



Curso Multimedia Home Platform 1.1.2

DSM-CC II. Stream Events & NPT

Enviando Events al receptor y reaccionando ante ellos

Curso Multimedia Home Platform 1.1.2

Copyright 2008 © Enrique Pérez Gil

Licensed under the ***Creative Commons Attribution-Non-Commercial-No Derivative Works 3.0 Unported License***. You may not use this file except in compliance with the License. You may obtain a copy of the License at:

<http://creativecommons.org/licenses/by-nc-nd/3.0/legalcode>

This is a human-readable summary of the License applied:

(<http://creativecommons.org/licenses/by-nc-nd/3.0/>)

You are free to Share, to copy, distribute and transmit the work **Under the following conditions:**

- **Attribution.** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial.** You may not use this work for commercial purposes.
- **No Derivative Works.** You may not alter, transform, or build upon this work.

For any reuse or distribution, you must make clear to others the license terms of this work. Any of the above conditions can be waived if you get permission from the copyright holder. Nothing in this license impairs or restricts the author's moral rights.

Introducción

- DSMCC puede contener otro tipo de objetos distintos a Ficheros y Directorios, como por ejemplo aquellos referidos como **Stream Event** Messages: Tipos de Eventos
- Sin embargo un Tipo de Evento no me vale de nada si no sé donde se indica la ejecución de un Evento ni en qué momento del tiempo debe producirse...
- Como vimos en la parte de Files un **Stream Event** Message **hereda** de Stream, es decir, que además de la lista de Tipos de Eventos, cada uno con su **nombre**, características y su **EventId único en el contexto del Object Carousel**, dispone de las referencias(taps) a los Streams donde por un lado van a residir los **Stream Event Descriptors**, esto es, los “sucesos”, y por otro los **NPT Descriptors**, es decir, las referencias de Tiempo NPT: Normal Play Time.
- **NPT** es una referencia de tiempo que viaja como “**NPT reference descriptors**” en un **Stream de tipo 0x0C DSM-CC Descriptors**, y que permite al Broadcaster establecer una secuencia temporal válida para sus contenidos.
- **NPT es una ayuda que aporta DSMCC al Broadcaster.**
- Los Stream Event Descriptors también viajan en un **Stream de tipo 0x0C DSM-CC Descriptors**

- **NPT** es una referencia de tiempo que viaja como “**NPT reference descriptors**” en un **Stream de tipo 0x0C DSM-CC Descriptors**, y que permite al Broadcaster establecer una secuencia temporal válida para sus contenidos.
- **NPT es una ayuda que aporta DSMCC al Broadcaster.**
- Podría parecer que esto no es muy útil, puesto que como ya hemos dicho, ya existe el PCR (Program Clock Reference) de MPEG y que sirve para sincronizar el tiempo de los STBs, el problema es que PCR sirve tan sólo a los decodificadores MPEG para establecer como la hora del deco el correct **System Time Clock: STC**, y no es posible usarlo de forma más flexible.
- Un código NPT es más flexible y fácil de usar que PCR, de manera que las aplicaciones no tienen que preocuparse, por ejemplo, con respecto a qué horario está siendo emitido el PCR.
- A pesar de su bondad no es recomendable su uso por su poca fiabilidad. Veremos más adelante.

- **¿ Pero qué me ofrece realmente NPT ?** Imaginemos por ejemplo que el contenido que se está emitiendo es editado y acortado en 3 puntos, y además, cuando se emite resulta que hay un par de anuncios. ¿ cómo sabe una aplicación en qué punto correcto de la emisión del contenido se encuentra ? El PCR es el tiempo normal, esto no le dice nada, pero veamos con un ejemplo qué ofrece NPT:
 - **M**: se representan los bloques del evento original
 - **ED**: se representan las partes eliminadas al editarlo
 - **AD**: emisiones de anuncios
 - **NPT**: valor de la NPT. Se ajusta a los valores de la media que se emiten, manteniendo la coherencia

M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
ED			x	x						x	x	x						x	x	x				
AD						x	x	x							x	x	x							
NPT	1	2	5	6	7	1	2	3	8	9	13	14	15	16	1	2	3	17	21	22	23	24		

- Con el valor NPT se almacena el **contentID**, que nos ayuda a
 - identificar los saltos en la emisión de un contenido: contentID distintos en medio de la secuencia
 - Identificar secuencias de NPT de distintos contenidos, por ejemplo un anuncio que se emite en dos partes
 - Identificar los límites de dos shows diferentes.
- El **Descriptor NPT** se debe de emitir una vez por segundo!

- Consiste en la posibilidad de lanzar eventos asociados al contenido. Por ejemplo se pueden usar para indicar que va a comenzar un nuevo programa.
- Los **DSMCC Stream Events Objects** son los **tipos de eventos** que pueden darse asociados a un Servicio y se describen en un Object Carousel mediante los **Stream Event Messages**. Dentro de la definición del mensaje, además de un **eventID** único en el Carousel y una descripción “Humana” se incluirán referencias al Stream que contendrá los **Stream Event Descriptors (los sucesos!)** y al que contiene los NPT Descriptors (Normal Play Time), **nuestra referencia de Tiempo!!!**
- Los eventos en sí, que se lanzan usando los **Stream Event Descriptor**, viajan en otros Elementary Streams distintos (0x0C) a los que soportan los Data y Object Carousels, y son los que “suceden”. Estos contienen:
 - una referencia a su tipo de evento: **eventID (ojo, si el tipo no existe se ignora el evento)**.
 - **Un valor NPT que indica el momento en que debe de ocurrir**
 - **Información útil específica para la app.**

- Al tener que ocurrir en un momento concreto, esto permite al broadcaster enviar más de un **Stream Event Descriptor** y así asegurarse de que llega.
- La especificación MHP indica que para un evento programado el Broadcaster **deberá emitir el Stream Event Descriptor una vez cada segundo durante los 5 segundos previos al suceso.**

DO-IT-NOW EVENTS

- Existe un modo de eventos muy usado denominado “do-it-now”. Por ejemplo: El tipo sería Gol, cuya definición estaría en el Object Carousel; cuando se marcarse, se publicaría un event do-it-now asociado el tipo Gol. Cuando le llega el evento a la aplicación MHP que se ha suscrito al evento esta puede, por ejemplo, abrir un mensaje en pantalla mostrando algún tipo de anuncio: la clásica pizza...etc

Stream Events y NPT

- Si un Stream Event Programado debe de lanzarse a un NPT value que ha pasado debido a, por ejemplo, Edición, entonces el evento se lanzará en cuanto llegue el siguiente NPT asociado con el valor superior al programado
- Sólo es necesario NPT para Stream Events programados. Los DO-IT-NOW no lo necesitan. Estos son más sencillos de usar, tanto para los broadcasters como para los desarrolladores, por lo que son los más comúnmente utilizados.
- **Recomendación: usar DO-IT-NOW** en lugar de programados, debido a que por ejemplo los valores NPT pueden corromperse cuando los contenidos saltan entre networks, o bien si los PCR se corrompen puede ocurrir que al re-construirse no se haga lo propio con los NPT. Mirad lo que dice la cabecera de la clase DSMCCStream (y relacionadas)

NOTE: The NPT mechanism and scheduled stream events that depend on it are known to be vulnerable to disruption in many digital TV distribution networks. Existing deployed network equipment that re-generates the STC is unlikely to be aware of NPT and hence will not make the necessary corresponding modification to STC values inside NPT reference descriptors. Applications should only use NPT where they are confident that the network where they are to be used does not have this problem.

NPT

- Veíamos que un Object Carousel puede contener Stream Messages que mediante TAPS pueden hacer referencia al MPEG Stream que contiene los valores NPT. Igualmente sabemos que DSMCC usa los siguientes tipos de Streams, entre los cuales tenemos el tipo **0x0C** que se supone contiene **DSMCC Descriptors** como lo son: **Stream Event Descriptors** y **NPT Descriptors**.
 - 0x0A: Multiprotocol Encapsulation: **Datos IP**
 - 0x0B: Object o Data Carousels (Ficheros, Event Objects...): **Stream Messages**
 - 0x0C: DSMCC Descriptors (**Stream Event Descriptors**, **NPT Descriptors**)
 - 0x0D: es necesario abrirlo para conocer el contenido
- Para obtener un Stream Message has de conocer la referencia al mismo para así poder crear el objeto **DSMCCStream** o **DSMCCStreamEvent** que lo describe.

`org.dvb.dsmcc.DSMCCStream \ org.dvb.dsmcc.DSMCCStreamEvent` (ved el API)

- ¿ Cómo se gestionan los eventos ? ¿ cómo accedo a los valores NPT ?
 - Con **org.dvb.dsmcc.DSMCCStreamEvent** sabré que tipos de eventos existen y podré suscribirme a sus “sucesos”!!
 - **org.dvb.dsmcc.DSMCCStream**, de quien hereda, me permitirá acceder a la NPT y sus valores.
- Sabemos que el BIOP de definición de **Stream Messages** puede contener TAPS que se refieren a:

STR_NPT_USE	0x000B	Stream NPT Descriptors
STR_STATUS_AND_EVENT_USE	0x000C	Stream mode y Stream Event Descriptors
STR_EVENT_USE	0x000D	Stream event Descriptors
STR_STATUS_USE	0x000E	Stream mode Descriptors
BIOP_ES_USE	0x0018	Video/Audio Stream
BIOP_PROGRAM_USE	0x0019	Service Reference

org.dvb.dsmcc.DSMCCStream

- El API DSMCCStream es el que sigue:
 - public **DSMCCStream**(org.dvb.dsmcc.DSMCCObject aDSMCCObject)
 - El DSMCCObject debe referenciar un Stream o StreamEvent Object Carousel BIOP Message
 - public **DSMCCStream**(java.lang.String path)
 - El path debe apuntar a un DSMCCObject que represente un Stream o StreamEvent Object Carousel BIOP Message
 - public **DSMCCStream**(java.lang.String path, java.lang.String name)
 - El path debe apuntar a un DSMCCObject que represente un Stream o StreamEvent Object Carousel BIOP Message
 - public long **getDuration**()
 - Duración en segundos del evento actual.

org.dvb.dsmcc.DSMCCStream

- API DSMCCStream

- public void **removeNPTListener**(org.dvb.dsmcc.NPTListener l)
- public void **addNPTListener**(org.dvb.dsmcc.NPTListener l)

Suscripción de los eventos de NPT para detectar las siguientes situaciones:

- **NPTPresentEvent**: Se ha detectado una referencia NPT
- **NPTRemovedEvent**: La referencia NPT se ha eliminado
- **NPTRateEvent**: El Ratio de frecuencia NPT ha cambiado
- **NPTDiscontinuityEvent**: Se ha detectado una discontinuidad

- public long **getNPT**()
 - Valor actual del NPT en segundos
- public org.dvb.dsmcc.NPTRate **getNPTRate**()
 - NPTRate (ver API)

org.dvb.dsmcc.DSMCCStream

- API DSMCCStream
 - public org.davic.net.Locator **getStreamLocator()**
 - Locator del Stream referenciado
 - public boolean **isAudio()**
 - public boolean **isData()**
 - public boolean **isMPEGProgram()**
 - public boolean **isVideo()**
 - Métodos para saber si el Stream referenciado es de tipo Audio, Datos, Service, Video

org.dvb.dsmcc.DSMCCStreamEvent

- DSMCCStreamEvent hereda de DSMCCStream. Lo que se usa para sincronizarse y gestionar Events es **DSMCCStreamEvent**.
- DSMCCStreamEvent representa los **Stream Event Objects**.
- El problema ya lo adelantamos viendo los constructores de DSMCCStream:
 - ¿ cómo puedo hacer referencia al Stream Event Object del Carousel ? Has de conocer su referencia/path para poder construirlo.

org.dvb.dsmcc.DSMCCStreamEvent

- Un Stream Event Object puede contener descripciones **de varios tipos de Eventos**.
 - El api se plantea de la forma siguiente: primero habremos de **obtener referencias al Stream Event Object (paths)** y después nos suscribimos en ella para recibir los eventos del tipo de evento que elijamos.
 - Un Tipo de Event tiene un ID único para todos los Tipos existentes, sin embargo el name es algo que es único sólo dentro de un DSMCCStreamEvent

org.dvb.dsmcc.DSMCCStreamEvent

- API DSMCCStreamEvent

- public **DSMCCStreamEvent**(DSMCCObject aDSMCCObject)
 - El DSMCCObject debe referenciar un OC DSMCC StreamEvent BIOP Object
- public **DSMCCStreamEvent**(String path, String name)
- public **DSMCCStreamEvent**(String path)
 - El path debe referenciar un OC DSMCC StreamEvent BIOP Object
- public synchronized void **unsubscribe**(int eventId, StreamEventListener l)
- public synchronized void **unsubscribe**(String eventName, StreamEventListener l)
- public synchronized int **subscribe**(String eventName, StreamEventListener l)
 - Suscripción a un evento concreto.Devuelve el ID con el que se le identifica unitariamente.

```
public interface StreamEventListener extends java.util.EventListener {  
    public void receiveStreamEvent(StreamEvent e);  
}
```

org.dvb.dsmcc.DSMCCStreamEvent

- API DSMCCStreamEvent
 - public String [] **getEventList()**
 - Lista de los eventos que cubre este DSMCCStreamEvent

org.dvb.dsmcc.StreamEvent

- El API
 - public java.lang.Object **getSource()**
 - El DSMCCStreamEvent del que proviene.
 - public String **getEventName()**
 - Nombre del evento
 - public int **getEventId()**
 - Event ID
 - public long **getEventNPT()**
 - Momento en que se lanzó en millis.
 - public byte[] **getEventData()**
 - Información que se emite con el evento. Mensaje...etc

Ejemplo simple de implementación en Ejercicio dsm2

```
package code4tv.mhp112.exercise_dsm2;

import org.dvb.dsmcc.DSMCCObject;
import org.dvb.dsmcc.DSMCCStreamEvent;
import org.dvb.dsmcc.StreamEventListener;

public class DSMCCEventSample implements StreamEventListener {

    private String eventIDA = "eventTypeA";
    private String eventIDB = "eventTypeB";
    private String eventIDC = "eventTypeC";
    private String dsmccURIPath = "event.txt";
    private DSMCCObject dsmcc = null;
    private DSMCCStreamEvent dsmccStreamEvent = null;

    public DSMCCEventSample() {
        try {
            dsmcc = new DSMCCObject(dsmccURIPath);
            dsmccStreamEvent = new DSMCCStreamEvent(dsmcc);
            dsmccStreamEvent.subscribe(eventIDA, this);
            dsmccStreamEvent.subscribe(eventIDB, this);
            dsmccStreamEvent.subscribe(eventIDC, this);
        } catch (Throwable err) {
            err.printStackTrace();
        }
    }

    public void receiveStreamEvent(org.dvb.dsmcc.StreamEvent evt) {
        if (evt.getEventName().equals(eventIDA)) {
            System.out.println("EVENT A RECEIVED");
        } else if (evt.getEventName().equals(eventIDB)) {
            System.out.println("EVENT B RECEIVED");
        } else if (evt.getEventName().equals(eventIDC)) {
            System.out.println("EVENT C RECEIVED");
        }
    }
}
```

ISO/IEC 13818-1	Part 1. Elementary Streams transport definition
ISO/IEC 13818-6	Part 6. Extensions for DSM-CC. Digital Storage Media Command and Control
ETSI EN 300 468	Digital Video Broadcasting (DVB);Specification for Service Information (SI) in DVB systems
ETSI EN 301 192	DVB specification for data broadcasting
ETSI TR 101 202	Implementation Guidelines for Data broadcasting
ETSI TR 101 162	Digital broadcasting systems for television, sound and data services; Allocation of Service Information (SI) codes for Digital Video Broadcasting (DVB) systems
ETSI TR 102 154	Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in Contribution and Primary Dist
ETSI TR 101 211	Guidelines on implementation and usage of Service Information (SI)
ETSI TR 101 200	Digital Video Broadcasting (DVB); A guideline for the use of DVB specifications and standards
DAVIC	Digital Audio Visual Council. davic 1.4.1
HAVI	Specification of the Home Audio/Video Interoperability (HAVi) Architecture
Interactivetvweb	http://www.interactivetvweb.org/
Wikipedia DSMCC	http://en.wikipedia.org/wiki/DSM-CC
MHP 1.1.2	Multimedia Home Platform, A068r1 & tam668r23_11xdraft_20061115
MHP 1.1.3	Multimedia Home Platform, A068r3
CDC 1.1	Connected Device Configuration (CDC) 1.1 (JSR=218).
PBP 1.1	Personal Basis Profile 1.1 (JSR 217)
MHP.org	www.mhp.org
INTRO MHP 1.1.3	tam1032r1-mhp-iptv-presentation