



Java Mobile Developers Forum  
<http://jmdf.java2me.org>

# Java ME per Palmari e PDA



Stefano Sanna  
<http://www.gerdavax.it>



# Parliamo di...

- Java ME: non solo cellulari...
- CLDC e MIDP su PDA
  - J9 su Windows Mobile
- CDC e Personal Profile
  - J9 su Windows Mobile e Symbian (Series 80)
- Alcune applicazioni d'esempio
- Conclusioni



## Il mondo dei PDA

- I PDA offrono caratteristiche interessanti per applicazioni heavy-mobile:
  - Display di grandi dimensioni: QVGA, VGA...
  - Input completo: tastiera + touchscreen + keypad
  - Connettività: IrDA, Bluetooth, Wifi, GPRS/EDGE/UMTS
  - Potenza hardware: processori fino a 600MHz, RAM fino a 128MB
  - Espandibilità: SD, MMC, CompactFlash, USB...
  - Accessori: barcode reader, RDIF reader...
  - Base installata in crescita, grazie alla diffusione di sistemi di navigazione satellitare

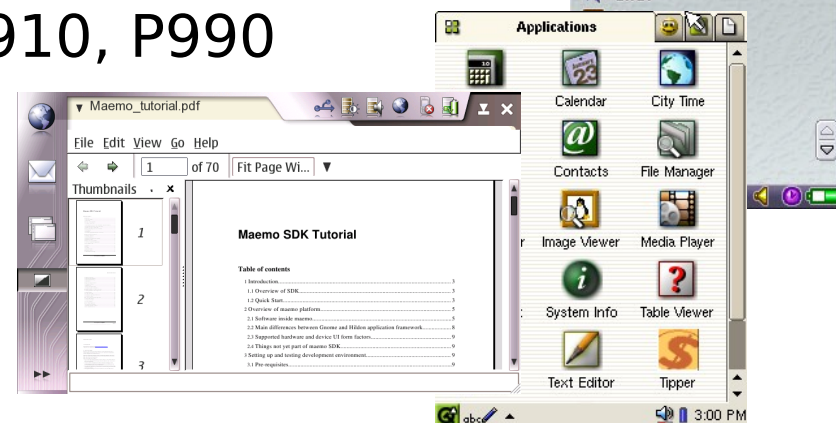


# Più potenza, meno limiti

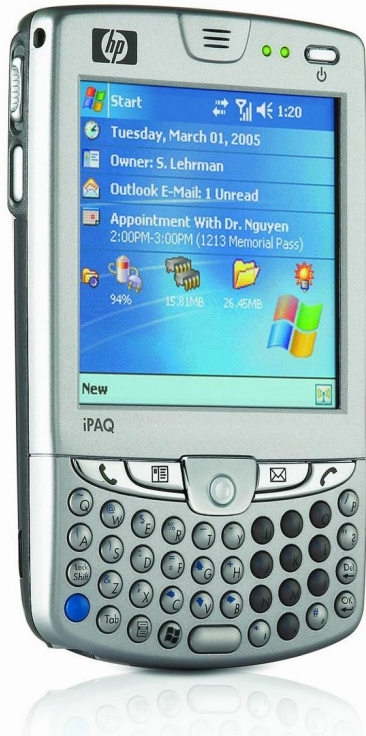
- Queste caratteristiche hanno implicazioni immediate:
  - Display: **cartografia**, multimedia, on-field operations
  - Input: interfacce più ricche e **intuitive**
  - Connettività: abbattimento dei costi (Wifi)
  - Potenza hardware: **VoIP**, cartografia, offline operations
- I PDA non sono cellulari potenziati: sono dispositivi diversi, con **specifici ambiti applicativi**

# Sistemi operativi per PDA

- Windows Mobile
  - HP iPAQ
  - Qtek 9xxx
- PalmOS
  - Palm Treo, LifeDrive, Tungsten...
- Symbian
  - Sony-Ericsson P800, P900/910, P990
  - Nokia 9300, 9500
- Linux
  - Sharp Zaurus (Qtopia)
  - Nokia 770 (Maemo)



# PDA: cosa offre il mercato



HP iPAQ 6515  
GPRS + BT+ WiFi + **GPS**

Qtek 9000:  
**UMTS** + BT + WiFi + **VGA**

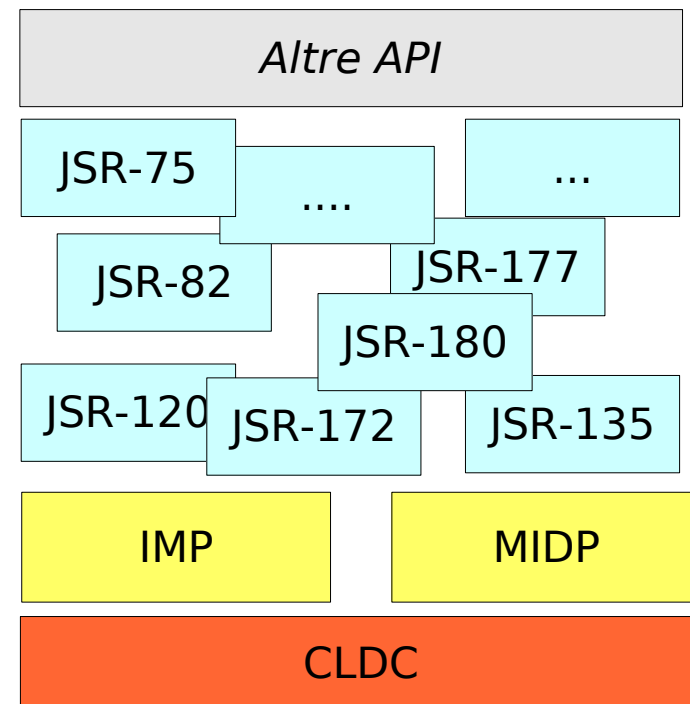
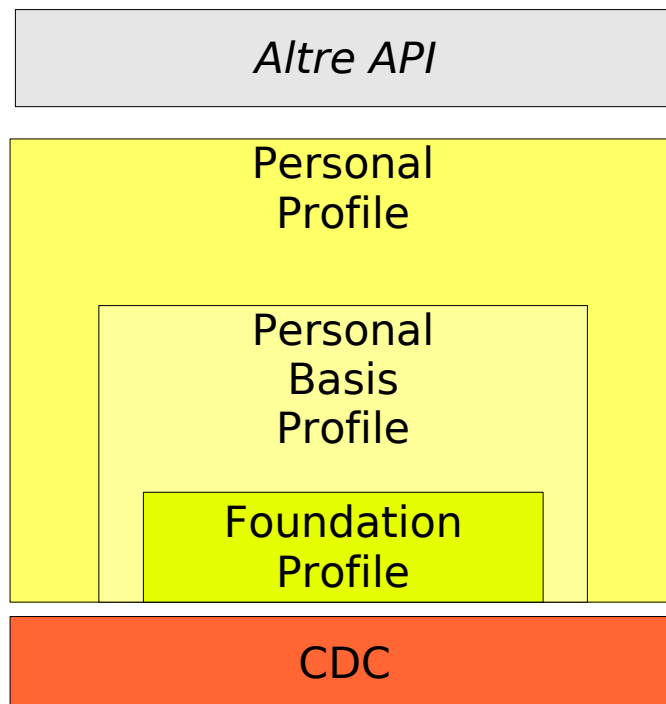


Qtek 9100  
EDGE + BT+ WiFi(**g**)



# Java ME: CDC e CLDC

- La piattaforma Java ME definisce attualmente due famiglie di dispositivi, corrispondenti alle configurazioni CDC e CLDC:





## MIDP su PDA... ieri

- Le prime versioni del Wireless Toolkit includevano una utility per il deployment delle MIDlet Suite su PalmOS
- Principali limitazioni:
  - Memoria disponibile estremamente limitata
  - Mancanza di sistema operativo multitasking
  - Complessità del processo di deployment: l'applicazione doveva essere compilata esplicitamente per la piattaforma Palm



## MIDP su PDA... oggi!

- I moderni runtime MIDP offrono supporto di ottimo livello:
  - Installazione semplice
  - Gestione dettagliata dei permessi assegnati a ciascuna applicazione
  - Interfaccia utente coerente con il sistema operativo
  - Buona integrazione con alcuni API native
- Non sempre le VM sono installate sul dispositivo: l'utente deve installarle manualmente o **acquistarle separatamente**



# Runtime MIDP per Pocket PC

- IBM WebSphere Everyplace Micro Environment
  - [http://www-128.ibm.com/developerworks/websphere/zones/wireless/weme\\_eval\\_runtimes.html](http://www-128.ibm.com/developerworks/websphere/zones/wireless/weme_eval_runtimes.html)
- ACCESS JV-Lite2 Wireless Profile
  - <http://www.accesschina.com.cn/products/midp.htm>
- Tao Group Intent Platform, Midlet Manager
  - <http://taogroup.com/main.php?pageid=254911.php&temptype=t1>



## J9: efficace ed economico

- IBM distribuisce il runtime WEME, basato sulla VM J9: evaluation (senza time bomb!) gratuita, licenza di utilizzo intorno ai 5-6euro per istanza!
- Principali caratteristiche:
  - Supporto:
    - CLDC 1.1, MIDP 2.0, JSR-75
  - Compatibilità:
    - Windows Mobile 2003, Windows Mobile 5.0, Palm OS, Linux
- Buone prestazioni, ottima integrazione con il sistema operativo

# Gestione MIDlet e permessi

**MIDlet List** 19.06

- HelloWSWorld - 1.0
- SerialMIDlet - 1.0
- MicroDynDNS - 1.0
- Sketch - 1.0**

Install Launch Manage Delete

Actions

**Choose Permission** 19.28

- Comm
- Contact List Read
- Contact List Write
- Datagram
- Datagram Receiver
- Event List Read
- Event List Write
- File Read
- File Write
- HTTP
- HTTPS
- Push Registry
- SSL
- Server Socket
- Socket
- ToDo List Read
- ToDo List Write
- Video Control - Snapshot

Actions Navigate

**Permission Interac** 19.28

If Sketch wants to activate a local port for connectivity to another computer, prompt me ...

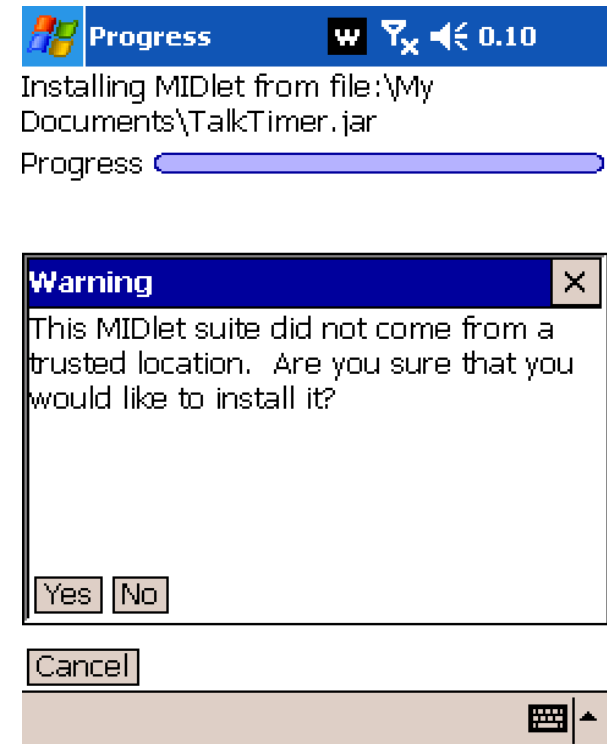
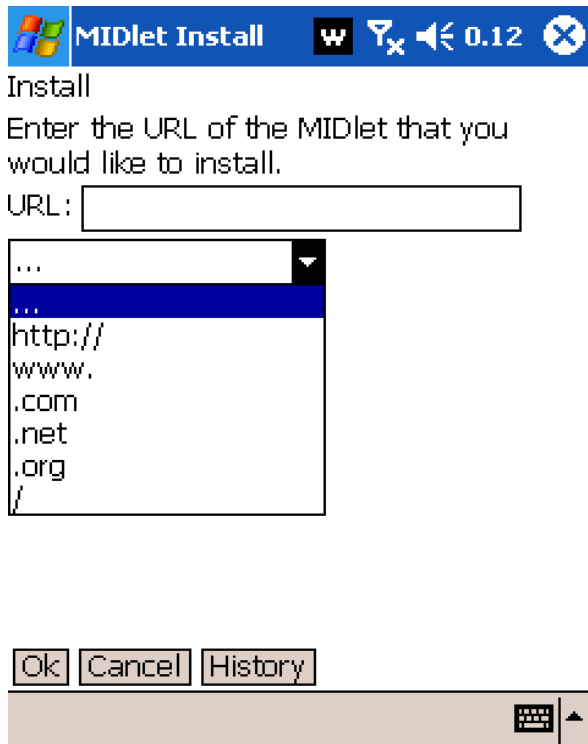
- Not allowed. Never prompt again.
- Each time the function is used (oneshot).
- Once when the application runs (session).
- Always allowed. Never prompt again (blanket).

Save Cancel



# MIDP: installazione

- E' possibile via rete o JAR file locale:





# Canvas: finalmente il pointer!

- Il Canvas su PDA supporta la gestione degli eventi relativi al puntatore (penna):
  - **hasPointerEvents()** e **hasPointerMotionEvents()** permettono di conoscere a runtime le caratteristiche del dispositivo
  - **pointerPressed()**, **pointerDragged()** e **pointerReleased()** sono i metodi di callback invocati quando lo stato del puntatore cambia. I metodi ricevono le coordinate x,y del punto al quale l'evento si riferisce
- Proviamo a scrivere uno Sketch pad...



# Sketch: il telaio...

```
public class Sketch extends MIDlet {  
    private SketchCanvas gui;  
  
    public Sketch() { gui = new SketchCanvas(); }  
  
    public void startApp() {  
        Display.getDisplay(this).setCurrent(gui);  
    }  
  
    public void pauseApp() { ... }  
  
    public void destroyApp(boolean unconditional) { ... }  
}
```



# SketchCanvas /1

```
public class SketchCanvas extends Canvas implements CommandListener {
    private Image mainBoard;
    private Image tempBoard;
    private Graphics mainG;
    private int lastX, lastY;
    private Command clearCommand = new Command("Clear", Command.SCREEN, 1);

    public SketchCanvas() {
        if (! (hasPointerEvents() && hasPointerMotionEvents())) {
            throw new RuntimeException("Can't use touch screen!");
        }
        init();
    }
}
```



# SketchCanvas /2

```
private void init() {
    setTitle("Sketch");
    mainBoard = Image.createImage(this.getWidth(), this.getHeight());
    mainG = mainBoard.getGraphics();
    addCommand(clearCommand);
    setCommandListener(this);
}

private void clear() {
    mainG.setColor(0xEEEEEE);
    mainG.fillRect(0, 0, this.getWidth(), this.getHeight());
}
```



# SketchCanvas /3

```
public void pointerPressed(int x, int y) {
    lastX = x;
    lastY = y;
    mainG.setColor(0xFF0000);
    mainG.drawLine(x, y, x, y);
    repaint();
}

public void pointerDragged(int x, int y) {
    mainG.setColor(0xFF0000);
    mainG.drawLine(lastX, lastY, x, y);
    lastX = x;
    lastY = y;
    repaint();
}
```



# SketchCanvas /4

```
public void paint(Graphics g) {  
    if (mainBoard != null) {  
        g.drawImage(mainBoard, 0, 0, Graphics.TOP | Graphics.LEFT);  
    }  
}  
  
public void pointerReleased(int x, int y) {  
    repaint();  
}
```





## Note sul Wireless Toolkit

- Il WTK fornisce emulatori “phone oriented”, senza supporto per l'input via penna. Per abilitarlo è sufficiente impostare un flag all'interno del file

WTK\_DIR\wtllib\devices\DEVICE\_NAME\DEVICE\_NAME.properties

ad esempio:

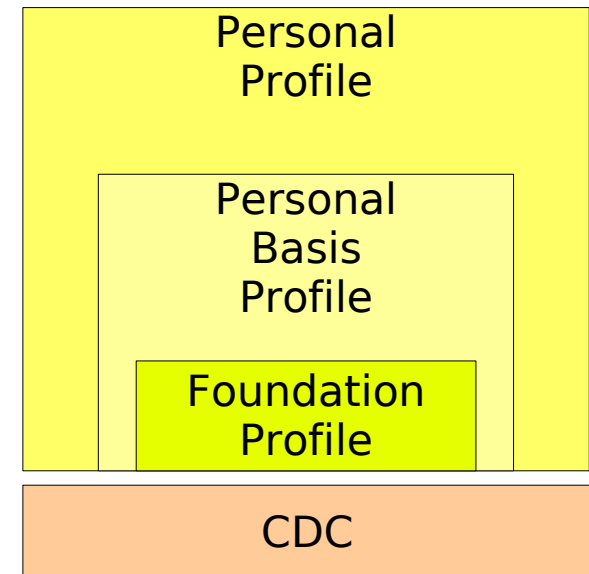
c:\WTK23\wtllib\devices\DefaultColorPhone\DefaultColorPhone.properties

**touch\_screen=true**



# CDC, FP, PBP, PP: la squadra!

- Rispetto al mondo MIDP, CDC e i relativi profili costituiscono un ecosistema complesso e modulare:
  - **CDC**:
    - Connected Device Configuration
  - **FP**:
    - Foundation Profile
  - **PBP**:
    - Personal Basis Profile
  - **PP**:
    - Personal Profile





# Chi fa cosa: CDC e FP

- **CDC** (JSR 36):
  - Fornisce una VM compatibile J2SE, con un subset della libreria standard (1.3 per CDC 1.0, 1.4 per CDC 1.1):
    - java.lang, java.lang.ref, java.lang.reflect, java.math
    - java.io, java.net, javax.microedition.io
    - java.security, java.security.cert
    - java.text, java.util, java.util.jar, java.util.zip
- **Foundation Profile** (JSR 46):
  - Completa i package forniti da CDC, per avere piena copertura della libreria standard 1.3.1



# Chi fa cosa: PBP e PP

- **Personal Basis Profile** (JSR 129):
  - Aggiunge il supporto base AWT per componenti lightweight e introduce il modello di applicazioni basato su Xlet:
    - java.awt, java.awt.event, java.awt.color, java.awt.image
    - java.beans, java.rmi, java.rmi.registry (per Xlet)
    - javax.microedition.xlet, javax.microedition.xlet.ixc
- **Personal Profile** (JSR 62):
  - Completa il supporto AWT con i componenti nativi:
    - java.applet, java.awt.datatransfer



## CDC e PP: vantaggi in breve

- Standard library molto più ricca, sottoinsieme della Standard Edition 1.4 (CDC 1.1)
- Sfruttano le prestazioni dei palmari e degli smartphone di fascia alta
- Supporto JNI: ciò che il runtime non offre può essere sviluppato ad hoc se necessario
- Possibilità di usare componenti nativi e di utilizzare altre librerie grafiche (HAVi, DVB...)
- Rivisitazione in chiave “JME” dell'esperienza maturata su Personal Java



# CDC: tre modelli applicativi

- Il top della gamma Java ME lascia ampia libertà allo sviluppatore, che può scegliere il modello applicativo più opportuno:
  - **Stand-alone**: il metodo main() costituisce l'entry point dell'applicazione
  - **Managed**: definendo una Xlet, gestita a sua volta da un XletManager, che riprende alcuni dei concetti delle MIDlet e, soprattutto, delle Xlet inizialmente introdotte con Java TV
  - **Embedded**: attraverso Applet eseguite all'interno dell'AppletViewer o di browser opportunamente esteso



## CDC/PP: alcune particolarità

- CDC offre supporto al Generic Connection Framework: si tratta di una implementazione ancora limitata e non tutti gli schema di connessione sono supportati (generalmente sono HttpURLConnection), talvolta con leggere differenze rispetto a CLDC/MIDP
- La VM dispone di una console e i runtime più diffusi permettono di impostare opportuni parametri di avvio per visualizzare lo standard output e ricevere comando attraverso lo standard input



## CDC: API opzionali

- Sono attualmente definite alcune API per CDC:
  - J2ME RMI Optional Package, (RMI OP); JSR 66
  - JDBC Optional Package for CDC/Foundation Profile API; JSR 169
  - Java TV API; JSR 927
  - Java Secure Socket Extension for CDC (JSSE)
  - Advanced Graphics and UI (fornirà supporto a Swing, Java 2D, Image I/O, Input Method Framework APIs for more capable mobile devices); JSR 209
- Per maggiori informazioni:
  - <http://java.sun.com/j2me/>



## Criticità VM su PDA

- Rispetto al runtime Java distribuito con un telefono cellulare, la VM per PDA è un... software di terze parti e può presentare diverse criticità:
  - Scarsa coerenza con il look & feel del sistema; non è possibile conoscere l'effettiva dimensione della porzione di schermo assegnata all'applicazione
  - Scarso sfruttamento delle risorse hardware disponibili (memoria, risoluzione)
  - Supporto JNI instabile
  - Forte dipendenza dalla versione del sistema operativo



# PersonalJava (Jeode) su WinCE

Dov'è l'icona di Windows

A cosa servono?

Perché i menù sono qui e non in basso?

La dimensione della finestra è stata impostata **MANUALMENTE**, tenendo conto anche della dimensione della barra di Windows CE: **FOLLIA!**

???????????????





## Portabilità: le regole di PP

- La specifica precisa che alcune funzionalità sono dipendenti dalla piattaforma. In alcuni casi l'invocazione di una certa funzionalità non produce alcun effetto, in altri casi la VM solleva una eccezione.
- Alcuni esempi:
  - `java.awt.Frame` potrebbe non essere ridimensionabile né collocabile a piacere sul display
  - gestione eventi e colori possono avere limitazioni
- Non tutte le restrizioni sono esplicite (`getButton()`)



# Runtime CDC/PP per Pocket PC

- Esmertec JBed
  - <http://www.esmertec.com/>
- NSICom CrE-ME 4.1
  - <http://www.nsicom.com/>
- Savaje SavajeOS
  - <http://www.savaje.com/>
- Mysaifu JVM
  - [http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index\\_en.html](http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html)



# Runtime CDC/PP per Pocket PC

- IBM
  - WebSphere Everyplace Micro Environment 5.7.x
    - Windows Mobile 2003 SE
    - MontaVista Linux
  - WebSphere Everyplace Micro Environment 6.0
    - Windows Mobile 5.0 (anche VGA!)
    - Windows Mobile 2003 Smartphone Edition
  - Nokia 9300, 9300i, 9500

[http://www-128.ibm.com/developerworks/websphere/zones/wireless/weme\\_eval\\_runtimes.html](http://www-128.ibm.com/developerworks/websphere/zones/wireless/weme_eval_runtimes.html)



## J9 su Windows Mobile

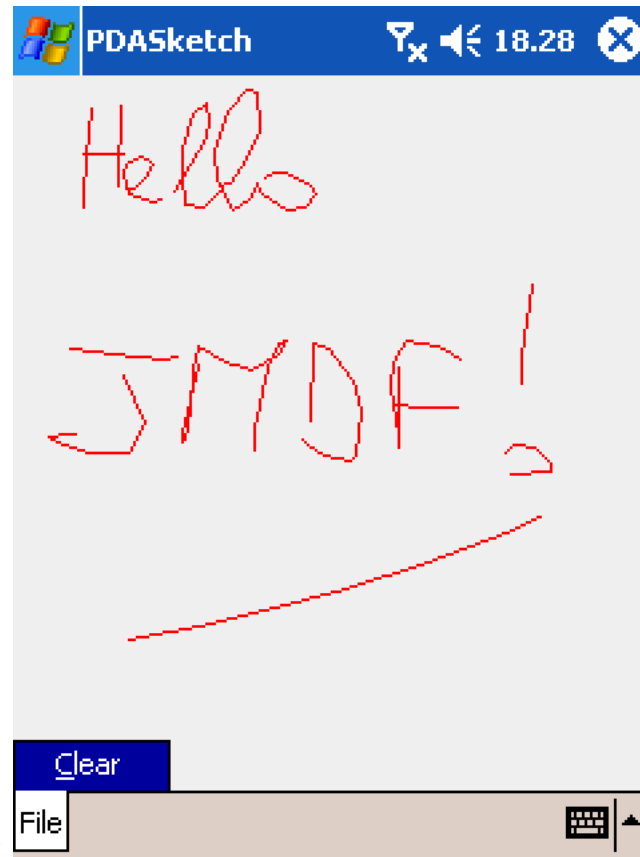
- La VM di IBM è coerente con il sistema operativo e permette di gestire in modo efficiente eventuali librerie native
- La presenza di AWT permette realizzare interfacce grafiche interessanti, anche per l'uso del touch screen senza penna
- Questo “Domotipad” potrebbe essere il prototipo di una applicazioni di domotica basata su Personal Profile



# PDASketch su J9

L'icona di Windows è a posto

Invoca correttamente la chiusura dell'applicazione



La dimensione della finestra è stata impostata **AUTOMATICAMENTE!**

La posizione dei menù è coerente con il L&F del sistema operativo



# PDASketch su J9

```
public void mouseDragged(MouseEvent mouseEvent) {
    mainG.setColor(new Color(0xFF0000));
    mainG.drawLine(lastX, lastY, mouseEvent.getX(), mouseEvent.getY());
    lastX = mouseEvent.getX();
    lastY = mouseEvent.getY();
    repaint();
}

public void mousePressed(MouseEvent mouseEvent) {
    lastX = mouseEvent.getX();
    lastY = mouseEvent.getY();
    mainG.setColor(new Color(0xFF0000));
    mainG.drawLine(lastX, lastY, lastX, lastY);
    repaint();
}
```



## Ambiente di sviluppo

- Non essendo necessaria (per ora) una fase di preverifica né un bootclasspath dedicato, è possibile utilizzare **qualsiasi** ambiente di sviluppo, purché disponga di una configurazione JDK **1.4.x**
- IBM distribuisce il WebSphere Device Developer, che include i runtime e un completo ambiente di sviluppo con gestione progetti CDC/PP
- In generale occorre:
  - **esperienza**, per tenere a mente quali metodi sono effettivamente implementati su CDC/FP/PBP/PP
  - **conoscenza** del runtime in uso



## PP in emulazione su Series 80

- Nokia fornisce Carbide.j, una suite completa di strumenti per il supporto allo sviluppo di applicazioni Java ME CLDC/MIDP e **CDC/PP**
- Carbide.j può essere utilizzato come strumento stand-alone o integrato in Netbeans, Eclipse, WebSphere Device Developer
- Per avere il supporto CDC/PP:
  - installare un SDK specifico per la Series 80 (S80\_DP\_2\_0\_PP\_SDK.zip)
  - aggiungere il nuovo SDK tra gli emulatori di Carbide.j



# PP in emulazione su Series 80





# Deployment su PDA

- file JAR contenente le classi e le eventuali risorse
- script **NOME\_APP.Ink** per l'avvio:

```
255#\Program Files\J9\PPR010\bin\j9.exe" "-jcl:ppro10" "-cp" "\My Documents\Domotipad.jar" org.jugsardegna.mobile.domotipad.Domotipad
```

**Tutto su una linea!**

```
255#  
"\Program Files\J9\PPR010\bin\j9.exe"  
"-jcl:ppro10"  
"-cp" "\My Documents\Domotipad.jar"  
org.jugsardegna.mobile.domotipad.Domotipad
```

Lunghezza script (inutile!) →

Percorso JVM →

Percorso del JAR →

Class Library in uso →

fn della main class →



# Deployment su Nokia Series 80

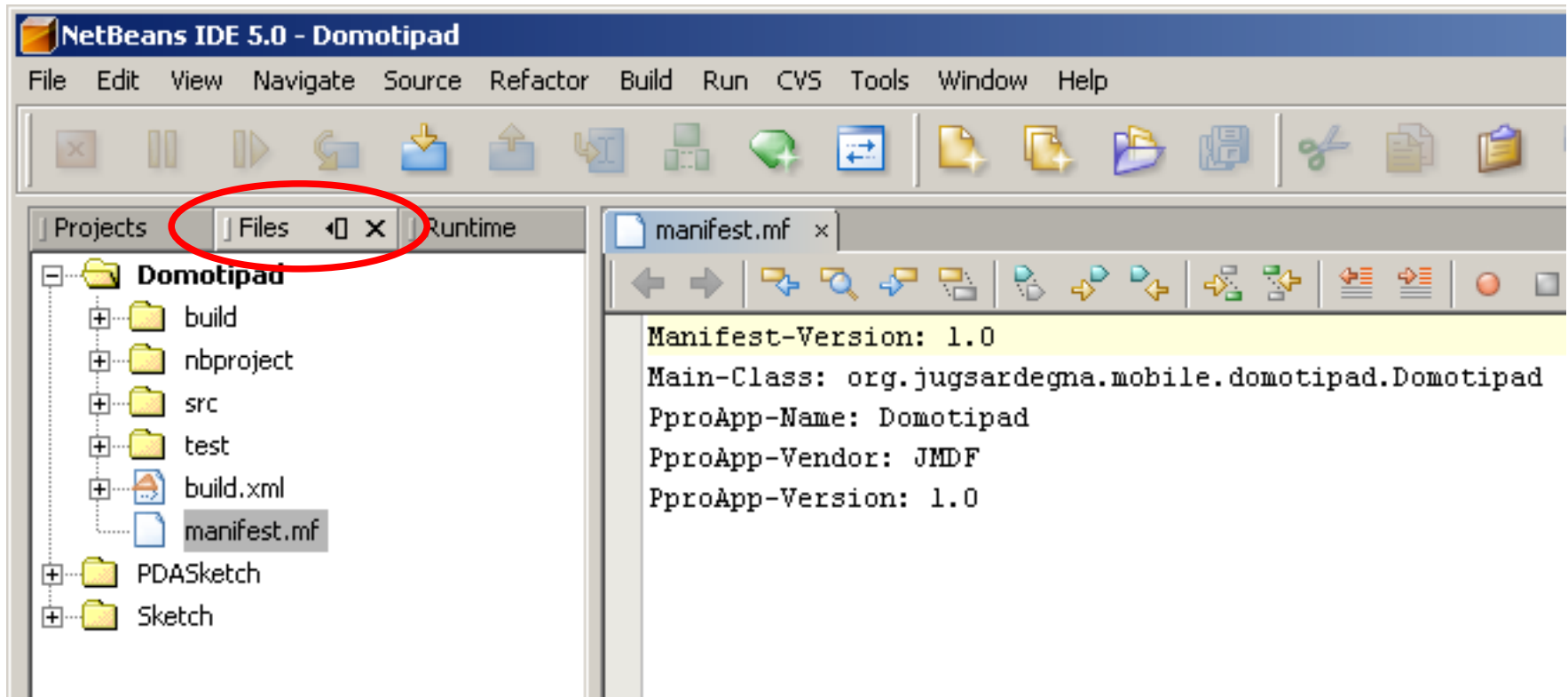
- In questo caso, il deployment assomiglia a quanto già visto in ambiente MIDP:
  - il file Manifest **deve** contenere le seguenti property:

```
Main-class: org.mycompany.MyApplicationMainClass  
PproApp-Name: MyApplication  
PproApp-Vendor: MyCompany  
PproApp-Version: 0.0.1
```

- Senza queste informazioni l'applicazione **non viene installata!**



# Gestione Manifest su Netbeans





## API opzionali

- Spesso CLDC/MIDP e CDC/PP coesistono all'interno del medesimo dispositivo: le librerie opzionali, però, sono specifiche di ciascun runtime!
- Non è possibile, ad esempio, avere la Wireless Messaging API su CDC
- La presenza di JNI su CDC/PP permette di superare alcune limitazioni e accedere direttamente alle funzionalità del sistema operativo



# Lettura dati GPS Bluetooth

- La JSR-82 è utilissima per accedere a GPS Bluetooth:
  - discovery del dispositivo
  - accesso allo stream NMEA attraverso RFCOMM
- Attualmente J9 non include il supporto all'API per Bluetooth, ma è possibile raggiungere il risultato effettuando:
  - discovery e associazione del dispositivo attraverso le funzionalità del sistema operativo
  - accesso stream attraverso la porta seriale esportata dal sistema e una opportuna implementazione di **JavaComm**



# Discovery GPS /1

Start 9.45

domenica 19 febbraio 2006

Imposta informazioni su utente

Nessun messaggio da leggere

Nessuna attività

Nessun appuntamento programmato

Disattiva Bluetooth

---

Collegamenti Bluetooth

**Impostazioni Bluetooth**

Nuovo

Impostazioni Blueto 9.46 ok

## Bluetooth: Generali

### Profilo attivo

Predefinito 🔍

### Stato Bluetooth

**Bluetooth è ATTIVATO**

Attiva

Disattiva

Usare [Bluetooth Manager](#) per impostare le connessioni.

Generali | Accessibilità | Servizi | Informazioni su

Impostazioni Blueto 9.46 ok

## Bluetooth: Servizi

### Servizi

- Trasferimento file ▲
- Scambio di informazioni ☰
- Porta seriale** □
- Server di rete personale ▼

### Impostazioni servizio

- Attiva servizio
- Richiede autorizzazione
- Richiede autenticazione (passepartout)
- Richiede crittografia

Avanzate...

Generali | Accessibilità | Servizi | Informazioni su

# Discovery GPS /2

Impostazioni Blueto 9.47

## Porta seriale Bluetooth

### Porte

Porta COM in entrata:

Porta COM in uscita:



La periferica predefinita per la connessione seriale è AKI. Visualizzare la schermata di selezione della periferica quando si ristabilisce la connessione.



Start 9.47

domenica 19 febbraio 2006

Imposta informazioni su utente

Nessun messaggio da leggere

Nessuna attività

Nessun appuntamento programmato

Disattiva Bluetooth

**Collegamenti Bluetooth**

Impostazioni Bluetooth

Nuovo

Connessione guidat: 9.52



## Connessione guidata Bluetooth

Rileva un dispositivo Bluetooth



AKI



BTGPS 1536EA Gerdalab 9300



Marlin



zazuo

Annulla Indietro Avanti

Visualizza



# Discovery GPS /3

Connessione guidata: 9.53

**Connessione guidata Bluetooth**  
Rileva un dispositivo Bluetooth

**BTGPS 1536EA**

**Selezione dei servizi**

- Serial Port

Selezionare i servizi offerti da questo dispositivo per i quali si desidera creare dei collegamenti.

**Sicurezza**

Usare una conness. sicura e criptata

**Annulla**   **Indietro**   **Avanti**



**Bluetooth Manager** 9.53

**Bluetooth**  
Collegamenti



BTGPS  
1536EA:  
Serial Port

Collegamenti	Connessioni attive	
Nuovo	Strumenti	Visualizza

**Bluetooth Manager** 9.54

**Bluetooth**  
Collegamenti



BTGPS  
1536EA:  
Serial Port

Collegamenti	Connessioni attive	
Nuovo	Strumenti	Visualizza



# Installazione JavaComm

- Il pacchetto (distribuzione Telio/CESerial) è costituito da quattro file:
  - **comm.jar + CESerial.jar**: contengono le classi Java. Devono essere inclusi nel classpath dell'applicazione
  - **commapi.dll**: è la libreria a caricamento dinamico nativa. Deve essere **copiata nella directory \Windows**
  - **javax.comm.properties**: è il file contenente le property. Deve essere **copiato nella directory contenente i file JAR**

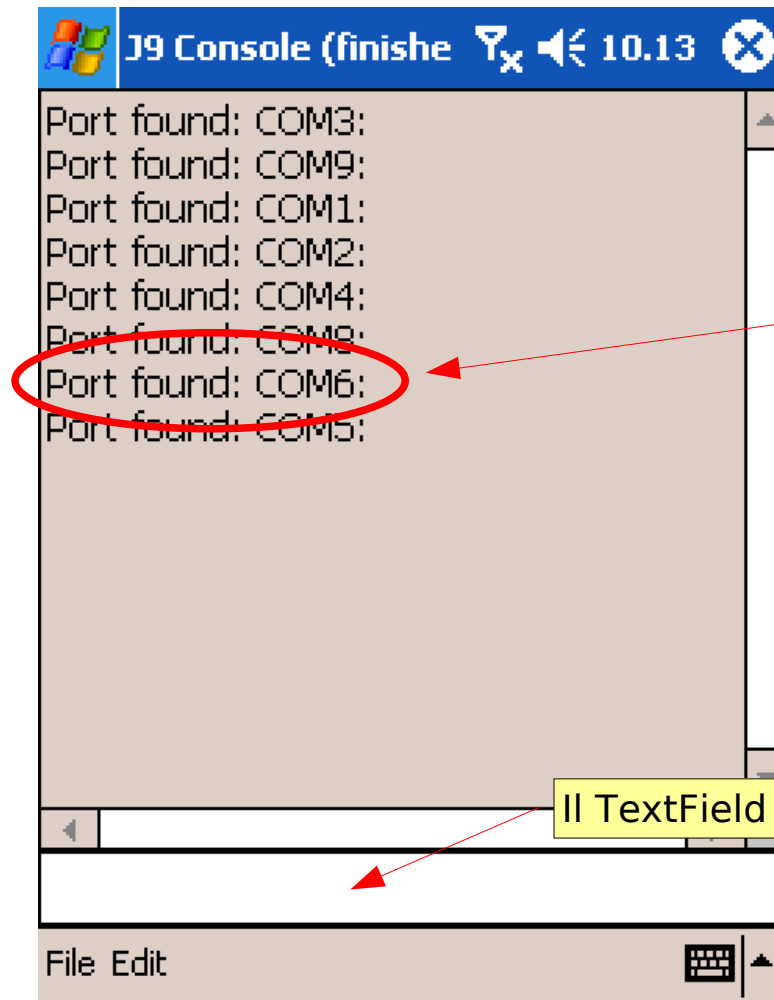


# Enumerazione delle porte

- Attenzione ai nomi delle porte: sistemi operativi diversi utilizzano schemi di naming differenti:

```
Enumeration ports = CommPortIdentifier.getPortIdentifiers();  
  
while (ports.hasMoreElements()) {  
  
    CommPortIdentifier id = (CommPortIdentifier) ports.nextElement();  
  
    System.out.println("Port found: " + id.getName());  
  
}
```

# Enumerazione delle porte



I ":" alla fine dell'identificatore sono specifici della libreria in uso!

Il TextField dello standard input! :-)



# GPSReader: apriamo la console

```
SerialPort port = (SerialPort)
    CommPortIdentifier.getPortIdentifier("COM6:").open("JMDF", 5000);
```

```
port.setSerialPortParams(4800, SerialPort.DATABITS_8, SerialPort.STOPBITS_1,
    SerialPort.PARITY_NONE); // 8N1@4800bps
```

```
BufferedReader buffer = new BufferedReader(new
    InputStreamReader(port.getInputStream()));
String sentence;
while((sentence = buffer.readLine()) != null) {
    System.out.println(sentence);
}
```



## Tips

- Il caricamento delle librerie native richiede una certa quantità di risorse: è consigliabile (e, talvolta, indispensabile!) effettuare un *touch* sulle librerie native affinché vengano caricate all'avvio dell'applicazione (per JavaComm: è sufficiente una enumerazione delle porte!)
- In alcuni runtime, il file browser vede come root directory la cartella “My Documents”: eventuali file posti in altre posizioni non saranno raggiungibili dall'utente



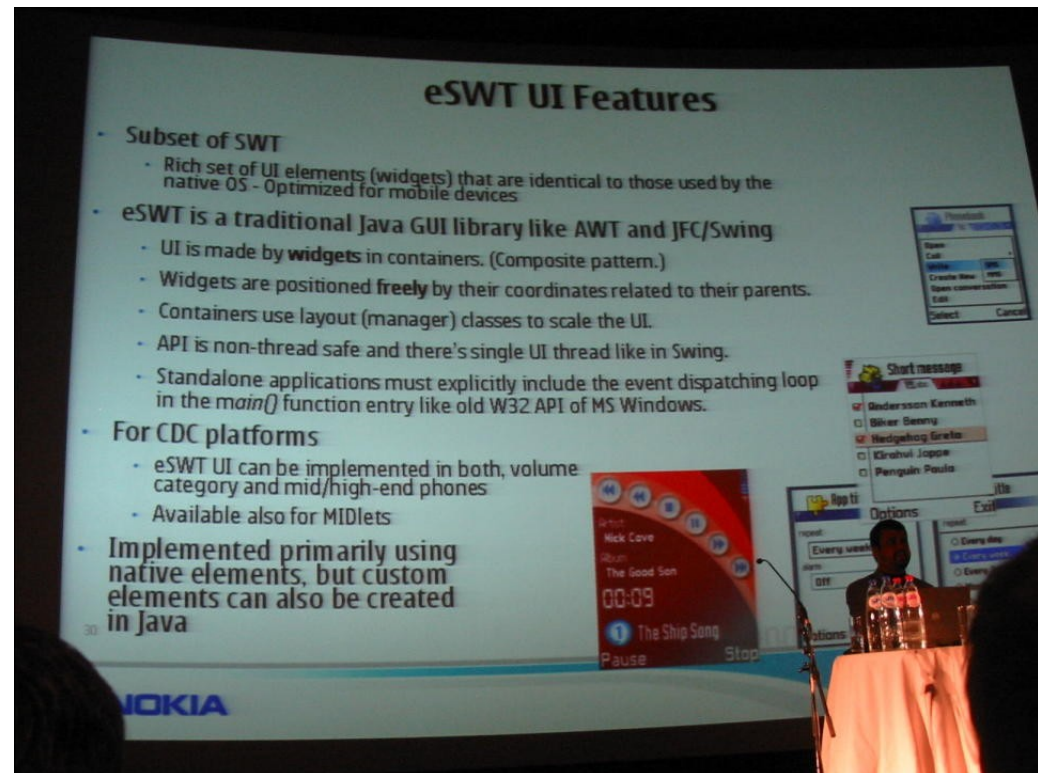
## Non solo AWT...

- Il supporto AWT base fornito dal PBP e la disponibilità di JNI permettono di utilizzare differenti toolkit di componenti grafici:
  - LWVCL:
    - è una libreria di componenti grafici lightweight, attualmente disponibile per J2SE, J2ME (PersonalJava, PersonalProfile, MIDP), SWT, .NET
  - eSWT:
    - è il porting ragionato del set di componenti nativi della piattaforma RCP, nato in seno al progetto Eclipse
    - prestazioni e L&F eccellenti!

## Non solo Series 80...

- A fine 2005 Nokia ha annunciato che le future generazioni dei terminali Series 60 includeranno un runtime CDC. E' iniziato il porting di eSWT per tali dispositivi.

Srikanth Raj a Java Polis 2005





# Conclusioni

- I PDA sono dispositivi insostituibili per usabilità, potenza, connettività e espandibilità: sono la soluzione ideale per numerosi scenari applicativi
- Java ME offre opportunità entry-level (MIDP) e avanzate (PP): come già avvenuto in passato, con risorse modeste si può sperimentare e acquisire dimestichezza con questa classe di device
- Un dubbio: avremo, in futuro, tutti i PDA Java-enabled senza necessità di acquistare software opzionale?

*Ringrazio Demis Castagna (Java Italian Portal, JUG Roma) per il supporto nei test su Windows Mobile 5*



# Specifiche

- PersonalProfile Programmer Guide
  - [http://java.sun.com/j2me/docs/pdf/PP\\_Programmer\\_Guide.pdf](http://java.sun.com/j2me/docs/pdf/PP_Programmer_Guide.pdf)
- Personal Basis Profile Programmer Guide
  - [http://java.sun.com/j2me/docs/pdf/PBP\\_Programmer\\_Guide.pdf](http://java.sun.com/j2me/docs/pdf/PBP_Programmer_Guide.pdf)
- Sun Microsystems fornisce implementazioni di riferimento per l'intero stack basato su CDC. Documentazione, codice e binari su:

<http://java.sun.com/j2me>



# Tools

- Nokia Series 80 Platform SDK's for Symbian OS, for Java (Personal Profile)
  - <http://www.forum.nokia.com/main/0,6566,034-367,00.html>
- Microsoft Remote Display Control
  - <http://www.microsoft.com/technet/prodtechnol/wce/downloads/ppctoys.msp#EGAA>
- IBM WebSphere Wireless Technologies
  - <http://www.ibm.com/developerworks/websphere/zones/wireless/>
- Java Programming on Palm
  - <http://www.palmos.com/dev/tech/java/developers.html>



**Grazie per l'attenzione :-)**

Farmer Clem meets the 21st C by lumix2004  
<http://www.sxc.hu/browse.phtml?f=profile&l=lumix2004>



# Java ME per Palmari e PDA

(Versione 1.1)

**(C) 2006 Stefano Sanna (gerdavax@tiscali.it)**

è garantito il permesso di copiare, distribuire e/o modificare questo documento seguendo i termini della Licenza per Documentazione Libera GNU, Versione 1.1 o ogni versione successiva pubblicata dalla Free Software Foundation. Una copia della licenza in lingua italiana è disponibile presso: <http://www.softwarelibero.it/gnudoc/fdl.it.html>

Realizzato in ambiente Windows con OpenOffice 2.0

*Tutti i marchi commerciali sono di proprietà dei rispettivi titolari e sono stati citati in questa presentazione a solo scopo illustrativo.*