

# PHILLIPS

The Isolate API for J2ME

Richard Houldsworth June 2003

# Why Small Devices Need Isolation

- Digital TV
  - MHP applications considered to logically run in their own VM.
  - OCAP defines conformance tests requiring Xlet isolation.
- Web browser devices
  - Applets in separate pages should be isolated.
- Mobile Phones
  - Phones need to support multiple separated MIDP application suites.
- End-user expectations for device stability
- API user is likely to be *application manager*, not downloaded Applet/Xlet/MIDlet

# What Isolation Gives You

- Reliable domain termination
  - `Isolate.halt()` vs. `ThreadGroup` termination (NB `destroy()` doesn't work).
- Simple solution to 'current domain' queries
  - `Isolate.currentIsolate()` vs.
  - complex and fragile `SecurityManager` stack-fishing/`ClassLoader/ThreadGroup` systems.
- Avoids inadvertent inter-domain effects
  - e.g. `Xlet1` calls middleware calls `Xlet2`.
  - e.g. Locking on shared classes.
- However, does mean more overhead:
  - Link comms vs. method invocation.
  - Overall memory footprint (depending on implementation).

# Transitioning to Isolates

- **Replace (typically) ClassLoader-based application manager with Isolate system**
  - setup lifecycle command/notification Links in initial messages.
  - Isolate main class is XletContext implementation
    - installs local SecurityManager, XletClassLoader.
    - creates Xlet and Thread to service Xlet commands over Link.
  - new Thread on parent to service Link from child.
- **Change resource proxies to use Links**
  - check for direct use of static fields.
- **Prototyping without a real J2ME implementation**
  - SE: use JSR-121 RI.
  - ME: use ‘simulated’ Isolate implementation based on Threads.

# J2ME Isolate API Strategy

- J2ME JSR for one or more Configurations/Profiles
- Pure subset of J2SE specification
- Subset to ‘what the platform provides’
  - simple to do
    - remove NIO, TransientPreference classes.
    - remove IsolateMessage types (e.g. SocketChannel...).
    - remove other SE-specific signature dependencies.
  - JSR-121 API can be directly pre-processed for this.
- Subset to ‘what makes sense’
  - a pure subset of above.
  - decisions based on functional requirements.
  - community feedback needed.

# Isolation on CLDC / MIDP

- Very size constrained
  - Isolation implementation must have ~0 overhead.
- All MIDlets within a MIDlet Suite share an execution environment.
- MIDlet Suites are standalone:
  - No (perceived) need for inter-suite comms.
  - No need for Isolation creation from MIDlets.
- Isolate API potentially useful only for application manager:
  - managing multiple MIDlet suites.
  - managing MIDlets+native applications.

# Isolation on CDC

- Larger, more complex systems
  - More capable Isolate API appropriate.
  - Has J2SE 1.3-era IO types: File, Socket.
  - Intelligent descriptor passing would add complexity to implementation.
- Many potential application models:
  - Personal Basis Profile adds:
    - Generic Xlet model (interface with init, start, pause, destroy methods).
    - Inter-Xlet Communication (IXC) – RMI-based.
  - Personal Profile adds:
    - Applet model.
    - Separate AWT component trees could be needed for browsers.

# Feature Requirements

| Function                          | CLDC/MIDP   | CDC/FP   |
|-----------------------------------|---|--|
| Isolate (start/stop)              | <b>Yes</b> – core functionality.  | <b>Yes</b> – isolate halt essential.   |
| Isolate (IO bindings)             | <b>No</b> – MIDlets won't use this.   | <b>No</b> – Xlets won't use this.  |
| Link/ IsolateMessage (core types) | <b>Probably</b> – MIDlet will probably need runtime comms with middleware. Inter-MIDlet comms not needed yet. | <b>Yes</b> - Xlet-Middleware comms essential. Inter-Xlet comms could be based on this.   |
| IsolateMessage (Descriptor-based) | <b>No</b> – Most types not present in CLDC. GCF used instead. Sharing between isolates considered unlikely.   | <b>No</b> - These types unlikely to be passed between Xlets and Middleware. RMI-type comms between Xlets does not require these types. |
| IsolateMessage (Visitor support)  | <b>No</b> – probably not used.  | <b>Maybe</b> – only useful if message types unknown, which is unlikely in typical use.   |
| EventLink/<br>EventDispatcher     | <b>Maybe</b> - CLDC currently favours use of state methods. Event system may be too heavyweight.              | <b>Probably</b> - asynchronous notification helps middleware, EventDispatcher is only for convenience.                                 |



# Summary

- Clear and present need in J2ME
  - Consumer Electronics level of stability.
  - Digital TV market has existing requirements.
  - Application Management expected to be main user.
- J2SE API can be adapted for J2ME
  - Smaller footprint APIs essential.
  - Potential for fast-track JSR.
  - Reference Implementation on J2ME VM required.
- API Subsets for CDC and CLDC
  - **Feedback please!**

