



# Java ME

## APIS Opcionals

### APIS Opcionals



# Wireless Messaging API (WMA)

## ♦ Wireless Messaging API (WMA)

- ♦ Permet enviar missatges de text (SMS) i missatges multimèdia (MMS) des de aplicacions Java
- ♦ Java Community: [JSR 120](#) (1.0), [JSR 205](#) (2.0)
- ♦ Paquet opcional basat en Generic Connection Framework (GCF)
- ♦ Connexions SMS:



Nº telèfon :Port

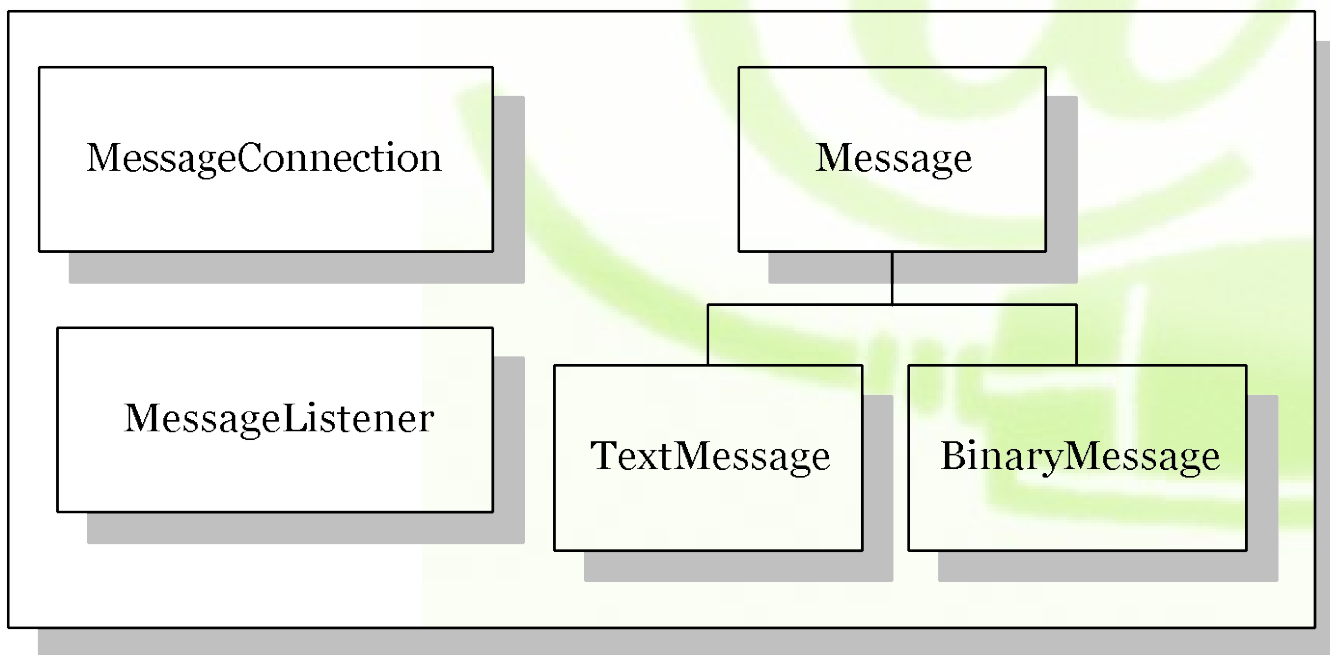
+346547387765:10000



# Wireless Messaging API (WMA)

## ♦ Arquitectura

- ♦ Paquets ([API javadocs](#))
  - `javax.microedition.io`
  - `javax.wireless.messaging`





# Wireless Messaging API (WMA)

## ♦ **Message**

- ♦ Interfície pare de les interfícies **TextMessage** i **BinaryMessage**
  - **BinaryMessage**: missatges binaris (MMS)
  - **TextMessage**: missatges de text (SMS)

## ♦ **MessageConnection**

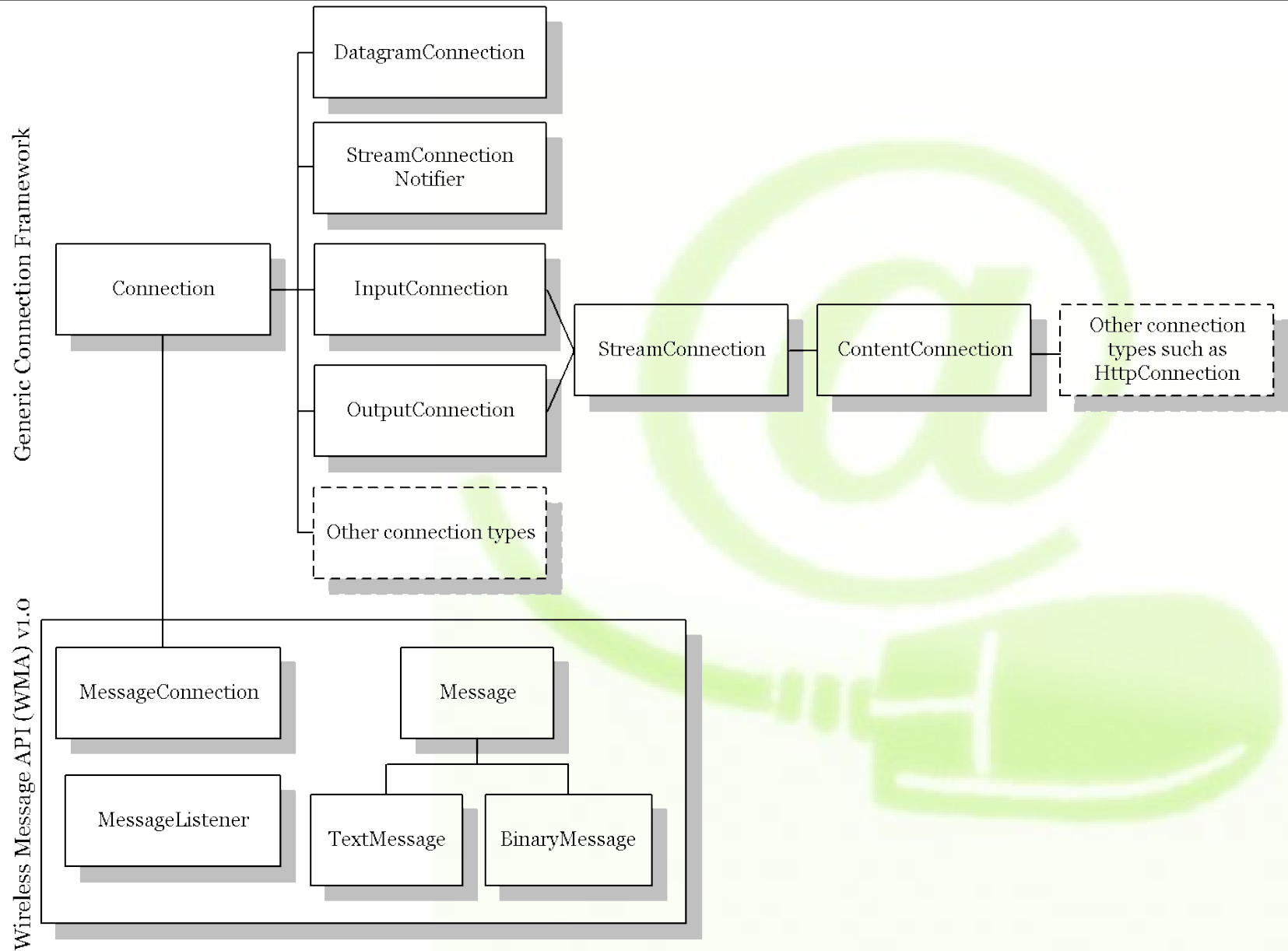
- ♦ Interfície filla de la interfície Connection de l'**API GCF**
- ♦ Proveeix una factoria per crear missatges

## ♦ **MessageListener**

- ♦ Control d'esdeveniments de missatges



## WMA i GCF





# Wireless Messaging API (WMA)

## Enviar un missatge de text

```
public void sendTextMessage(String msg, String url) {  
    Try {  
        MessageConnection mc = (MessageConnection) Connector.open(  
            " +34674345654:10000");  
        TextMessage tmsg = (TextMessage) mc.newMessage  
            (MessageConnection.TEXT_MESSAGE);  
        if (url != null)  
            tmsg.setAddress(url);  
        tmsg.setPayloadText(msg);  
        mc.send(tmsg);  
    }  
    catch(Exception e) {  
        // Handle the exception...  
        System.out.println("sendTextMessage " + e);  
    }  
}
```



# Wireless Messaging API (WMA)

## ♦ 2 tipus de connexions (URLs)

- ♦ **Client:** només pot enviar missatges

```
(MessageConnection)Connector.open("sms://+5121234567:5000");
```

- ♦ **Servidor:** pot enviar i rebre missatges

```
(MessageConnection)Connector.open("sms://:5000");
```

- ♦ **El servidor** s'ha d'implementar per fils d'execució. Un fil d'execució per rebre missatges (pare) i fils d'execució fills per processar els missatges rebuts.
- ♦ L'esquema URL no és específic per a un sol protocol (SMS). Pot suportar múltiples protocols (anomenats Wireless Message Protocols)



# Sun Wireless Toolkit

## ♦ Podem executar directament l'emulador

```
$ cd /home/sergi/downloads/WTk2.5.2/bin
$ ./ktoolbar
```

- ♦ Open Project -->
  - Escolliu el Sample **WMADemo**

## ♦ Run Over The Air (OTA)

- ♦ Simula una instal·lació real
  - Project --> Run Via OTA

## ♦ Consola WMA

- ♦ Menú File -> Utilities -> WMA Console
- ♦ Podem enviar i rebre missatges des de la consola

## ♦ Configuració de propietats

- ♦ \$WTK\_HOME/lib/internal.config





# Sun Wireless Toolkit

## ◆ Configuració SMS

```
$ cat internal.config | grep sms
com.sun.midp.io.j2me.sms.DatagramHost: localhost
com.sun.midp.io.j2me.sms.CBSPort: 24680
com.sun.midp.io.j2me.sms.PhoneNumber=+5551212
wireless.messaging.sms.smsc: +17815511212
```

- ◆ Emulació de missatges per enviament de missatges datagrama (sockets)

## ◆ Permisos

- ◆ SecurityException

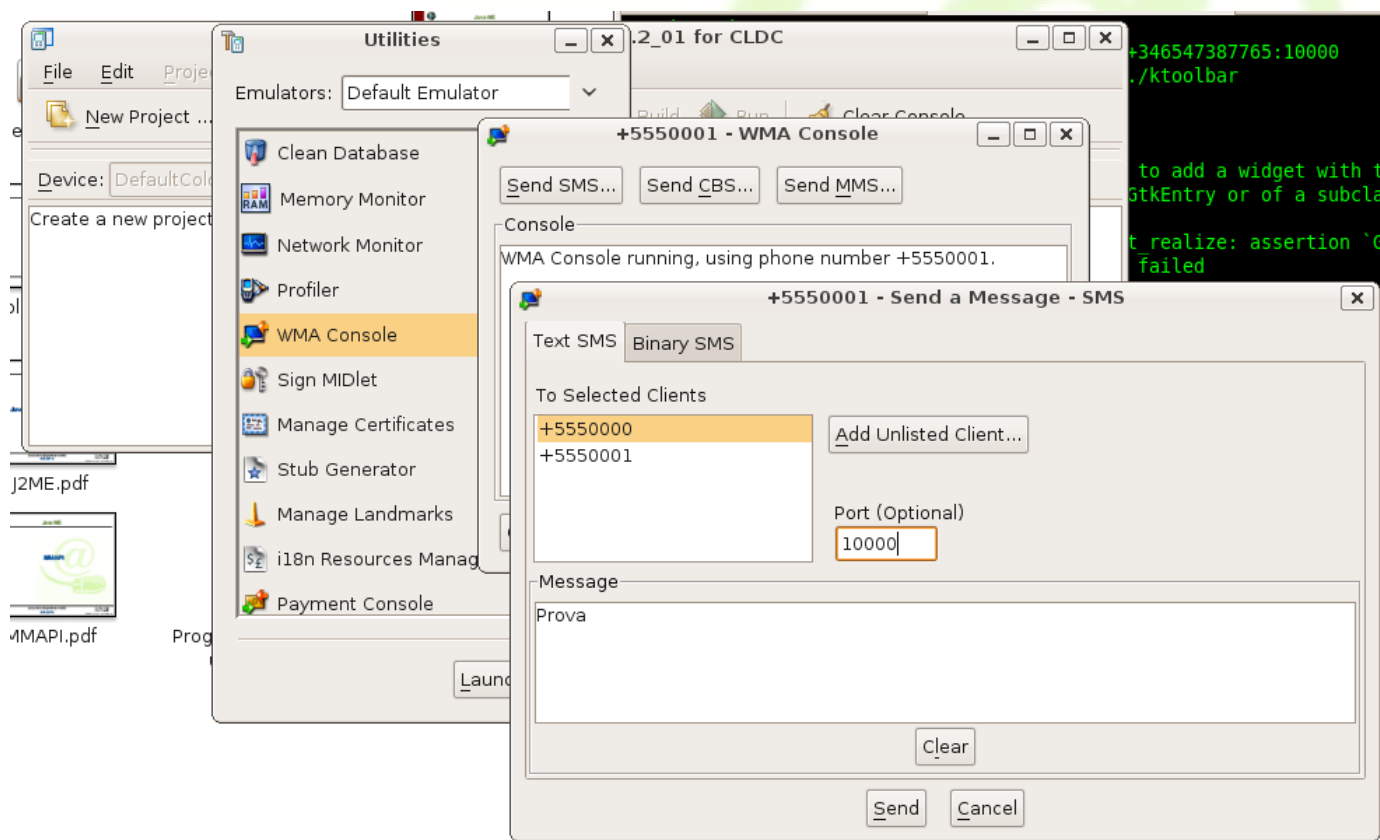
```
#
# Permissions to use specific SMS features
com.sun.midp.io.j2me.sms.permission.receive: true
com.sun.midp.io.j2me.sms.permission.send: true

# Permissions to use connection handlers
javax.microedition.io.Connector.sms: true
javax.microedition.io.Connector.cbs: true
```



# Sun Wireless Toolkit

- **Nº de telèfon del emulador:**
- **Console WMA**





# Wireless Messaging API (WMA)

## ♦ Missatgeria P2P

- ♦ WMA proporciona la possibilitat d'enviar missatges entre mòbils (peers)
- ♦ El mòbil receptor ha de tenir una aplicació que sàpiga llegir els missatges que enviem
- ♦ **NOTA:** Si no indiqueu el port, s'utilitzarà l'aplicació per defecte de recepció de missatges.

## ♦ JSMS API

- ♦ Hi han APIs en Java SE que permeten connectar a SMS Service Centers per enviar missatges a mòbils
- ♦ Aquestes aplicacions es poden “portar” a Java ME utilitzant GCF

## ♦ Recursos

- ♦ <http://developers.sun.com/mobility/midp/articles/wma2/>



# Activació d'aplicacions per Push

## ♦ Push vs Pull

### ♦ Model Pull

- Quan un usuari vol tenir dades ha de estirar (pull) les dades. Ha d'entrar a l'aplicació i sol·licitar les dades

### ♦ Model Push

- Les dades són empeses (push) cap a l'usuari
- L'usuari pot rebre dades en el moment que aquestes estiguin disponibles sense necessitat d'haver sol·licitat les dades.



# Activació d'aplicacions per Push

## ◆ Connexions entrants

- ◆ Podem activar una aplicació al rebre una connexió d'entrada
  - Sockets, datagrames, missatges SMS/MMS...
- ◆ Normalment el mòbil no tindrà un IP fixe (sockets i datagrames no són gaire adequats)
- ◆ El numero de telèfon si es fixe. Podem utilitzar SMS
- ◆ 2 opcions:
  - Registrar la connexió estàticament al fitxer JAD
  - Dinàmicament, utilitzant l'API de PushRegistry



# Activació d'aplicacions per Push

## Registrar la connexió estàticament al fitxer JAD

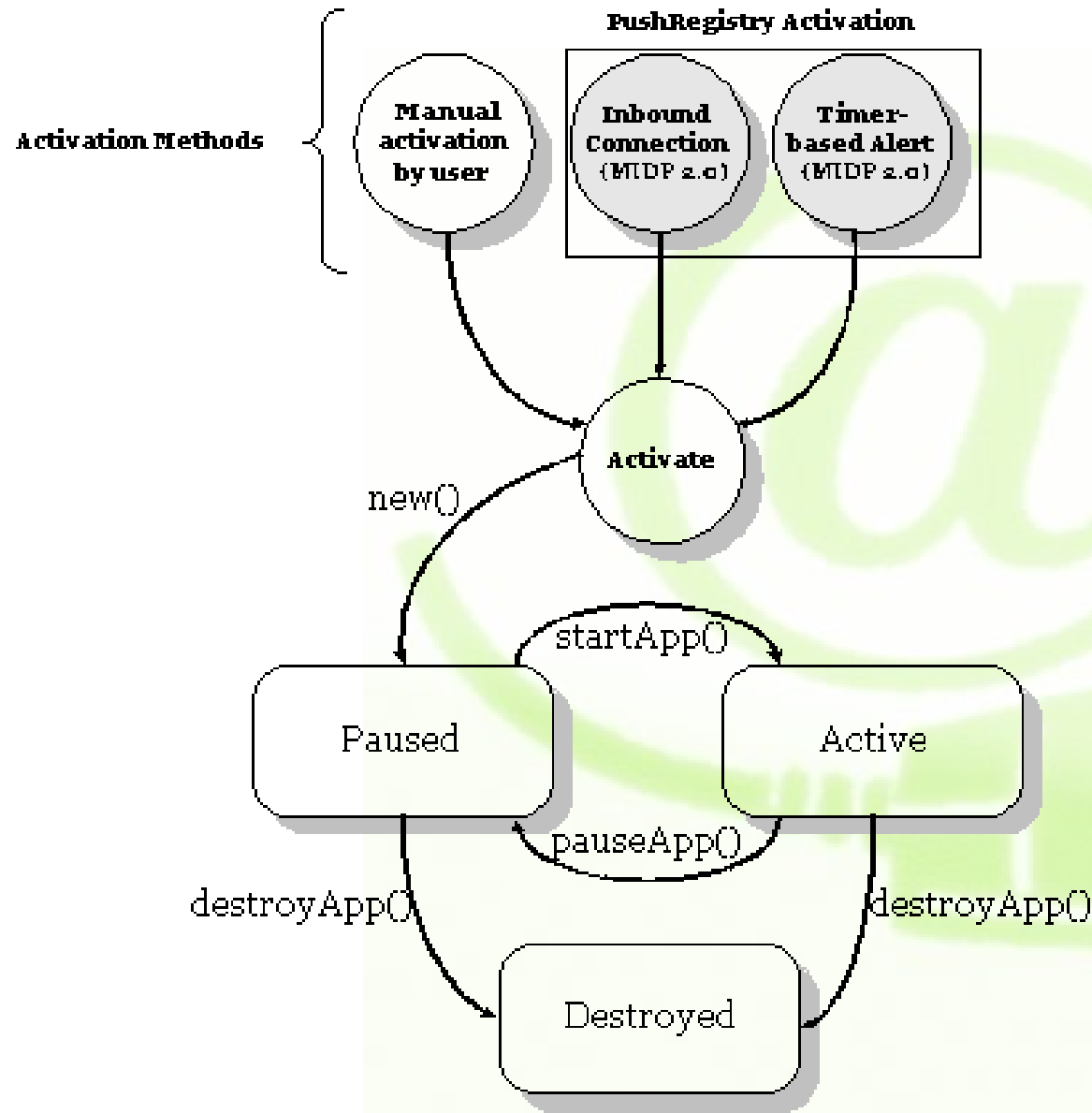
```
MIDlet-Push-1: sms:///6226,tw.subastas.micro.MIDletPrincipal,*
```

## Dinàmicament, utilitzant l'API de PushRegistry

```
PushRegistry.registerConnection(url, nombreClaseMIDlet, remitentesPermitidos);
```



# Activació d'aplicacions per Push





# Netbeans PushRegistry

WMADemo1

Project Configuration: DefaultConfiguration Manage Configurations...

Category:

- General
- Platform
- Abilities
- Application Descriptor
- Build
  - Sources Filtering
  - Compiling
  - Libraries & Resources
  - Obfuscating
  - Creating JAR
  - Signing
  - Generating Javadoc
- Running
- Deploying

Attributes MIDlets **Push Registry** API Permissions

☐ Use Values from "DefaultConfiguration"

Push Registry Entries:

MIDlet Class	Sender IP	Connection String
example.sms.SMSReceive	*	sms://:50000
example.cbs.CBSReceive	*	cbs://:50001
example.mms.MMSReceive	*	mms://:example.mms.MMSDemo

Add...  
Edit...  
Remove  
Move Up  
Move Down

OK Cancel Help





# Bluetooth Optional API

## ♦ Bluetooth Optional API

- ♦ Permet als desenvolupadors, crear aplicacions que utilitzin la tecnologia Bluetooth
- ♦ Java Community Process: [JSR 82](#)
- ♦ També anomenada Java platform access Bluetooth through the Java APIs for Bluetooth Wireless Technology (JABWT o BTAPI)
- ♦ Bluetooth pot ser utilitzar per intercanviar fitxers, fotos, business cards o qualsevol tipus de dades entre dos dispositius mòbils.



# Bluetooth Optional API

## ♦ L'emulador suporta bluetooth

### ♦ Demos WTK

- Obex Demo
- Bluetooth demo

### ♦ OBEX

- És l'abreviatura de OBject Exchange
- És un protocol de comunicacions que facilita l'intercanvi de fitxers binaris entre dispositius.
- El protocol es mantingut per la "Infrared Data Association" i ha estat adoptat per Bluetooth.
- Inicialment s'utilitzava amb sistemes d'infrarojos.

### ♦ WTK demo:

Ctrl+Shift+N > Samples > MIDP > Bluetooth Demo



# PDA Optional Package

## ♦ 2 components

### ♦ Personal Information Management (PIM)

- Permet accedir a la informació de l'agenda i als contactes (address book)

### ♦ FileConnection (FC) JSR 75

- Dona accés als fitxers del dispositiu mòbil
- Els fitxers poden estar emmagatzemats a la memòria interna o en memòries externes/secundàries.
- Paquet: **javax.microedition.io.file**, .
- Necessari si les aplicacions Java volen accedir a imatges, cançons, vídeos o altres fitxers del dispositiu.
- Necessita permisos.

## ♦ Suportat per l'emulador de Sun Wireless Toolkit



# PDA Optional Package

## ♦ PDAP Demo

- ♦ **Ctrl + Alt + N > Samples > MIDP > PDAP Demo**
- ♦ Sistema de fitxers de l'emulador:

`/home/sergi/netbeans-6.1/mobility8/WTk2.5.2/j2mewtk_template/appdb/DefaultColorPhone/filesystem`

## ♦ Memòria de l'emulador:

- ♦ `WTk_HOME/j2mewtk_template/appdb`
- ♦ Conté informació com les bases de dades RMS



# Altres API

## ♦ Location API for J2ME

- ♦ Llibreries per facilitar la creació d'aplicacions location-aware
  - Poden utilitzar el GPS o el mòbil directament (triangulació d'estacions mòbils)
  - [Google Latitude](#)

## ♦ Mobile 3D Graphics API for J2ME 1.1

- ♦ Binding per tal de poder utilitzar OpenGL (requeriment de maquinari)

## ♦ Advanced Multimedia Supplements (AMMS) API

- ♦ Permet crear aplicacions multimedia avançades.  
JSR-234



# Altres API

- ♦ **Content Handler API (JSR-211)**
  - ♦ Permet a les aplicacions invocar altres aplicacions segons una URL, un tipus de contingut MIME o per identificadors.
- ♦ **Security and Trust Services API (JSR 177 i JSR 219)**
  - ♦ Suport per a criptografia i altres serveis de seguretat
- ♦ **CDC**
  - ♦ RMI (Remote Method Invocation)
  - ♦ JDBC
  - ♦ Java TV™ API



## 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>