Integrating Mobile Devices and Emulators onto the Eclipse IDE with TmL

Christian Kurzke
TmL Team Lead

Mauren Brenner

Fábio Fantato

Daniel Franco
Tools for mobile Linux (TmL)

• History
  - December 2006: creation review
  - March 2007: short talk at EclipseCon 2007
  - Source code available in CVS repository

• Objective
  - Provide support for development of mobile applications on the Eclipse IDE

• Related projects
  - CDT, all other DSDP subprojects (MTJ, NAB etc.)
Project Scope

• Current scope
  - Device Framework supporting devices and emulators
  - VNC Viewer for display visualization

• Future scope
  - Simulated end-to-end environment
TmL and TM *(future)*

- TM: Target Management, another DSDP subproject
  - RSE: Remote Systems Explorer
  - Adapters
TmL Components

• Device Framework
  - Integration of devices and emulators to Eclipse IDE
  - Supports real, physical devices
  - Supports device emulators

• VNC Viewer
  - Graphic display visualization capabilities
  - SWT component
  - VNC client (VNC protocol, also known as RFB)
Device Framework

- Generic framework
  - Extension point mechanism
  - Generic classes and interfaces

- Target users
  - Device vendors
  - Emulator developers
  - SDK developers, where SDKs often include emulators

- Sample implementations

http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.tml.device/?root=DSDP_Project
Definition of “Device”

• A device means a piece of hardware with a certain purpose or functionality, e.g. a mobile phone, a set-top box, a single board computer etc

• Abstract description of a device or emulator
  - Real, physical device
  - Emulator

• Device plug-in
  - Device or emulator properties
  - Provides a wizard to create instances
  - Contains components used by all instances
    - Scripts
    - Binaries
Definition of “Service”

- A service is the implementation of functionalities or capabilities offered by each device

- Framework can provide common services
- Device services: start, stop, reset, halt, flash
- Emulator services: start, stop, restart, deploy

- Service plug-in
  - Contains service-specific components
    - Scripts
    - Wizards
Device Framework Architecture
Extension Points

org.eclipse.tml.device

org.eclipse.tml.service

org.eclipse.tml.serviceDefinition

org.eclipse.tml.state
Device Extension Point

**org.eclipse.tml.device**

<table>
<thead>
<tr>
<th>Extension Element Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the properties of &quot;device&quot;. Required fields are denoted by &quot;*&quot;.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id*</td>
<td>org.eclipse.tml.device.qemureact.qemureactDevice</td>
</tr>
<tr>
<td>name*</td>
<td>QEmuReact Mobile Device</td>
</tr>
<tr>
<td>description</td>
<td>Mobile Emulator for QEMUARM</td>
</tr>
<tr>
<td>version</td>
<td>0.2.0</td>
</tr>
<tr>
<td>provider</td>
<td>Eclipse.org</td>
</tr>
<tr>
<td>copyright</td>
<td>Motorola Inc. 2007</td>
</tr>
<tr>
<td>handler</td>
<td>org.eclipse.tml.device.qemureact.handler.QEmuF</td>
</tr>
<tr>
<td>icon</td>
<td>icons/full/obj16/qemureact.gif</td>
</tr>
</tbody>
</table>
Service Extension Point

org.eclipse.tml.service

id*: org.eclipse.tml.service.stop.stopService
name*: Stop Service
description: Service to Stop a mobile device
version: 0.2.0
provider: Eclipse.org
copyright: Motorola Inc. 2007
handler: org.eclipse.tml.service.stop.handler.StopService
icon: icons/full/obj16/stop.gif
Device and Service Plug-ins
Service Definition Extension Point

org.eclipse.tml.serviceDefinition
State Extension Point

- Example states
  - Off
  - Idle
  - Connected

- State extensions
  - Each developer can create one’s own set of states

example services

DemoDevice
StartService
StopService
ConnectService
DisconnectService

org.eclipse.tml.state
State Transitions

There is a set of state transitions for each device and service
An Example Device Plug-in
Creating Instances of a Device Plug-in

Instance creation wizards

Instance properties page

Instance view
Device Properties and Instance Properties

Device Properties

- Editable using Preferences page
- A set of properties per device plug-in
- Rules according to XML definition

Properties.config

<instance>
  <host>127.0.0.1</host>
  <port>5900</port>
</instance>

<parameters>
  <param id="1" name="L" value="."
  <param id="1" name="-m" value="256"
  <param id="1" name="-vnc" host="y" />
</parameters>

<location read-only="y">
  <path>/emuv/bin</path>
  <bin>run.bat</bin>
</location>
<emulator>

Instance properties

- Rules according to XML definition
- Editable using Properties page
- Saved as metadata in workspace

Properties.config

<host>127.0.0.1</host>
<port>5900</port>
VNC Viewer

http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.fml.vnc/?root=DSDP_Project
VNC Viewer Architecture

• SWT component
  - Graphical display viewer
  - Can be used standalone or within an Eclipse view

• VNC client
  - VNC protocol (or RFB, Remote Frame Buffer)
  - “Feeds” the SWT component
VNC Viewer: Ideas for the Future

- Configurable skins
- Multiple displays
- Keyboard maps
- Extensible protocol
Demo

http://wiki.eclipse.org/DSDP/TML/How_to_configure_TmL_demo
Simulated End-to-End Environment (*future*)

- Complete network infrastructure
- Connection among mobile devices as well as back-end servers
- Network nodes are devices and emulators implemented by means of the Device Framework
- Suitable environment to test mobile enterprise applications
- A potential testbed for innovative applications
Where Could TmL Be Useful?
TmL Project Resources

Project web site:
http://www.eclipse.org/dsdp/tml
Project wiki:
http://wiki.eclipse.org/DSDP/TML
Users newsgroup:
eclipse.dsdp.tml
Developer mailing list:
dsdp-tml-dev@eclipse.org
CVS repositories:
http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.tml.device/?root=DSDP_Project
http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.tml.vnc/?root=DSDP_Project
TmL demo:
http://wiki.eclipse.org/DSDP/TML/How_to_configure_TmL_demo

Suggestions and contributions are welcome! 😊
Questions & Answers