

# nanoStream Event Handling & Logging

(c) 2009-2011, nanocosmos gmbh

<http://www.nanocosmos.de>

[info@nanocosmos.de](mailto:info@nanocosmos.de)

## General Information

nanoStream supports event handling and debug logging for extended information about the encoding and streaming behaviour.

## Applications – GUI and command line

The command line encoder LiveEncCmd.exe supports the command line option /L <logfile> which writes an event log to the log file.

## High Level Events

- OnStop: will be called when the stream stops due to an error or file end.  
OnEvent: will be called on special events, for example:
- stream reconnect
  - time out situations / network error
  - authentication events (wrong password)

## Plugin Interface (Active-X / NPAPI / Javascript)

Event Logging can be activated by the interface function **SetLog**( *filename* ) . Set a local file name for which your application has write access, e.g. in the temporary folder.

## Log / Debug Configuration Registry Settings

Key: HKEY\_CURRENT\_USER\Software\DebugNano\ <Module Name>

Example Module Names:

- nanoStream.ax (Main nanoStream Plugin / ActiveX)
- nRtmpRenderer.ax (RTMP Streaming Sender)

Key: HKEY\_CURRENT\_USER\Software\DebugNano\ nRtmpRenderer.ax

### File name

Sets the output file name. The folder must exist.

Value name: LogToFile

Value type: REG\_SZ / String

Valid values: a valid output file name to enable file logging or an empty string

### Logging level

A higher value increases the amount of logging messages sent, and messages get more detailed.

Value name: TRACE

Value type: REG\_DWORD

Valid values:

0 – minimal logging

...

9 – maximal logging

## Handling High Level Events with Active-X

### OnStop:

- define event handler method

```
C# Example: void axRTMPActiveX1_OnStop(object sender,
AxRTMPActiveX.IRTMPActiveXEvents_OnStopEvent e) {
    MessageBox.Show(e.ToString() + " - " + e.result);
}
```

- register the event handler method:

```
C# Example: axRTMPActiveX1.OnStop += new
AxRTMPActiveX.IRTMPActiveXEvents_OnStopEventHandler(axRTMPActiveX1_OnStop);
```

Possible return values:

- 1: start streaming error
- 2: connection lost
- 3: no input data, timeout
- 4: license timeout
- >64: connection lost, reconnecting attempt (result - 64 shows the reason for connection loss)

### OnEvent:

- define event handler method:

```
axRTMPActiveX1.OnEvent += new
```

```
AxRTMPActiveX.IRTMPActiveXEvents_OnEventEventHandler(axRTMPActiveX1_OnEvent);
```

- register the event handler method:

```
void axRTMPActiveX1_OnEvent(object sender, AxRTMPActiveX.IRTMPActiveXEvents_OnEventEvent e)
{
    MessageBox.Show(e.ToString() + " - " + e.type + " - " + e.result);
}
```

- possible type codes:

- 10: rtmp statistics; result is a JSON-Object

- JSON Example: {"outputBufferSize" : 262000, "outputBufferFillness" : 1000, "outputBitrateDefault" : 320000, "outputBitrateFallback" : 350000, "videoBitrate" : 300000, "roundTripTime" : 33}

- outputBufferSize: the maximum available buffer (in bytes)

- outputBufferFillness: how much of the buffer is used (in bytes)

- outputBitrateDefault: the estimated bitrate the packets are send to the stream URL, this value is estimated by using tcp packet statistics gathered with a tcp sniffer, only available when running as admin

- outputBitrateFallback: like outputBitrateDefault, but not as accurate; does not use a tcp sniffer, but is always available
  - videoBitrate: the bitrate used by the video encoder at the moment
  - roundTripTime: the average time in ms, which is needed to receive an acknowledgement for a sent packet
- 
- 11: rtmp connection status; result is a JSON-Object
    - Example: `{"connectionStatus" : "connected"}`
    - Possible values: "connected", "disconnected" and "reconnecting"

See the Javascript API and nanoStream.html sample for an example implementation.

## Further Information and Services

Further information and extended services are available upon request.

Please contact us for further information.