



Curso MHP 112 Ejercicios

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.

- **Es necesario una instalación tomcat/servidor web en la máquina.**
- **El planteamiento es disponer de ejercicios en los que se pueda trabajar con libertad de movimientos sin perjuicio de los anteriores. Por ello para asegurar la integridad de algunas prácticas el comienzo de las nuevas consiste en “copiar y renombrar” anteriores**
- **Ejercicios ejecutados en Boadilla del Monte, Madrid, España.**
- **Es importante disponer de los fuentes de las librerías STUBS configurados en Eclipse.**

Ejercicios XLETS-LIFE CYCLE

Ejercicios Bloque CICLO-1

Ejercicio ciclo1

Descripción

- Desarrollad un Xlet (code4tv.mhp112.exercise_ciclo1.Exercise_ciclo1.java) que, sin efectuar ninguna acción añadida, muestre los mensajes de cada estado en la consola.
- Jugad sacando la lista de aplicaciones con el mando y quitándola para ver como pasa a **paused** y se reactiva, o cambiando de canal para ver como se destruye y reinicia...

Objetivo: familiarizarse con el ciclo de vida de los Xlets

Comentarios a la solución

- Ved el fichero /classes/exercise_ciclo1.ait
- Si lo dejamos **present** aparece en Strong 5510 al pulsar APP en el Canal Madrid pero no así en Antena 3 donde sí hay un Launcher emitido por Antena 3
- Si se pone **bound** a 0, cuando cambia de canal hace **paused** y luego arranca.
- Con **bound** a 1 destruye el xlet al cambiar de canal.
- Probad cambiando configuración en ait de **bound** y **present** y cambiad de canal, sacad la lista de canales y ocultadla, pulsad botón de lista de aplicaciones (APP en Strong 5510).
- Ved exercise_ciclo1.txt

Ejercicio ciclo2

Descripción

- Desarrollar un Xlet(`code4tv.mhp112.exercise_ciclo2.Exercise_ciclo2.java`) que no llegue a estado Loaded.
- El alumno puede copiar el contenido de ciclo1 y partir de él renombrando paquete y clase.

Objetivo: familiarizarse con el ciclo de vida de los Xlets

Comentarios a la solución

- Si se quita el **implements** pero tiene todos los métodos de Xlet: funciona!!!!
- Pongamos ahora un runtime exception: el objeto no se llega a instanciar: NO LLEGA A LOADED!!
- Ved `exercise_ciclo2.txt`

Ejercicio ciclo3

Descripción

- Desarrollad un Xlet(code4tv.mhp112.exercise_ciclo3.Exercise_ciclo3.java) pase desde Loaded a Destroyed. Hacedlo por dos caminos.
- El alumno puede copiar el contenido de ciclo1 y partir de él renombrando paquete y clase.

Objetivo: familiarizarse con el ciclo de vida de los Xlets

Comentarios a la solución

- Deben de en el método initXlet:
 - a) Lanzar la Exception XletStateChangeException en initXlet o
 - b) Llamar a XLetContext.destroyXlet(). Que hagan las dos.
- En el caso a) NO se llama a destroyXlet() pero después se llama **startXlet()!!!!!!**
- En el caso b) **Se llama a startXlet()!!!!!!** antes de matarlo!!! Y lo mata porque si cambiáis de canal veréis que no hace el destroy()
- Si incluimos un System.out del xlet vemos que cuando se cambia de canal el objeto Xlet se descarta, observad el id del mismo en System.out
- Ved exercise_ciclo3.txt

Ejercicio ciclo4

Descripción

- Desarrollar un Xlet(`code4tv.mhp112.exercise_ciclo4.Exercise_ciclo4.java`) que estando en Paused cuando se le quiera arrancar pase a Destroyed
- El alumno puede copiar el contenido de ciclo1 y partir de él renombrando paquete y clase.

Objetivo: familiarizarse con el ciclo de vida de los Xlets

Comentarios a la solución

- Deben de : guardar el XletContext para poder después llamar a `notifyDestroyed()` en el `startXlet()` (paso de Paused a Destroyed en `startXlet()`)
- Vale desde Active igualmente.
- Ver `Exercise_ciclo4.txt` se ve como lo mata.
- Opción b) Ahora haced que se quede en Paused llamando a la exception `XletStateChangeException`. Se queda en Pause
- Ved `exercise_ciclo4.txt`

Ejercicio ciclo5

Descripción

- Desarrollar un Xlet(`code4tv.mhp112.exercise_ciclo5.Exercise_ciclo5.java`) que pase de Active a Paused a los 5 segundos de estar activo.
- El alumno puede copiar el contenido de ciclo1 y partir de él renombrando paquete y clase.

Objetivo: familiarizarse con el ciclo de vida de los Xlets

Comentarios a la solución

- Deben de arrancar un Thread y hacer que se pare a los 5 segundos.
- Hacemos un Xlet que no se para 5 segundos sino que espera un poco en cada ciclo y pregunta...y así hasta los 5”
- Ver `Exercise_ciclo5.txt`
- Variante: si queréis hacer que muera: llamar a `ctx.notifiedDestroyed()`

Ejercicios Bloque CICLO-2

Ejercicio ciclo6

Descripción

- Desarrollar un Xlet(`code4tv.mhp112.exercise_ciclo6.Exercise_ciclo6.java`) que muestre TODAS las propiedades disponibles en el XletContext y en System. Igualmente habrá de imprimir los parámetros pasados al Xlet
 - Queremos saber qué versión de MHP y que elementos opcionales soporta.
 - El alumno puede copiar el contenido de ciclo1 y partir de él renombrando paquete y clase.
- **Nota:** *se aporta `code4tv.mhp112.mhpproperties.MHPPProperties` donde tenéis unos `String[][]` que os pueden ayudar!!! . Se encuentra en el directorio: `util_mhpproperties` (siempre se sigue el mismo esquema: `util_[nombre_paquete]`, nombre clase = nombre paquete con las mayúsculas).*

Objetivo

conocer como acceder a los argumentos y visibilidad de las propiedades del entorno

Comentarios a la solución

- Habrá problemas de permisos. Hablaremos en el capítulo de Seguridad....Aplicaciones firmadas.....
- Ver `Exercise_ciclo6.txt` para ejecución con seguridad ON y `Exercise_ciclo6(err security).txt` para seguridad ON

Ejercicios APP SIGNALLING

Ejercicios Bloque APPSIG-1

Ejercicio apsig1

Descripción

- Describir la secuencia de búsquedas de los ficheros **/startup.init** y **/local/startup.props** asumiendo que la AIT ha proporcionado los siguientes 2 transport_protocol_descriptor con la siguiente estructura de directorios/zips

```
http://www.uno.org  
  apps/graphics.zip  
  appsutils  
  front/frames/pages.zip
```

```
http://www.dos.org  
  jdbc/drivers.zip
```

Objetivo: familiarización con los transport_protocol_descriptor

Ejercicio apsig1. solución

- La AIT ha proporcionado **2 transport_protocol_descriptors** de forma que tenemos la siguiente estructura de directorios:

- <http://www.uno.org>

apps/graphics.zip

appsutils

front/frames/pages.zip

- <http://www.dos.org>

jdbc/drivers.zip

La búsqueda de un fichero startup.init sería:

Como /startup.init en <http://www.uno.org/apps/graphics.zip>

Como <http://www.uno.org/appsutils/startup.init>

Como /startup.init en <http://www.uno.org/front/frames/pages.zip>

Como /startup.init en <http://www.dos.org/jdbc/drivers.zip>

La búsqueda de un fichero local/startup.init sería:

Como /local/startup.init en <http://www.uno.org/apps/graphics.zip>

Como <http://www.uno.org/appsutils/local/startup.init>

Como /local/startup.init en <http://www.uno.org/front/frames/pages.zip>

Como /local/startup.init en <http://www.dos.org/jdbc/drivers.zip>

Ejercicio apsig2

Descripción

- Ahora imaginemos que la AIT ha proporcionado un **DVB-J Application Location Descriptor** con la siguiente información. Describid los lugares en los que se buscaría el Xlet y sus paths completos.

```
base_dir: /home
classpath:
    data1/classes
    /wise/data2
xlet:
    myweb.Mxlet
```

- **Objetivo:** familiarización con los transport_protocol_descriptor



Ejercicio apsig2. solución

- La AIT ha proporcionado un **DVB-J Application Location Descriptor** con la siguiente información:

base_dir: /home

Classpath:

data1/classes;

/wise/data2

xlet:

myweb.Mxlet

Hemos de buscar:

A: /home/myweb/Mxlet.class

B: /home/data1/classes/myweb/Mxlet.class

C: /wise/data2/myweb/Mxlet.class

Como A en <http://www.uno.org/apps/graphics.zip>

Como A en <http://www.uno.org/appsutils/home/myweb/Mxlet.class>

Como A en <http://www.uno.org/font/frames/pages.zip>

Como A en <http://www.dos.org/jdbc/drivers.zip>

Como B en <http://www.uno.org/apps/graphics.zip>

Como B en <http://www.uno.org/appsutils/home/data1/classes/myweb/Mxlet.class>

Como B en <http://www.uno.org/font/frames/pages.zip>

Como B en <http://www.dos.org/jdbc/drivers.zip>

Como C en <http://www.uno.org/apps/graphics.zip>

Como C en <http://www.uno.org/appsutils/wise/data2/myweb/Mxlet.class>

Como C en <http://www.uno.org/font/frames/pages.zip>

Como C en <http://www.dos.org/jdbc/drivers.zip>

Ejercicios MHP GRAPHICS I DEVICES-SCENES

Ejercicios Bloque G1

Ejercicio graph7

Descripción

- Desarrollad un XLET (code4tv.mhp112.exercise_graph7.*) que obtenga toda la información que se pueda de la configuración actual de los Devices de Gráficos, Vídeo y Background.
- El alumno puede copiar el contenido de ciclo1 y partir de él renombrando paquete y clase.

Objetivo

- Comienzo de familiarización con los componentes Gráficos del sistema, en especial con los referentes a la configuración

Comentarios a la solución

- ¿ Donde hacerlo ? en initXlet, startXlet....Hagamos un Thread!!!
- Introducimos el concepto de Thread en startXlet
- Hemos de imprimir la información....System.out.
- El resultado es un HEmulatedGraphicsDevice: Configuración obtenida:
 - pixelAspectRatio=56:45
 - pixelResolution=720x576,
 - is16x9=false,
 - screenAreaX=[0.0,0.0,1.0,1.0]
 - interlaced=true,
 - filter_flickering=true,
 - realPixelAspectRatio=48:45
- Ved Exercise_graph7.txt
- Se aporta paquete **util_screenconf** para imprimir las configuraciones.

Ejercicio graph8

Descripción

- Desarrollad un XLET (code4tv.mhp112.exercise_graph8.*) que obtenga todas las posibles configuraciones que admite cada uno de los Devices de Gráficos, Vídeo y Background.
- El alumno puede copiar el contenido de graph7 y partir de él renombrando paquete y clase

Objetivo

- Comienzo de familiarización con los componentes Gráficos del sistema, en especial con los referentes a la configuración

Comentarios a la solución

- Nótese que nos da configuraciones con el mayor número de propiedades REQUIRED, es decir, que ofrece flexibilidad.
- Ved Exercise_graph8.txt

Ejercicios Bloque G2

Ejercicio graph9

Descripción

- Desarrollad un XLET (code4tv.mhp112.exercise_graph9.*) que establezca la configuración siguiente:
 - Graphics: Pixels cuadrados, resolución de 800x600, escalado de imágenes
 - Video: Alineado con Graphics
 - Background: de color azul, y además poderlo cambiar después.
- Utilizad los métodos de cada Device, no los de HScreen.
- El alumno puede copiar el contenido de graph8 y partir de él renombrando paquete y clase

Objetivo

- Familiarización con los componentes Gráficos del sistema, en especial con los referentes a la configuración de HScreen y Devices

Comentarios a la solución

- RECOMENDACIÓN: leeros la documentación de los métodos Java!!!!
- Hemos de implementar los métodos relacionados con reserva de recursos.
- No soporta Matte!
- Si hay tiempo: Una vez hecho repetir estableciendo la configuración con la opción de HScreen.getBestConfiguration(...)
- Ved Exercise_graph9.txt

Ejercicios Bloque G3

Ejercicio graph9b

Descripción

- Utilizad la gestión de eventos de cambios de configuración en los Devices para readaptar los mismos.
- De donde partimos: mecanismo a aplicar en general para ejercicios **variantes(o extensions) como este (a,b,c...)** salvo que se diga lo contrario:
 - copiad el Exercise_graph9.java en el mismo directorio como Exercise_graph9b.java y realizad la modificación sobre este.
 - Para ejecutarlo: cambiad en el exercise_graph9.ait la clase Exercice_graph9 por Exercise_graph9b

Objetivo

- Familiarización con los componentes Gráficos del sistema, en especial con los referentes a la configuración de HScreen y Devices

Comentarios a la solución

- Ojo: eliminamos los listeners en el destroy!!

Ejercicios Bloque G4

Ejercicio graph10

Descripción

- Desarrollad un XLET (code4tv.mhp112.exercise_graph10.*) en el que:
 - Caso A) puesto que nuestra app va a tener mucho texto, queremos una GraphicsDevice con configuración 14:9 y resolución 720 x 576; además deseamos colocar nuestra aplicación de tal forma que tenga Fondo Azul y se sitúe en (0,0, 200,200)
 - Caso B) Además se sitúe en el (40,40, 70%,70%)
 - Caso C) Además queremos asegurar que se va a ver entera
 - Caso D) Además queremos asegurar que el texto se va a ver correcto, y ojo, situamos texto en los bordes del área!

Común: Queremos los métodos boolean siguientes para conocer el estado del Xlet: started(), paused(), destroyed().

Cada caso se le indicará al Xlet como un parámetro de arranque

- El alumno puede copiar el contenido de graph9b y partir de él renombrando paquete y clase

Objetivo

- Uso de HScene: creación y configuración. Asunción del problema de OverScan
- Gestión correcta de destrucción de recursos y gestión de los mismos

Ejercicio graph10

Comentarios a la solución

- Hacemos prueba para todos los casos
- **Hay que configurar en condiciones la HGraphicsDevice!!!**
- No sale el background a azul ? hay que establecer el backgroundmode= HScene.BackgroundMode_FILL
- **Listener para ver cambios en HScreen**
- Poner a setVisible(false) en paused
- **Listeners para saber si me han cambiado el HScene de aspecto**
- Liberar recursos en destroyXlet
- Controlar el ciclo de vida
- Al final, si hay tiempo: hay que hacer un Thread en el que cada 3" cambie de configuración de forma cíclica
- Al final, si hay tiempo: probar un Resize de HScene

NO DA TIEMPO: CON HACER LA CONFIGURACIÓN DE LA HSCENE 1 VEZ VALE.

Ejercicios MHP GRAPHICS II WIDGETS

Ejercicios Bloque G2-1

Ejercicio graph21

Descripción

- Desarrollad un XLET (code4tv.mhp112.exercise_graph21.*) en el que puesto que nuestra app va a tener mucho texto, queremos un GraphicsDevice con una configuración 14:9 y resolución 720 x 576; además deseamos colocar nuestra aplicación de tal forma que tenga Fondo BLUE y se sitúe en (15%,15%, 70%,70%)
- Después: Creamos dos componentes de texto donde se puedan introducir números enteros y dos botones: uno para multiplicar y otro para finalizar la aplicación: Solo creadlos no hagáis nada más.
- El alumno puede copiar el contenido de graph10 y partir de él renombrando paquete y clase

Objetivo: Uso de Widgets: creación y configuración

Comentarios a la solución

- Donde establecer los componentes ?: cuando creamos la HScene
- Qué componentes usar ?: HSingleLineEntry y HTextButton
- Al establecer el texto a los botones tenemos que especificar estados !: ayuda para la configuración de los componentes por parte de HAVI.
- Los componentes de texto NO se ven!!: establezcamos un color del fondo: yellow
- No se ve el color de fondo!: setBackgroundMode(HVisible. FILL o NO_FILL).

(continúa)

Ejercicio graph21

Comentarios a la solución

- El fondo de los botones coincide con el del fondo: pongámosles un fondo verde al de multiplicar y uno rojo al de salir. Pasa exactamente igual que con los textos: hay que establecer el `backgroundMode`.
- Se aporta el paquete `WidgetFactory`.

Ejercicios Bloque G2-2

Ejercicio graph21b

Descripción

- Creemos un título: “tabla de multiplicar”: Fondo negro y color amarillo (no uséis WidgetFactory)
- Creemos un pie: “copyright mi nombre”: Fondo negro y color amarillo (no uséis WidgetFactory)
- Establezcamos las fuentes adecuadas para el resto de componentes existentes.
- Ahora haced que la scene sea transparente!!
- Y también los botones!

Seguimos trabajando en el ejercicio 21.

Objetivo: Uso de Widgets: creación y configuración

Comentarios a la solución

- Qué componente para un Texto?: HText.
- Una vez hecho: ¿ No notáis “vibración” en el título y subtítulo ? En TV en modo Entrelazado (presentación de pares/impares en cada hertzio de forma alternativa) las líneas horizontales vibrarán si usamos **colores demasiado saturados (poco gris) o que contrasten demasiado**). Cambiemos el negro por: (0,100,0). Ahora el que vibra es el Yellow. Cuando en el video sale blanco es complicado pero al menos se puede mitigar.
- Una vez hecho cambiad si queréis las creaciones de Textos por llamadas a WidgetFactory.

Ejercicios Bloque G2-3

Ejercicio graph22

Descripción

- XLET: code4tv.mhp112.exercise_graph22.*.
- Queremos cambiar el aspecto de TODOS los HTextButton haciendo que siempre se pinte un fondo de imagen difuminada.
 - Disponéis de un método en utils_imagefactory que os da una imagen pasándole un montón de parámetros....
- El trabajo ha de hacerse creando un HLook vuestro para el botón.
- El alumno puede copiar el contenido de graph21 y partir de él renombrando paquete y clase

Objetivo: Uso de Widgets: Familiarización con HLook

Comentarios a la solución

- Interesante: establecemos un HTextLook y no hace nada.
- Veamos si le establecemos el foco: requestFocus() y además le decimos: setBordersEnabled(true)
- Se aporta utils_imagefactory

Ejercicios Bloque G2-4

Ejercicio graph22b

Descripción

- Usad Private HLook Data para almacenar la imagen.
- Origen: seguimos en el 22. Creamos dos clases nuevas copiando las originales y trabajamos sobre ellas: Exercise22b y MyHTextLookb

Objetivo

- Uso de Widgets: Familiarización con HLook

Comentarios a la solución

Ejercicios Bloque G2-5

Ejercicio graph23

Descripción

- XLET: code4tv.mhp112.exercise_graph23.*.
- Establecer el orden de navegación de los componentes usando setFocusTraversal:
Modelo:

```
    <-> Param a <-> multiply <->  
          |           |  
          <-> Param b <-> end <->
```
- El alumno puede copiar el contenido de graph22b y partir de él renombrando paquete y clase.

Objetivo:

- Uso de Widgets: Familiarización con HNavigable

Comentarios a la solución

- **Nadie tiene el foco!!!!** ¿ cómo se obtiene el foco ? Cuando se presenta la HScene se hace un requestFocus(). La HScene no es visible por defecto -> nadie tiene el foco!!!!
- Cuando se obtiene el foco vemos que el primer parámetro aparece recuadrado. Probemos los movimientos...Funciona!!!!
- Probad a entrar en una caja de texto y teclead algo...¿ Cómo salgo del modo edición ? Con la tecla Exit.

Ejercicio graph23b

Descripción

- Establecer un HLook común para los HSingleLineEntry de manera que se impriman en consola los cambios que se reciban en **widgetChanged**
- Seguimos en el Ejercicio 23.

Objetivo

- Uso de Widgets: Familiarización con HLook.

Comentarios a la solución

- Creamos MyHSinglelineEntryLook

Ejercicio graph23c

Descripción

- Añadid dos textos transparentes de manera que en uno diga que componente pierde el foco y en el otro diga quien lo obtiene
- Modificad MyHTextLookb y MyHSingleLineEntry para que impriman el estado en que se encuentra el componente: NORMAL_STATE, FOCUSED_STATE...etc
- Seguimos en el Ejercicio 23b.

Objetivo

- Uso de Widgets: Familiarización con HNavigable

Comentarios a la solución

- Nota: FocusEvent.getOppositeComponent() no funciona!!!
- Para el segundo se aporta util_statename que nos da la descripción del estado.

Ejercicios Bloque G2-6

Ejercicio graph24

Descripción

- XLET: code4tv.mhp112.exercise_graph24.*.
- Implementad la operación de multiplicar y la de end.
- El resultado de la multiplicación colocadlo en otro texto. Si da error la operación que muestre “error!”
- Includ dos botones de forma que cada uno se encargue de borrar el carácter anterior a donde se encuentra el cursor de cada HSingleLineEntry
- El alumno puede copiar el contenido de graph23c y partir de él renombrando paquete y clase

Objetivo

- Uso de Widgets: Familiarización con HActionable, Gestión de eventos de teclado, Problemática de Edición de texto

Comentarios a la solución

- Atención al setFocusTraversal. Hay que añadir los nuevos botones
- Atención: ¿ **cómo borramos caracteres estando dentro de un HSingleLineEntry** ? Lo veremos en un par de ejercicios.
- Existen muchas posibilidades de configuración de un HSingleLineEntry...caracteres que acepta, etc...Luego veremos DVBTxtLayout
- Cuando hacéis el exit el XLET se queda en el deco: si dáis a APP lo podéis ver y lanzar de nuevo.

Ejercicios Bloque G2-7

Ejercicio graph25

Descripción

- XLET: code4tv.mhp112.exercise_graph25.*.
- Haced un componente HToggleButton para que cuando esté pulsado multiply multiplique y cuando no reste.
- El alumno puede copiar el contenido de graph24 y partir de él renombrando paquete y clase

Objetivo

- Uso de Widgets: Familiarización con HSwitchable, Gestión de eventos de teclado

Comentarios a la solución

- Ojo: al cambiar el multiply por un HToggleButton necesitaría un HGraphicLook
- Ojo: Necesitamos imágenes para mostrar. Las creamos.
- Ojo: no vale crear las imágenes con el Toolkit pues sólo sobre imágenes creadas con Component.create podemos usar el Graphics para pintar....le pasamos el scene!!!! Y pintamos el texto MULT y RESTA

Ejercicios Bloque G2-8

Ejercicio graph26

Descripción

- Implementad un nuevo HSinglelineEntry que ofrezca la función de borrar el carácter anterior cuando pulsemos BACK (VK_F9)
- Haced que si el carácter está al principio quite el foco al componente y se lo pase al siguiente componente
- Y después de lo anterior: ¿ y si queremos que el componente sólo admita caracteres numéricos ?
- Usad el ejercicio 25 para probar: bastará con cambiar un tipo de HSinglelineEntry por otro.

Objetivo

- Uso de Widgets: Familiarización con HActionable, Gestión de eventos de teclado, Problemática de Edición de texto

Comentarios a la solución

- ¿ **Cómo borramos caracteres ?** Ver **DIGITV Compatible STBs for Italy D-BOOK V.1.pdf**
BACK es VK_F9, p30
`public static final int VK_F9 = 0x78;`
En P38: Focus is lost:
 1. When object is empty (0 characters) and the keyEvent.VK_F9 is pressed

Ejercicio graph26

Comentarios a la solución

- Ojo con la versión de Stubs con la que compiléis: si lo hacéis con una incorrecta, por ejemplo: stubs113, los valores siguientes aparecen incorrectos, con lo que el Xlet no funciona bien:

```
System.out.println(">>>>>>>> RESERVED_ID_MAX:"+java.awt.AWTEvent.RESERVED_ID_MAX );
System.out.println(">>>>>>>> TEXT_START_CHANGE:"+org.havi.ui.event.HTextEvent.TEXT_START_CHANGE);
System.out.println(">>>>>>>> ITEM_LAST:"+org.havi.ui.event.HItemEvent.ITEM_LAST );
System.out.println(">>>>>>>> ADJUST_LAST:"+org.havi.ui.event.HAdjustmentEvent.ADJUST_LAST );
System.out.println(">>>>>>>> ADJUST_FIRST:"+org.havi.ui.event.HAdjustmentEvent.ADJUST_FIRST );
```

Compilando con 113 stubs oficiales de davic/havi/dvb:

```
[2#3:1] >>>>>>>> RESERVED_ID_MAX:1999
[2#3:1] >>>>>>>> TEXT_START_CHANGE:20
[2#3:1] >>>>>>>> ITEM_LAST:19
[2#3:1] >>>>>>>> ADJUST_LAST:6
[2#3:1] >>>>>>>> ADJUST_FIRST:1
```

Compilando con 112 stubs oficiales de davic/havi/dvb:

```
[3#2:1] >>>>>>>> RESERVED_ID_MAX:1999
[3#2:1] >>>>>>>> TEXT_START_CHANGE:2019
[3#2:1] >>>>>>>> ITEM_LAST:2018
[3#2:1] >>>>>>>> ADJUST_LAST:2005
[3#2:1] >>>>>>>> ADJUST_FIRST:2000
```

- Clase hecha: se aporta paquete: util_mysinglelineentry
- Numéricos: HKeyboardInputPreferred.INPUT_NUMERIC
- ¿ Cómo hacer que el componente HSingleLineEntry se ponga en modo edición al tomar el foco ?
Ver `private void initCharManagement()` en MyHSingleLineEntry

Ejercicios Bloque G2-9

Ejercicio graph27

Descripción

- XLET: code4tv.mhp112.exercise_graph27.*.
- Queremos que nuestra HScene ofrezca como fondo una imagen difuminada que además queremos poder variar usando un HRange cambiando alguno de los parámetros de creación de imágenes proporcionado en util_imagefactory.getFillBack
- El componente HRange y el botón “Aplicar” que se encargará de establecer la imagen de fondo de acuerdo a los parámetros establecidos por HRange se encontrarán en un mismo contenedor, el cual aparecerá cuando se pulse un botón creado a tal efecto en la zona de Multiplicación.
- Cuando esto ocurra debe desaparecer todo lo que hay en pantalla y se debe presentar el componente de configuración de Fondo. Por supuesto debemos de ofrecer otro botón de cerrar para volver a nuestra pantalla de multiplicaciones.
- El alumno puede copiar el contenido de graph26 y partir de él renombrando paquete y clase

Objetivo

- Uso de Widgets: Familiarización con HAdjustmentValue, Gestión de eventos de teclado, HContainers

Ejercicio graph27

Comentarios a la solución

- Se deben de meter todos los componentes de la multiplicación en un HContainer, para que sea sencillo ocultarlos.
- Se deben de crear los nuevos así mismo en otro HContainer.
- Para la imagen tenemos dos opciones: imagen de fondo en la Scene o hacemos un HContainer: MyHContainer. Mejor la Scene.
- Se aporta WidgetFactory para la creación del Range.

Ejercicios Bloque G2-10

Ejercicio graph28

Descripción

- XLET: code4tv.mhp112.exercise_graph28.*.
- El cambio entre la vista de Multiplicación y de Establecimiento de fondo se hará pulsando la tecla roja del mando a distancia.
- El alumno puede copiar el contenido de graph27 y partir de él renombrando paquete y clase

Objetivo

- Uso de Widgets: Familiarización con HAdjustmentValue, Gestión de eventos de teclado, HContainers

Comentarios a la solución

- Ojo con la gestión de la suscripción cuando nos ocultamos en el pause
- Ojo con la gestión de la suscripción cuando nos destruimos

Ejercicios Bloque G2-11

Ejercicio graph29

Descripción

- XLET: code4tv.mhp112.exercise_graph29.*.
- Queremos ver la lista de teclas de HRcEvent soportadas cargándolas en un HListGroup. Además queremos que cuando se carguen, se muestren las imágenes de representación si las hay.
 - Desde *VK_COLORED_KEY_0* a *VK_SUBTITLE*
- Cuando la representación sea un Color se habrá de mostrar una imagen de 32x32 del mismo
- Hacemos un Xlet Limpio que hereda de **code4tv.mhp112.basexlet.BaseXletAdv**. Los componentes se crean en **setComponents()**. **Se irán viendo más posibilidades y reglas correctas de codificación en los sucesivos ejercicios (foco...).**

Objetivo

- Uso de Widgets: Familiarización con HItemValue, Gestión de Teclado

Comentarios a la solución

- De donde sacamos los códigos de teclas ? Veamos HRcKeyEvent y KeyEvent
- Mejor hacemos bucle de 0 a 4000...
- Es curioso que aunque dice que no lo soporta, sí que recibimos en su momento el BACK (VK_F9)!!!

Ejercicios Bloque G2-12

Ejercicio graph30

Descripción

- XLET: code4tv.mhp112.exercise_graph30.*.
- Creemos las imágenes con una transparencia de 50 (0-255, 255 es opaca)
- Queremos que HScene tenga la imagen difuminada del graph28. (pongamos valor del HRangeValue a 120.
- El alumno puede copiar el contenido de graph27 y partir de él renombrando paquete y clase
- Haced setBackground(Blue) y NO-BACKGROUND-FILL

Objetivo

- Uso de Widgets: Familiarización con Transparencias

Comentarios a la solución

- Primero veremos que sorprendentemente el fondo de las imágenes es Azul ¿ Cómo es posible ? Si el mode de pintado de HScene es NO_BACKGROUND_FILL? Hemos de quitar el setBackground(Color.Blue).
- Pero queremos ver el BACKGROUND. ¿ cómo hacerlo ? Veamos graph30b

Ejercicio graph30b

Descripción

- Estamos trabajando en graph30.
- Establezcamos un HBackgroundDevice con Color = Blue
- Establezcamos un color de fondo del Componente HRangeValue que nos permita ver el Background con un nivel de transparencia medio, para lo cual le pedimos un Color con un nivel de transparencia del 50% hasta el Background a HGraphicsConfiguration. (0% significa 100% transparente):
HGraphicsConfiguration
public Color **getPunchThroughToBackgroundColor**(int percentage)
- Es decir, que lo que se pinte con este Color es como una ventana hacia atrás que atraviesa el Video Layer y llega hasta el Background Layer.

Objetivo

- Uso de Widgets: Familiarización con Transparencias

Comentarios a la solución

- NO FUNCIONA ¿ por qué ? Porque no se debe soportar: ved **mhp 1.1.2 A0068r1/11.1.4.2**

Ejercicios Bloque G2-13

Ejercicio graph31

Descripción

- XLET: code4tv.mhp112.exercise_graph31.*.
- Se trata de ejemplificar los diferentes modelos de Porter-Duff utilizando Componentes que heredan de HComponent o mediante HLook.
- Queremos un par de componentes solapados para cada tipo de Porter-Duff. Además queremos ver en pantalla las opciones de Alpha y Rule establecidas
- Copiad ejercicio ya hecho en src_done. Lo estudiamos.

Objetivo

- Uso de Widgets: Familiarización con Transparencias

Comentarios a la solución

- No funciona el uso de DVBAAlphaComposite ni con HLook ni heredando de HVisible.
- Interesante el tema del setXORMode: no funciona.

Ejercicios Bloque G2-14

Ejercicio graph32

Descripción

- XLET: code4tv.mhp112.exercise_graph32.*.
- Creamos un HToggleGroup para las propiedades de Horizontal, Vertical, Corner y Line_Orientation.
- Creamos un HRangeValue para controlar el espacio entre caracteres
- Creamos un HMultilineEntry y un HText
- Cada vez que cambie alguna de las propiedades se actualizarán las correspondientes al DVBTextLayoutManager que hemos asociado al HText
- Igualmente cada vez que cambiemos texto en el multiline este cambiará en el HText
- El alumno puede copiar el contenido de graph29 y partir de él renombrando paquete y clase.

Objetivo

- Uso de Widgets: Familiarización con DVBTextlayoutManager (y widgets)

Comentarios a la solución

- Atención al **repaint()**: Si no lo haces no refresca el nuevo modelo de reglas.

Ejercicios MHP PERSISTENT STORAGE

Ejercicios Bloque PSTO-1

Ejercicio psto1

Descripción

- XLET: code4tv.mhp112.exercise_psto1.Exercise_psto1.
- Grabará un fichero llamado pers1.txt en su zona ORGID/APPID
- Imprimirá los valores de Permisos y Atributos que tiene.
- Le cambiará la priority a máxima, la fecha de expiración a hoy y le dará permisos a todos los de su organización de lectura/escritura.
- Imprimimos de nuevo los valores de Permisos y Atributos que tiene.
- Finalmente: borramos el fichero (no queremos dejar cosas en el deco....)
- Hacedlo AUTOSTART
- El alumno puede copiar el contenido de ciclo5 y partir de él renombrando paquete y clase.

Objetivo

- Familiarización con el API de Persistent Storage

Comentarios a la solución

- Con Security ON no podemos ni acceder a dvb.persistent.root Ved Exercise_psto1 (security on).txt
- Hay que desactivarla. Ved Exercise_psto1 (security off).txt

Ejercicios Bloque USPREF-1

Ejercicio user1

Descripción

- XLET: code4tv.mhp112.exercise_user1.Exercise_user1.
- Imprime las Preferences del sistema
- Se suscribe a los cambios de situación de las preferences e imprime los eventos y los nombres de las que han cambiado
- Cambie la Preference del Post Code. Y si ha ido bien que imprime las preferencias otra vez para verlo.
- El alumno puede copiar el contenido de ciclo5 y partir de él renombrando paquete y clase.

Objetivo

- Familiarización con el API de User Preferences.

Comentarios a la solución

- El objeto Preference tiene un toString() que se encarga de imprimirlo “legiblemente” (en teoría).
- Sin seguridad magnífico. **Con seguridad NO es posible** más que leer las prefs que se indicaron en Security: User Language, Parental Rating, Default Font y Country Code.
- Ved Exercise_user1 (sin seguridad).txt y Exercise_user1 (CON seguridad).txt

Ejercicios XML PROCESSING

Ejercicios Bloque XML-1

Ejercicio xml1

Descripción

- XLET: code4tv.mhp112.exercise_xml1.Exercise_xml1.
- Leer un Documento XML y escribir su contenido.
- Veamos el ejercicio. No es objetivo del curso el API indicado pero sí ofrecer un punto de partida:
 - OrderHandler: hereda de DefaultHandler y es el que recibirá los eventos en el parsing.
 - CreateParser: crea el Parser usando el OrderHandler pasado.
 - Order.xml: fichero XML
 - Exercise_xml1: el Xlet ☺

Objetivo

- Familiarización con el XML Parsing API soportado por MHP

Comentarios a la solución

- **Ejecutemos haciendo que el parser sea Validador:** factory.setValidating(true). Obtenemos Exercise_xml1 (1) .txt. Que vemos ?
 1. El Parser del Deco es **Piccolo**: [www.bluecast.com](http://piccolo.sourceforge.net/), <http://piccolo.sourceforge.net/>
 2. No nos deja indicar que factory.setValidating(true); En la especificación **NO obliga** a que sea Validating.

(continúa)

Ejercicio xml1

Comentarios a la solución

- En los requerimientos MHP se indica que:
 - An implementation **MAY** support validation of XML documents against a DTD...
 - Most J2ME platforms **will likely provide a non-validating XML parser**, as it is **much less resource intensive than its validating counterpart**.
- Y para cuando se ofrece un non validating Parser:
 - Although there is no requirement to parse external entities, the parser **MUST** notify the application of unparsed entities via the DefaultHandler method `skippedEntity`.
- Por lo tanto hagamos en `CreateParser` que **no se pida un parser Validador** y ejecutemos de nuevo:
 - `factory.setValidating(false);`
- ¿ Qué obtenemos ? **Ved Exercise_xml1 (2).txt**
 - **Ahora sí hace el Parsing** aunque hay un pequeño problema cuando hace parsing del price!!
Can't parse the PRICE element - java.text.ParseException: Unparseable number: "\$19.99"
- Resolvamos el problema:
Cambiemos
`floatValue = NumberFormat.getCurrencyInstance().parse(strValue).floatValue();`
por
`floatValue = NumberFormat.getCurrencyInstance(java.util.Locale.US).parse(strValue).floatValue();`

(continúa)

Ejercicio xml1

Comentarios a la solución

- Ahora Sí. Ved [Exercise_xml1 \(3\).txt](#)

En realidad podría no haber funcionado si atendemos a las MHP 1.1.2 A0068r1 specs:

15.4 Locale support

Support of resources for the following locales is required:

- One guaranteed one (EN.UK).
- Zero (or more) implementation dependant ones.

Ejercicios Privileged Runtime Code Extension

Ejercicios Bloque PRIVILEGED-1

Previo. Operativa con STB

- **Paso 1:** En /com/vars.bat estableced la variable HTTPCLASSESDIR, que en una instalación tomcat podría ser:

C:\tomcat4.0\webapps\root\mhpclasses

Será donde se guardarán los recursos del Xlet a Ejecutar.
- **Paso 2:** Haced el Xlet normalmente salvo por que:
 - Los parámetros de arranque **NO** se van a establecer en el ait de vuestro Xlet sino en el de un **Xlet paralelo vacío** que se llamará como queráis pero que habrá de heredar de **code4tv.mhp112.remotexlet.RefXlet** o bien podrá ser este mismo pero en este caso habréis de tener cuidado de cambiar el ait cada vez que cambiéis de xlet...
- **Paso 3:** Se habrán de incluir como parámetros de arranque del RefXlet los siguientes:

classname:::nombre.de.vuestra.clase

uri:::http://10.0.0.5:8080/mhpclasses/

URI donde se encuentran las clases de vuestro XLET

Previo. Operativa con STB

- **Paso 4:** Ejecución. En lugar de llamar al comando **u XletAppendix** se llamará a :
uh EmptyRefXletAppendix VuestroXletAppendix
Por ejemplo: `uh rjm6 jm04`

Ejercicio rxml1

Descripción

- XLET: code4tv.mhp112.exercise_rxml1.Exercise_rxml1
- Carguemos remotamente el Exercise_xml1 usando el framework code4tv.mhp112.remotexlet
- Hacemos que nuestro xlet “lanzadera” herede de RefXlet.

Objetivo

- Familiarización con el API Priviledged Code Extension

Comentarios a la solución

- Ved Exercise-rxml1.txt
- Contenido principal de la AIT:

```
#Xlet parameters list
param = "classname:::code4tv.mhp112.exercise_xml1.Exercise_xml1"
param = "uri:::http://10.0.0.5:8080/mhpclasses/"
param = "scene:::4"

#Initial class name (fully qualified name) MANDATORY
class "code4tv.mhp112.exercise_rxml1.Exercise_rxml1"
```

Ejercicios Bloque PRIVILEGED-2

Ejercicio rjm6

Descripción

- XLET: code4tv.mhp112.exercise_rjm6.Exercise_rjm6
- Carguemos remotamente el Exercise_jm04 usando el framework code4tv.mhp112.remotexlet
- Hacemos que nuestro xlet “lanzadera” herede de RefXlet.

Objetivo

- Familiarización con el API Privileged Code Extension

Comentarios a la solución

- Carga remota de Exercise_jm04. Java Media Framework

Ejercicios INTER-XLET COMMUNICATION

Ejercicios Bloque INTERX-1

Ejercicio inter1c, inter1s, inter1i

Descripción

- Desarrollaremos dos Xlets independientes:
code4tv.mhp112.exercise_inter1c.Exercise_inter1c
code4tv.mhp112.exercise_inter1s.Exercise_inter1s
- Crearemos un paquete, como si fuera otro Xlet, que usarán ambos:
code4tv.mhp112.exercise_inter1i

Este paquete contendrá un interface y una clase:

- **code4tv.mhp112.exercise_inter1i.RemoteInterServer**, el cual implementará java.rmi.Remote e incluirá un método que devuelva un objeto del tipo:
- **code4tv.mhp112.exercise_inter1i.InterServerData** que será serializable.
- El Xlet servidor (Exercise_inter1s) habrá de publicarse como un objeto remoto.
- El cliente (Exercise_inter1c) deberá acceder al objeto remoto y obtener un objeto del tipo InterServerData. Acto seguido se destruirá.
- Los Xlets ids serán:
Inter1s: app 0x14 0x29
Inter1c: app 0x15 0x30

(continúa)

Ejercicio inter1c, inter1s, inter1i

Descripción

- En primer lugar se hará upload del Server **en modo “Present”** usando el ait siguiente:
`/classes/exercise_inter1s.ait`
- En segundo Lugar se hará upload del Client **en modo “Present”** usando el ait siguiente:
`/classes/exercise_inter1c.ait`
- **Un vez subidos los dos se pulsará APP en el mando y se ejecutará el Server. Una vez se haya publicado el objeto se pulsará de nuevo APP y se ejecutará el Client.**
- Para destruir sendos Xlets se pueden usar los comandos de la consola: `/com/kclient.bat` y `/com/kserver.bat`
- El alumno puede copiar el contenido de ciclo5 para cada uno de los 2 Xlets y partir de él renombrando paquete y clase.

Objetivo

- Familiarización con el API Inter-Xlet Comm.

Comentarios a la solución

- Para poder subir dos Xlets simultáneamente en el ait del client incluimos el contenido del ait del Server al final del mismo. (ver `Exercise_inter1c.ait`) Podemos usar `k` y `kc.bat` para destruirlos.
- Ved `Exercise_inter1c.txt`
- **Sólo funciona si es el Xlet el Servidor Remote, no vale crear un objeto auxiliar.**
- **Otro detalle: no es necesario desactivar la seguridad.**

Ejercicios MHP DVB System Information

Ejercicios Bloque DVBSI-1

Ejercicio sidvb1

Descripción

- XLET: code4tv.mhp112.exercise_sidvb1.Exercise_sidvb1
- Queremos obtener las SINetwork a partir de los dos métodos provistos por SIDatabase y comparar los resultados. Queremos analizar los datos de SIInformation y de los métodos propios obviando los de tipo retrieval(...)
- Nuestro XLET heredará de **code4tv.mhp112.basexlet.BaseXletAdvButtons** (util_basexlet).

Forma de proceder:

Métodos a sobre-escribir

- public void setComponents(): creamos los componentes si necesitamos (siempre super.setComponents() antes), nuestro aspecto gráfico. getHScene() devuelve la HScene
- public void xletPaused(): se llamará cuando pase a paused
- public void xletDestroyed(): se llamará cuando pase a destroyed
- public void componentShown(): se llamará cuando efectivamente la HScene se haya mostrado.
- public void go(): se llamará cuando se pulse el botón de ejecutar de la clase base

Parámetro del XLET siempre

- param = "scene::::4"

INIT.

```
public void initXlet(XletContext ctx) throws XletStateChangeException {
    super.initXlet(ctx);
    ...
}
```

- El Xlet se publicará con Control = present

(continúa)

Ejercicio sidvb1

Objetivo

- Familiarización Acceso a SI con DVB.

Comentarios a la solución

- Hay que controlar en el pause y en el destroy la parada de lectura de información.
- Según la doc. retrieveActualSINetwork debería devolver sólo una instancia pero devuelve 3!, las mismas que retrieveSINetworks
- Ver salida Exercise_sidvb1-a1.txt y Exercise_sidvb1-a2.txt para cada método
options::::[1,t][2,f][3,f][4,f][5,f][6,f][7,f][8,f][9,f][10,f][11,f][12,f][13,f][14,f][15,f][16,f][17,f][18,f]
options::::[1,f][2,f][3,f][4,f][5,f][6,t][7,f][8,f][9,f][10,f][11,f][12,f][13,f][14,f][15,f][16,f][17,f][18,f]
- NO podemos poner los params en diferentes líneas en la ait porque son muchos y no funciona.

Ejercicios Bloque DVBSI-2

Ejercicio sidvb1b

Descripción

- Queremos obtener todos los SITransportStreams de los métodos provistos por SIDatabase y SINetwork y comparar los resultados.
- Seguimos con sidvb1

Objetivo

- Familiarización Acceso a SI con DVB.

Comentarios a la solución

- Ver salida Exercise_sidvb1-b.txt
- options:::[1,t][2,f][3,t][4,f][5,f][6,f][7,f][8,f][9,f][10,t][11,f][12,f][13,f][14,f][15,f][16,f][17,f][18,f]

Ejercicios Bloque DVBSI-3

Ejercicio sidvb1c

Descripción

- Queremos obtener todos los SIBouquet y ampliar la información de los TransportStream con la llamada que aporta SIBouquet para obtener los SITransportStreamBAT
- Seguimos con sidvb1

Objetivo

- Familiarización Acceso a SI con DVB.

Comentarios a la solución

- Ver salida Exercisesia_**sidvb1-c**.txt
- No hay Bouquets: no encuentra la tabla
- options:::[1,f][2,f][3,f][4,f][5,t][6,f][7,f][8,f][9,f][10,f][11,f][12,t][13,f][14,f][15,f][16,f][17,f][18,f]

Ejercicios Bloque DVBSI-4

Ejercicio sidvb1d

Descripción

- Queremos obtener la información del SITransportStreamDescription de la TSDT del TS actual
- Seguimos con sidvb1

Objetivo

- Familiarización Acceso a SI con DVB.

Comentarios a la solución

- Ver salida Exercise_sidvb1-d.txt
- NO encuentra la tabla TSDT. Y NO Es opcional!!!
- options:::[1,f][2,f][3,f][4,f][5,f][6,f][7,f][8,t][9,f][10,f][11,f][12,f][13,f][14,f][15,f][16,f][17,f][18,f]

Ejercicios Bloque DVBSI-5

Ejercicio sidvb1e

Descripción

- Queremos obtener la información de los **Actual SIService** (la petición para Services en general necesita del original network id) de SIDatabase y de SITransportStream
 - No obtengáis los tipos de PMTService e SIEvent
 - la llamada para la network Actual como para las de NIT dan lo mismo
- Seguimos con sidvb1

Objetivo

- Familiarización Acceso a SI con DVB.

Comentarios a la solución

- Ver salida Exercise_sidvb1-e.txt
- Los Services no los ponemos pues hay que pasar el network ID
 - `if (this.info.DVBInfoDoSIServices() && false){// lo deactivo pq hay que pasar el original network ID`
- LOS ACTUAL SERVICES SÍ FUNCIONA!!!!
- Para sacar los TS necesitamos el 3
- `options:::[1,f][2,f][3,t][4,f][5,f][6,f][7,t][8,f][9,f][10,f][11,f][12,f][13,f][14,f][15,f][16,f][17,f][18,t]`

Ejercicios Bloque DVBSI-6

Ejercicio sidvb1f

Descripción

- Queremos obtener la información del PMTService de SIDatabase y de SIService
 - Ejecutamos la llamada de SIService que devuelve PMTService
- Seguimos con sidvb1

Objetivo

- Familiarización Acceso a SI con DVB.

Comentarios a la solución

- Ver salida Exercise-sidvb1-f.txt
- options:::[1,f][2,f][3,f][4,t][5,f][6,f][7,t][8,f][9,f][10,f][11,f][12,f][13,f][14,f][15,t][16,f][17,f][18,f]

Ejercicios Bloque DVBSI-7

Ejercicio sidvb1g

Descripción

- Queremos obtener la información de PMTElementaryStream de los PMTService
- Seguimos con sidvb1

Objetivo

- Familiarización Acceso a SI con DVB.

Comentarios a la solución

- Ver salida Exercise_sidvb1-g.txt
- Da error. No encuentra una tabla
- options:::[1,f][2,f][3,f][4,t][5,f][6,f][7,f][8,f][9,f][10,f][11,t][12,f][13,f][14,f][15,f][16,f][17,f][18,f]

Ejercicios Bloque DVBSI-8

Ejercicio sidvb1h

Descripción

- Queremos obtener la información de SIPresent Events de los SIService
 - (los sacaremos partiendo de los Transport Stream porque los Service directos no ofrecen los mismos pues hay que pasarles un original network id)
- Seguimos con sidvb1

Objetivo

- Familiarización Acceso a SI con DVB.

Comentarios a la solución

- Ver salida Exercise_sidvb-h.txt
- options:::[1,f][2,f][3,f][4,f][5,f][6,f][7,t][8,f][9,f][10,f][11,t][12,f][13,f][14,f][15,f][16,t][17,f][18,f]

Ejercicio sidvb1i

Descripción

- Queremos obtener la información de SIFollowing Events de los SIService en un rango amplísimo
- Seguimos con sidvb1

Objetivo

- Familiarización Acceso a SI con DVB

Comentarios a la solución

- Ver salida Exercise_sidvb-i.txt
- options:::[1,f][2,f][3,f][4,f][5,f][6,f][7,t][8,f][9,f][10,f][11,t][12,f][13,f][14,t][15,f][16,f][17,f][18,f]

Ejercicios Bloque DVBSI-9

Ejercicio sidvb1j

Descripción

- Queremos obtener la información de los Descriptors de los SINetwork
- Seguimos con sidvb1

Objetivo

- Acceso a SI

Comentarios a la solución

- Ver salida Exercise_sidvb-j.txt
- options::::[1,f][2,f][3,f][4,f][5,f][6,t][7,f][8,f][9,t][10,f][11,f][12,f][13,f][14,f][15,f][16,f][17,f][18,f]

Ejercicios MHP Java TV System Information

Ejercicios Bloque JAVSI-1

Ejercicio sijav1

Descripción

- XLET: code4tv.mhp112.exercise_sjav1.Exercise_sjav1
- Queremos obtener toda la información de SIManager y respecto a los TransportStreams derivada de los TransportStreamCollection
- La información se volcará al System.out.
- Nuestro XLET heredará de **code4tv.mhp112.basexlet.BaseXletAdvButtons** (util_basexlet).
Forma de proceder:
Ver sidvb1
- El Xlet se publicará con Control = autostart
- El alumno puede copiar parte del contenido de sidvb1 y partir de él renombrando paquete y clase.

Objetivo

- Acceso a SI con JavaTV

(continúa)

Ejercicio sijav1

Comentarios a la solución

- Tanto con seguridad como sin ella no da la información de TS, sólo Bouquets, y tampoco lo hace bien: DATA_UNAVAILABLE (esto tb puede darse por problemas de seguridad)
- OJO: java.lang.NullPointerException

```
// [3#66:2] at com.adb.tv.service.SIManagerImpl.getSupportedDimensions(Unknown Source)
// [3#66:2] at code4tv.mhp112.exercise_sjav1.JavaTVInfo.printSIManagerInfo(Unknown Source)
```

Ofrece más información con la seguridad off aunque es raro pues en teoría es posible leer todo del SI.
- options::::[1,f][2,f][3,t][4,f][5,f][6,f][7,f][8,f][9,f][10,f][11,f][12,f]
- NO hay TS
- Ved salida Exercise-sjav1-a.txt

Ejercicios Bloque JAVSI-2

Ejercicio sijav1b

Descripción

- Queremos obtener toda la información respecto a los Bouquet derivada de BouquetCollection
- Seguimos con sijav1.

Objetivo

- Acceso a SI con JavaTV

Comentarios a la solución

- Tanto con seguridad como sin ella no da la información de TS, sólo Bouquets, y tampoco lo hace bien: DATA_UNAVAILABLE (esto tb puede darse por problemas de seguridad)
- NO hay Bouquets
- options:::[1,t][2,f][3,f][4,f][5,f][6,f][7,f][8,f][9,f][10,f][11,f][12,f]
- Ved salida Exercise-sjav1-b.txt

Ejercicios Bloque JAVSI-3

Ejercicio sijav1c

Descripción

- Queremos obtener la información de los Network existentes a partir de la NetworkCollection
- Seguimos con sijav1

Objetivo

- Acceso a SI con JavaTV

Comentarios a la solución

- Ved salida Exercise-sjav1-c.txt
- options:::[1,f][2,t][3,f][4,f][5,f][6,f][7,f][8,f][9,f][10,f][11,f][12,f]
- No hay networks

Ejercicios Bloque JAVSI-4

Ejercicio sijav1d

Descripción

- Queremos obtener la información de los Service existentes a partir de SIManager. Sin entrar en los ServiceDetails
- Seguimos con sijav1

Objetivo

- Acceso a SI con JavaTV

Comentarios a la solución

- options::::[1,f][2,f][3,f][4,t][5,f][6,f][7,f][8,f][9,f][10,f][11,f][12,f]
- Ved salida Exercise-sjav1-d.txt

Ejercicio sijav1e

Descripción

- Queremos obtener la información de los ServiceDetails asociados a todos los Service que podamos identificar incluyendo los ServiceComponent y ServiceDescription. No entramos en el ProgramSchedule de ServiceDetails.
- Hay que hacer un Filtro!
- Seguimos con sijav1

Objetivo

- Acceso a SI con JavaTV

Comentarios a la solución

- options:::[1,f][2,f][3,f][4,t][5,f][6,t][7,t][8,f][9,f][10,t][11,f][12,f]
- Ved salida Exercise-sjav1-e.txt

Ejercicios Bloque JAVSI-5

Ejercicio sijav1f

Descripción

- Queremos obtener la información de ProgramEvents de los ProgramSchedule asociado a todos los Service que podamos identificar. Incluyendo de los ProgramEvents encontrados sus ServiceComponent y ProgramEventDescription...
- Queremos los current nada más.
- Seguimos con sijav1

Objetivo

- Acceso a SI con JavaTV

Comentarios a la solución

- Ved salida Exercise-sjav1-f.txt
- options:::[1,f][2,f][3,f][4,t][5,f][6,t][7,f][8,t][9,f][10,f][11,t][12,t]

Ejercicios MHP SI, Section Filtering

Ejercicios Bloque SFSI-1

Ejercicio sf1

Descripción

- XLET: code4tv.mhp112.exercise_sf1.Exercise_sf1
- Deseamos obtener toda la tabla PAT
- Nuestro XLET heredará de **code4tv.mhp112.basexlet.BaseXletAdvButtons** (util_basexlet).
Forma de proceder:
 - Ver sidvb1
- El Xlet se publicará con Control = autostart
- El alumno puede copiar parte del contenido de sidvb1 y partir de él renombrando paquete y clase.

Objetivo

- Familiarización con Section Filtering

Comentarios a la solución

- Recordemos que la tabla PAT va siempre en el PID = 0
- Recordemos que el valor de la Section de la tabla PAT table_id= 0x0000
- Ver Exercise_sf1.txt

Ejercicio sf1b

Descripción

- Deseamos obtener para el Canal La Primera (que tiene Apps) las tablas de App Signalling que contienen Streams de tipo Private Sections!!!
 - 1.- Ejecutad Exercise_sidvb1-i para saber el Locator de “La primera”
 - 2.- Extraemos la PMTElementaryStreams del Service de La Primera usando el Locator anterior
 - 3.- Cuando tengamos un ES de tipo Private Sections (0x0005) obtenemos las Sections de tipo **table_id = 0x74**, **la AIT**
- Seguimos en sf1

Objetivo

- Familiarización con Section Filtering

Comentarios a la solución

- Ver Exercise_sf1b.txt (resultado)
- El método de getStreamType() = 11 para Object o Data Carousel y 5 = Private Sections
- Filtramos solo los Stream de tipo 5. En modo Threaded!!
- Creamos el SectionFilterGroup cada vez que filtramos un PID de APP, si no da problemas, aún habiendo hecho detach.
- Atención al FinishFilterListener

Ejercicios MHP PROTOCOLS

Ejercicios Bloque PROT-1

Ejercicio tls1

Descripción

- XLET: code4tv.mhp112.exercise_tls1.Exercise_tls1
- El Xlet imprimirá el contenido de una URL pública usando el protocolo **http**
 - <http://www.verisign.com>
 - <http://www.register.com>
 - <http://www.terra.es>
- El alumno puede copiar ciclo5 y partir de él renombrando paquete y clase.
- El Xlet arrancará como autostart.

Objetivo

- Familiarización con seguridad en canal de retorno.

Comentarios a la solución

- Ved IOFactory.post(...)
- Ojo con la seguridad. Hay que desactivarla.
- Ved Exercise_tls1 http .txt

Ejercicio tls1b

Descripción

- XLET: code4tv.mhp112.exercise_tls1.Exercise_tls1
- El Xlet imprimirá el contenido de una URL pública usando el protocolo **https**
 - <https://www.verisign.com>
 - <https://www.register.com>
 - <https://www.terra.es>
 - <https://www.inem.es>
- Seguimos con tls1.
- Simplemente cambiad la URL por https.
- El Xlet arrancará como autostart.

Objetivo

- Familiarización con seguridad en canal de retorno. JSSE

Comentarios a la solución

- Ved Exercise_tls1b https .txt

Ejercicios Bloque PROT-2

Ejercicio tls1c

Descripción

- XLET: code4tv.mhp112.exercise_tls1.Exercise_tls1
- Seguimos con tls1b.
- Estableced el siguiente Handler y ejecutad:

```
URL.setURLStreamHandlerFactory(new URLStreamHandlerFactory() {  
    public URLStreamHandler createURLStreamHandler(String protocol) {  
        if (protocol.equals("https"))  
            return new com.sun.net.ssl.internal.www.protocol.https.Handler();  
        else  
            return null;  
    }  
});
```

Objetivo

- Familiarización con seguridad en canal de retorno. JSSE

Comentarios a la solución

- La clase NO se encuentra
- Ved Exercise_tls1c https .txt

Ejercicios Bloque PROT-3

Ejercicio tls1d

Descripción

- XLET: code4tv.mhp112.exercise_tls1.Exercise_tls1
- Seguimos con tls1c.
- Modificamos el Handler para bajarnos la clase **com.sun.net.ssl.internal.www.protocol.https.Handler** remotamente aprovechando las posibilidades de Remote Priviledged.
- Includid en el directorio http de bajada de classes el contenido de refs/FP1.1-jsse-cdc.jar
- Pasad la URL de donde bajarse la clase como parámetro al XLET. (recordad que acaba en /)
- Pasad el nombre de la clase handler como parámetro al XLET.

Objetivo

- Familiarización con seguridad en canal de retorno. JSSE

Comentarios a la solución

- NO nos deja instanciar la clase! : java.lang.RuntimeException: Export restriction: this JSSE **implementation is non-pluggable** at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Unknown Source) Caused by: java.lang.RuntimeException: Export restriction: this JSSE implementation is non-pluggable.
- Ved Exercise_tls1d https .txt

Ejercicios Bloque PROT-4

Ejercicio tls1e

Descripción

- XLET: code4tv.mhp112.exercise_tls1.Exercise_tls1
- Seguimos con tls1d.
- Modificamos el Handler para bajarnos la clase **com.sun.net.ssl.internal.www.protocol.https.Handler** remotamente aprovechando las posibilidades de Remote Priviledged.
- Procedemos a usar la versión **Doméstica de JSSE** que Sí es pluggable:
 - <http://java.sun.com/products/archive/jsse/>
 - refs\jsse-1_0_3_04-domestic\jsse1.0.3_04\lib\all\jsse.jar, jnet.jar y jcert.jar

Objetivo

- Familiarización con seguridad en canal de retorno. JSSE

Comentarios a la solución

- Funciona salvo en dos casos: verisign y register que dan otro tipo de problemas.
 - Verisign:**
`javax.net.ssl.SSLProtocolException: SSL message with unsupported version occurred`
at tv.osmosys.security.net.ssl.SSLInputRecord.read(Unknown Source)
 - Register:**
`java.io.IOException: unsupported keyword OID.2.5.4.17`

(continúa)

Ejercicio tls1e

Comentarios a la solución

- Ved Exercise_tls1e https .txt
- En /logs/"sun.com.net HTTP requests security.txt" se pueden ver las peticiones efectuadas para bajarse el Handler y todo lo que necesita.
- **OJO:** al imprimir la salida con `System.out.println(new String(bytesreaded));` se corta cuando encuentra un acento en la página

[2#70:1] P

[2#70:1] i¿°

Si en lugar de eso imprimís de la siguiente forma imprime todo:

```
StringBuffer sb = new StringBuffer();
for (int i =0;i<by.length;i++)
    sb.append((char)by[i]);
System.out.println(sb.toString());
```

Ejercicios Bloque PROT-5

Ejercicio tls1f

Descripción

- XLET: code4tv.mhp112.exercise_tls1.Exercise_tls1
- Seguimos con tls1e.
- Queremos imprimir los Ciphers suites soportados. Validad con *11.8.6 DVB Extensions for Cryptography*
- Igualmente queremos imprimir los providers.
- ¿ Cual es la SSLSocketFactory ?

```
System.out.println("Default SSocket Factory: "+javax.net.ssl.SSLSocketFactory.getDefault());
```

Objetivo

- Familiarización con seguridad en canal de retorno. JSSE

Comentarios a la solución

- Es correcto.
- Ved Exercise_tls1f https .txt
- Para bajarse el Xlet completo se puede ejecutar el ejercicio: rtls1, basado en el Priviledge Remote Download Framework

Ejercicio tls1g

Descripción

- Ejecutad tls1f con la seguridad activada

Objetivo

- Familiarización con seguridad en canal de retorno. JSSE

Comentarios a la solución

- No nos permite establecer la Factory.

```
[2#2:2] java.lang.SecurityException: No Permissions found for PID 0x20002 (java.lang.RuntimePermission setFactory)
[2#2:2]     at com.adb.security.AppSecurityManager.checkPermission(Unknown Source)
[2#2:2]     at com.adb.security.AppSecurityManager.checkPermission(Unknown Source)
[2#2:2]     at java.lang.SecurityManager.checkSetFactory(Unknown Source)
[2#2:2]     at java.net.URL.setURLStreamHandlerFactory(Unknown Source)
```

- Ved Exercise_tls1g https .txt
- Hay que disponer de permisos para poder establecer la Factory si deseas acceder de esta forma.
- Por otro lado si dispones de ellos es probable que sea necesario disponer de otra información para establecer los parámetros correctos de configuración para realizar llamadas HTTPS.

Ejercicios Bloque PROT-6

Ejercicio tls1h

Descripción

- XLET: code4tv.mhp112.exercise_tls1.Exercise_tls1
- Seguimos con tls1f (desactivamos la seguridad)
- Vamos a realizar las peticiones HTTPS usando Apache HTTP Client.
- Includ como fuentes del Xlet el directorio: refs\apache-commons\src\java
- Cread una función que reciba la URL y haga el trabajo.

Objetivo

- Familiarización con seguridad en canal de retorno. JSSE

Comentarios a la solución

- Funciona y además el problema que teníamos con Register ha desaparecido, permaneciendo el de Verisign:
Verisign:
`javax.net.ssl.SSLProtocolException: SSL message with unsupported version occured`
`at tv.osmosys.security.net.ssl.SSLInputRecord.read(Unknown Source)`
- Ved Exercise_tls1h https.txt

Ejercicios Bloque PROT-7

Ejercicio tls2

Descripción

- XLET: code4tv.mhp112.exercise_tls2.Exercise_tls2
 - Desarrollar un Xlet que imprima el contenido de una URL pública usando **el protocolo SSL**
- Sockets:**
- https://www.verisign.com
 - https://www.register.com
 - https://www.terra.es
 - https://www.inem.es
- Podéis copiar tls1 y modificar nombre de paquete, clase y demás.

Objetivo

- Familiarización con seguridad en canal de retorno. JSSE

Comentarios a la solución

- La conexión con Sockets no tiene mayor complicación. Parece más efectivo realizar peticiones http apoyándose en Sockets directamente antes que HTTPS.
- Ved Exercise_tls2.txt

Ejercicios MHP Service Context

Ejercicios Bloque SERCTX-1

Ejercicio sc1

Descripción

- XLET: code4tv.mhp112.exercise_sc1.Exercise_sc1.java
- Desarrollar un Xlet() que acceda al ServiceContext, y se suscriba como listener de manera que imprima
 - La información de eventos que se sucedan, incluyendo en la misma la causa del evento allí donde esté disponible.
 - Los ServiceContentHandlers y los Locators que gestionen,
 - Todo la info que ofrezca Service.
- Realizad el trabajo de forma asíncrona en una clase aparte del Xlet .
- Cambiad de canales con el mando y cada vez que cambiéis imprimid la información de ContentHandlers y demás.
- Haced el Xlet Service Bound = 0 y Control = autostart
- Origen: heredad de code4tv.mhp112.basexlet.BaseXletAdvButtons.
- Tendrá dos botones: uno con el que se lanzará el Thread y otro que matará el XLET.

Objetivo

- Familiarización con el API ServiceContext

Comentarios a la solución

- NO hay diferencia de ejecutar con Seguridad a hacerlo sin Seguridad.
- Ved Exercise_sc1.txt

Ejercicios Bloque SERCTX-2

Ejercicio sc1b

Descripción

- Ampliemos el ejercicio anterior proporcionando la lista de canales disponibles en una lista de selección (Java Media SIManager) de forma que el usuario pueda cambiar de canal.

Objetivo

- Familiarización con el API ServiceContext

Comentarios a la solución

- Ejecutamos con seguridad: Exercise_sc1b-security.txt. Da un problema de permisos: No podemos cambiar de canal.
- Ejecutamos ahora sin seguridad: Ver Exercise_sc1b.txt

Ejercicios MHP Interaction Channels

Ejercicios Bloque INTERCHA-1

Ejercicio ichan1

Descripción

- XLET: code4tv.mhp112.exercise_ichan1.Exercise_ichan1.java
- Seleccionemos un Interaction Service definido mediante una AIT residente en nuestro servidor web.
- Cread un fichero ichannel.ait vacío.
- Probar con un **javax.tv.locator.Locator** y un **org.davic.net.dvb.DvbLocator**
- Se parte de Xlet Vacío. El alumno puede copiar el ciclo5 y partir de él renombrando lo que proceda.

Objeto

- Familiarización con Interaction Channels

Comentarios a la solución

- **El SIManager NO acepta** un Locator con una url http://.....
- Ved Exercise_ichan1.txt
- No acepta Interaction Channel Apps.
- El Locator org.davic....NO lo entiende!!!

Ejercicios MHP APPs Listing & Launching

Ejercicios Bloque APPLIS-1

Ejercicio lis1

Descripción

- XLET: code4tv.mhp112.exercise_lis1.Exercise_lis1.java
- XLET que obtenga toda la información posible de las APPs a las que tenga acceso vía AppsDatabase
- No lo hagáis bound to service.
- Ponedlo autostart
- Imprimid las propiedades cada vez que cambie algo en AppsDatabase
- Partid de ciclo5 (Un Xlet básico)

Objetivo

- Familiarización con el API de APP Listing & Launching

Comentarios a la solución

- El objeto AppsDatabase no muta al cambiar de canal. El listener permanece.
- Ved Exercise_lis1.txt

Ejercicios Bloque APPLIS-2

Ejercicio lis2

Descripción

- XLET: code4tv.mhp112.exercise_lis2.Exercise_lis2.java
- Arrancará un Xlet disponible y queremos saber como evoluciona su ciclo de vida. Ejecutadla solo si no está ya arrancada y si el estado no es INVALID.
- Ejecutad el lis1 para obtener una referencia de Xlet buena para acceder por AppID: ORGID Y APPID. La cadena de La 1 de TVE es buena opción pues tiene bastantes XLETs, la EPG sería buena opción.
- Pasad el appid/orgid como parámetros al XLET y obtenedlos con util_getparams
- Haced el lis2 PRESENT para que podamos ejecutarlo en la cadena que interese.
- Hasta conseguirlo haced que intente lanzar el XLET cada vez que se arranque (después de pause)
- Haced “Salir” en el EPG. Veremos su cambio de estado.
- Partid de ciclo5 (Un Xlet básico)

Objetivo

- Familiarización con el API de APP Listing & Launching.

Comentarios a la solución

- Con seguridad ON es posible controlar el ciclo de vida de las del Service
- **Un detalle sin importancia (☺):** getXletProperty(“dvb.app.id”) y getXletProperty(“dvb.org.id”) te devuelve el String del valor en hexadecimal, si quieres compararlo con el AppID debes pasarlo a int!!!: Integer.decode(“0x”+ctx.getXletProperty(“dvb.org.id”).intValue());
- Ved Exercise_lis2.txt. Cambiad de canal de la 1 a la 2 para que la appsdatabase incluya la EPG.

Ejercicios MHP Stored Services & Cached Apps

Ejercicios Bloque STOSER-1

Ejercicio cach1

Descripción

- XLET: code4tv.mhp112.exercise_cach1.Exercise_cach1.java
- Punto de partida: copiad el lis1 y renombrad.
- Mostrad las ExtendedAttributes.
- Además queremos saber qué StoredServices existen, y conocer toda la información posible de los mismos (hasta AppIDs).
- Como no os saldrán demasiados (☺) probad con el filtro a null, para que salgan todos e imprimís la información de los normales también.
- Hacemos que sólo se imprima la información una vez y después se destruye el Xlet. De esta forma podemos lanzarlo n veces en el mismo contexto.
- Además es Service bound y Present
- Interesante:
 - » Name :**Canal Ingenieria**
 - » [3#2:1] Locator :dvb://22d4.a.6e
 - » [3#2:1] ServiceType :DATA_BROADCAST (es de tipo IP Signalling/MPE)
 - » Name :**GUIDE PLUS+**
 - » [3#38:1] Locator :dvb://22d4.a.10d
 - » [3#38:1] ServiceType :DATA_BROADCAST
 - » Name : **¿? No lo define**
 - » [3#38:1] Locator :dvb://22d4.a.ffe
 - » [3#38:1] ServiceType :DATA_BROADCAST

Ejercicio cach1

Objetivo

- Familiarización con Storage & Cache API

Comentarios a la solución

- Interesante: el locator del Xlet es siempre el del service **+.0**:
 - `dvb.transport.oc.locator :dvb://22d4.3f4.213.0`
- Ved Exercise_cach1.txt

Ejercicio cach2

Descripción

- XLET: code4tv.mhp112.exercise_cach2.Exercise_cach2.java
- Queremos crear un StoredService llamado: Code4TV
- Haced que cada vez que se ejecute si ya está creado lo elimine y si no lo cree.
- Si hay tiempo:
 - Cuando lo cree queremos almacenar nuestro Exercise_cach1. Hay que hacer el Application Descriptor File.
 - Punto de partida: copiad Exercise_cach1 y renombrad, así disponemos de los apis de información.
 - Seguimos el mismo esquema: service bound y present, una ejecución y destroy.
 - Cuando es creado NO aparece en la lista de canales de la TV pues no tiene APPS.
 - Recordemos incluir en el AIT file ambos xlets. Y Recordemos que primero se sube el cach1 solo con su ADF y luego cach2 con la referencia a aquel en el AIT file (recuerda inter-xlet).

Objetivo

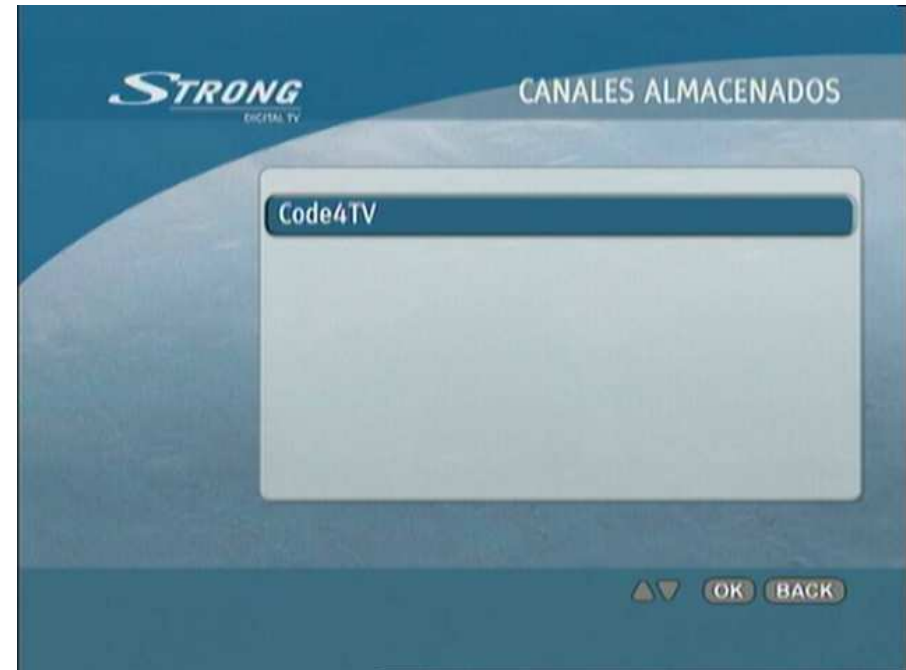
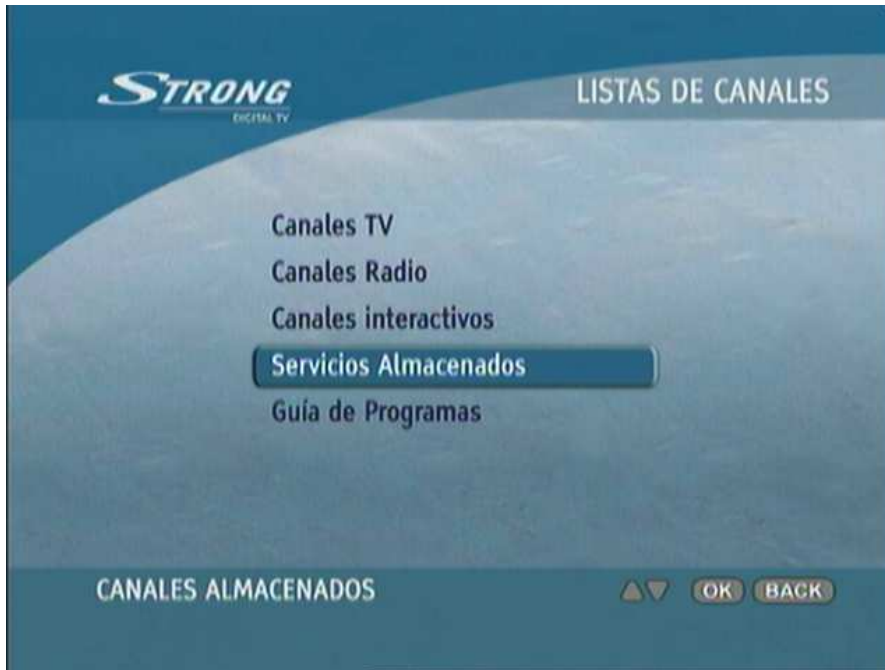
- Familiarización con Storage & Cache API

Comentarios a la solución

- Hay que desactivar la seguridad. Ved Exercise_cach2 con security.txt
- Hay que hacer el Application Descriptor File. Veamos si hay suerte y podemos prescindir del application_storage_descriptor. Ved /logs/dvb.storage.00000014.0029. **NO funciona** hay que hacer el **application_storage_descriptor**.
- Recordemos ejecutar dos veces para crear el service y eliminarlo después.

Comentarios a la solución

- Podemos crear el Service, aunque no la App al no disponer de un App Signalling real



Ejercicios MHP Return Channel Connection API

Ejercicios Bloque RC-1

Ejercicio rcapi1

Descripción

- XLET: code4tv.mhp112.exercise_rcapi1.Exercise_rcapi1.java
- 1º Imprima la información que pueda de los RCInterface existentes.
- 2º Intente una conexión con un ConnectionRCInterface. No podrá pues el Strong no tiene modem. Intentadlo con la que tenga.
- Usad todos los APIS Listener: tanto de Resource Management del RCInterfaceManager como de la conexión. Liberad si no lográis conectar.
- Hacedlo autostart.
- Partid de ciclo5 (Un Xlet básico)

Objetivo

- Familiarización con el Return Channel Connection API

Comentarios a la solución

- Con Security ON recordemos que no se pueden establecer conexiones.
- Ved Exercise_rcapi1 (strong sec on).txt, Exercise_rcapi1 (strong sec off).txt
- Interesante el interface: **org.dvb.net.rc.EthRCInterface**
- Ved Exercise_rcapi1 (engel sec on).txt, Exercise_rcapi1 (engel sec off).txt
- Interesante el interface: org.dvb.net.rc.ModemRCInterface
- **Ojo:** llamando a mi móvil sin seguridad NO PREGUNTA!!!! Y LLAMA!!!!

Ejercicios MHP. Java Media Framework

Ejercicios Bloque JMF-1

Ejercicio jm00

Descripción

- XLET: code4tv.mhp112.exercise_jm0o.Exercise_jm00.java
- Queremos saber la lista de Clases Manejadoras de Datasources en función del nombre del **protocolo** usado.
- Queremos saber lista de clases manejadoras de **tipo de contenido**.
- Intentad cargar cada clase encontrada indicando si se pudo o no.
- Partid de ciclo5.
- Hacedlo autostart

Objetivo

- Familiarización con JMF

Comentarios a la solución

- Recordemos los mime/types (FORMATS & RESTRICTIONS)
- Protocolos: http, https, dvb, file, ocap
- Para los protocols la cosa va bien, para los handlers no es obvia la clase y no funciona.
- Ved Exercise_jm00.txt

Ejercicios Bloque JMF-2

Ejercicio jm00b

Descripción

- Añadir al ejercicio anterior un método que obtenga el ServiceMediaHandler del DVBSERVICE que se está presentando y nos imprima TODOS los Control que ofrece.

Objetivo

- Familiarización con JMF

Comentarios a la solución

- Ved Exercise_jm00b.txt

Ejercicios Bloque JMF-3

Ejercicio jm00c

Descripción

- Añadir al ejercicio anterior un método que usando MediaSelectControl establezca un nuevo Stream de Audio mediante el método `select(Locator)`.
- Hemos de ejecutar antes `sidvb1` con los parámetros siguientes para obtener los Locators:
 - `scene:::4;options:::[1,f][2,t][3,f][4,t][5,f][6,f][7,f][8,f][9,f][10,f][11,t][12,f][13,f][14,f][15,f][16,f][17,f][18,f]`
 - Salida en `Exercise-jm00c-SIDVB1.txt`
- Hemos de saber cual es el Stream de Audio actualmente seleccionado para poner el alternativo.
- Lo ideal es pasar los Locators como parámetros al `Xlet`.

Objetivo

- Familiarización con JMF

Comentarios a la solución

- Resultados obtenidos estando sintonizados en LaSexta:
 - Service Locator: `dvb://22d4.2.154`
 - Streams audio disponibles:
 - » `dvb://22d4.2.154.5df`
 - » `dvb://22d4.2.154.5e0`

(continúa)

Ejercicio jm00c

Comentarios a la solución

- **FUNCIONA! PERO:** si hacéis select con un solo Locator -> SOLO reproducirá ese Locator!! En nuestro caso hemos dejado solo el Audio !!!!!
- Hay que hacer un select(Locator[]) con todos los que queremos reproducir).
- Ved Exercise_jm00c.txt

Ejercicios Bloque JMF-4

Ejercicio jm01

Descripción

- XLET: code4tv.mhp112.exercise_jm01.Exercise_jm01.java. Partid de jm00 renombrando.
- Queremos saber los Control que ofrece el Player que gestiona el DVB Service actual indicando si tenemos Component gráfico.
- ¿ Cual es la resolución full-screen ?
- A partir del **AWTVideoSizeControl** obtened toda la información relativa a Video Size (hay 3 métodos...)
- A partir del **VideoPresentationControl** obtened toda la información relativa a Video Size (hay 6 métodos...) y clipping y scaling factors
- A partir de **BackgroundVideoPresentationControl** queremos obtener toda la información del VideoTransformation Actual.
- A partir de **VideoFormatControl** queremos conocer toda la información que ofrezca
- Hacedlo Service bound = 1, es preferible dada la gestión intra-canales de los Players y AutoStart.
- Hereda de **BaseXletAdvButtons (util_basexlet)**:
 - Forma de operar: Basta con implementar *public void go()*;
- Haced una clase que reciba el XletContext y dentro de ella se obtenga y se haga todo.
- Cuando esté ejecutando: cambiad el aspect ratio del STB y lanzad el proceso de nuevo.

Objetivo

- Familiarización con JMF al manejar el Video

(continúa)

Ejercicio jm01

Comentarios a la solución

- Aportamos util_jmfcontrols
- Curiosamente no hay ningún Component gráfico asociado a ningún Control sin embargo lo importante es el método getVisualComponent del Player!!!! Ved como sí existe un tv.osmosys.javax.media.ComponentBasedPlayerControll de forma que podremos mostrar el contenido en un AWT Component
- Si cambiamos el aspect ratio del DECO se actualizan los valores de VideoFormat a 2:
 - [3#73:1] getActiveFormatDefinition _AFD_____ :AFD_4_3
 - [3#73:1] getAspectRatio _____ASPECT_RATIO___ :ASPECT_RATIO_4_3
 - [3#73:1] getDecoderFormatConversion __DFC_____ :DFC_PROCESSING_UNKNOWN
 - [3#73:1] getDisplayAspectRatio_____DAR_____ :DAR_16_9
- Si lo dejamos normal en 4:3
 - [3#72:1] getActiveFormatDefinition _AFD_____ :AFD_4_3
 - [3#72:1] getAspectRatio _____ASPECT_RATIO___ :ASPECT_RATIO_4_3
 - [3#72:1] getDecoderFormatConversion __DFC_____ :DFC_PROCESSING_UNKNOWN
 - [3#72:1] getDisplayAspectRatio_____DAR_____ :DAR_4_3
- Estos cambios NO afectan a la información de los otros controls, para ellos no ha habido ninguna transformación.
- La app se mueve manteniendo su posición respecto al video con contenido.
- Ved Exercise_jm01.txt

Ejercicio jm02

Descripción

- XLET: code4tv.mhp112.exercise_jm02.Exercise_jm02.java
- Deseamos reescalar el vídeo de la señal usando **AWTVideoSizeControl** con los siguientes datos:
 - source[x=50,y=50,width=200,height=300]
 - dest[x=300,y=80,width=200,height=200]
- Deseamos ofrecer al usuario además un botón que le permita cambiar de modo: Transformado o Normal cada vez que se pulse
- Imprimid la resolución actual, la solicitada y la obtenida
- Cada vez que se produzca el cambio de escalado de video imprimid la info que se obtenía en jm01
- Haced una clase que reciba el Control que necesita y que tenga un método que cambie el estado de visión de trans a normal y viceversa cada vez que se le llame.
- Hacedlo Service bound = 1 es preferible dada la gestión intra-canales de los Players, y AutoStart.
- Copiad jm01 y renombrad. Sobrescribid setComponents() para incluir el código de construcción del nuevo botón. public void go(); imprimirá toda la información cuando lo deseemos (run)

Objetivo

- Familiarización con JMF al manejar el Video.

(continúa)

Ejercicio jm02

Comentarios a la solución

- Es posible que no se obtenga un **AWTVideoSizeControl**, cambio de canal.
- Hemos solicitado y obtenido lo siguiente:

Obtained Size :

source[x=50,y=50, **width=200**,height=300]

dest[x=300,y=80, **width=200**,height=200]

scaleX=1.0, scaleY=0.6666667

Y hemos pasado de:

[2#31:1] getVideoSize _____:java.awt.Dimension[width=720,height=576]

[2#31:1] getInputVideoSize_____:java.awt.Dimension[width=720,height=576]

[2#31:1] getActiveVideoArea_____ :org.havi.ui.HScreenRectangle[0.0,0.0,0.0,0.0]

[2#31:1] getActiveVideoAreaOnScreen___:org.havi.ui.HScreenRectangle[0.0,0.0,0.0,0.0]

[2#31:1] getTotalVideoArea_____ :org.havi.ui.HScreenRectangle[0.0,0.0,1.0,1.0]

[2#31:1] getTotalVideoAreaOnScreen____:org.havi.ui.HScreenRectangle[0.0,0.0,1.0,1.0]

[2#31:1] getClipRegion_____ :java.awt.Rectangle[x=0,y=0,width=720,height=576]

a

– [2#31:1] getVideoSize _____:java.awt.Dimension[width=199,height=299]

– [2#31:1] getInputVideoSize_____ :java.awt.Dimension[width=720,height=576]

– [2#31:1] getActiveVideoArea_____ :org.havi.ui.HScreenRectangle[0.4166565,0.0,0.1388855,0.0]

– [2#31:1] getActiveVideoAreaOnScreen___:org.havi.ui.HScreenRectangle[0.4166565,0.0,0.1388855,0.0]

– [2#31:1] getTotalVideoArea_____ :org.havi.ui.HScreenRectangle[0.06944275,0.08679199,0.277771,0.52082825]

– [2#31:1] getTotalVideoAreaOnScreen_ :org.havi.ui.HScreenRectangle[0.4166565,0.1388855,0.277771,0.34721375]

– [2#31:1] getClipRegion_____ :java.awt.Rectangle[x=49,y=49,width=199,height=299]

- Los datos de ActiveVideoArea no son válidos, Es posible que no llegue información de píxeles con contenido en la señal.

(continúa)

Ejercicio jm02

Comentarios a la solución

Obtained Size :

source[x=50,y=50,width=200,height=300]

dest[x=300,y=80,width=200,height=200]

scaleX=1.0, scaleY=0.6666667

En Normalised sería: [X= 300/720= 0.4166565, Y = 80/576=0.1388888, W=200/720=0.277777, H=200/576=0.3472222]

- **ActiveVideoArea = ActiveVideoAreaOnScreen**

- getActiveVideoArea:org.havi.ui.HScreenRectangle[0.4166565,0.0,0.1388855,0.0]

KO: De donde salen los datos ?. Ya antes de hacer la transformación los valores eran 0.

X= 0.4166565 = 300 / 720 -> posición X en normalizadas.

- **TotalVideoArea**

- getTotalVideoArea:org.havi.ui.HScreenRectangle[0.06944275,0.08679199,0.277771,0.52082825]

- getTotalVideoArea:org.havi.ui.HScreenRectangle[**0.06944275=50/720** ,**0.08679199 = 50/576** ,**0.277771 = 200/720** ,**0.52082825 = 300/576**]

OK: coordenadas normalizadas del video “Cortado”: source

- **TotalVideoAreaOnScreen**

- [2#31:1] getTotalVideoAreaOnScreen_:org.havi.ui.HScreenRectangle[0.4166565,0.1388855,0.277771,0.34721375]

OK: coordenadas normalizadas de la ventana de video en el Screen: Destiny

Hay alguna diferencia mínima.

Ejercicio jm02b

Descripción

- Ampliamos jm02 con el tipo de transformación **BackgroundVideoPresentationControl**
- Idéntica operativa pero ahora hacemos una transformación distinta
- Includ un parámetro del Xlet que nos diga la trans que queremos:

```
public static final int AWTSIZE    =0;  
public static final int BACK      =1;
```

Objetivo

- Familiarización con JMF al manejar el Video.

Comentarios a la solución

- Ver Exercise_jm2b.txt

Ejercicio jm02c

Descripción

- Ampliamos jm02b con el tipo de transformación PAN & SCAN
- Idéntica operativa pero ahora hacemos una transformación distinta

```
public static final int AWTSIZE    =0;  
public static final int BACK      =1;  
public static final int PANSCAN    =2;
```

Objetivo

- Familiarización con JMF al manejar el Video.

Comentarios a la solución

- Aparentemente hace la transformación pero no se aprecia ninguna modificación en los mismos, y por lo tanto tampoco en el video.
- Ver Exercise_jm2c.txt

Ejercicio jm02d

Descripción

- Ampliamos jm02c con el tipo de transformación **Component**
- Idéntica operativa pero ahora hacemos una transformación distinta
- Dos opciones: una sin modificar el Video de origen y otra modificando.....

```
public static final int AWTSIZE           =0;
public static final int BACK              =1;
public static final int PANSCAN          =2;
public static final int COMPONENT        =3;
public static final int COMPONENT_CLIP  =4;
```

Objetivo

- Familiarización con JMF al manejar el Video.

Comentarios a la solución

- Recordemos que:
 - Si HAY ComponentBasedPlayerControl entonces se podrá ver el vídeo en AWT
 - EL VideoPresentationControl es OPCIONAL por si queremos ajustar antes el video original.
- **OJO:** después de obtener el componente hay que hacer repaint() en su contenedor o en la scene!!!!

(continúa)

Ejercicio jm02d

Comentarios a la solución

- Si dejamos service bound =0 entonces hay problemas con la situación de los Control, es preferible almacenar el estado y reproducirlo al cambiar de canal desde 0.
- Exercise_jm2d-noclip.txt y Exercise_jm2d-clip.txt
- getVideoSize() devuelve el tamaño de vídeo original, no toma en cuenta las transformaciones al AWT Component.

Ejercicio jm03

Descripción

- XLET: code4tv.mhp112.exercise_jm03.Exercise_jm03.java
- Oigamos algo de música! org.havi.ui.HSound.
- Pasemos el path como parámetro del Xlet: http://10.0.0.5:8080/escoba.mp2. escoba.mp2 está en /heavy_resources/escoba.mp2
- Hemos de cargar la música sin “molestar” al usuario:-> Modo Threaded, mostrando un mensaje que indique que estamos haciendo el Loading.
- Hacedlo de forma que lea los bytes o bien una URL: (Ayuda: IOFactory.getBytesFile(path))
- Hagamos que sea Bound = 0 y Autostart
- Hereda de BaseXletAdvButtons.

Objetivo

- Familiarización con JMF al manejar HSound.

Comentarios a la solución

- Con Security ERROR. Sin él OK
java.lang.SecurityException: No Permissions found for PID 0xd0002 (java.net.SocketPermission 10.0.0.5:8080 connect,resolve)

(continúa)

Ejercicio jm03

Comentarios a la solución

- La experiencia intentando obtener los datos de un InputStream cuando no conocemos el tamaño es que crear buffers muy importantes dan error!!! OJO:
 - error de LAN en Strong:===00004280=====LAN911x software reset=====
- Finalmente conocer el tamaño es importante dado que reduce los buffer temporales
- Ved Exercise_jm03.txt

Ejercicio jm03b

Descripción

- Seguimos con jm03. Probad con /heavy_resources/alice.mp3

Objetivo

- Familiarización con JMF al manejar HSound.

Comentarios a la solución

- El fichero está tomcat4.0/webapps/root/alice.mp3
- ENTIENDE MPEG-1 LAYER 3, !!!!!!!!!!!!!!! MP3 !!!!!!!!!

Ejercicio jm04

Descripción

- XLET: code4tv.mhp112.exercise_jm04.Exercise_jm04.java
- Oigamos más música pero usemos los Players. Según la doc sólo podíamos oír sonidos con HSound y usando JMF con ficheros...
- Pasad el path mediante un parámetro del XLET.
- Probad pasando una URL: `http://10.0.0.5:8080/escoba.mp2`
- Colocamos escoba.mp2 en nuestro directorio base del fuente y
- Le pasamos el path `file:/home/escoba.mp2` (**file: sólo se usa con direccionamiento absoluto**)
- Solicitemos primero un Realized, y después un Prefetch! Haced que el Thread devuelva en el callback el Player ya Prefetched! Y así pueda ser el sustituto del HSound.
- Queremos saber cuando acaba la canción: EndOfMediaEvent!
- Queremos saber lo que dura el Clip en segundos y comenzar a la mitad!
- Hacedlo Bound = 1, AutoStart

Objetivo

- Familiarización con JMF al manejar Locators y Players

(continúa)

Ejercicio jm04

Comentarios a la solución

- No tenemos duración definida con escoba.mp2 de forma que no podemos arrancar a la mitad.
- Ahora cuando hacemos stop se para y luego sigue donde se quedó no como el HSound!!!
- Con URL no Funciona: Exercise-jm04-error_url.txt. Con File sí funciona! Problema: el tamaño del fichero. Un momento!!!, en las MHP 1.1.2 specs se dice lo siguiente:
 - 11.11.12 Support for the HTTP protocol in DVB-J
 - In MHP terminals where an HTTP protocol (clause 6.3.7.1 or clause 6.3.7.2) is supported, the following classes and methods shall support the HTTP protocol concerned. In MHP terminals where the HTTPS protocol (clause 6.3.7.3) is supported, the following class and methods shall support that protocol.
 - **The constructor for javax.media.MediaLocator for referencing audio files intended to be played from memory.**
 - **Methods on javax.media.Manager accepting javax.media.MediaLocator as input parameters for constructing JMF players for audio files intended to be played from memory.**
- Si hacemos la pertinente modificación en nuestro ejercicio. Vemos que lamentablemente NO funciona: Exercise-jm04 error_url.txt. No encuentra Player.

(continúa)

Ejercicio jm04

Comentarios a la solución

- Sin embargo si vemos el log observamos un matiz interesante:

Leyendo de file:/

- [2#10:1] path:file:/home/escoba.mp2 usingMediaLocator:true
- [2#10:1] Content Type:**audio/mpeg**

Leyendo de URL:

- [3#94:1] path:http://10.0.0.5:8080/escoba.mp2 usingMediaLocator:true: usingMediaLocatorWithURL true
- [3#94:1] Content Type:**audio/x-mpeg**
- [3#94:2] javax.media.NoPlayerException

El content-type podría ser el problema

En [tomcat_dir]\conf\web.xml ajustamos el mime para que devuelva audio/mpeg y probamos de nuevo

FUNCIONA!!!!

Ved Exercise-jm04-OK_URL.txt

Por supuesto **con la seguridad activa** y sin permisos para acceder al host falla:

```
[2#2:1] path:http://10.0.0.5:8080/escoba.mp2 usingMediaLocator:true: usingMediaLocatorWithURL true
[2#2:2] java.io.IOException: <DataSrc loc = http://10.0.0.5:8080/escoba.mp2, content = null>: connect() failed
[2#2:2]         at tv.osmosys.javax.media.protocol.http.DataSource.connect(Unknown Source)
[2#2:2]         at javax.media.Manager.createDataSource(Unknown Source)
```

Ejercicios MHP DSM-CC. Sistema de Archivos

Ejercicios Bloque DSMCC1-1

Ejercicio dsm1

Previo

Disponemos de un directorio `heavy_resources/` en el que residen una serie de imágenes para usar.

Descripción

- XLET: `code4tv.mhp112.exercise_dsm1.Exercise_dsm1.java`
- Leamos un fichero imagen en modo síncrono y establezcámoslo como fondo de la HScene
- El Fichero a leer: `images/mandrill.jpg` [se crea estructura bajo el directorio del src del ejercicio: `exercise_dsm1/images/exercise_dsm1/mandrill.jpg`]
- Probad los siguientes casos:
 - `images/exercise_dsm1/mandrill.jpg`
 - `/images/exercise_dsm1/mandrill.jpg`
 - `/home/images/exercise_dsm1/mandrill.jpg`
- Cread un método **`readFileSync()`**. Recibimos el **path** del fichero como parámetro del XLET
- Partimos de `BaseXletAdvButtons`. Escribid en el `System.out` para mostrar OK u ERROR.
- Usad `BorderLayout`. `contenedor.setLayout(new BorderLayout());`
 - `message = Utils.getHText(0, 0, 100, 50, "Reading", Color.gray, HVisible.BACKGROUND_FILL, Color.white, fBody);`
 - `contenedor.add(message,"Center");`
- Hacedlo `Bound = 1, AutoStart`
- Probad al final con un GIF a ver que pasa.

(continúa)

Ejercicio dsm1

Objetivo

- Familiarización con DSMCC

Comentarios a la solución

- Resultados
 - images/exercise_dsm1/mandrill.jpg : ok
 - /images/exercise_dsm1/mandrill.jpg: error
 - /home/images/exercise_dsm1/mandrill.jpg: ok
- El método **public static byte[]** getFileBytesContent(DSMCCObject dsmccObj) imprime un LOG de información respecto a formatos de URLs. Como vemos, si intentamos crear un Locator de tipo File con javax.tv.locator.Locator da error!!! Es obvio que sólo le gustan referencias de contenido streaming, sin embargo con javax.media.MediaLocator es perfecto.
- En IOFactory basta cambiar los valores de las siguientes variables para deshabilitar el LOG:
 - public static boolean *PRINT_PATH* = true;
 - public static boolean *TEST_LOCATOR* = true;

Ejercicio dsm1b

Descripción

- Leamos un fichero en modo asíncrono.
- Creemos un método distinto: `readFileAsync()`; y recibamos como parámetro del xlet el modo de carga: `sync` o `async`
- Imprimid:

```
System.out.println("!----->AbsolutePath "+dsm.getAbsolutePath());
System.out.println("!----->CanonicalPath "+dsm.getCanonicalPath());
System.out.println("!----->Path "+dsm.getPath());
System.out.println("!----->URL "+dsm.getURL());
```

Objetivo

- Familiarización con DSMCC

Comentarios a la solución

- En modo asíncrono da la siguiente exception del tipo: No nos permite leer en modo asíncrono sin montar un ServiceDomain:

```
org.dvb.dsmcc.InvalidPathNameException: Cannot find ServiceDomain for
DSMCCObject(images/exercise23/mandrill.jpg) (probably ServiceDomain is detached).
[2#86:2] at org.dvb.dsmcc.DSMCCObject.asynchronousLoad(Unknown Source)
```
- Ved: Exercise-dsm1-error-servicedomain.txt

(continúa)

Ejercicio dsm1b

Comentarios a la solución

- **Ojo:** la llamada `synchronousLoad()` falla igualmente. Es curioso que después llama a `asynchronousLoad`
- Para probar la lógica hacemos un truco: funciona igual y en modo `async`.
`IOFactory.asynchronousLoad(dsmccObj, this)`
- ¿ **Será capaz de leer un GIF ?** Según MHP 1.1.2 no está obligado. **Lo lee.**

Ejercicios Bloque DSMCC1-2

Ejercicio dsm2

Descripción

- XLET: code4tv.mhp112.exercise_dsm2.Exercise_dsm2.java
- Queremos obtener del Programa actual todos los ES PMTElementaryStream, y sobre aquellos que sean del tipo Object Carousel, montar el ServiceDomain y listar el contenido del mismo.
 - Primero: retrievePMTService para el Service del programa actual.
 - Segundo: retrievePMTElementaryStreams para ese PMTService y así obtenemos los PMTElementaryStream del tipo que interesa
- Recordemos: **0x0B**: Object o Data Carousels (Ficheros, Event Objects...): **Object Carousel y Data Carousel Messages**
- Hereda de BaseXletAdvButtons. Basta con implementar **go()**;
- Hacedlo Bound = 1 y Present.

Objetivo

- Familiarización con DSMCC

Comentarios a la solución

- Ved Exercise-dsm2.txt

Ejercicio dsm2b

Descripción

- Una vez funcione dsm2 leed algunos de los ficheros de forma asíncrona, a ver si ahora funciona!!!!

Objetivo

- Familiarización con DSMCC

Comentarios a la solución

- Ved Exercise-dsm2b.txt

Ejercicios MHP Tuning API

Ejercicios Bloque TUNING-1

Ejercicio tun1

Descripción

- XLET: code4tv.mhp112.exercise_tun1.Exercise_tun1.java
- Desarrollad un Xlet que Imprima la información que pueda a partir de StreamTable: Locators, TS, Services, ES....
- Partimos de un xlet vacío: ciclo5

Objetivo

- Familiarización con el Tuning API

Comentarios a la solución

- Ved Exercise_tun1.txt
- Es curioso lo que tarda en acceder a toda la información.

Ejercicio tun2

Descripción

- XLET: code4tv.mhp112.exercise_tun2.Exercise_tun2.java
- Queremos:
 1. Mostrar los NetworkInterfaces Existentes con toda la información posible de los mismos.
 2. Efectuar una sintonización a un TS concreto distinto del TS que está sintonizado en el momento.
 3. Poder hacer el tuning por Locator o por TS.
 4. Que en el caso de Locator el tuning se haga a un **Service** del TS destino.
 5. Includ como parámetro al Xlet: "bylocator" = true/false
 6. Que sólo tuncie una vez y que el proceso de Tuning se efectúe en modo Threaded en clase aparte (Runnable)
- Hacedlo PRESENT y bound =1
- Partid de tun1 (copiadlo y renombrad)

Objetivo

- Familiarización con el Tuning API

Comentarios a la solución

- **Ojo:** org.davic.net.Locator no se puede instanciar directamente: hay que usar: org.davic.net.dvb.DvbLocator.

(continúa)

Ejercicio tun2

Comentarios a la solución

- Con la seguridad activada no podemos hacer Tuning: Ved Exercise_tun2(security on).txt
- Desactivad después y ejecutad. Ved Exercise_tun2-locator.txt y Exercise_tun2-ts.txt para cada modo.
- En mi ejercicio sintonizo un Locator o un TS....Para el caso del Locator me apoyo en el método de impresión de información de tun1 para devolver un Locator de un Service que nos interese. Ojo a la obtención del id a partir del locator: `int getTSIDFromLocator(loc)`
- Como habréis podido observar en el **Exercise_tun2** después de sintonizar, incluso con Locator en el que aparece un Service, no vemos **NADA**. Así que cuando esto ocurre **el Player no se re-inicia por defecto!!!** Es uno de los efectos colaterales de este API. Lo que habría que hacer es acto seguido seleccionar un Service con java TV Service Selection API.
- Hay que estar seguros de que los canales que nos devuelve el API están correctamente sintonizados en el deco. Puede ocurrir que debas “re-sintonizar” el STB.

Ejercicio tun2b

Descripción

- Seguimos con tun2
- Haced que en el caso de Locator haga el Select del Service con JavaTV después de haber hecho tuning del nuevo TS

Objetivo

- Familiarización con el Tuning API

Comentarios a la solución

Ejercicios MHP Internet Clients

Ejercicios Bloque INTERNET-1

Ejercicio iclient1

Descripción

- XLET: code4tv.mhp112.exercise_iclient1.Exercise_iclient1.java
- Obtened los Internet Client Services Disponibles y muestre toda la información que pueda de ellos.
- Además: si encuentra alguno, cambiemos de canal hacia él.
- Xlet básico, partid de ciclo5.
- Hacedlo present

Objetivo

- Familiarización con el Internet Client API

Comentarios a la solución

- Como vemos en la salida: Exercise_iclient1.txt no se soportan los Internet Clients.

Ejercicios MHP SATSA

Ejercicios Bloque SATSA-1

Ejercicio sat1

Descripción

- XLET: code4tv.mhp112.exercise_sat1.Exercise_sat1.java
- Abramos una conexión con nuestro smart card reader. (recordad meter vuestra tarjeta/DNI Electrónico)
- Nos suscribimos al Listener del SmartcardReader
- Xlet básico, partid de ciclo5.
- Hacedlo Autostart.

Objetivo

- Conexión a Smartcard

Comentarios a la solución

- Si intentamos usar openDefaultConnection nos da un error el STB Strong 5510 indicándonos que no dispone de ese método la clase SmartCardReader. Curiosamente este deco incorpora la parte de SmartcardReader de MHP 1.1.3
- Usemos openRawConnection, pero cambiando el path del proyecto para compilar con los Stubs de MHP 1.1.3: \libs\havi_dvb_davic\mhp1.1.3 tam685r16_mhp11_javadoc\mhp113-havi_dvb_davic_tam685.jar
- Exercise_sat1.txt

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