



# Adaptive Bitrate Streaming with Wowza Media Server

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

## **Abstract**

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

Wowza Media Systems, the leader in media server software, provides reliable, scalable, and secure live and on-demand streaming. Wowza Media Server 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 did not 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, end 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 Media Server and a set of streams or files for ABR, it is possible to provide this user experience.

## How Adaptive Bitrate Works

A common misconception is that Wowza Media Server 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 Media Server, which then sends the rendition that the player requests.

To deliver adaptive bitrate streaming to end 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 Media Server uses SMIL files to provide information that various players need to play the best available stream.

When the 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, the player does not waste bandwidth playing a rendition that is larger than the screen. Most players make the right selection, but there are exceptions.

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 will attempt to switch to a lower bitrate rendition. If it detects that more bandwidth is available, and it can play the

higher rendition, it will switch to the higher bandwidth. Additionally, if the screen size changes because the player changes orientation or goes to full-screen, the player may also change renditions to suit this new option.

## **Stream Requirements**

Wowza Media Server supports a variety of software and hardware encoders. Some encoders are not capable of producing 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 the video stream that do not 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 in the set of ABR streams and the time codes must match across all renditions. The encoder used to produce the streams must maintain the keyframe alignment in the streams that it sends to Wowza Media Server or that it records for on-demand playback. The user must ensure that the encoder supports multiple aligned streams (some encoders do not support this feature) and is configured properly. Wowza Media Server cannot 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 needs to reset or restart any rendition, it should either reset all renditions or ensure that time codes remain aligned with all other renditions.

If the encoder cannot produce aligned streams or if the network between the encoder and server has insufficient bandwidth, another option available for live ABR implementation is to use the Wowza Transcoder AddOn, assuming the Server has adequate processor capacity. The 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 is not compatible with the players. In this case, the stream can be transcoded to compatible formats. These streams can also be recorded for on-demand playback.

For any on-demand streams, the underlying file system must be capable of delivering 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 is a problem with either, the stream will fail to switch properly, and end users will experience buffering, frame skipping, or the stream will fail to play.

Other important considerations:

- The keyframe interval, that is, the time between keyframes, should be low. A low keyframe interval will allow the player to switch streams quickly. If the player has to wait too long for the next keyframe, the end 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 end user is small. However, it is 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 is no point in having renditions that cannot be played because of incompatible encoder settings. It is 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 cannot play, playback may fail completely, resulting in 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 may 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 is 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.

## Wowza Server Configuration

The only extra configuration required on Wowza Media Server to serve ABR streams to end users is to make the SMIL file available for reference by the player.

Using the Wowza Transcoder AddOn, the SMIL file can be produced automatically by the transcoder when it starts. While the format is identical to the text version, the automatically generated SMIL file is referenced in the player with an `ngpr:` prefix. If this feature in the Wowza Transcoder AddOn is not being used, the file must be produced manually and is referenced in the player with a `smil:` prefix.

As mentioned previously, a SMIL file is an XML file that describes each of the renditions available for playback. For manually created files, the file name has a .smil filename extension. In most cases, this file will go in the Application content directory, or, for on-demand streams, in the same directory as the on-demand files.

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>

```

The Wowza Transcoder AddOn produces SMIL files with full codec and frame size information. The SMIL file produced is called a Name Group. It is referenced with an ngrp: prefix.

### Name Group Example

http://<wowza-ip>:1935/live/ngrp:mystream\_all/playlist.m3u8

For more information about SMIL files and the Wowza Transcoder AddOn, see [How to create and play SMIL files with streams created by Wowza Transcoder](#).

## Player Configuration

In most cases, configuring a player is simply a matter of providing the player with the URL for the SMIL file it will use. From there, the player will detect that the stream is an ABR stream and play it as such.

### Example URLs

#### Apple HLS

http://<wowza-ip>:1935/live/smil:mystream.smil/playlist.m3u8

## Adobe HDS

<http://<wowza-ip>:1935/live/smil:mystream.smil/manifest.f4m>

## Microsoft Silverlight

<http://<wowza-ip>:1935/live/smil:mystream.smil/Manifest>

JW Player and Flowplayer require a special SMIL file for RTMP ABR playback that contains extra information needed for RTMP playback. Wowza Media Server is capable of producing the required file from the same SMIL file used for HTTP playback. For JW Player to switch correctly, it needs the frame width as well as the system bitrate for each rendition.

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

## JW Player 6 and Flowplayer

<http://<wowza-ip>:1935/vod/smil:sample.smil/jwplayer.smil>

```
<smil>
  <head>
    <meta base="rtmp://example.com:1935/live/_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>
```

## JW Player 5

JW Player 5 used an .rss file format.

<http://<wowza-ip>:1935/vod/smil:sample.smil/jwplayer.rss>

```
<rss version="2.0" xmlns:media="http://search.yahoo.com/mrss/"
xmlns:jwplayer="http://developer.longtailvideo.com/">
  <channel>
    <item>
      <title>sample.smil</title>
      <description>sample.smil</description>

      <jwplayer:streamer>rtmp://example.com:1935/live/_definst_</jwplayer:streame
r>

      <media:group>
```



```
width="320"/>
<media:content url=" mp4:sample_400.mp4" bitrate="528"
width="640"/>
</media:group>
</item>
</channel>
</rss>
```

For other players, please refer to the player documentation.

## Summary

Wowza Media Server supports ABR for live and on-demand playback, and Wowza Media Server administrators should use ABR for the best possible streaming experience. The 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 Media Server on sufficiently powerful hardware.

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

Wowza Media Server makes all renditions of an ABR stream available to the player, but it is the player that will determine which rendition to play and when to switch to a different rendition. The server responds with the rendition that is requested.

## References

[How to do adaptive bitrate streaming](#)

[How to debug encoder multi-bitrate keyframe alignment](#)