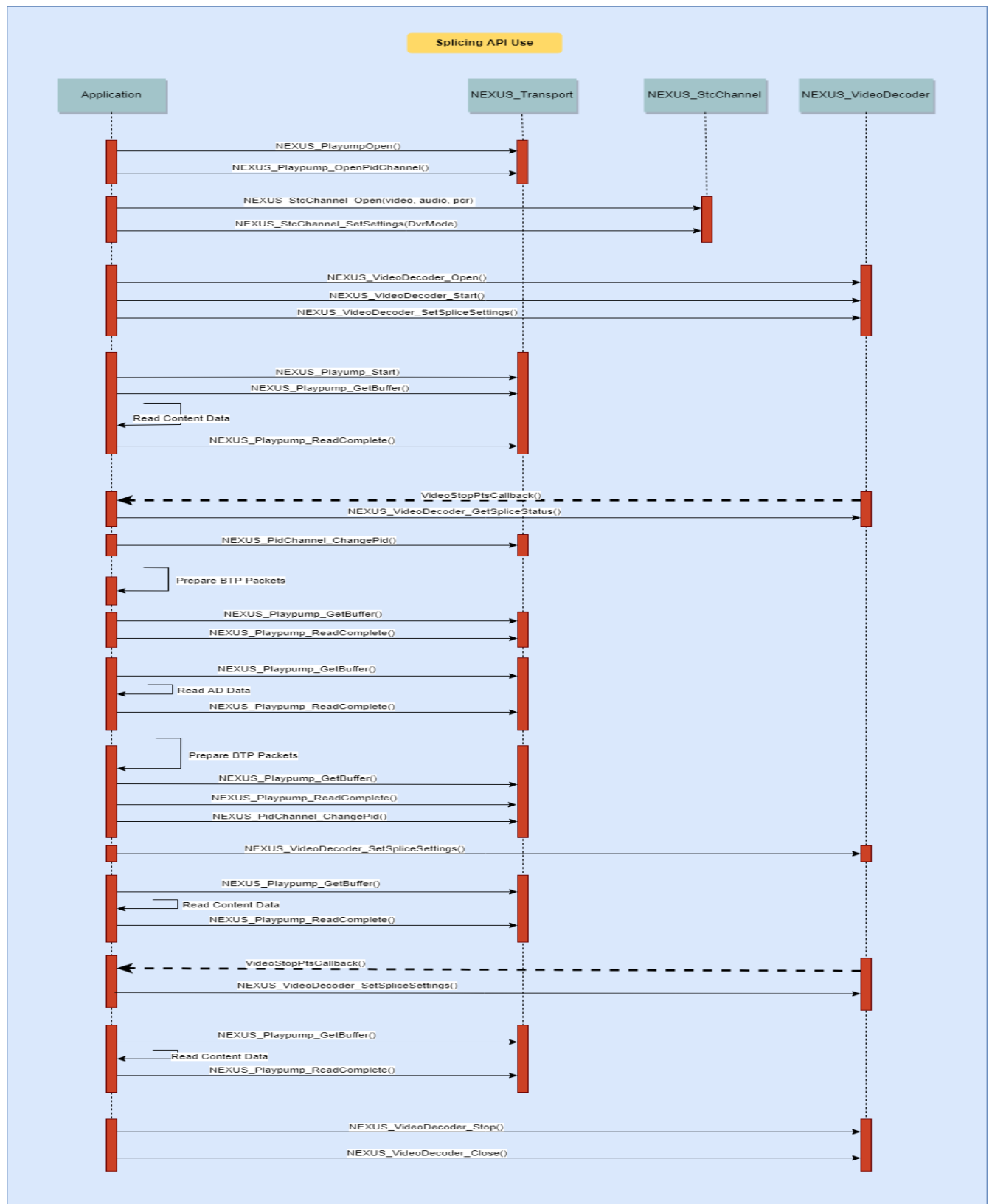
6. Going from RC to MC in live mode

This transition is very similar to the playback mode. Application must use NEXUS_(Audio/Video)Decoder_SpliceStopFlow and NEXUS_(Audio/Video)Decoder_SpliceStartFlow API to replace PidChannelHandle. Instead of PTS offset BTP packets, PCR offset BTP packets should be used in live mode.

7. Playing back to back RC clips

If several RC clips are fitted in one window of MC, the application should play these clips one after another sending PTS offset BTP packet in playback mode or PCR offset BTP packet in live mode to adjust PTS as needed. Note that the combined duration of all RC clips should exactly match the time between stop splice point and start splice point in MC. To transition back to MC use the appropriate sequence from this document after finishing the last RC clip.

Playback call sequence diagram



Live call sequence diagram

