
Notes:

- For step-by-step instructions on how to update your Wowza Streaming Engine media server software, see Update your Wowza Streaming Engine installation.
- Docker images for Wowza Streaming Engine 4.8.0 are now available. Additional Cloud platform updates are coming soon.
- If using the Wowza Streaming Engine 4.8.0 installer to install a new instance of Wowza Streaming Engine, you must install additional dependencies in order to ingest SRT MediaCaster streams or use the generic SRT stream targets destination to publish an SRT stream. To address this, install the latest version of the Microsoft Visual C++ Redistributable for Visual Studio appropriate to your operating system. This is not an issue for pre-existing installations of Wowza Streaming Engine that are updated to Wowza Streaming Engine 4.8.0.

Contents

New features and functionality in Wowza Streaming Engine 4.8.0
Breaking changes in Wowza Streaming Engine 4.8.0
Detailed list of changes in Wowza Streaming Engine 4.8.0
Known issues in Wowza Streaming Engine 4.8.0

New features and functionality in Wowza Streaming Engine 4.8.0
Breaking changes in Wowza Streaming Engine 4.8.0

If you're updating from a version of Wowza Streaming Engine earlier than 4.8.0, the following file and library changes break functionality if you don't take steps to address them.

Wowza Streaming Engine Java API breaking changes

Version 4.8.0 introduced the following breaking changes to the Wowza Streaming Engine Java API:

- Public methods removed from IPushPublish and all core classes implementing IPushPublish:
  - isParent(), isTargetedStream(), handleParentStreamDisconnet()
- Public method removed from IPushPublishProfileUtil and all core classes implementing IPushPublishProfileUtil:
  - isTargetedStream()
- Public method removed from IPushPublishStreamMonitorClient and all core classes implementing IPushPublishStreamMonitorClient:
  - isTargetedStream()

Update to Java 9

Wowza Streaming Engine 4.8.0 is built on Java 9 (OpenJDK Java SE JRE 9.0.4), and can be used with Java versions 9-12. This enables you to use your Wowza Streaming Engine media server software with JavaSE 11, which is a long-term supported version. However, because of changes between Java versions, if your Wowza Streaming Engine workflow uses custom modules or plugins, we recommend that you test them in a non-production environment prior to updating your production media server software or Java version. For more information, see Update to Java 9.

Detailed list of changes in Wowza Streaming Engine 4.8.0

Changes since 4.7.8 release

- Fixed an issue with restarting long-running source streams that could trigger a memory leak in the MPEG-DASH and CMAF live packetizers.
- Fixed an issue that caused uninitialized memory to be returned to users if a
Fixed a Java non-heap memory leak with the creation and destruction of the thumbnailer.

- Removed the Google protobuf library.
- Fixed several security vulnerabilities in Wowza Streaming Engine Manager.
- Restored `HttpStreamerItem.getLiveStreamPacketizer()` and `HttpStreamerItem.getLiveStreamRepeater()` in the Wowza Streaming Engine Java API. These were accidentally removed in Wowza Streaming Engine 4.7.8. This restores the functionality of the ModuleStreamResolver module.
- Restored the leading forward slash in responses to `Request.getRequestURI()` in the Wowza Streaming Engine Java API. This was accidentally removed in Wowza Streaming Engine 4.7.8.
- Updated the Wowza Streaming Engine Manager Tomcat web.xml file to use the default HTTP Header Security Filter.
- Updated the XStream library to version 1.4.11.1.
- Fixed an issue with audio-only and video-only streams packaged into MPEG-TS files for HLS that resulted in improper segment duration values.
- Added support for SSL in outgoing HTTPS connections (MediaCasterConnection), such as pushing a DASH stream.
- Fixed an issue with ingesting MPEG-TS with AC3 audio when there are multiple PES packets.
- Fixed a problem with timecode rollover for long-running RTMP push publishing streams that need to send ABS timecode information.
- Fixed an issue with DVB Teletext decoding for Cyrillic option 2 and option 3 so that it now correctly resolves character selection.
- Improved error handling for the de-serialization of bad RTMP streams.
- Added support for the MutlicastInterfaceAddress in .stream files when ingesting MPEG-TS/RTP streams.
- Fixed an issue with Common Encryption (CENC) for CMAF so that streams are properly encrypted for playback over MPEG-DASH and work with Widevine DRM and Playready DRM.
- Added support for the default H.265 transcoder template in Wowza Streaming Engine Manager.

### Live Stream Encoding, Transcoding, and Transrating

- Updated libvpx to version 1.8.1 on both Linux and Windows.
- Fixed a transcoder issue with the sampling rate and channel number not being set for the Speex decoder.
- Fixed an issue with NVIDIA hardware-accelerated decoding that could result
in the delivery of incorrect or corrupt frames downstream.

- Added support for the default H.265 transcoder template in the Wowza Streaming Engine Manager interface.

**WebRTC**

- Updated WebRTC session timeouts so clients won’t disconnect as quickly when sources drop.
- Added IMediaStreamNameAliasProvider3 to resolve stream name aliases for WebRTC sessions. The following methods were added to support the use of stream name aliasing:
  - `String resolvePlayAlias(IApplicationInstance appInstance, String name, WebRTCSession webrtcSession)`
  - `String resolveStreamAlias(IApplicationInstance appInstance, String name, WebRTCSession webrtcSession)`
  - `String resolvePlayAlias(IApplicationInstance appInstance, String name, IWebSocketSession webSocket)`
  - `String resolveStreamAlias(IApplicationInstance appInstance, String name, IWebSocketSession webSocket)`
- Made the following updates to WebRTC logging, handling, and configuration:
  - Added logging for UDP sessions.
  - Added logging for UDP/TCP disconnects.
  - Updated the timeout handler so that it correctly times out inactive sessions.
  - Fixed an issue that prevented passing through codec updates for WebRTC connections in the system. Depending on your publishing clients source encoder, this could lead to frequent calls to IMediaStreamActionNotify3’s onCodecInfoVideo method. To turn this off, you can use the webrtcCodecUpdates property (WebRTC/Properties Name: webrtcCodecUpdates Type: boolean default: true).

**HLS, Low-Latency HLS, and CMAF**

- Added support for Low-Latency HLS EXT-X-SKIP tag and _HLS_skip query on playlist requests.
- Added support for the HLS media playlist EXT-X-RENDITION-REPORT tag. The tags are included for each ABR rendition in both standard HLS CMAF and low-latency HLS CMAF media playlists.
- Added support for Low-Latency HLS segment HOLD-BACK attribute on EXT-X-SERVER-CONTROL tag.
- Added an application property to allow enabling and control of the segment HOLD-BACK value reported in Apple HLS CMAF media playlists (location: Application.xml:Application/HTTPStreamer/Properties, name: cupertinoSegmentHoldBack, type: Double, default: none)
- Added an application property to allow control of the PART-HOLD-BACK value reported in Apple Low-Latency HLS media playlists (location: Application.xml:Application/HTTPStreamer/Properties, name: cupertinoPartHoldBack, type: Double, default: 3x CMAF chunk duration)

**Stream Targets (Push Publishing)**

- Fixed issue where stream targets unnecessarily sent IGMP join when pushing out a stream. This caused PAT and PMT errors and caused extra network traffic.
- Fixed a null pointer exception that occurred when initializing SSL configurations for some stream targets.
- Fixed an issue where Push Publish session destroy was incorrectly calling session create notification.
- Improved the performance of underlying child stream target sessions by updating the ModulePushPublish to immediately start “parent” targets (GroupManager, WowzaCDN, WowzaStreamingCloud) on application start and stop the parent targets on application stop.
- Fixed the following issues with the stream target group manager:
  - Improper handling of the playbackHost parameter for child profiles that require it, such as Akamai targets on the Akamai MSL4 network.
  - Problems arising from using a calculated bitrate as part of the output streamName when "(bitrate)" is specified in the streamName parameter. Now the source stream name is used as the fallback.
  - Failure to properly shut down the push sessions when stopping or restarting MediaCaster source streams that were the source of the transcoded streams being pushed.

**Live Stream DVR Playback (Wowza nDVR)**

- Fixed an issue with DVR recording with 48kHz audio and DASH playback that resulted in bad segment duration values in the DASH MPD.

**SRT**

- Fixed an issue with SRT streams using the Medicacaster where stream targets weren’t notified correctly of streams’ changes in status.
• Added support for setting the following SRT properties in a .stream file:
  ○ sendBufferSize
  ○ receiveBufferSize
  ○ sendBufferSizeUDP
  ○ receiveBufferSizeUDP
• Fixed an issue that could cause Wowza Streaming Engine to crash when initializing SRT.

Known issues in Wowza Streaming Engine 4.8.0

When an RTP jitter buffer is enabled for an incoming RTSP MediaCaster source, errors repeat after source is temporarily unavailable

When jitter is enabled, if an incoming RTSP MediaCaster source becomes temporarily unavailable due to network issues, the following errors appear in the logs repeatedly, even when the source stream is available again.

```java
ERROR 500 -
at com.wowza.wms.rtp.depacketizer.RTPDePacketizerWrapperPacketSorter$PacketSender.run(RTPDePacketizerWrapperPacketSorter.java:117)
```

Solution

There is no fix available at this time, but you can stop the repeated errors by manually disconnecting the stream or restarting the application or server.

For a detailed list of known issues that are still in effect, see the Wowza Streaming Engine known issues.