



Curso Multimedia Home Platform 1.1.2

Internet Clients. Internet Access Profile

Curso Multimedia Home Platform 1.1.2

Copyright 2008 © Enrique Pérez Gil

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

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

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

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

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

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

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

Introducción

- Veamos las opciones de MHP en cuanto al Internet Profile (MHP 1.1.2 A068r1)

Area	Specification	Enhanced Broadcast Profile 2	Interactive Broadcast Profile 2	Internet Access Profile 2
Broadcast channel protocols				
	6.2.2, "MPEG-2 Sections"	M	M	M
	6.2.5, "DSM-CC User-to-User Object Carousel"	M	M	M
	IP Multicast stack based on: 6.2.6, "DVB Multiprotocol Encapsulation", 6.2.7, "Internet Protocol (IP)" 6.2.8, "User Datagram Protocol (UDP)" 6.2.10, "IP signalling"	O	Ro	M
Interaction channel protocols				
TCP/IP	6.3.3, "Transmission Control Protocol (TCP)" 6.3.2, "Internet Protocol (IP)"	-	M	M
UDP/IP	6.3.2, "Internet Protocol (IP)" 6.3.9, "User Datagram Protocol (UDP)"	-	M	M
DSM-CC U-U RPC	6.3.4, "UNO-RPC" 6.3.5, "UNO-CDR" 6.3.6, "DCM-CC User to User"	-	O	O
HTTP	6.3.7.1, "HTTP 1.1"	-	O	O
HTTP	6.3.7.2, "MHP profile of HTTP 1.0"	-	M	M
DNS	6.3.10, "DNS"	-	M	M
HTTPS	6.3.7.3, "HTTPS"	-	M	M
Interaction Channel File System	6.4.1, "File system implemented only via the interaction channel"	-	M	M
DSMCC / HTTP hybrid	6.4.2, "Hybrid between broadcast stream and interaction channel"	-	M	M

Introducción

- Veamos las opciones de MHP en cuanto al Internet Profile (MHP 1.1.2 A068r1)

Area	Specification	Enhanced Broadcast Profile 2	Interactive Broadcast Profile 2	Internet Access Profile 2
DVB-J				
Others	11.9.1, "Timer Support"	M	M	M
	11.9.2, "User Settings and Preferences API"	M	M	M
	11.9.3, "Profile and version properties"	M	M	M
	11.9.4, "Non-CA smart card API"	M	M	M
	11.9.5, "XML parsing API"	M	M	M
Stand-alone applications	11.12.2, "Stored services"	M	M	M
Support for DVB-HTML	11.13, "Support for DVB-HTML"	O (note 1)	O (note 1)	O (note 1)
Internet access	11.14, "Internet access"	-	-	M
	17, "Internet access clients"	-	-	M
NOTE 1: DVB-HTML is a single option which shall only be considered included if all parts of it are included.				
NOTE 2: Recommended to be implemented as part of WWW browser but outside the scope of the MHP conformance regime				

Introducción

- Observamos que DVB-HTML es opcional y que HTTP 1.1 también
- Con respecto a los **Internet Clients**, en MHP 1.1.2 A0068r1 /17.2 se describen los requerimientos mínimos:
 - Email client: **SMTP** for sending email (see RFC 821 [107]) or **HTTP** to a WebMail server.
 - IETF RFC 281: RFC 821 1982 "SIMPLE MAIL TRANSFER PROTOCOL" (SMTP)
 - Si es mediante HTTP: Cláusula 3.2.2 of RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1"
 - **Protocols for receiving email are implementation dependent.**
 - Usenet news NNTP (RF 977 [108]) or HTTP to a WebNews server.
 - IETF RF 977 1982 "Network News Transfer Protocol"
 - IETF RFC 1738 1994 "Uniform Resource Locators (URL)".
- ¿ Donde se encuentran los APIS ?

org.dvb.internet

Veamos el detalle

org.dvb.internet

- Los Internet Clients van a funcionar como si se tratasen Services, lo cual implica en primer lugar que cuando se seleccione uno de ellos las aplicaciones existentes morirán (quizá no). Dicho de otra forma: **son Stored Services específicos cuyo Service Handler será especial: en realidad serán los mismos Handlers y los propios Services los que ofrecerán la funcionalidad:** No existirá una app dentro del Stored Service sino que el mismo Stored Service ofrece la funcionalidad
- La forma en que se seleccionan es mediante **el Java TV Selection API.**
- Los elementos principales son los dos siguientes:
 - public interface **InternetClientService** extends **javax.tv.selection.Service**.
 - Es el objeto Service **Base** devuelto cuando se selecciona uno de ellos
 - public interface **InternetClient** extends **javax.tv.service.selection.ServiceContentHandler**
 - Este es el componente Handler base que se encarga de manejar el contenido del Service seleccionado en el ServiceContext: es decir, aquel que me da el API del Internet Client elegido.
 - Recordemos el ServiceMediaHandler, encargado de la presentación completa de un canal.

org.dvb.internet

- De InternetClientService obtendremos en la práctica instancias de los siguientes tipos:
 - public interface **WWWBrowserService** extends InternetClientService
 - public interface **UsenetClientService** extends InternetClientService
 - public interface **EmailClientService** extends InternetClientService
- Y paralelamente de InternetClient obtendremos los correspondientes Handlers:
 - public interface **EmailClient** extends InternetClient
 - public interface **UsenetClient** extends InternetClient
 - public interface **WWWBrowser** extends InternetClient
- Las funcionalidades las ofrecen los Internet Client Handlers y los mismos Services

Pero aún no sabemos como seleccionar estos Services!

Security

- Para poder seleccionar estos services se debe tener `javax.tv.service.selection.SelectPermission`, es decir, Signed app & PRF.

Seleccionando un Internet Client Service

- Recordemos como se obtenían los Services en el api de Java TV service:

```
Javax.tv.service.SIManager sm = javax.tv.service.SIManager.createInstance();
```

```
ServiceList lista = sm.filterServices(ServiceFilter filtro);
```

....

- Pues bien, dentro del paquete **org.dvb.internet** disponemos del siguiente filtro con los tres tipos de Services que podemos seleccionar.
- Basta construir el filtro pasando el tipo de Internet Client Deseado.

```
public final class InternetServiceFilter extends ServiceFilter{  
    public static final int EMAIL_CLIENT = 1;  
    public static final int WWW_CLIENT = 2;  
    public static final int NEWS_CLIENT = 3;  
  
    public InternetServiceFilter(int service_type)  
}
```

Veamos los APIS en detalle

InternetClientService API

- public boolean **canRunApplication()**;
 - Indica si la aplicación (Internet Client) se puede ejecutar sin que la nuestra (la que lo llama) muera.
- public SIRequest **retrieveDetails**(SIRequestor requestor);
 - Siempre fallará, el Requestor recibirá un DATA_UNAVAILABLE
- public String **getName()**;
 - Nombre corto del Service. "Implementation Dependent"
- public boolean **hasMultipleInstances()**;
 - Siempre false. Nos decía si el Service está disponible en diferentes Transports.
- public ServiceType **getServiceType()**;
 - Devuelve una implementación InternetServiceType del tipo que corresponda.
- public InternetClientService[] **getSupportedClientServices()**;
 - Podría ocurrir que un mismo Service de un tipo, por ejemplo, WWWBrowser, también soporte el de EmailService. Devuelve los Services a los que "representa", incluyendo al que preguntamos

InternetClientService API

- Al margen disponemos de la gestión de Listening de eventos asociados a la situación del ServiceContext ya vista en Java TV Service Context, la cual conviene tener muy en cuenta para saber en qué estado se encuentran los InternetClient seleccionados.
- Veamos a continuación los APIs de los Services instanciados. Nótese que además de estos veremos los Handlers...o mejor dicho los Client.

EmailClientService API

- public String **getUserEmailAddress()**
 - Dirección de correo del usuario. **Mismo valor que si leemos la User Preference: User @**
- public void **addToAddressBook**(String address, String name)
 - Se añade el mail de un destinatario.
- public void **setInitialMessage**(String to, String subject, String messageBody)
 - Valores por defecto de destinatario, subject y body. **Válido sólo durante la vida de esta instancia**

UsenetClientService API

- public void **subscribe**(String newsgroup)
 - Añade una suscripción a un newsgroup a lista de suscripciones.
- public void **setInitialMessage**(java.net.URL message)
 - Msg inicial cuando el UsenetClient comienza. **Válido sólo durante la vida de esta instancia**
- public void **setInitialGroup**(java.net.URL group);
 - Establece el grupo inicial a presentar cuando se arranca el UsenetClient. **Válido sólo durante la vida de esta instancia**

WWWBrowserService API

- public String[] **getAcceptedMediaTypes()**
 - Array de MIME soportados indicando además versión de HTML: *text/html; version = "4.0"*.
 - Si el Browser soportase diferentes versiones de HTML devolvería varias instancias de: *text/html; version = "4.0"*.
- public String[] **getSupportedPlugins()**
 - Lista de plug-ins instalados.
- public String **getUserAgent()**
 - Devuelve el User-Agent HTTP Header.
- public boolean **areFramesSupported()**
 - Si el Browser soporta Frames y estas están habilitadas por el User.
- public void **setHomepage**(java.net.URL defaultUrl)
 - Home Page del Browser. Se usará al arrancar si no hay InitialURL
- public void **setInitialURL**(java.net.URL initialUrl)
 - URL a lanzar cuando el Browser arranque. **Válido sólo durante la vida de esta instancia**

WWWBrowserService API

- public void **addBookmark**(java.net.URL bookmarkUrl, java.lang.String name)
- public void **addBookmark**(javax.tv.locator.Locator locator, String name)
 - Gestión de Bookmarks.
 - Cuando se selecciona un Locator que apunta a un Service el resultado es el mismo que si se seleccionara el Service por el usuario o mediante el Java TV Selection API.

InternetClient. API Común a todos.

- public InternetClientService **getService()**;
 - Service al que pertenece el InternetClient Handler, por ejemplo: WWWBrowserService
- public void **addInternetClientListener**(InternetClientListener l);
- public void **removeInternetClientListener**(InternetClientListener l);
 - Gestión de suscripción a eventos del InternetClient

```
public interface InternetClientListener extends java.util.EventListener {  
    public void receiveInternetClientEvent(InternetClientEvent event);  
}
```

Veamos los tipos de eventos recibidos:

- **InternetClientFailureEvent** [extends InternetClientEvent]: se suelen ofrecer los 3 siguientes.
 - **PermissionDeniedEvent** [extends InternetClientFailureEvent]: no hay acceso a la URL en el Server.
 - **URLUnavailableEvent** [extends InternetClientFailureEvent]: URL no disponible.
 - **CancelledByUserEvent** [extends InternetClientFailureEvent] : la operación la canceló el usuario.
 - **InternetClientSuccessEvent** [extends InternetClientEvent]: la operación ha ido bien.
- public Locator[] **getServiceContentLocators()**;
 - Devuelve los Locator que apuntan a las partes que componen el Service (audio, video...), en nuestro caso el array sólo contiene el Locator devuelto por getService().getLocator()

EmailClient API

- public void **createMessage**(String to, String subject, String messageBody)
 - Método asíncrono.
 - Se verá como ha ido por el Listener API de InternetClientListener
 - Si falta alguno de los datos se le preguntará al usuario.

UsenetClient API

- public void **selectMessage**(URL message)
 - Método asíncrono.
 - Se verá como ha ido por el Listener API de InternetClientListener
 - Presenta el mensaje indicado por la URL

- public void **selectGroup**(URL group)
 - Método asíncrono.
 - Se verá como ha ido por el Listener API de InternetClientListener
 - Presenta los mensajes del NewsGroup indicado

WWWBrowser API

- public void **goToURL**(URL url)
 - Método asíncrono.
 - Se verá como ha ido por el Listener API de InternetClientListener
 - Muestra en el Browser la URL indicada.

Ejercicios Bloque INTERNET-1

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