

## Flash CS3 Video Encoder und Cue Points

Sie können mit Flash-Videodateien eine Reihe unterschiedlicher Cue Points verwenden. Mithilfe von ActionScript können Sie mit Cue Points interagieren, die Sie entweder beim Erzeugen einer FLV-Datei einfügen oder mit ActionScript in Echtzeit erstellen:

- **Cue Points für die Navigation:** Sie fügen beim Kodieren der FLV-Datei Cue Points für die Navigation in den FLV-Datenstrom und das FLV-Metadatenpaket ein. Mithilfe von Cue Points für die Navigation können Benutzer zu einem bestimmten Teil einer Datei springen.
- **Cue Points für Ereignisse:** Sie fügen beim Kodieren der FLV-Datei Cue Points für Ereignisse in den FLV-Datenstrom und das FLV-Metadatenpaket ein. Sie können Code für die Verarbeitung von Ereignissen verfassen, die an festgelegten Punkten während der FLV-Wiedergabe ausgelöst werden.
- **ActionScript-Cue-Points:** Externe Cue Points, die Sie mithilfe von ActionScript-Code erstellen. Sie können Code verfassen, mit dem diese Cue Points in Abhängigkeit von der Wiedergabe des Videos ausgelöst werden. Diese Cue Points sind weniger genau als die

eingebetteten Cue Points, da sie im Videoplayer separat verwaltet werden. Die Abweichung beträgt hierbei in etwa eine Zehntelsekunde.

Mit Cue Points für die Navigation wird an der angegebenen Cue-Point-Position ein Schlüsselbild erstellt. Mit ActionScript-Code können Sie dann die Wiedergabeposition eines Video Players auf diese Stelle setzen. Sie können bestimmte Punkte in einer FLV-Datei festlegen, an die Benutzer dann springen können. Wenn das Video beispielsweise aus mehreren Kapiteln oder Segmenten besteht, können Sie die Videowiedergabe steuern, indem Sie in die Videodatei Cue Points für die Navigation einbetten.

Sollten Sie eine Anwendung erstellen wollen, in der Benutzer Cue Points auswählen können, sollten Sie die Cue Points beim Kodieren der Datei einbetten und keine ActionScript-Cue-Points verwenden. Es empfiehlt sich, die Cue Points in die FLV-Datei einzubetten, da sie dann exakter angewählt werden können.

Sie können mit ActionScript-Code auf Cue-Point-Parameter zugreifen. Cue-Point-Parameter sind Teil des Ereignisobjekts, das zusammen mit dem `cuePoint`-Ereignis empfangen wird.

## HD Video und der Flash Player

Als Neuerung gibt es einen Hardware-unterstützten Fullscreen Mode. Dadurch sind deutliche Performance-Verbesserungen und eine bessere Bildqualität möglich. Genutzt wird DirectX (Windows) bzw. OpenGL (Mac). Statt das Hochskalieren dem Flash Player zu überlassen, wird dies vom Betriebssystem übernommen.

Außerdem werden mehrere Prozessoren beim Videodecoding genutzt (Multithreaded video decoding). Der VP6 Codec läuft auf einem Mehrprozessor-System in einem eigenen Thread. Im Hauptthread finden das Rendering und das Post Processing des Videos statt. Damit ist das Betrachten von Video in HD-Auflösung (1080p) auf modernen Dualcore-Rechnern möglich. Insgesamt ein sinnvoller Schritt, um die mittelmäßige Performance bei der Videowiedergabe im Vollbildmodus und bei hochauflösenden Videos (HD) in den Griff zu bekommen.

## H.264- und HE-AAC-Unterstützung

Seit Ende Juli während des Flash Player 9 Updates wird auch der populäre Videostandard H.264 und der Audiocodec High Efficiency AAC (HE-AAC) von Flash unterstützt!

H.264 kommt z.B. in Blu-Ray und HD-DVD-Videoplayern zum Einsatz. Auch Apple setzt mit QuickTime (inklusive iPod, Apple TV und iTunes) auf H.264 und AAC. Des Weiteren nutzen viele große TV-Sender H.264 und werden sich sicherlich über den verringerten Encodingaufwand freuen. Insgesamt reicht das Einsatzgebiet von H.264 von Mobile (3GP) bis hin zu hochauflösendem Content (z.B. HDTV), also weit über das Thema „Webvideo“ hinaus. Alle gängigen Dateitypen (.mp, .m4v, .m4a, .mov, .3gp) werden vom Flash Player unterstützt.

### Flash Player „Moviestar“

Das Update für den Flash Player mit dem Codenamen „Moviestar“, das die H.264-Unterstützung bringt, ist für Windows, Mac und Linux verfügbar. Die aktuell vorliegende Version wartet mit folgenden Features auf:

- Wiedergabe von H.264-Videos und Audio mit dem HE-AAC Codec
- Multicore-Unterstützung für Vector Rendering
- Fullscreen-Modus mit Hardwareunterstützung
- Ein Cache zum Cachen von Komponenten, z.B. des Flex Frameworks
- Microsoft Active Accessibility (MSAA) Support (nur Windows)

### Metadaten von H.264-Videos im Flash Player

Mit der Möglichkeit, Videos mit H.264-Codec im Flash Player abzuspielen, sind einige neue Informationen in den Metadaten hinzugekommen. Im Folgenden eine Übersicht über die wichtigsten Neuerungen:

#### **audiocodecid = mp4a**

Die ID des Audiocodecs ist „mp4a“ bei AAC, „mp3“ bei MP3

#### **videocodecid = avc1**

Die ID des Videocodecs lautet „avc1“ für den H.264 Codec

#### **audiosamplerate = 44100**

Die Samplerate der Tonspur. Es sind 8 bis 96 KHz möglich

#### **videoframerate = 24**

Die Framerate des Videos

#### **audiochannels = 1**

Die Anzahl der Audiokanäle

#### **duration = 70.1010430839002**

Dauer des Videos in Sekunden

#### **avclevel = 31**

Dieser Wert liegt zwischen 10 und 51. Es handelt sich dabei um verschiedene Level. Ein Level von H.264 beschränkt die Variablen eines Datenstroms, wie etwa die maximale Auflösung oder die Bitrate. 10 entspricht dabei dem Level 1.0, 51 entspricht dem Level 5.1.

Fortsetzung auf nächster Seite

### **avcprofile = 77**

Hier wird Auskunft über das H.264-Profil gegeben. *avcprofile* kann folgende Werte annehmen: 66 (= baseline), 77 (= main), 88 (= extended), 100 (= High), 110 (= High 10), 122 (= High 4:2:2), 144 (= High 4:4:4)

### **aacat = 2**

Laut Tonic Uro müsste der Parameter *aatype* heißen, es wird aber *aacat* ausgegeben. Der Wert steht für drei verschiedene AAC-Profile. 0 ist das Profil „AAC Main“, 1 steht für AAC LC (Low Complexity) und 2 für SBR (Spectral Band Replication, also HE-AAC)

### **seekpoints = 0,2.083,4.167,6.25,8.333, ...**

In den Metadaten ist ein Array mit *seekpoints* enthalten, die genau angesteuert werden können (*NetStream.seek()*). Dies ist neu, eine solche Liste gibt es bei *.flv* nicht

### **moovposition = 40**

Die Position (in Bytes) des *moov atom* in der Videodatei

### **Hinweis: H.264-Dateien werden erst nach vollständigem Download abgespielt**

Wer erste Versuche mit H.264-Videos und der neuen Beta des Flash Players unternommen hat, wird eventuell auf folgendes Problem gestoßen sein: Das Video wird erst abgespielt, wenn die Datei vollständig aus dem Internet heruntergeladen worden ist. Dieses Problem kennt der eine oder andere vielleicht von Apples QuickTime, wo dies manchmal ebenfalls der Fall ist. Die Ursache liegt am so genannten „moov atom“, das eine Art Inhaltsverzeichnis über die Datenstruktur der Videodatei darstellt. Das *moov atom* ist der Index einer MPEG-4-Datei, der Informationen zu den enthaltenen Videodaten und deren Datenstruktur beinhaltet. Dieser meist ein oder zwei Kilobyte große Index kann entweder am Anfang oder am Ende einer Datei stehen:

- Haben Dateien den Index am Ende der Datei, hat dies im Web zur Folge, dass zuerst die komplette Datei heruntergeladen werden muss, bevor sie abspielbar ist. Der Player weiß nicht, wie er die Daten zu „interpretieren“ hat.
- Stehen die Informationen jedoch am Anfang der Datei – also vor den eigentlichen Videodaten – ist der so genannte Progressive Download problemlos möglich. Da die Informationen über die Datenstruktur von Beginn an bekannt sind, können die Videos noch während des Downloads abgespielt werden.

### **H.264 Encoding für das Web**

Möchte man H.264-Videos für den Flash Player produzieren, sollte man aufgrund der obigen Schilderungen darauf achten, Dateien mit dem Index am Dateianfang zu erstellen. Leider sind die Anwendungen von Adobe (z.B. Premiere oder After Effects) dazu noch nicht in der Lage.

Nutzt man z.B. QuickTime als Encoder, gibt es dort seit langem die Option „Schnellstart“ (engl.: fast start) unter der Rubrik „Für Internet-Streaming vorbereiten“. Hat man diese Funktion aktiviert, wird das „moov atom“ an den Anfang geschrieben. Hat man bereits eine fertige Datei, kann man das „moov atom“ über einige Tools an den Anfang verschieben. Es gibt z.B. das C-Programm mit dem Namen „qt-faststart.c“ von Mike Melanson. Von Renaun Erickson gibt es den QTIndexSwapper, der mit AIR geschrieben worden ist (Adobe Integrated Runtime). Außerdem soll es bereits helfen, die Datei einfach mit QuickTime zu öffnen und mit *Speichern unter* neu abzuspeichern.