This guide describes many relevant concepts and requirements that are important to understand when you start using nDVR, which provides the ability to record a live stream with Wowza Streaming Engine™ media server software while simultaneously allowing users to play or pause a live stream, rewind to a previously recorded point, and resume viewing at the current live point. nDVR works with the HDS, HLS (Cupertino), Smooth Streaming, and MPEG-DASH protocols. Custom configuration enables you to control the archive strategy and availability of your recorded streams. nDVR requires that the audio and video in an incoming stream be aligned.

Setup for client playback of recorded streams is similar to playback of live streams from Wowza Streaming Engine and can be easily added to your existing server configuration.

**Notes:**
- This article applies to Wowza Streaming Engine and Wowza Media Server™ software version 3.5 and later.
- Wowza Streaming Engine 4.7.6 or later is required to use nDVR for MPEG-DASH streaming.

**Contents**

**Streaming concepts**
- Stream types
- Supported media
- Supported protocols and players
- Adaptive bitrate delivery
- Playback URLs

**Installation and configuration**
- Licensing nDVR
- Application configuration
- Content storage
- Archive strategy
- DVR window duration

**Best practices**
Streaming concepts

Wowza nDVR can be configured to support multiple workflows and supports:

- **Stream types** - The **live** stream type as well as the **liverepeater-edge** stream type, which is used in live repeater (origin/edge) configurations. To set up a Wowza nDVR live stream repeater, see [Set up a Wowza Streaming Engine live stream repeater for Wowza nDVR](#).

- **Video and audio codecs**
  - **Video**: H.264, video-only streams
  - **Audio**: AAC, MP3, audio-only streams

- **Protocols** - A Wowza media server can deliver H.264-encoded streams using the following HTTP streaming protocols:
  - Adobe HTTP Dynamic Streaming (Adobe HDS)
  - Apple HTTP Live Streaming (Apple HLS)
  - Microsoft Smooth Streaming
  - (Wowza Streaming Engine 4.7.6 or later) Dynamic Adaptive Streaming over HTTP (MPEG-DASH)

- **Players** - To play recorded streams, you can use Wowza™ Player, JW Player, or develop your own custom player. You can also use the following example players, which are included in your Wowza media server software installation, to test playback of your...
recorded files:
- **Microsoft Silverlight** ([install-dir]/examples/LiveDVRStreaming/SilverlightPlayer/player.html)
- **Strobe**: ([install-dir]/examples/LiveDVRStreaming/FlashHTTPPlayer/player.html)
- **MPEG-DASH**: ([install-dir]/examples/LiveVideoStreaming/MPEGDASHPlayer/player.html)

**Note**: The supported player technologies must also support your media:
- Microsoft Silverlight requires that both audio and video be present for playback. For live streaming only, the keyframe frequency must be between 1 and 4 seconds (2 seconds is recommended). Anything greater than 4 seconds may cause playback to freeze.
- Apple iOS devices require that video is encoded using H.264 format and AAC or MP3 stereo audio. Profile and level support varies depending on the device, see the Apple technical specifications for the devices that you plan to target.

- **Adaptive bitrate delivery** - When using Wowza nDVR, you can leverage Synchronized Multimedia Integration Language (SMIL) files for adaptive bitrate delivery. The SMIL workflow requires an encoder that can generate multiple bitrate streams from the same source with properly aligned keyframes.

  **Important**: The streams used in adaptive bitrate streaming must have aligned keyframes. Wowza nDVR creates chunks based on the keyframe alignment and the adaptive bitrate switching occurs on chunk boundaries.

- Wowza nDVR can be configured using Wowza Streaming Engine Manager or via the Java API.

**Playback URLs**

**Notes:**
- In the following examples, **[wowza-ip-address]** is the IP address or domain name of the server running Wowza Streaming Engine.
- Playback URLs can include uppercase ‘DVR’ or lowercase ‘dvr’. Wowza Streaming Engine application names are case-sensitive.
- The smil: prefix is used when playing back a group of stream renditions using SMIL files. To see an example SMIL file, see **Stream adaptive bitrate content with Wowza Streaming Engine**. If you’re using SMIL files to create adaptive bitrate, use the adaptive bitrate playback URL formats, below.
Use the following URL formats to play your nDVR recorded streams over the supports HTTP protocols:

**HDS (Flash playback):**

Double-click [install-dir]/examples/LiveDVRStreaming/[playerType]/player.html, enter the playback URL using one of the formats below and then click **Connect**.

- **Single bitrate:** http://[wowza-ip-address]:1935/dvr/myStream/manifest.f4m?DVR
- **Adaptive bitrate (SMIL file):** http://[wowza-ip-address]:1935/live/smil:myStream.smil/manifest.f4m?DVR

**HLS (Cupertino streaming, iOS playback):**

Enter the playback URL, using one of the formats below, into the Safari web browser on the device:

- **Single bitrate:** http://[wowza-ip-address]:1935/dvr/myStream/playlist.m3u8?DVR
- **Adaptive bitrate (SMIL file):** http://[wowza-ip-address]:1935/live/smil:myStream.smil/playlist.m3u8?DVR

**Smooth Streaming (Silverlight players):**

Double-click [install-dir]/examples/LiveDVRStreaming/SilverlightPlayer/player.html, enter the playback URL using one of the formats below, and then click **Connect**.

- **Single bitrate:** http://[wowza-ip-address]:1935/dvr/myStream/manifest?DVR
- **Adaptive bitrate:** http://[wowza-ip-address]:1935/live/smil:myStream.smil/manifest?DVR

**MPEG-DASH:**

Double-click [install-dir]/examples/LiveDVRStreaming/MPEGDASHPlayer/player.html, enter the playback URL using one of the formats below, and then click **Connect**.

- **Single bitrate:** http://[wowza-ip-address]:1935/dvr/myStream/manifest.mpd?DVR
- **Adaptive bitrate:** http://[wowza-ip-address]:1935/live/smil:myStream.smil/manifest.mpd?DVR
Installation and configuration

Licensing nDVR

**Perpetual and Subscription licenses for Wowza Streaming Engine** provide for unlimited connections to the Wowza Streaming Engine instance and enable use of the nDVR technology that’s integrated with each licensed instance.

**Note:** If you purchased a license for Wowza Streaming Engine or Wowza Media Server before January 1, 2015, contact [sales@wowza.com](mailto:sales@wowza.com) to learn more about how to license nDVR.

Application configuration

To configure nDVR functionality for live streaming, see Set up and run nDVR in Wowza Streaming Engine.

Content storage

When nDVR starts recording a live stream, a content storage directory is created. The content storage directory location is defined by the `DVR/StorageDir` property in the `Application.xml` file. The default location that nDVR writes to is `[install-dir]/dvr/[[app-name]]/_definst_/[[streamName]].[n]`.

If the application name `[[app-name]]` is `live` and the stream name `[[streamName]]` is `myStream`, with the default settings in `Application.xml`, the storage directory will be located at `[install-dir]/dvr/live/_definst_/myStream.0`.

If you set the `DVR/StorageDir` property to `E:/myStorage` and your stream name is `myStream`, the storage directory will be located at `E:/myStorage/dvr/_definst_/myStream.0`.

Directory structure

When nDVR records a live stream, it creates a new folder in the content storage directory every 600 seconds (10 minutes) by default. The folder naming convention is `HHHH_MM_SS` where `H` = hours, `M` = minutes, `S` = seconds. As an example, if nDVR records a live stream for 1
hour and 10 minutes, the content storage directory will contain the following folders:

- 0000_00_00
- 0000_10_00
- 0000_20_00
- 0000_30_00
- 0000_40_00
- 0000_50_00
- 0001_00_00
- 0001_10_00
- 0001_20_00
- manifest.txt

Each time-based folder contains .m4fa (audio information) and .m4fv (video information) files and a .txt (manifest) file. If the incoming stream sends metadata, a .m4fm (metadata) file is also created. These files are based on the MPEG-4 format and together represent a chunk of recorded audio and video content that can be used for playback of your recorded stream. Using the folder named 0000_00_00 as an example, the files that describe the first chunk are:

- A-0000_00_00_000.m4fa
- dvrManifest_0000_00_00.txt
- M-0000_00_00_000.m4fm
- V-0000_00_00_000.m4fv

The files in a time-based folder are named using the folder-naming convention, with A (audio), V (video), or M (metadata) pre-pended to the filename and a timestamp value (in milliseconds) appended to the filename (A-HHHH_MM_SS_mmm).

**Important:** Don’t delete or move any of these files or playback errors will occur.

**Archive strategy**

You can configure an archiving strategy that tells the DVR store what to do with an old stream when a new stream of the same application instance and stream name starts (for example, when the encoder restarts). The strategy is controlled by the `DVR/ArchiveStrategy` property in `Application.xml`. You can set this property to one of the following values:

- **append:** Specifies that nDVR will append to the existing recording. This is the default setting.
- **delete**: Specifies that the previous recording will be deleted and replaced with the new recording.

- **version**: Specifies that a new content storage directory will be created. If the stream name is `myStream`, the first recording is written to `[install-dir]/dvr/[application-name]/myStream.0`, the second recording is written to a directory named `myStream.1`, and so on. Whenever there’s a recording of the same stream name, the previous recording is preserved and not deleted or overwritten.

**DVR window duration**

You can specify the duration (in seconds) of material available from the DVR store for DVR playback. The duration is a floating DVR window that starts at the current live point and works backwards. Older data that falls outside of the specified duration value is purged from the file system. The duration value is controlled by the **DVR/WindowDuration** property in `Application.xml`. The default **WindowDuration** value of 0 means that there’s no DVR window applied and all data is available for DVR playback.

**Best practices**

**Aligned audio and video**

Input streams with short keyframe intervals and aligned audio and video work best. nDVR expects that the audio and video in an incoming live stream is aligned. When the audio and video are out of alignment beyond the allowable tolerance, you’ll see the following WARN message in `[install-dir]/logs`:

```
```

If you have out-of-alignment issues, you can set properties in `Application.xml` to try to compensate for them; however, nDVR can’t fix these problems. See **dvrAllowableAVPacketDelta** in Configure advanced properties for nDVR. Note that this property is meant to help in situations where audio and video aren’t closely aligned. If you already have an out-of-alignment issue, increasing the default value is likely to cause more problems. Increasing this value will increase the number of chunks that aren’t recorded, which may cause the overall quality to be unacceptable.
Keyframe size

HTTP streaming has a latency of 3 chunks, which is why smaller keyframe sizes are often recommended. Smaller keyframe sizes result in more data in the playlist. Consider playlist size and the frequency at which data is being sent from the server when determining your encoding settings.

Dependable live stream

nDVR requires a reliable live stream. When troubleshooting nDVR, first test that your live workflow is successful without nDVR enabled.

Recording length

nDVR has a recording limit of 30 hours. In real-world deployments, serving playlist data this large won’t work. Also, if a problem occurs during the recording (disk full, network, out-of-memory, etc.), the entire recording could be comprised. An extremely large manifest file may also cause performance issues because the entire manifest file must be loaded into memory. Select a strategy that records a reasonable length so that small manifest files can be loaded and unloaded quickly. A longer recording is a viable workflow if only a small part of the recording is specified for playback. For example, a 24-hour recording with a 1- or 2-hour playback window results in a much smaller playlist size vs. a playlist for the entire 24-hour recording. The playlist will be repeatedly requested every few seconds (based on keyframe size); therefore, an extremely large playlist size is problematic for the player and should be avoided. For more information about how to request a portion of your recording for playback, see Use nDVR playlist request with the Wowza Streaming Engine Java API.

OSMF player recommendations

We recommend using OSMF version 2.0. Users have reported buffering issues when using earlier versions. The Strobe example player installed with Wowza Streaming Engine and Wowza Media Server (version 3.5 and later), located in [install-dir]/examples/LiveDVRStreaming/FlashHTTPPlayer/player.html, uses the default values for the buffering parameters.

Built-in safeguards

nDVR has multiple built-in safeguards built that require no additional configuration.

If nDVR detects an invalid chunk, it will try to fix it by recalculating if the clock was reset. Or, if it can’t fix it, the chunk will be thrown out. For example, an invalid chunk could be caused by a problematic input stream where the video gets far ahead of the audio. If this occurs, it’s reported in the logs with warning messages like the following:
FAQs

What's an example of flexible viewer start time with a live event?

nDVR is suited for workflows where you want to record a live stream and have playback flexibility. If you have a one-time live event, or perhaps a recurring event such as a house of worship, you can broadcast a live event and make it available for real time viewing and accommodate late arrivals:

1. Users start viewing from the start time of the live stream, rewind to hear a particular part of the video again, and then continue streaming from the current live point.
2. If users join late, they can start viewing after the time the live stream has started, but start from the beginning and continue until the live stream ends. In this scenario, users can start at any time after the live stream has started--1 minute later or 1 hour later.
3. Users can start viewing the recording after the live stream has stopped. In this scenario, playback is streaming the previously recorded content as video on demand (VOD).

When do I need to use nDVR for recording?

nDVR is suited for workflows where you want users to be able to pause, resume, or rewind during a live recording. Other recording methods supported by Wowza Streaming Engine are suited for workflows where you want to create a recording contained in one MP4 or FLV file for playback as video on demand only with no playback during the time of the recording.

Do I need a separate Wowza Streaming Engine application if I already have my live workflow set up?

No, it’s not necessary. You can easily integrate nDVR with your existing live workflow. For instructions, see Set up and run nDVR in Wowza Streaming Engine. If you have multiple workflows such as live streams that won’t recorded or if you need to apply different DVR recording parameters for different input streams, then you can set up an application for each
How do I play a recorded video after the live event and recording is finished?
Use the same Wowza Streaming Engine application and keep the StreamType set to live, even though the live stream is no longer being published. The URL is the same as if you were playing back the live video. See “Playback” in Set up and run nDVR in Wowza Streaming Engine for sample playback URLs.

How do I specify playback start and end times?
For more information about how to control playlist requests, which have a start time and an optional end time, see Use nDVR playlist request API.

What happens to my recording if my live stream disconnects?
 nDVR resumes the recording after the live stream reconnects. The streamTimeout property specifies the amount of time that Wowza Streaming Engine will wait for packets before it stops recording. The default property value is tuned to account for when encoders disconnect and then restart. For more information, see streamTimeout in Configure advanced properties for nDVR.

Can I play recorded streams from nDVR over RTSP, RTP, RTMP?
No. Playback is currently supported over HLS (Cupertino), MPEG-DASH, HDS, and Smooth Streaming.

Can I play a previous version of a recorded stream?
If the ArchiveStrategy property value is set to version, then you can specify which version you want to play in the playback URL. For example, the playback URL

http://localhost:1935/live/myStream.3/playlist.m3u8?dvr

specifies that "version 3" of the recorded stream name myStream can be streamed to an Apple iOS device from the Wowza Streaming Engine live application.

To learn more about how to control your recordings, see Use nDVR recording with the Wowza Streaming Engine Java API.

What does dvrstreamingpacketizer do?
This packetizer takes a live incoming stream and saves the stream data through the nDVR recording process. As part of the configuration required to enable nDVR, add the
**dvrstreamingpacketizer** property value to the **Streams/LiveStreamPacketizers** container in **Application.xml**. You must also add additional live stream packetizers property values (cupertinostreamingpacketizer, smoothstreamingpacketizer, sanjosestreamingpacketizer, mpegdashstreamingpacketizer) to the container to enable live streaming to supported players/devices.

**What happens if I don't set a store location?**

nDVR sets the default store location as:

```bash
In a live repeater (origin/edge) configuration, this property must be left empty. For more information, see Set up live stream repeater for use with nDVR (origin/edge).
```