



# Adaptive Bitrate Streaming with Wowza Streaming Engine

[www.wowza.com](http://www.wowza.com)

---

Copyright © 2013 – 2014 Wowza Media Systems, LLC. All rights reserved.

Third-party trademarks are trademarks or registered trademarks of their respective owners.

## **Abstract**

Adaptive bitrate (ABR) streaming is a method commonly employed for adjusting media streams to available bandwidth in order to provide the best possible streaming experience. Written with the new user in mind, this paper explains why ABR is used, how Wowza Streaming Engine™ software handles ABR, the requirements for live and on-demand streams, and how to configure Wowza Streaming Engine to deliver ABR streams.

Wowza® Media Systems, the leader in media server software, provides reliable, scalable, and secure live and on-demand streaming. Wowza Streaming Engine software enables organizations of all sizes to engage millions of viewers worldwide by delivering the highest quality video and audio to any connected device.

## Introduction—Why Adaptive Bitrate?

Before high-speed Internet connections became widely available and before mobile devices could view streaming media, it was acceptable to provide two or more bitrate renditions of a media stream. Buttons on players would let users manually select the optimal rendition to play. Some players could detect the available bandwidth automatically and then play the suitable rendition.

These methods didn't always provide the best possible streaming experience. If the available bandwidth changed while receiving the stream, the user had to wait for buffering or view a low-quality stream. The only way to switch streams was to select another rendition manually.

Today, users expect high-quality streams on their smartphones, tablets, desktop computers, and Internet-enabled TVs. They expect players to adapt to changing network conditions automatically without any intervention and without buffering or skipped frames. With the proper configuration of Wowza Streaming Engine and a set of streams or files for ABR, it's possible to provide this user experience.

## How Adaptive Bitrate Works

A common misconception is that Wowza Streaming Engine determines which rendition to send to a player. Actually, the server makes all the renditions available. The player determines which rendition to play and requests the chosen rendition from Wowza Streaming Engine, which then sends the rendition that the player requests.

To deliver adaptive bitrate streaming to users, configure the player with a list of each available bitrate and size. Or the player can use a configuration delivered in a Synchronized Multimedia Integration Language (SMIL, pronounced "smile") file. A SMIL file is an XML file that describes each available rendition of a stream. Wowza Streaming Engine uses SMIL files to provide information that players need to play the best available rendition.

When a player has this information, it begins playing the best rendition that it can play based on the currently available bandwidth and, in some instances, the screen resolution. When using screen resolution to determine the optimal rendition to play, a player doesn't waste bandwidth playing a rendition that's larger than the screen. Most players make the right selection.

While a stream is playing, the player monitors the delivery rate of the stream and the available bandwidth. If it detects that the available bandwidth is less than the current bitrate, it tries to switch to a lower bitrate rendition. If it detects that more bandwidth is available, and it can play the higher bitrate rendition, it switches to the higher bandwidth. Additionally, if the screen size changes because the player changes orientation or goes to full screen mode, the player may also change renditions to suit this new option.

## Stream Requirements

Wowza Streaming Engine supports a variety of software and hardware encoders. Some encoders can't produce aligned streams, or the network connection from the encoder to the server may not be capable of sustaining enough bandwidth to deliver all of the renditions necessary for ABR streaming.

With ABR streaming, switching between renditions occurs at keyframe boundaries. Keyframes are complete frames in a video stream that don't require any information from previous or subsequent frames. For ABR streams to switch properly without interruption, each rendition must be keyframe aligned. This means that for any keyframe, all images and timecodes in the set of ABR streams must match across all renditions. The encoder that produces the streams must maintain the keyframe alignment in the streams that it sends to Wowza Streaming Engine or that it records for on-demand playback. The user must ensure that the encoder supports multiple aligned streams (some encoders don't support this feature) and is configured properly. Wowza Streaming Engine can't correct alignment problems between multiple incoming streams.

For live streams, the encoder must produce all of the renditions without overloading resources, and it must have enough bandwidth so that none of the renditions are delayed. If a delay occurs for one or more renditions, the user experience will suffer as the player waits for the delayed rendition during the switch. If the encoder must reset or restart any rendition, it should either reset all renditions or ensure that timecodes remain aligned with all other renditions.

If the encoder can't produce aligned streams or if the network between the encoder and server has insufficient bandwidth, another option available for live ABR implementations is to use Wowza Transcoder AddOn, assuming the Wowza server has adequate processor capacity. Wowza Transcoder AddOn can take a single incoming stream and produce multiple keyframe-aligned renditions for delivery as ABR streams. This AddOn can also be used in cases where the incoming stream format isn't compatible with players. In this case, the stream can be transcoded to compatible formats. These streams can also be recorded for on-demand playback.

For on-demand streams, the underlying file system must be able to deliver any of the renditions on demand without delay. Each time a stream is played, the file is accessed on the file system. For a busy server, this means there could be many concurrent read operations. To handle this type of load, the normal practice is to use a RAID file system that increases read performance.

Keyframe alignment and timely delivery are the two most important aspects of ABR streaming. If there's a problem with either, the stream won't switch properly, and users will experience buffering, frame skipping, or the stream won't play.

Other important considerations:

- The keyframe interval (the time between keyframes) should be low. A low keyframe interval allows a player to switch streams quickly. If the player has to wait too long for the next keyframe, the user experience will degrade. A setting of 1 – 5 seconds between keyframes works well.

- The audio track should match across all renditions. While some large changes in video quality can pass almost unnoticed, even a very small change in audio is noticeable. Many encoders use the same audio track across all renditions.
- Ensure that there are enough renditions so that when switching occurs, the noticeable change for the user is small. However, it's not a good idea to have too many renditions. Depending on the range of players supported, a group of 3 – 5 renditions, split evenly from lowest-to-highest bitrate, works well.
- Ensure that each rendition is playable separately by the target players. There's no point in having renditions that can't be played because of incompatible encoder settings. It's better to have a separate set of renditions for different player types than to offer an incompatible rendition to any given player. If the player tries to switch to a rendition that it can't play, the result is a distinctly negative user experience.
- Some players have special requirements. For example, Samsung Smart TV requires that all renditions have the same encoding settings regarding H.264 Profile and Level and Frame Size and that they can only differ in bitrate between renditions. For ABR streams to be compliant with the Apple App Store for delivery to iOS clients on mobile networks, there must be a rendition that's no more than 64kbps. To accommodate these kinds of manufacturer requirements, separate SMIL files can be used to target different types of users and different playback platforms. For example, there can be one SMIL file with low bitrate renditions for mobile users and a different file for desktop or set-top boxes.

## SMIL Files Configuration

The only extra configuration required in Wowza Streaming Engine to serve ABR streams to users is to make a SMIL file that describes each of the live or on-demand renditions available for playback available for reference by the player. The SMIL file can be produced using either the SMIL Files feature in Wowza Streaming Engine Manager or manually and is referenced by the player with a **smil:** media type prefix in playback URLs.

As mentioned previously, a SMIL file is an XML file that describes each of the renditions available for playback. The file name has a **.smil** file name extension. In most cases, this file will go in the Wowza server content directory. For on-demand streams, it's typically stored in the same directory as the on-demand file renditions.

In its simplest form, the SMIL file provides a stream name and bitrate information for each rendition. It can also provide frame size information and codec information for each rendition.

## Simple SMIL File Example

```
<smil>
  <head>
  </head>
  <body>
    <switch>
      <video src="hi-livestream" system-bitrate="500000"/>
      <video src="lo-livestream" system-bitrate="250000"/>
    </switch>
  </body>
</smil>
```

## SMIL File with Audio-Only Rendition

```
<smil>
  <head>
  </head>
  <body>
    <switch>
      <video src="hi-livestream" system-bitrate="500000"/>
      <video src="lo-livestream" system-bitrate="250000"/>
      <video src="lo-livestream" system-bitrate="250000">
        <param name="audioOnly" value="TRUE" valuetype="data"/>
      </video>
    </switch>
  </body>
</smil>
```

With live audio-only renditions, extra configuration is required to packetize the audio chunks for Apple HLS streaming.

## SMIL File with Codec and Frame Size Information

```
<smil>
  <head>
  </head>
  <body>
    <switch>
      <video src="mp4:sample_400.mp4" width="320" height="240" system-
bitrate="528000">
        <param name="videoCodecId" value="avc1.66.12" valuetype="data"/>
        <param name="audioCodecId" value="mp4a.40.2" valuetype="data"/>
        <param name="audioBitrate" value="128000" valuetype="data"/>
        <param name="videoBitrate" value="400000" valuetype="data"/>
      </video>
      <video src="mp4:sample_800.mp4" width="640" height="480" system-
bitrate="928000">
        <param name="videoCodecId" value="avc1.77.31" valuetype="data"/>
        <param name="audioCodecId" value="mp4a.40.2" valuetype="data"/>
        <param name="audioBitrate" value="128000" valuetype="data"/>
        <param name="videoBitrate" value="800000" valuetype="data"/>
      </video>
    </switch>
  </body>
</smil>
```

## Example Playback URLs

### Apple HLS

`http://[wowza-ip-address]:1935/myApplication/smil:myStream.smil/playlist.m3u8`

### Adobe HDS

`http://[wowza-ip-address]:1935/myApplication/smil:myStream.smil/manifest.f4m`

### Microsoft Smooth Streaming

`http://[wowza-ip-address]:1935/myApplication/smil:myStream.smil/Manifest`

### MPEG-DASH Streaming

`http://[wowza-ip-address]:1935/myApplication/smil:myStream.smil/manifest.mpd`

## Transcoder AddOn Configuration

If you use Wowza Transcoder AddOn to produce ABR streams from a single incoming live stream, the ABR streams are grouped into a logical group (called a "Stream Name Group") using the transcoder templating system. A Stream Name Group is defined in the transcoder template and is referenced by the player with an **ngrp**: media type prefix in playback URLs.

### Stream Name Group Example Playback URLs

#### Apple HLS

`http://[wowza-ip-address]:1935/live/ngrp:myStream_all/playlist.m3u8`

#### Adobe HDS

`http://[wowza-ip-address]:1935/live/ngrp:myStream_all/manifest.f4m`

#### Microsoft Smooth Streaming

`http://[wowza-ip-address]:1935/live/ngrp:myStream_all/Manifest`

#### MPEG-DASH Streaming

`http://[wowza-ip-address]:1935/live/ngrp:myStream_all/manifest.mpd`

You can configure Wowza Streaming Engine software to generate a SMIL file that references the ABR streams in a transcoder Stream Name Group. The SMIL file format is identical to the text version, except that it has full codec and frame size information for each transcoded or transrated rendition.

## Player Configuration

In most cases, configuring a player is simply a matter of providing it with the URL of the SMIL file or Stream Name Group to use. From there, the player detects that the stream is an ABR stream and plays it as such.

### JW Player and Flowplayer

JW Player 6 and Flowplayer require a special SMIL file that has extra information needed for RTMP ABR playback (RTMP dynamic streaming). For JW Player to switch correctly, it needs the frame width and system bitrate for each rendition.

Flowplayer uses the same format, a SMIL file, as JW Player 6.

```
<smil>
  <head>
    <meta base="rtmp://example.com:1935/vod/_definst_" />
  </head>
  <body>
    <switch>
      <video src="mp4:sample_400.mp4" width="320" height="240" system-
bitrate="528000" />
      <video src="mp4:sample_800.mp4" width="640" height="480" system-
bitrate="928000" />
    </switch>
  </body>
</smil>
```

### Other Players

For other players, refer to the player documentation.

## Summary

Wowza Streaming Engine supports ABR for live and on-demand playback, and Wowza Streaming Engine administrators should use ABR for the best possible streaming experience. Wowza Transcoder AddOn allows administrators with non-ABR encoders, or with limited bandwidth between encoder and server, to create live ABR streams when running Wowza Streaming Engine on sufficiently powerful hardware.

Keyframe alignment and timely delivery are the two most important aspects of ABR streaming. If there's a problem with either, then the stream will fail to switch properly and users will experience buffering or skipping or, worst case, the stream will fail to play.

Wowza Streaming Engine makes all renditions of an ABR stream available to the player, but it's the player that determines which rendition to play and when to switch to a different rendition. The server responds with the rendition that's requested.

## References

[How to do adaptive bitrate streaming](#)

[How to set up and run Wowza Transcoder AddOn for live streaming](#)

[How to create and play SMIL files with streams created by Wowza Transcoder](#)

[How to use JW Player with adaptive bitrate streaming from Wowza Streaming Engine](#)

[How to use Flowplayer with Wowza Streaming Engine](#)