



MHP. Entorno de Trabajo MHP 1.1.2

Stubs 1.1.2

- Este proyecto se publica con el ánimo de ayudar a la comunidad de desarrolladores MHP

This entire project is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of the software updated and/or generated or the use of the documentation provided

El proyecto completo se proporciona sin ningún tipo de garantía expresa o implícita. En ningún caso los autores serán responsables de cualquier posible daño supuestamente derivado ni del uso del software, ni de su actualización/modificación, ni de aquel generado ni de la documentación proporcionada.

- Un recordatorio: base MHP 1.1.2

PBP1.1

jvm

Davic

org.davic

Basic MPEG concepts
Resource management
Access to CA information
Tuning API: org.davic.net.tuning

DVB,

org.dvb

Access to SI/PSI
Application management
Access to DSM-CC object carousels
Return channel session management
Inter-Xlet communication: org.dvb.io.ixc
DVB-HTML
Internet clients

Havi

org.havi

Video/graphics integration
UI widgets for consumer systems and TV screens

JavaTV

javax.tv

Xlet classes and infrastructure
Service selection
SI/PSI access
Media control extensions

Seguridad

https/ssl

Java Media Framework

javax.media

Presentación de contenido Media

DOM, Document Object Model.

org.w3c.dom

Sólo para DVB-HTML

XML, XML Processing (Parsing)

org.xml.sax

javax.xml

SATSA, Smartcard comm

javax.microedition.io.apdu

En definitiva, siguiendo la última documentación de MHP 1.1.2:

tam668r23_11xdraft_20061115.pdf

www.mhp.org

...los Stubs MHP 1.1.2 contienen lo siguiente:

- Cryptographic API as defined in MHP 1.1.2: 11.8.5
- APIs for return channel security as defined in MHP 1.1.2: 11.8.2
- SATSA requirements as specified in MHP 1.1.2:11.9.4 Non-CA smart card API.
- The XML parsing API as specified in MHP 1.1.2: 11.9.5 XML parsing API
- HAVI, DVB, DAVIC as specified in MHP 1.1.2. Includes also DVB-HTML e Internet Clients
- Java TV & Java Media as specified in MHP 1.1.2
- DOM as specified in MHP 1.1.2: 8.10 Document Object Model (DOM)

Bien, pero:

¿ Qué JVM ?

¿ Cómo compilo ?

¿ Con qué librerías ?

Veamos la propuesta que os hacemos

JVM. PBP 1.1

- En MHP 1.1.2 Specs: 1.1.2 Blue Book A0068r1.pdf, dice:
 - 11 DVB-J Platform
 - 11.1 The Java Platform
 - The Java platform is defined in **PBP 1.1**
- ...y PBP1.1 es:
 - PBP 1.1 "Personal Basis Profile 1.1 for the J2ME Platform (JSR 217)"
 - Refs:
 - <http://www.jcp.org/>
 - <http://java.sun.com/products/personalbasis/>,
 - Implementación de referencia: `pbp-1_1-fr-ri.zip`, `/refs/pbp11/pbp-1_1-fr-ri.zip`

JVM. PBP 1.1

- Basado en el Connected Device Configuration (CDC) 1.1 (JSR=218). Veamos qué dice esta especificación

/refs/cdc/CDC 1.1 runtime-RI.pdf

- **A Java runtime environment** is an implementation of Java technology for a specific target platform...
- The **Connected Device Configuration** (CDC) Java runtime environment is an implementation of Java technology **for connected devices**. These include **mobile** devices like **PDA**s and **smartphones** as well as **attached** devices like **set-top boxes**, **printers** and kiosks.
- CDC target devices can vary widely based on their features and purpose. FIGURE 1-1 describes some CDC target device categories and organizes them by their **two most important characteristics: purpose** (fixed or general) and **mobility** (mobile or attached).

JVM. PBP 1.1

- Basado en el Connected Device Configuration (CDC) 1.1 (JSR=218). Veamos qué dice esta especificación. (y 2)

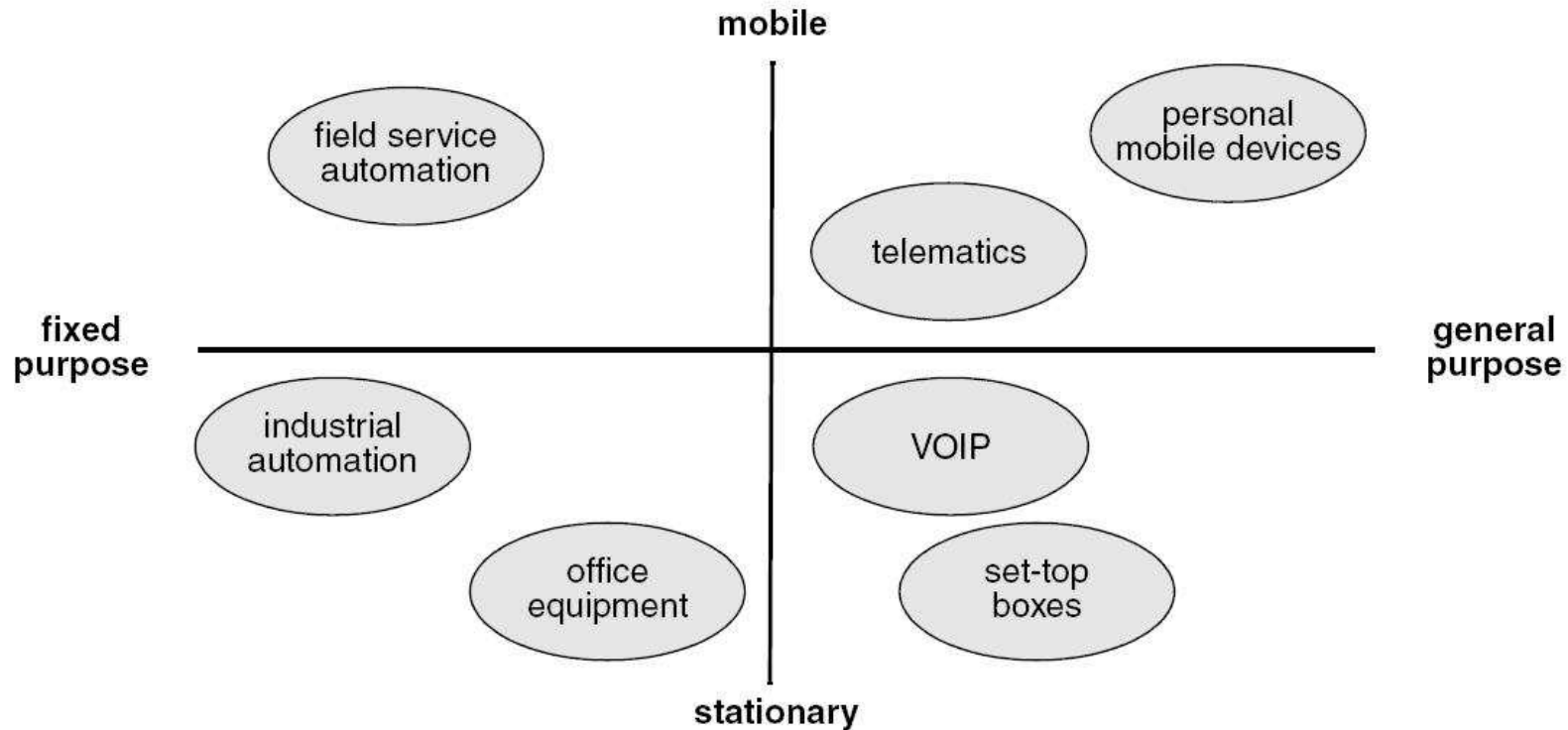


FIGURE 1-1 CDC Target Device Categories

JVM. PBP 1.1

- Basado en el Connected Device Configuration (CDC) 1.1 (JSR=218). Veamos qué dice esta especificación. (y 3)

Categoría	Descripción	Opciones
<i>Configuration</i>	Defines the most basic Java class library and Java virtual machine capabilities for a broad range of devices.	* Connected Device Configuration (CDC, JSR-218) supports connected devices like smart phones, set-top boxes and office equipment.
<i>Profile</i>	Defines additional APIs that support a narrower range of devices. A profile is built on a specific configuration	<p>* Foundation Profile (JSR-219) provides application-support classes like network and I/O support platforms without a standards-based GUI system.</p> <p>* Personal Basis Profile (JSR-217) provides a standards-based GUI framework for supporting lightweight components. In addition to the same application support classes provided by Foundation Profile, Personal Basis Profile includes support for the xlet application model.</p>

JVM. PBP 1.1

- Es decir, que todo deriva de la configuración básica de CDC. La cual está basada en **Java SE 1.4.2**
- El Runtime de PBP 1.1 (JSR 217) está basado en el Foundation Profile (JSR 219), al cual añade una serie de opciones. En JavaDoc de PBP 1.1 vemos:
 - Personal Basis Profile 1.1 is based upon Foundation Profile 1.1 (JSR-219). Specifically, Personal Basis Profile 1.1 *includes* Foundation Profile 1.1.

JVM. PBP 1.1

Librerías del Runtime

- Las clases que usaremos como librerías del runtime serán las que encontramos en su implementación de referencia: **refs/pbp11/pbp-1_1-fr-ri.zip** y recogidas en el siguiente punto como un solo jar:

`/libs/pbp11/pbp11.jar`

Fuentes

- Puesto que está basado en el jdk1.4.2 vamos a hacer referencia a los fuentes de esta JVM

`jdk14/src.zip`

Refs:

<http://java.sun.com/j2se/1.4.2/download.html>

JVM. PBP 1.1

Fuentes no incluidas

- No los tenemos del paquete javax.microedition.*

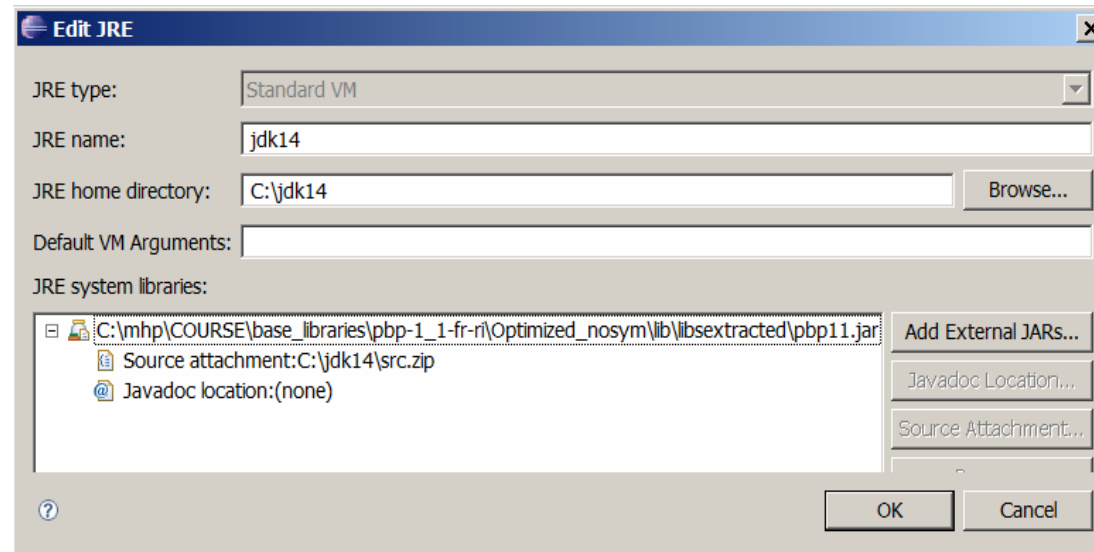
Herramienta de trabajo

- Eclipse europa configurado para trabajar con proyectos Java. Se necesitará el J2SE 1.6.x

www.eclipse.org

Entorno de Desarrollo y Compilación

- Necesitamos compilar con las librerías indicadas del PBP 1.1 + seguridad, que veremos más adelante su configuración.
 - **Paso 1:** Instalamos JDK 1.4.2 en c:/jdk14 (por ejemplo). En Eclipse lo instalamos como jdk14
 - **Paso 2:** Establecemos el bootclasspath del JDK1.4.2 a la siguiente librería:
/libs/pbp11/pbp11.jar.
 - **Paso 3:** Establecemos los fuentes de pbp11.jar como los incluidos en el jdk1.4.2: src.zip

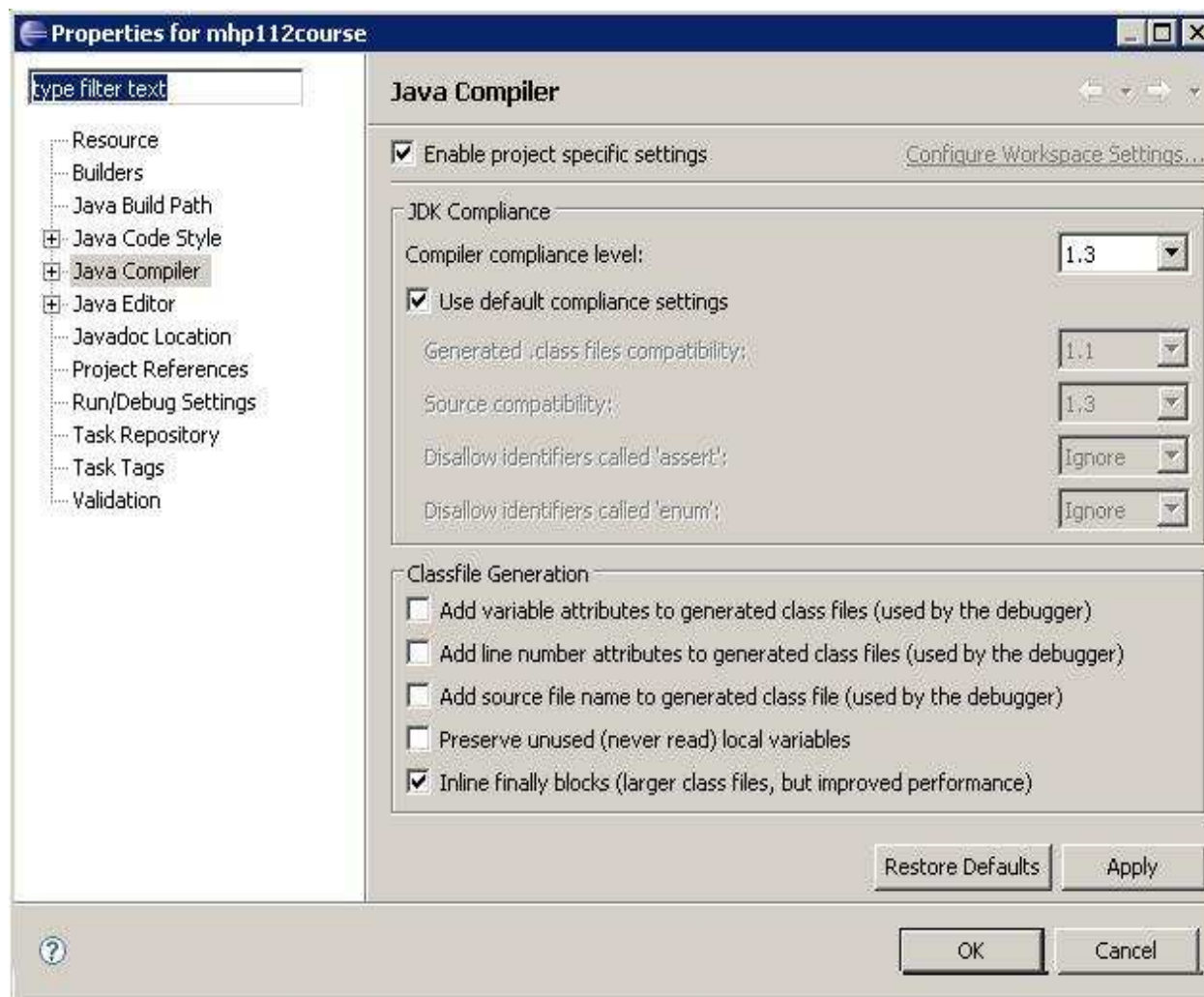


Entorno de Desarrollo y Compilación

- **Paso 4:** para trabajar generando los proyectos del entorno, por ejemplo dom112, necesitamos el jdk142 real, lo que hacemos es copiar el jdk142 con otro nombre de directorio, por ejemplo jdk14n, y lo instalamos en Eclipse como jdk14n. De esta forma podemos compilar nuestros proyectos de Stubs con jdk14n y nuestros Xlets con jdk14.

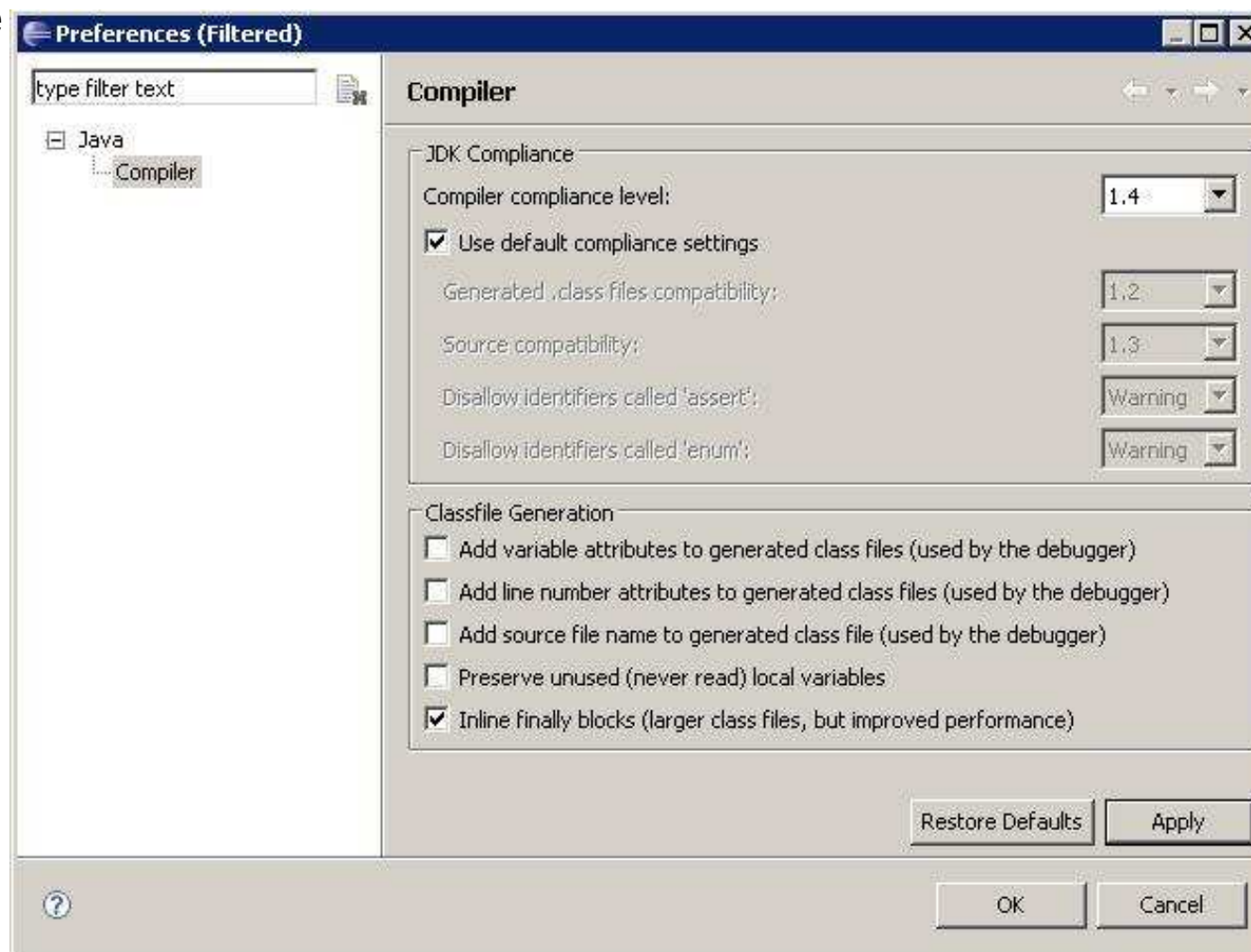
Entorno de Desarrollo y Compilación

- **Paso 5:** Opciones de compilación. OK para:
 - MHP 1.0.2
 - MHP 1.1.2



Entorno de Desarrollo y Compilación

- **Paso 5:** Opciones de compilación.
- También funcionan para MHP 1.0.2 y 1.1.2 aunque se recomienda establecer la menos restrictiva: compliance level 1.3





DVB, HAVI & DAVIC

Stubs para los paquetes de DVB, HAVI y DAVIC.

Stubs DAVIC, DVB y HAVI

Davic:

MHP 1.1.2 specs: "DAVIC 1.4.1 Specification Part 9: Information Representation", refs/davic

Havi:

MHP 1.1.2 specs:

HAVi v1.1 Chapter 8, 15-May-2001,
HAVi v1.1 Java L2 APIs, 15-May-2001
HAVi v1.1 Chapter 7, 15-May-2001

DVB:

MHP 1.1.2 Specs

- Recientemente (abril-mayo 2008) **mhp.org** ha publicado los mhp 1.1.2 stubs.
 - <http://www.mhp.org/fullspeclist.htm>
- Estos stubs se suponen basados en la también recién publicada documentación de MHP 1.1.2 (abril-mayo 2008) :
 - http://www.mhp.org/specs/tam668r23_11xdraft_20061115.zip
- Los Stubs publicados incluyen las correcciones descritas en:
 - http://www.mhp.org/specs/tm3570.tam0941r2.mhp1.1.2_errata1.clean.pdf

Stubs DAVIC, DVB y HAVI

- En los nuevos Stubs publicados:
 - Los siguientes paquetes aunque **sí están en la Documentación NO están en los Stubs**, no obstante code4tv los ha incluido desarrollándolos a partir de la documentación.
 - org.dvb.dom.css (4 files)
 - org.dvb.dom.dvbhtml (17 files)
 - org.dvb.dom.environment (3 files)
 - org.dvb.dom.event (1 files)
 - **Hay Nuevos componentes:**
 - package org.dvb.smartcard (4 files)
 - package org.dvb.spi (6 files)
 - org.dvb.application.storage.ApplicationStorageController (1 file)
 - org.dvb.application.storage.ApplicationStorageListener (1 file)
 - org.dvb.security.provider.CryptographicServiceProviderProvider (1 file)
 - org.dvb.ui.BufferedAnimation (1 file)
 - **Hay componentes Actualizados: comprobad la documentación.** Lo han sido por ejemplo:
 - org.dvb.security.provider.ProviderManager
 - org.dvb.security.pkcs11.DVBPkcs11Provider

Stubs HAVI, Incoherencia resuelta

- En PBP1.1 la clase `java.awt.event.KeyEvent` NO tiene el siguiente constructor, que está deprecated, por lo que las clases siguientes de HAVI no compilan al tener un constructor basado en aquel : `org.havi.ui.event.HRcEvent`, `org.havi.ui.event.HKeyEvent`

```
public HRcEvent(Component source, int id, long when, int modifiers, int keyCode) { super(source, id, when, modifiers, keyCode); }  
public HKeyEvent(Component source, int id, long when, int modifiers, int keyCode) { super(source, id, when, modifiers, keyCode); }
```

/refs/pbp11/pbp-1_1-fr-spec-jdoc/pbp-spec/java/awt/event/KeyEvent.html

- **Solución:** comentar dicho constructor

Stubs DAVIC, DVB y HAVI

Librerías y fuentes

- /libs/havi_dvb_davic/havi_dvb_davic112-1.1.jar
- /libs/havi_dvb_davic/havi_dvb_davic112-1.1.jar_src.jar



JavaTV 1.0 & JMF 1.0

Creando un paquete JavaTV / JMF coherente con PBP1.1

Stubs JMF: Java Media Framework

- MHP 1.1.2 specs
 - Java Media Player Specification Part of ISBN:1-892488-25-6: "Java" Media Framework API Version 1.0 specification"
- refs:
- http://java.sun.com/products/java-media/jmf/1.0/apidocs/jmf1_0-spec.pdf
 - <http://java.sun.com/products/java-media/jmf/1.0/>
- **No existe una implementación de referencia por sí sola...**

PERO...

Stubs JMF: Java Media Framework

- **SI existe Implementacion válida dentro de Java TV 1.0.** Ver JavaTVOverview.pdf incluida en JavaTV:
 - *“The Java TV API uses the Java Media Framework (JMF) 1.0 APIs for managing the broadcast media pipeline. The JMF APIs provide a foundation for the Java TV API by defining a set of APIs and a framework for displaying time-based media that are independent of transport mechanism, transport protocol, and media content type.”*
- **Sin embargo la implementación de Referencia está basada en los jdk1.1.8, 1.2.2 y depende de PersonalJava.** Los fuentes de la implementación dependen de las librerías siguientes
 - **PersonalJava 3.1:** classes.zip
ref: <http://java.sun.com/products/personaljava/pj-emulation.html>, refs/javaTV1.0/pjee3_1-win-nonrom.exe
 - **Parser de SUN:** parser.jar
ref: www.apache.org, incluido en tomcat3.0.zip
 - **DOM** de MHP 1.1.2: /libs/dom112/dom112.jar. **Es necesaria para el parser.jar**

Stubs Java TV 1.0

- En esta situación no podemos tener un proyecto de los APIS de JavaTV/JMF que compilen únicamente con el PBP 1.1 y los APIs autorizados de MHP.
- Por un lado hemos de desarrollar con la seguridad de trabajar contra PBP 1.1, y por otro además hay que revisar el código de JavaTV para incluir los cambios descritos en las especificaciones MHP 1.1.2 : Annex A.
- Lo que haremos será primero crear un proyecto partiendo de los fuentes de JavaTV/JMF que haremos que compile correctamente en MHP 1.1.2, y después aplicaremos los cambios descritos en la documentación.
- Durante el proceso LOS APIS PÚBLICOS DE JavaTV/JMF PERMANECERÁN INTACTOS, de manera que aseguramos que los Xlets compilarán siempre correctamente
- Vayamos viendo las dependencias encontradas y las soluciones aplicadas para eliminarlas.

Stubs Java TV 1.0

- Punto de partida:
 - JavaTV/JMF source code:
 - [jtv-1_0-src]/javatv_fcs/src/javax (**JavaTV, javax.tv**)
 - [jtv-1_0-src]/javatv_fcs/src/com (**SUN impl, com.sun.tv**)
 - [jtv-1_0-src]/javatv_fcs/jmflite/src/javax, (**JMF, javax.media**)
 - [jtv-1_0-src]/javatv_fcs/jmflite/src/com (**SUN impl, com.sun.media**)
 - Libraries de partida(objective: eliminarlas al máximo)
 - PersonalJava 3.1** classes.zip
 - Parser de SUN** parser.jar
 - dom112.jar
 - Compilamos contra PBP 1.1

Stubs Java TV 1.0

Dependencia de Jdk 1.2.2

- Thread.stop() inexistente en PBP 1.1
 - Afecta a:
 - **javax.tv.carousel.CarouselFile**-> Inner FileWatcher extends Thread
 - com.sun.tv.media.util.Thread extends com.sun.tv.media.util.MediaThread
 - com.sun.media.amovie.AMController extends com.sun.tv.media.util.MediaThread
 - com.sun.tv.media.MediaController llamando a:
 - Inner PrefetchWorkThread extends MediaThread
 - Inner RealizeWorkThread extends MediaThread
 - Inner StopTimeThread extends MediaThread
 - Inner TimeStartThread extends MediaThread
 - com.sun.media.amovie.ActiveMovie extends Thread
 - com.sun.tv.media.util.ThreadedEventQueue extends com.sun.tv.media.util.MediaThread
 - **Solución:** Comentamos todas las referencias a stop();

Stubs Java TV 1.0

Dependencia de Jdk 1.2.2

- java.awt.Panel inexistente en PBP 1.1
 - Afecta a:
 - com.sun.media.amovie.VisualComponent extends Panel
Además provoca error en las llamadas:
 - addNotify()
 - removeNotify()
 - setOwner(Component)
 - Y en la clase **com.sun.tv.DisplayManager** en la sentencia:
(c instanceof com.sun.media.amovie.VisualComponent) (line 62)
 - Y en **com.sun.tv.XletContainer** en la sentencia:
(c instanceof com.sun.media.amovie.VisualComponent) (line 50)
 - **Solución:** Cambiamos java.awt.Panel por java.awt.Container

Stubs Java TV 1.0

Dependencia de Jdk 1.2.2

- Java.awt.Component.show() inexistente en PBP 1.1
 - Afecta a: **com.sun.tv.DisplayManager**
 - **Solución:** Cambiamos show() por setVisible(true);

Stubs Java TV 1.0

Dependencia de Jdk 1.2.2

- Los métodos **java.util.Date.setHours()/setMinutes(...)/setSeconds(...)** inexistentes en PBP 1.1 (*es curioso que jdk 1.4.2 sí los mantiene*)

- Afecta a: **com.sun.tv.receiver.SIEmulator (line 154)**

```
Date startDate = new Date();
startDate.setHours((int)((startTime - (startTime % 100)) / 100));
startDate.setMinutes((int)(startTime % 100));
startDate.setSeconds(0);
```

- **Solución:** Sustituimos lo anterior por

```
int h = (int)((startTime - (startTime % 100)) / 100);
int m = (int)(startTime % 100);
Date startDate = new Date( new Date().getTime()+ h * 60 * 60 * 1000 + m * 60 * 1000 );
```

Stubs Java TV 1.0

Dependencia de PersonalJava 3.1

- Para eliminar toda dependencia del runtime de PJava nos quedamos con el mínimo set de clases (ver abajo) para poder compilar, y ya **no** tenemos en cuenta la librería completa de Pjava
- Ahora el objetivo es eliminar la dependencia de estas clases.

com.sun.util.DefaultPTimer

com.sun.util.DefaultTimerItem

com.sun.util.PTimer

com.sun.util.PTimerScheduleFailedException

com.sun.util.PTimerSpec

com.sun.util.PTimerWentOffEvent

com.sun.util.PTimerWentOffListener

Stubs Java TV 1.0

Dependencia de PersonalJava 3.1

- La clase **com.sun.tv.net.EncapDatagramSocketImpl.java** hereda de **java.net.DatagramSocketImpl** y esta última de Pjava NO tiene los tres métodos abstractos que SI tiene la nueva **java.net.DatagramSocketImpl** de PBP1.1:

joinGroup (SocketAddress, NetworkInterface)

peekGroup (SocketAddress, NetworkInterface)

peekData (DatagramPacket)

Sin embargo esta clase no es usada por nadie.

Solución: eliminamos de los fuentes **com.sun.tv.net.EncapDatagramSocketImpl.java**

Stubs Java TV 1.0

Dependencia de PersonalJava 3.1

- La clase abstracta `javax.tv.util.TVTimer.java` usa `com.sun.tv.timer.TVTimerImpl.java`, y `com.sun.tv.timer.TVTimerImpl` usa varias de Pjava, por ejemplo, `com.sun.util.PTimerSpec`.

`javax.tv.util.TVTimer:`

```
public static TVTimer getTimer() {  
    ...  
    theTimer = new com.sun.tv.timer.TVTimerImpl();  
    ...  
}
```

`com.sun.tv.timer.TVTimerImpl` sólo se usa por esta clase

Solución:

- Quitamos la referencia a la misma de `javax.tv.util.TVTimerSpec.java` comentando ese try.
- Eliminamos la clase `com.sun.tv.timer.TVTimerImpl.java`

Stubs Java TV 1.0

Última Dependencia de PersonalJava 3.1

- La clase `javax.tv.util.TVTimerSpec.java` usa `com.sun.util.PTimerSpec.java`.

com.sun.util.PTimerSpec sólo se usa por esta clase

PTimerSpec es una implementación de SUN de `javax.tv.util.TVTimeSpec`.

Solución:

- Eliminamos la referencia a **com.sun.util.PTimerSpec** de `javax.tv.util.TVTimerSpec.java` sustituyendo el tipo por el de la propia `TVTimerSpec`. Ver en tabla el único cambio realizado además de la eliminación del import.

Antes:	Después:
<pre>private PTimerSpec ptimer = null; private Vector listeners = null; public TVTimerSpec() { ptimer = new PTimerSpec(); listeners = new Vector(); }</pre>	<pre>private TVTimerSpec ptimer = null; private Vector listeners = null; public TVTimerSpec() { ptimer = new TVTimerSpec(); listeners = new Vector(); }</pre>

Stubs Java TV 1.0

Dependencia de SUN Java Parser

- La clase **com.sun.tv.receiver.ReceiverFile** necesita el parser de SUN incluido en tomcat3.0:

Sin embargo esta clase no es usada por nadie.

- **Solución:** la eliminamos de los fuentes

Además se elimina la dependencia de dom112.

Stubs Java TV 1.0

- Ahora aplicamos los cambios descritos en **Annex A MHP 1.1.2, A068r1**. Sólo se incluyen aquellas que modifican ficheros .java en Javadoc o código.

- Librerías y Fuentes:

/libs/javatv1.0-jmf/javatv10.jar

/libs/javatv1.0-jmf/javatv10_src.jar



XML Processing

Stubs Java XML processing

- MHP 1.1.2 specs
 - 11.9.5 XML parsing API: The XML parsing API is defined in clause 2, "JAXP Subset", of JSR-172: que es la especificación de **"1.0 J2ME Web Services"**

ref: <http://www.jcp.org/aboutJava/communityprocess/final/jsr172/index.html>, j2me_web_services-1_0-fr-spec.pdf

- j2me_web_services-1_0-fr-spec, clause 2, dice :
 - The goal of this optional package is to define a strict subset wherever possible of the XML parsing functionality defined in JSR-063 JAXP 1.2 [2] **that can be used on the Java 2 Micro Edition Platform (J2ME) [1].**

Stubs Java XML processing

- Disponemos de la implementación de J2ME Web Services 1.0
- Se han comprobado las clases y coinciden plenamente con las especificadas en J2ME Web Services 1.0: javax.xml.parsers, org.xml.sax, org.xml.sax.helpers

– refs:

<http://www.sun.com/software/communitysource/j2me/>

<http://www.sun.com/software/communitysource/j2me/wsa/download.xml>

j2me_web_services-1_0-src.zip

(interesting) <http://www.hcilab.org/documents/tutorials/JSR172/jsr172.html>

- Librerías y Fuentes en:

libs/xml/j2me_xml_ri.jar

libs/xml/j2me_xml_ri_src.jar



SATSA

Stubs SATSA

- En las últimas specs MHP 1.1.2:

11.9.4 Non-CA smart card API

The API for access to non-conditional smart cards shall be comprised of the following:

- the "SATSA-APDU" optional package defined by SATSA [106] (APDUConnection)
- the class `javax.microedition.apdu.APDUPermission` defined in clause B.1.2.1 of that document.
- the **org.dvb.smartcard** package defined in Annex AM "(normative): Smart card reader API" on page 1278.

The present document does not require support for U(SAT).

refs:

<http://java.sun.com/products/satsa/>

<http://jcp.org/aboutJava/communityprocess/final/jsr177/index.html>, [j2me_satsa-1_0-fr-spec.pdf](#)

Stubs SATSA

- Reference Implementation: Existe una implementación de referencia que trabaja contra el jdk 1.4.2
 - SATSA 1.0: This binary version of the Reference Implementation is based on the SATSA 1.0 specification. This is a complete working binary, bundled with the CLDC 1.0.4 RI and the MIDP 2.0 RI.
 - Refs:
 - <http://java.sun.com/products/satsa/index.jsp>
 - satsa-1_0-windows.zip
- **No obstante**, de todo lo anterior, sólo necesitamos añadir 2 clases que no están en los Stubs porque **org.dvb.smartcard** ya sí está en **MHP 1.1.2** (antes sólo en 1.1.3). Por otro lado **SATSA** se basa en el **Generic Connection Framework** (javax.microedition.io) que está en el **PBP 1.1**

`javax.microedition.apdu.APDUConnection`

`javax.microedition.apdu.APDUPermission`

Stubs SATSA

- Realizamos un proyecto ad-hoc incluyendo las dos clases que desarrollamos a partir de la documentación. De esta forma disponemos de los fuentes y TODA la documentación de los mismos.

- Librerías y Fuentes

libs/satsa10/satsa10.jar

libs/satsa10/satsa10_src.jar



DOM

Stubs DOM

- Necesario para DVB-HTML
- MHP 1.1.2 specs: “8.9 Document Object Model (DOM)”

Table 33: Supported DOM modules

DOM Module		Where specified	Required
Package	Feature String		
Level 2 Core	Core	Document Object Model (DOM) Level 2 Core Specification [96]	Yes
	XML		No
Level 2 Views	Views	Document Object Model (DOM) Level 2 Views Specification [100]	Yes
Level 2 Stylesheets	StyleSheets	Document Object Model (DOM) Level 2 Style Specification [99].	No
Level 2 CSS stylesheets	CSS	Document Object Model (DOM) Level 2 Style Specification [99].	No
	CSS2		Yes
Level 2 Events	Events	Document Object Model (DOM) Level 2 Events Specification [98].	Yes
	UIEvents		Yes
	MutationEvents		Yes
DVB-HTML	DVBHTML	See clause 8.9.4	Yes
DVB Events	DVBEvents	See clause 8.9.2	Yes
DVB Key Events	DVBKeyEvents	See clause 8.9.3	Yes
DVB CSS	DVBCSS	See clause 8.9.8.1	Yes
DVB Environment	DVBEnvironment	See clause 8.9.7	Yes

Stubs DOM

- Referencias Java Bindings

<http://www.w3.org/TR/DOM-Level-2-Core/>

<http://www.w3.org/TR/DOM-Level-2-Core/java-binding.zip>

<http://www.w3.org/TR/DOM-Level-2-Views/>

<http://www.w3.org/TR/DOM-Level-2-Views/java-binding.zip>

<http://www.w3.org/TR/DOM-Level-2-Events/>

<http://www.w3.org/TR/DOM-Level-2-Events/java-binding.zip>

<http://www.w3.org/TR/DOM-Level-2-Style/>

<http://www.w3.org/TR/DOM-Level-2-Style/java-binding.zip>

- Librerías y Fuentes

`/libs/dom112/dom112.jar`

`/libs/dom112/dom112_src.jar`



Seguridad

Seguridad: Return Channel Security & Cryptographic API

- En la documentación A0068r1, Capítulo “**11.8.2 APIs for return channel security**” se especifica:
 - This API is defined in the J2ME Security (JSSE) Optional Package Specification v1.0 which is found in FP 1.1 [121]. It includes the package `javax.microedition.io`.
ref: <http://java.sun.com/products/foundation/>, fp-1_1_2-mrel-ri-secop.zip, **jsse-cdc.jar**
- En la documentación A0068r1, Capítulo “**11.8.5 Cryptographic API**” se especifica que:
 - This API is defined in the J2ME Security (JCE) Optional Package Specification v1.0 part of FP 1.1 [121].
ref: <http://java.sun.com/products/foundation/>, fp-1_1_2-mrel-ri-secop.zip, **jce.jar**

Nota: Las librerías de seguridad de pbp-1.1-secop y FP1.1.2 son las mismas.

Seguridad: Return Channel Security & Cryptographic API

- Librerías

libs/security/jce.jar

libs/security/jsse-cdc.jar

- Código fuente no disponible pero las specs son:

ref: <http://java.sun.com/products/foundation/>, [cdc_secop_jaas-1_0_1-mrel-spec.zip](#), [cdc_secop_jce-1_0_1-mrel-spec.zip](#)

Seguridad: DVB Extensions for Cryptography

- En la documentación A0068r1, Capítulo “**DVB Extensions for Cryptography**” se especifica que:
 - The most recent version of "Java2 Standard Edition" includes new APIs providing the ability to use PKCS11 tokens (such as smart cards) in Java applications. These new APIs simplify the way in which Java applications can handle removable smart cards. They are also needed to login into PKCS11 token for non-key related operations such as encryption and decryption.
 - The present document is based on J2ME Personal Basis Profile (PBP 1.1 [116]) which does not provide these new APIs. For this reason, the following new packages are defined in the present document, see Annex A1.
 - org.dvb.security
 - org.dvb.auth.callback
 - org.dvb.net.ssl
 - org.dvb.security.pkcs11
- Todos estos APIs están incluidos en los Stubs de org.dvb.
- Estos APIs tienen dependencia de JCE y JAAS