



Java ME

MMAPI



MMAPI

♦ Paquets

```
javax.microedition.media  
javax.microedition.media.control
```

- ♦ Dissenyat per a suportar diversos continguts multimèdia i proporcionar mecanismes de captura d'àudio i vídeo
- ♦ La implementació proporcionada per Sun proporciona suport bàsic per a la reproducció simple d'àudio.
- ♦ Especificació de Java: **JSR 135**
- ♦ Similituds amb l'API Generic Connection Framework (GCF)

♦ No tots els dispositius suporten MMAPi:

<http://developers.sun.com/mobility/device>



MMAPI

♦ Característiques

- ♦ Suporta la **generació de tons, reproducció i enregistrament** de medis multimèdia (**àudio i vídeo**).
- ♦ **Bon rendiment** (*small footprint*): funciona amb les estrictes restriccions de memòria dels dispositius CLDC.
- ♦ **Independent** del **protocol** i del **contingut** (protocol and content agnòstic): La API no està esbiaixada cap a cap protocol o tipus de mitjà específic.
- ♦ **Flexible**: Els desenvolupadors poden limitar el suport a certs tipus de suport.
- ♦ **Extensible**: Es poden afegir noves funcionalitats de forma fàcil sense trencar la compatibilitat cap endarrere. És fàcil afegir suport per a nous formats



MMAPI

♦ Versions

- ♦ MMAPI té dos parts
 - Part integrada en el perfil MIDP 2.0
- ♦ MMAPI 1.2
 - Implementació opcional
 - <http://java.sun.com/javame/reference/apis/jsr135/>

```
javax.microedition.media  
javax.microedition.media.control  
javax.microedition.media.protocol
```



Exemple de suport multimèdia

♦ Suport multimedia

- ♦ <http://www.forum.nokia.com/devices/N95>
- ♦ **Vídeo formats:** 3GPP formats (H.263), H.264/AVC, MPEG-4, RealVideo 7,8,9/10
- ♦ **Formats àudio:** AAC, AAC+, AMR-NB, AMR-WB, eAAC, eAAC+, M4A, MIDI Tones (polifònics), Mobile XMF, MP3, MP4, RealAudio 7,8,10, SP-MIDI, True tones, WAV, WMA
- ♦ **Audio Features:** Audio Equalizer, Audio Recording AAC stereo, 48kHz, Audio Streaming, Music Player, Stereo
- ♦ **Video Features:** Video Call, video Editor, Video Player, Video Recorder, Video Sharing, Video Streaming
 - [Nokia](#)





Processament Multimèdia

♦ 2 parts

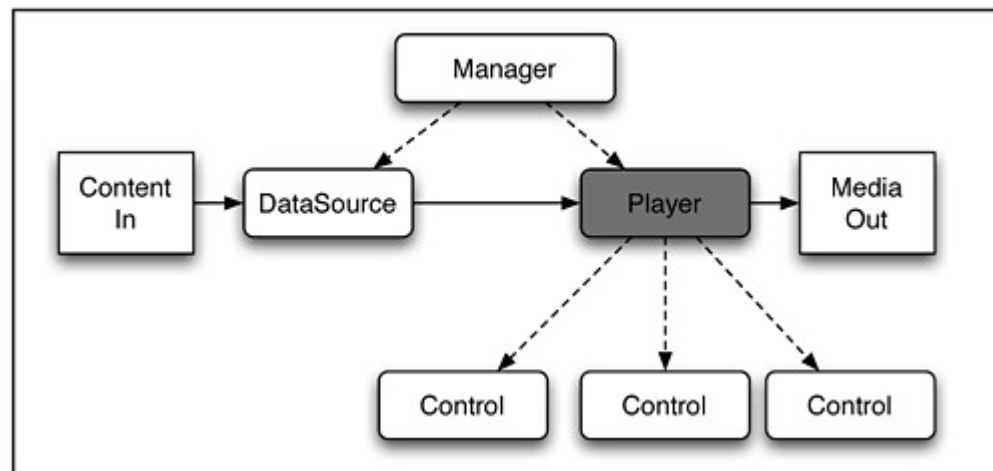
- ♦ **Tractament del protocol:** obtenir la informació d'una font local o remota (streaming server) per a ser processada per un sistema multimèdia
- ♦ **Tractament del contingut:** analitzar sintàcticament (parsing) o descodificar les dades multimèdia i presentar els resultats en un dispositiu de sortida com un altaveu de sortida o una pantalla.



Arquitectura

♦ Jerarquia de classes

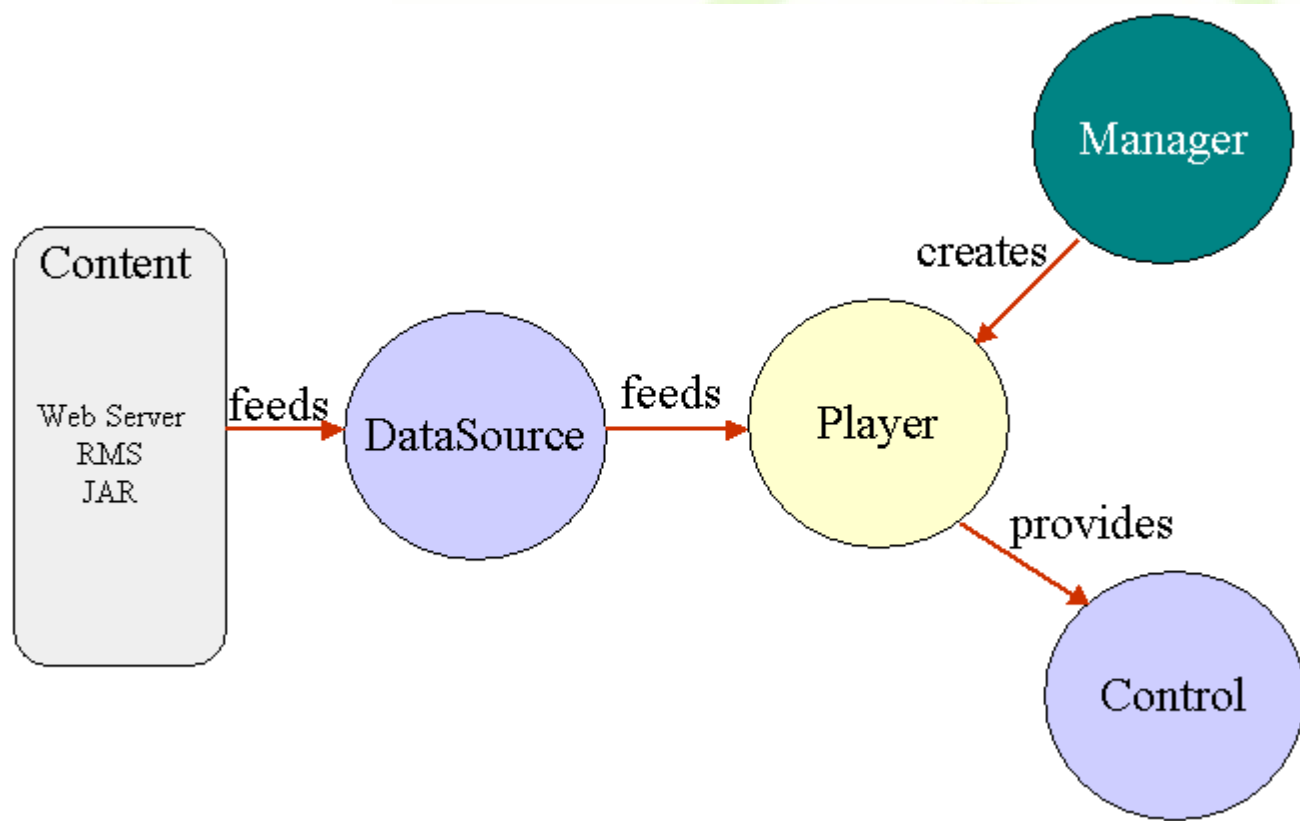
- ♦ **Manager:**
- ♦ És una classe **factoria** estàtica encarregada de crear objectes MMAPI.
- ♦ **Player:** conté les funcionalitats comunes a tots els tipus de reproductor multimèdia
- ♦ **DataSource:** Origen de les dades multimedia
- ♦ **Control:** conté les funcionalitats bàsiques dels diferents tipus de controladors





Funcionament

```
Player player = Manager.createPlayer(String url);
```





Manager

```
static Player createPlayer(String locator)
```

♦ Hi ha 3 tipus de “locator”:

- ♦ **Reproducció (media playback):** les reproduccions poden ser d'àudio o vídeo.
- ♦ **Captura (media capture):** Les captures poden ser de àudio o de vídeo.
- ♦ **Reproductor MIDI o de tons:** Per executar MIDIs o tons.

♦ Els locator són URI

```
Manager. createPlayer("http://www.yourwebsite.com/audio/song.wav")
```



Manager

Una sèrie d'exemples:

Tipus de medi	Exemple
Captura d'àudio	"capture://audio" captura àudio del dispositiu de captura d'àudio per defecte.
Captura d'àudio	"capture://devmic0?encoding=pcm" captura d'àudio del dispositiu devmic0 i codificació PCM
Captura de vídeo	"capture://video" captura vídeo del dispositiu de captura de vídeo per defecte.
Captura de vídeo	"capture://devcam0?encoding=rgb888&width=100&height=50" captura vídeo del dispositiu de captura de vídeo devcam0 amb unes mides específiques.
Escollar la radio	"capture://radio?f=105.1&st=stereo" en estèreo i al 105FM
rtp media	"rtp://host:port/type"
Executar un to	"device://tone" proporciona un reproductor per a executar tons
Executar un midi	"device://midi" proporciona un reproductor per a executar un MIDI

- ♦ El format multimèdia l'especifica el MIME-Type.

`http://host/sample.mid`

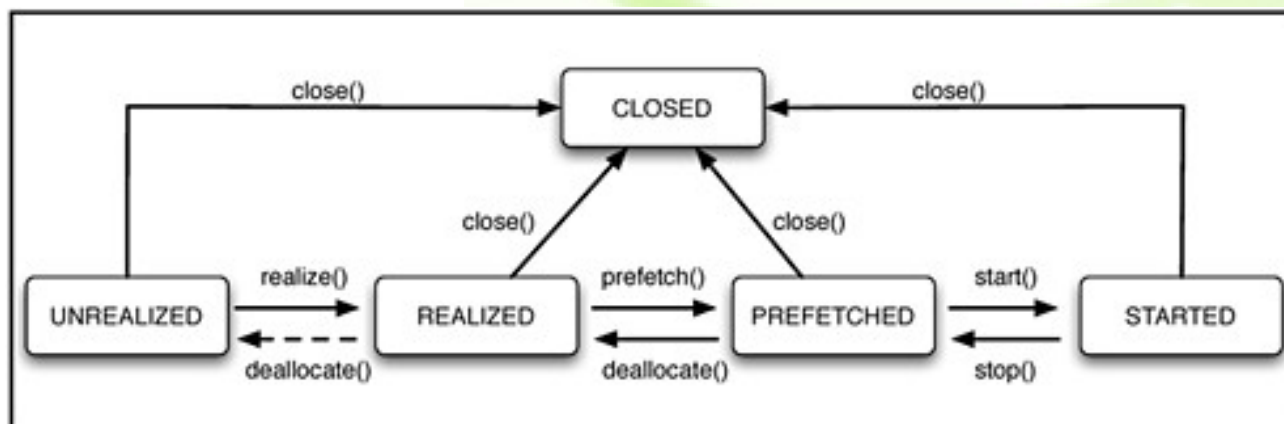
- ♦ Un servidor web associarà el MIME-Type **audio/midi** i per tant el mòbil crearà un reproductor de fitxers MIDI.



Cicle de vida d'un reproductor

♦ Estats

- ♦ **UNREALIZED**: Estat del reproductor al ser creat
- ♦ **REALIZED**: generat pel mètode *realize()*. Inicialitza la informació necessària per adquirir els recursos multimèdia
- ♦ **PREFETCHED**: generat pel mètode *prefetch()*. Estableix les connexions de xarxa i realitza altres tasques d'inicialització.
- ♦ **STARTED**: generat pel mètode *start()*. Es reproduïxen les dades multimèdia. Al finalitzar torna a l'estat PREFETCHED.
- ♦ **CLOSED**: generat pel mètode *close()*





Interfície Player

♦ Exemple bàsic de reproductor

- ♦ El mètode *start* retorna tant bon punt s'ha iniciat la reproducció. La reproducció continua en segon terme i finalitza automàticament quan s'arriba al final de les dades.

```
try {  
    Player p = Manager.createPlayer(  
        "http://www.fileformat.info/format/mpeg/sample/567fd6a0e0da4a8e81bdeb870de3b19c/DELTA.MPG");  
    p.realize();  
    GUIControl gc;  
    if ((gc = (GUIControl) p.getControl("GUIControl")) != null) {  
        Form form = new Form("My GUI");  
        form.append((Item) gc.initDisplayMode(GUIControl.USE_GUI_PRIMITIVE, null));  
        Display.getDisplay(this).setCurrent(form);  
    }  
    p.start();  
} catch (MediaException pe) { } catch (IOException ioe) { }
```



Interfície Player

♦ Altres mètodes

- ♦ TIME_UNKNOWN --> Live media
- ♦ MediaTime: el temps actual del mitjà que s'està executant
- ♦ TimeBase serveix per sincronitzar reproductors

String getContentType ()

long getDuration ()

long getMediaTime ()

long setMediaTime (long now)

int getState ()

TimeBase getTimeBase ()

void setTimeBase (TimeBase master)

void setLoopCount (int count)



Controls

♦ Interfície Control

- ♦ Defineix que un objecte és de tipus Control. Un objecte de Control és utilitzat per a realitzar operacions específiques amb un mitja multimèdia.

♦ Interfície Controllable

- ♦ Player implementa la interfície Controllable

```
Control getControl(java.lang.String controlType)
```

```
Control[] getControls()
```

- Múltiples controls poden ser implementats en un mateix objecte.



Controls

- ♦ **API base de Java ME només ofereix 2 controls**
 - ♦ ToneControl
 - ♦ VolumeControl
- ♦ **MMAPI 1.2**
 - ♦ FramePositioningControl | **GUIControl**
 - ♦ MetaDataControl | MIDIControl
 - ♦ PitchControl | RateControl
 - ♦ RecordControl | StopTimeControl
 - ♦ TempoControl | ToneControl
 - ♦ VideoControl | VolumeControl



Control d'esdeveniments

◆ **PlayerListener**

- ◆ Permet controlar els esdeveniments d'un reproductor
- ◆ Només conté un mètode:

```
void playerUpdate(Player player, String event, Object eventData)
```

◆ **Esdeveniments**

```
PlayerListener.BUFFERING_STARTED | PlayerListener.BUFFERING_STOPPED  
PlayerListener.CLOSED | PlayerListener.DEVICE_AVAILABLE  
PlayerListener.DEVICE_UNAVAILABLE | PlayerListener.DURATION_UPDATED  
PlayerListener.END_OF_MEDIA | PlayerListener.ERROR  
PlayerListener.RECORD_ERROR | PlayerListener.RECORD_STARTED  
PlayerListener.RECORD_STOPPED | PlayerListener.SIZE_CHANGED  
PlayerListener.STARTED | PlayerListener.STOPPED  
PlayerListener.STOPPED_AT_TIME | PlayerListener.VOLUME_CHANGED
```




Control d'esdeveniments

- ♦ **2 mètodes de Player permeten afegir/treure control d'esdeveniments:**

```
void addPlayerListener (PlayerListener listener)  
void removePlayerListener (PlayerListener listener)
```

- ♦ Ha tenir en compte:
 - Hi han noms d'esdeveniments específics per als diferents estats del reproductor (CLOSED, STARTED i STOPPED)
 - L'esdeveniment `DEVICE_UNAVAILABLE` succeïx quan hi ha una trucada entrant. L'esdeveniment `DEVICE_AVAILABLE` succeïx quan la trucada finalitza.



Protocols i formats

- ♦ **MMAPI no obliga a utilitzar cap protocol o format de dades multimèdia específic**
 - ♦ Normalment les implementacions garanteixen el suport de com a mínim un format i un protocol.
 - ♦ Hi han mètodes per saber els protocols i els formats suportats
 - `Manager.getSupportedContentTypes()`
 - `Manager.getSupportedProtocols()`
 - ♦ Si s'intenta crear un reproductor per a un protocol no suportat es llança l'excepció *MediaException*
 - ♦ Totes les funcionalitats són també opcionals. Un dispositiu pot suportar reproducció però no gravació (o control de volum).



Generació de tons

♦ Mètode estàtic de la classe Manager

```
try {
    // Do central de 4000 milisegons i volumen 100%
    Manager.playTone((ToneControl.C4, 4000, 100);
}
catch(MediaException me) {
}
```

♦ Tons de 0 (C-1) a 127 (G9)

- C-D-E-F-G-A-B és una octava de musical
- Do-Re-Mi-Fa-Sol-La-Si



$$\text{tone} = 12 * \log_2 (\text{freq}/220) + 57$$

Note	Frequency (Hz)	MIDI Tone
A4	440.00	69
A#	466.16	70
B	493.88	71
C	523.25	72
C#	554.36	73
D	587.33	74
D#	622.25	75
E	659.25	76
F	698.45	77
F#	739.98	78
G	783.99	79
G#	830.60	80



Reproducció MP3

♦ Exemple de reproducció d'un MP3

```
try {  
    Player p = Manager.createPlayer ("http://server/somemusic.mp3");  
    p.setLoopCount(5);  
    p.start();  
}  
catch(IOException ioe) {  
}  
catch(MediaException e) {  
}
```

- ♦ SetLoopCount estableix quants cops s'executarà la canço. -1 estableix un loop infinit.



Reproducció MP3

♦ Control de volum

```
Player p;  
VolumeControl vc;  
try {  
    p = Manager.createPlayer("http://server/somemusic.mp3");  
    p.realize();  
    // Obtenir el control de volum i establir el volum al màxim?  
    vc = (VolumeControl) p.getControl("VolumeControl");  
    if(vc != null) {  
        vc.setVolume(100);  
    }  
    p.prefetch();  
    p.start();  
}  
catch(IOException ioe) {  
}  
catch(MediaException e) {  
}  
...
```



Exemple de font de dades

♦ Reproduir wav de la base de dades RMS

```
...
RecordStore store;
int id;
// Reproduir des de un RecordStore
try {
    InputStream is = new ByteArrayInputStream(store.getRecord(id));
    Player player = Manager.createPlayer(is, "audio/X-wav");
    p.start();
}
catch (IOException ioe) {
}
catch (MediaException me) {
}
...
```



Reproducció de vídeo

```
...
Player p;
VideoControl vc;
try {
    p = Manager.createPlayer("http://server/somemovie.mpg");
    p.realize();
    // Obtenir el control de volum
    vc = (VideoControl) p.getControl("VideoControl");
    ....
    p.start();
}
catch(IOException ioe) {
}
catch(MediaException me) {
}
...
```



Reproducció de vídeo

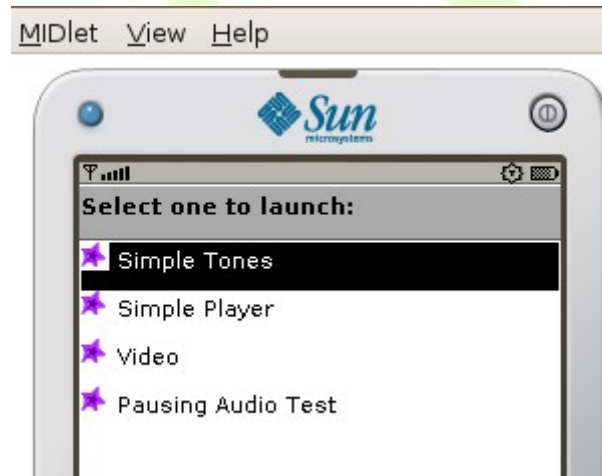
```
try {
    Player p = Manager.createPlayer(
        "
http://www.fileformat.info/format/mpeg/sample/567fd6a0e0da4a8e81bdeb870de3b19
    ");
    p.realize();
    GUIControl gc;
    if ((gc = (GUIControl) p.getControl("GUIControl")) != null) {
        Form form = new Form("My GUI");
        form.append((Item)
gc.initDisplayMode(GUIControl.USE_GUI_PRIMITIVE, null));
        Display.getDisplay(this).setCurrent(form);
    }
    p.start();
} catch (MediaException pe) {
} catch (IOException ioe) {
}
```




Netbeans

♦ Exemple de MMAPAPI

- ♦ Creeu un nou projecte des de el menú File o Ctrl+Shift+n
- ♦ A Samples > Java ME (MIDP) > **MMAPI Demos**
- ♦ Hi ha múltiples exemples d'àudio i vídeo. Executeu amb Run.





J2ME Wireless Toolkit

♦ L'emulador suporta:

♦ Àudio: PCM i WAV

- **MIDI:** Type 0 (single track), Type 1 (multiple tracks) i SP-MIDI

♦ Vídeo: MPEG-1

♦ Limitacions:

- Es poden crear múltiples reproductors però només un player de MIDI o tons pot entrar a l'estat `realize()` -> `MediaException`
- Es poden executar múltiples `playTone()` però només es garanteixen 4. Es pot executar `playTone()` i un MIDI a l'hora
- El número de reproductors de vídeo i wav està limitat pel heap (`-Xheapsize`)
- Només es pot fer una captura d'àudio simultàniament



MMAPI i fils d'execució

♦ Executar un reproductor en un nou fil d'execució

```
new Thread(new Runnable() {  
    public void run() {  
        //""Poseu aquí el codi del player  
    }  
}).start();
```

♦ Reproduir un to

```
new Thread(new Runnable() {  
    public void run() {  
        Manager.PlayTone(ToneControl.C4, 100,100);  
    }  
}).start();
```



Sons lliures

♦ Sons de Barcelona



- ♦ Escoltar sons gravats a Barcelona a partir d'un mapa de Google.

♦ The free sound project



- ♦ Gran base de dades lliures de sons sota una llicència lliure **Creative Commons Sampling Plus License**
- ♦ Tots dos són projectes de la Universitat Pompeu Fabra de Barcelona.
 - <http://mtg.upf.edu/>
- ♦ Utilitzarem sons d'aquesta base de dades.



Música i vídeo lliure

- ♦ **Hi han múltiples opcions per accedir a la cultura lliure...**
 - ♦ <http://www.musicallibre.net/>
 - ♦ <http://creativecommons.org/audio/>
 - ♦ <http://creativecommons.org/video/>
 - ♦ <http://www.magnatune.com/>
 - ♦ <http://www.jamendo.com>
 - ♦ <http://www.royaltyfreemusic.com/public-domain/creative->
- ♦ **Buscar continguts lliures**
 - ♦ <http://search.creativecommons.org/>



Reconeixement 3.0 Unported

Sou lliure de:



copiar, distribuir i comunicar públicament l'obra



fer-ne obres derivades

Amb les condicions següents:



Reconeixement. Heu de reconèixer els crèdits de l'obra de la manera especificada per l'autor o el llicenciador (però no d'una manera que suggereixi que us donen suport o rebeu suport per l'ús que feu l'obra).

- Quan reutilitzeu o distribuïu l'obra, heu de deixar ben clar els termes de la llicència de l'obra.
- Alguna d'aquestes condicions pot no aplicar-se si obteniu el permís del titular dels drets d'autor.
- No hi ha res en aquesta llicència que menyscabi o restringeixi els drets morals de l'autor.

Advertiment

Els drets derivats d'usos legítims o altres limitacions reconegudes per llei no queden afectats per l'anterior
Això és un resum fàcilment llegible del text legal (la llicència completa).

<http://creativecommons.org/licenses/by/3.0/deed.ca>