nDVR in Wowza Streaming Engine™ media server software lets you record a live stream while simultaneously allowing viewers to play or pause the live stream, rewind to a previously recorded point, and resume viewing at the current live point when playing streams on any device or player that support the HLS (Cupertino), MPEG-DASH, HDS, and Smooth Streaming protocols.

**Notes:**
- This article is for use with 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.

**Tutorial**

**About nDVR**

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.

**Configure an application in Wowza Streaming Engine Manager**

This section provides the steps for setting up an application for live streaming with nDVR using Wowza Streaming Engine Manager.

1. Click the **Applications** tab and then click **Add Application** in the contents panel.
2. On the **Add Application** page, click the **Live** button to select that application type.
3. In the **New Application** dialog box, enter a name, such as **dvr**, and then click **Add**.
4. Expand the dvr application in the contents panel, click nDVR, and then click Enable nDVR.

**Note:** Wowza Streaming Engine must be properly licensed to enable nDVR. For details, see About live stream nDVR.

5. Click Restart Now to apply the changes.

The Status field will show that nDVR is enabled.

6. Click Edit to configure nDVR settings for the application. You can use these settings to customize your application configuration to control nDVR behavior.

**DVR store directory**

Specifies the top-level folder where DVR streams are stored. This can contain an explicit path (for example, C:/myDvr) or environment variable (for example, ${com.wowza.wms.context.VHostConfigHome}). The following environment variables are supported.

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>${com.wowza.wms.AppHome}</td>
<td>Application home directory</td>
</tr>
<tr>
<td>${com.wowza.wms.ConfigHome}</td>
<td>Configuration home directory</td>
</tr>
<tr>
<td>${com.wowza.wms.context.VHost}</td>
<td>Virtual host name</td>
</tr>
<tr>
<td>${com.wowza.wms.context.VHostConfigHome}</td>
<td>Virtual host config directory</td>
</tr>
<tr>
<td>${com.wowza.wms.context.Application}</td>
<td>Application name</td>
</tr>
<tr>
<td>${com.wowza.wms.context.ApplicationInstance}</td>
<td>Application instance name</td>
</tr>
</tbody>
</table>

The DVR store directory is a folder structure containing the application name, the
application instance name, the stream name, and a version number. For example, if **DVR Store Directory** is set to `C:/myDvr`, and the incoming live stream name is `myStream`, streams for the `dvr` application would be stored in `C:/myDvr/dvr/_definst_/myStream.0`, `C:/myDvr/dvr/_definst_/myStream.1`, and so on.

**Streaming options**

Specifies how an incoming live stream can be played by indicating whether the `?DVR` query parameter must be appended to playback URLs.

**Recording options**

Select **Start recording on startup** to automatically begin recording a live stream that's published to the application for DVR when the application starts. If you want to manually control when the recording starts, you must use the [nDVR recording API](#).

**Archive method**

Specify what to do with a recording if a new stream with the same stream name is published to this application. The default option (**append**) appends the new stream to the existing recording and is best to use as a failsafe mechanism to account for disruptions in the current live stream (such as encoders restarting). The **delete** option deletes the previous recording and starts a new one. The **version** option starts recording the new stream to a new folder in the DVR store directory.

**DVR window duration**

Specify the amount of recorded content in the DVR store that's available for DVR playback.

**Note:** You can record up to 30 hours of content for DVR playback; however, you may encounter performance and playback issues if you make longer durations of content available for playback. For best practices guidance, see [Recording length](#).

- To specify that a live stream can be played using the DVR playback controls in a player, select **DVR streaming only**. This option means that the `?DVR` query parameter must be appended to playback URLs.
- To specify that a live stream can be played with or without DVR playback controls, select **Live and DVR streaming**. This option means that the live stream is available for playback with or without the `?DVR` query parameter appended to playback URLs.
URLs. If ?DVR isn’t appended to the playback URL, the live stream can still be played but DVR playback isn’t supported.

- To make the entire recording available for playback, select **All material available**. This option enables viewers who join the live stream in-progress to rewind and watch the stream from the beginning. This option also preserves the live stream recording after the stream ends.
- To make only part of the live stream available for playback, select **Use window duration** and then specify the duration in **Hours:Minutes:Seconds**. The minimum supported duration value is 60 seconds. This option enables viewers who join the live stream in-progress to rewind the stream for the duration that you specify and watch from that point forward. The duration window is a “floating window” that always ends at the current live point. Recorded data that falls outside this window is purged from the DVR store.

7. Click **Save**, and then click **Restart Now** to apply changes to nDVR settings for the application.

**Note:** Additional nDVR configuration properties are documented in [Configure advanced properties for Wowza nDVR](#).

**Configure an application in Application.xml**

This section provides the steps for setting up an application for live streaming with DVR capabilities by configuring an application’s Application.xml file. This is an alternative to configuring the application in Wowza Streaming Engine Manager.

1. Create the application folder `[install-dir]/applications/dvr`
2. Create the configuration folder `[install-dir]/conf/dvr` and copy `[install-dir]/conf/Application.xml` to this new folder.
3. Edit the newly copied `Application.xml` file and make the following changes (some of these settings may already be present):

   a. Set the `/ property to:

      ```
      ...
      ```

   b. Set the `/ property to:

      ```
      ...
      ```

   This setting enables DVR content to originate from the recorded store for this Wowza Streaming Engine instance and be streamed to any HTTPStreamer. In addition to DVR streaming, you can enable live streaming for the application by
adding additional packetizers. For example:

c. Set the / property to:

d. Set the / property to:

e. Set the HTTPStreamers property to the streaming protocols that you want to stream to. For example:

4. Customize additional nDVR properties for your application in Application.xml to control the archive method and availability of your recorded streams.

- Specify the top-level folder where DVR streams are stored. See DVR store directory.

- Specify what to do with a recording if a new stream with the same stream name is published to this application. See Archive method.

- Specify the amount of recorded content (in seconds) in the DVR store that’s available for DVR playback. To make all of the recorded content available for DVR playback, enter 0. See DVR window duration.

Additional nDVR configuration properties are documented in Configure advanced properties for Wowza nDVR.

5. Restart Wowza Streaming Engine to apply the changes.

Publish the stream to Wowza Streaming Engine

Publish a live stream from your encoder or camera to the dvr application by following the instructions in Connect a live source to Wowza Streaming Engine.

If you selected Start recording on startup, the dvr application will automatically record the stream when the encoder begins sending it, and users can seek to earlier segments of the live stream. If you want to manually control when the recording starts, you must use the Wowza nDVR recording API.

Publish the stream to Wowza Media Server
If you're using Wowza Media Server, do the following:

1. Using a text editor, edit the `[install-dir]/conf/admin.password` file and add a user name and password that will be used to start and stop publishing of the stream. Following is an example of the file with the username `myUser` and password `myPassword`:

   ```
   # Admin password file (format [username][space][password])
   # username password
   myUser myPassword
   ```

2. Restart the server.
3. Publish the live stream from your encoder or camera to the `dvr` application on the server.
4. Start the stream in Stream Manager:

   a. Enter the following URL in a web browser:
      ```
      http://[wowza-ip-address]:8086/streammanager
      ```
   b. In the Security dialog box, enter the user name and password from the `admin.password` file.
   c. In the Stream Manager webpage, click the `[start-receiving-stream]` link just below the `dvr` application folder.
   d. In the Start Receiving Stream dialog box, for MediaCaster Type, select `rtp-record`.
   e. In the Stream Name field, enter the stream name used by the encoder or camera.
   f. Click OK.

When the stream starts, it will also be recorded to the DVR store.

Playback

Use the following playback URLs for testing and playing your streams.

Notes:

- In the following examples, `[wowza-ip-address]` is the IP address or domain name of the server running Wowza Streaming Engine.
- If your player is secure (HTTPS), your stream URL must also be secure.

HDS playback URLs

- **DVR**: To play the live stream with DVR capabilities, use the following URL format:
http://[wowza-ip-address]:1935/dvr/myStream/manifest.f4m?

Live: To play the live stream without DVR capabilities, you must select the Live and DVR streaming option for the application. Then you can use a URL with the following format:
http://[wowza-ip-address]:1935/dvr/myStream/manifest.f4m

HLS (Cupertino) playback URLs

- DVR: To play the live stream with DVR capabilities, use the following URL format:
  http://[wowza-ip-address]:1935/dvr/myStream/playlist.m3u8?
- Live: To play the live stream without DVR capabilities, you must select the Live and DVR streaming option for the application. Then you can use a URL with the following format:
  http://[wowza-ip-address]:1935/dvr/myStream/playlist.m3u8

Smooth Streaming playback URLs

- DVR: To play the live stream with DVR capabilities, use the following URL format:
  http://[wowza-ip-address]:1935/dvr/myStream/Manifest?
- Live: To play the live stream without DVR capabilities, you must select the Live and DVR streaming option for the application. Then you can use a URL with the following format:
  http://[wowza-ip-address]:1935/dvr/myStream/Manifest

MPEG-DASH playback URLs

- DVR: To play the live stream with DVR capabilities, use the following URL format:
  http://[wowza-ip-address]:1935/dvr/myStream/manifest.mpd?
- Live: To play the live stream without DVR capabilities, you must select the Live and DVR streaming option for the application. Then you can use a URL with the following format:
  http://[wowza-ip-address]:1935/dvr/myStream/manifest.mpd

Playback in Wowza Streaming Engine Manager

Note: The test players in Wowza Streaming Engine Manager are also available online on our Wowza Test Players webpage.

DVR playback is only supported by the Adobe HDS, Apple HLS, Microsoft Smooth Streaming, and MPEG-DASH test players. To access the test players in Wowza Streaming Engine Manager, click Test Players in the upper-right corner of the dvr application page. Then select the
respective player (Adobe HDS, Apple HLS, MS Smooth, or MPEG-DASH), enter one of the playback URLs as the Stream URL, and click Start.

For Apple HLS streams, your browser must support HLS and/or HTML5. You can also test Apple HLS by using the Mobile tab in the Test Players window and emailing the URL to your iOS device or by entering the playback URL directly in Safari or QuickTime 10 on a computer running OS X 10.6 (Snow Leopard) or later.

Playback with example players

This section describes example players that you can use to test your live stream with DVR capabilities. The example players are installed with Wowza Streaming Engine. You can also download the example players to another computer and extract them from the compressed (zipped) folder.

**Download example players for Wowza Streaming Engine**

-OR-

**Download example players for Wowza Media Server**

To use the example players, do the following.

1. Open `[install-dir]/examples/LiveDVRStreaming/[playerName]/player.html`. Where:
   - `[install-dir]` is the location of your Wowza Streaming Engine software installation or the location of the downloaded example players.
   - `[playerName]` is one of the following: FlashHTTPPlayer, SilverlightPlayer, or MPEGDASHPlayer.

   **Note:** For Apple HLS playback, open the `[install-dir]/examples/LiveVideoStreaming/IOSMacOSX/player.html` in the Safari web browser.

2. Enter the playback URL as the Stream URL, and then click Connect.

Using the example players remotely

To test the stream from a remote computer, copy the example player folder `[install-dir]/examples/LiveDVRStreaming/FlashHTTPPlayer`, `[install-dir]/examples/LiveDVRStreaming/SilverlightPlayer`, or `[install-dir]/examples/LiveDVRStreaming/MPEGDASHPlayer`. Where:

- `[install-dir]` is the location where you extracted the example players.
dir]/examples/LiveDVRStreaming/MPEGDASHPlayer) to a web server, and then connect to the corresponding playback URL:

- **HDS:** http://[web-server]/FlashHTTPPlayer/player.html
- **Smooth Streaming:** http://[web-server/SilverlightPlayer/player.html
- **MPEG-DASH:** http://[web-server/MPEGDASHPlayer/player.html

## Troubleshooting

### Logging

nDVR uses the Wowza Streaming Engine log files in `[install-dir]/logs`. For more information about logging capabilities, see the [Wowza Streaming Engine User Guide](#) (PDF).

Running Wowza Streaming Engine in standalone mode enables you to view these messages in real-time from the console window. For a list of messages, scenarios that may cause them, and suggestions for resolution, see [Troubleshoot error messages in Wowza Streaming Engine](#).

Logging messages in `wowzastreamingengine_access.log` indicate if Wowza Streaming Engine and nDVR are configured properly. If so, the logs will show nDVR being initialized and configuration information being read from `Application.xml`. All messages, including warnings and errors, are logged in the `wowzastreamingengine_access.log` file.

### Troubleshooting tests

**Test #1: Playback basic live streaming without nDVR**

*Use one of the following articles to verify that your basic live streaming setup is working properly without nDVR recording enabled. The DVR/Recorders property should be disabled in the Application.xml file.*

- [Set up live streaming using an RTMP-based encoder in Wowza Streaming Engine](#)
- [Set up live streaming using an RTSP/RTP-based encoder in Wowza Streaming Engine](#)
- [Publish and play a live stream (MPEG-TS based encoder) in Wowza Streaming Engine](#)
- [Set up live streaming using a native RTP encoder with SDP file in Wowza Streaming Engine](#)
- [Stream over MPEG-DASH with Wowza Streaming Engine](#)
Test #2: Play a single recorded stream

After you verify that Wowza Streaming Engine is configured correctly for live streaming and you can play the stream successfully, enable recording again and test the playback of a recorded stream using the provided sample video and player. Make sure that you can **record and playback with the basic single server scenario** before you try to set up an origin/edge configuration or playback recorded groups (StreamNameGroups created by Transcoder or SMIL files).

1. Re-enable recording in `Application.xml` by setting the `DVR/Recorders` property to:

2. Use one of the above articles about setting up live streaming to test playback of a single recorded stream.

If you can’t playback the newly recorded stream, use the following troubleshooting checklist:

1. Verify that a directory was created to store your recorded files after publishing the live stream to Wowza Streaming Engine. By default, a directory named `dvr` is created at `[install-dir]/dvr`.

2. Verify that the `/` property in `Application.xml` is set to:

   ```
   
   ```
   
   This setting enables nDVR.

3. Verify that the `/` property in `Application.xml` is set to:

   ```
   
   ```
   
   Without this value, the `dvr` directory can’t be created.

   **Note:** If the live source that provides input to Wowza Streaming Engine stops or disconnects, it might prevent nDVR from recording.

4. Verify that the `/` property in `Application.xml` is set correctly. If not configured properly, a log message will indicate that nDVR can’t record and why. For more information about error messages and common causes, see [Troubleshoot error messages in Wowza Streaming Engine](#).
Note: This property shouldn't be set on edge servers if you're setting up the more complex live stream repeater (origin/edge) configuration for use with nDVR. Be sure to carefully follow the instructions in Set up a Wowza Streaming Engine live stream repeater for nDVR to avoid errors.

5. Verify that your Archive Method setting is correct. Check the Archive Method setting for the application in Wowza Streaming Engine Manager (/ property setting in Application.xml) if the DVR storage directory doesn't have your recorded files. If not configured properly, a log message will indicate that nDVR can't record and why.

6. Look for messages in the access.log file that indicate Wowza nDVR is initialized and has the required configuration information. The following are sample successful log messages where the application name is dvr and stream name is myStream.

```
INFO application app-start _definst_ dvr/_definst_
INFO session connect-pending 192.168.1.50 -
INFO session connect 192.168.1.50 -
INFO stream create - -
INFO stream publish myStream -
INFO server comment - LiveStreamDvrRecorderBase.initProperties[dvr/_definst_/myStream] : properties:
INFO server comment - DvrStreamManagerBase.initProperties[dvr/_definst_/myStream] : {Properties: isDvrPacketizer: true}
INFO server comment - DvrStreamManagerBase.initStorage[dvr/_definst_/myStream] : storeName:dvrfilestorage isRecorder:true hasStorage:true
INFO server comment - LiveStreamDvrRecorder.init[dvr/_definst_/myStream]: audioOnlyChunkTargetDuration: 2000 ms
INFO server comment - LiveStreamDvrRecorder.init[dvr/_definst_/myStream]: dvrWaitForCodecTime: 8000 ms
INFO server comment - LiveStreamDvrRecorder.init[dvr/_definst_/myStream]: recordAudio: true
INFO server comment - LiveStreamDvrRecorder.init[dvr/_definst_/myStream]: recordVideo: true
INFO server comment - LiveStreamDvrRecorder.init[dvr/_definst_/myStream]: recordData: true
INFO server comment - MediaStreamMap.getDvrRecorder[dvr/_definst_/myStream]: Create DVR Recorder: dvrrecorder
INFO server comment - LiveStreamDvrRecorder.resetStream[dvr/_definst_/myStream]
INFO server comment - DvrStreamStoreBase.resetTimeMap[dvr/_definst_/myStream/myStream.0] : resetTime=0
INFO server comment - DvrPacketHandler.handlePacket[]: Video codec:H264 isCompatible:true
INFO server comment - DvrPacketHandler.handlePacket[]: Audio codec:AAC isCompatible:true
```
Problems with #1, #2, and #3 above will be reported in the access.log file. Failure could be due to malformed Application.xml or an invalid or missing property in Application.xml. If you don’t see the success messages, see Troubleshoot error messages in Wowza Streaming Engine for more information about how to troubleshoot nDVR startup error messages.
7. Verify that nDVR is recording. If it is, folders are created in your storage directory and they will have files with .m4fa and .m4fv file name extensions.

8. Verify that your live stream uses supported audio and video codecs.

9. Verify that the / property in Application.xml is set to:

   ```xml
   <StreamType>default</StreamType>
   ```

   In Application.xml, the default StreamType is default. This is the setting for video on demand (VOD) and configures Wowza Streaming Engine to look for video files in the [install-dir]\content directory instead of a live stream. If this property is set to default instead of live, no DVR-related errors will be recorded in the access.log file and no DVR-related directories will be created.

Test #3: Playback of recorded files with live stream repeater (origin/edge) configuration

Carefully follow the instructions in Set up a Wowza Streaming Engine live stream repeater for nDVR.

1. Verify that the Application.xml file is configured correctly on the origin server. For more information, see Configuring the origin server.

2. Verify that the Application.xml file is configured correctly on each edge server. For more information, see Configuring the edge server.

   - On the origin server, the HTTPStreamers property must be set to:

     ```xml
     <HTTPStreamers />
     ```

     This enables the origin to stream DVR audio and video chunks to the edge servers.

     On the edge server, this property must be set to the protocols that you want to stream DVR to. For example:

     ```xml
     ```

   - The / property must be left empty on the edge server because only the origin server stores the data.
Setting the `/` property to `dvrstreamingrepeater` on both origin and edge will cause errors. It’s only required for edge servers.

3. Test live playback from the origin server. Configuration of additional LiveStreamPacketizers required.

4. Test live playback from the edge server. Configuration of additional HTTPStreamers required.

5. Test DVR playback from the origin server.

6. Test DVR playback from the edge server.

Verify archive method

If you set the **Archive Method** property to **version**, a new directory (for example, `myStream.1`) is created when the directory `myStream.0` already exists when:

- The stream is disconnected and then restarted.
- The stream times out and then reconnects.

If you want all recordings to be in the same directory, set the **Archive Method** setting to **append** (the default setting).

If you want the current recording of the same stream name to replace a previous recording, set the **Archive Method** to **delete**.

Manage encoder disconnects and restarts

nDVR depends on a continuous live stream. Encoders can unexpectedly disconnect and are often configured to automatically reconnect. You can specify the amount of time that nDVR waits for packets before it stops recording (the default value is 5 minutes). This timeout is meant to account for when encoders disconnect and then restart. See `streamTimeout`.

Audio/video alignment

nDVR expects the audio and video in an incoming stream to be aligned. Input streams with short keyframe intervals and aligned audio and video work best.

The following error in the log indicates an audio/video alignment problem.

```
ERROR server comment
DvrStreamStoreBase.storeChunks[dvr/_definst_/myStream/myStream.0] : Skipping chunk. A/V packet times differ by 3025 ms, more than allowed 2000 ms. aTime=3749903889 vTime=3749906914
```
Alignment problem will cause chunks to not be recorded, which will degrade the recording. The best way to fix this is upstream from nDVR (typically the encoding process). While we strongly recommend that you fix any audio/video alignment issues before recording, you can set nDVR properties to try and compensate for this problem. These properties won’t fix alignment problems, but may help you to work around them. See `dvrPacketSortTime` and `dvrAllowableAVPacketDelta`.

Recording is missing audio or video

If audio and/or video is missing from a recording, the audio and/or video might not have been present when the stream codecs were being identified. For example, if only video was detected, then only video would be recorded and not the audio. You can increase the amount of time that nDVR waits to receive audio and video codec information after streaming has started. See `dvrWaitForCodecTime`.

Adaptive bitrate stream alignment

Information about nDVR chunks is recorded to log files by default without turning on additional debugging properties. This information enables you to verify that individual streams in an adaptive bitrate streaming group have properly aligned packets.

The following shows sample log lines for an application named `liveDVR` and stream named `livestream1`. The values are start times for audio, video, and data chunks for both nDVR timescale and packet timescale.

```
LiveStreamDvrRecorder.endChunk[liveDVR/_definst_/livestream1]: Add chunk: ind:0 a/v/k: packets: 84/30/0 durations: 1950/1983/-1 dvrTimes:3/0/-1 pTimes:680182/680179/0
```

More resources

- Configure advanced properties for nDVR
- Set up a Wowza Streaming Engine live stream repeater for nDVR
- Troubleshoot error messages in Wowza Streaming Engine