Developing Occasionally Connected Applications Using Oracle 9iLite

Adding Mobility to Enterprise Application

Shawn Casey, Intel Corporation
Sharad Singhani, Oracle Corporation
Agenda

- Trends Towards Mobile Computing
- Occasionally Connected Computing
- Converting DB Apps to OCC Apps
- Conclusion
**2006 Laptop WiFi Environment**

**Projected 50M Intel® Centrino™ Mobile Technology based PCs, 2003-06**

- **Mobility Vectors**
- **Performance**
- **Battery Life**
- **Form Factor**
- **Wireless**

**Pentium® II Processor**

- **1998-1999**
- **200 million Subscribers**

**A first: Laptop sales beat desktops**

Thursday, July 3, 2003 Posted: 9:56 AM EDT (1356 GMT)

NEW YORK (AP) -- Notebook computer sales surpassed sales of desktop computers for the first time in May thanks to an increasing desire for mobility, according to a survey by market research firm NPD Group.

Notebooks accounted for more than 54 percent of the nearly $500 million in retail computer sales in May, the Port Washington, New York, concern said Wednesday. That compares with January 2000, when laptops represented less than 25 percent of sales volume.

**Intel engaged with 50 worldwide Public WLAN service providers; Plan to enable 15,000 Hotspots by year end '03**

© 2003 Intel Corporation. All rights reserved.
© 2003 Oracle Corporation. All rights reserved.
*Third party brands and names are the properties of their respective owners*
Occasionally Connected Computing

Benefits

Increased User Productivity
Transparent User Experience
Consistent User Experience
Increased efficiency and utilization

Features

Access Your Data Anytime
Dynamically manage intermittent connections
Common technologies/mechanisms
Run All Day Unwired

Vector

Offline Data Management
Seamless Connectivity
Multiple Platform Support
Power and Performance Management

OCC enables applications to overcome mobility challenges

© 2003 Intel Corporation. All rights reserved.
© 2003 Oracle Corporation. All rights reserved.
*Third party brands and names are the properties of their respective owners
“Consumers want more from their mobile devices and applications, especially in the future. Enhancements need to provide greater utility in order to lure consumers to the buying table.”

*Mobilizing the Consumer, Randy Giusto, IDC. – 8/02*
Mobility Inflection Point

Wireless Roaming Stressing Traditional Applications
Resets, Reboots, Service Disruptions, Refreshes, etc.

+ Volume Mobile Platforms
Laptops, Handhelds, Phones

= Opportunity to “Mobilize” Applications
  • Develop an Occasionally Connected Computing SW Arch.
  • Develop to Standards for the Robustness IT Requires
Occasionally Connected Computing Architecture

- Mobile Client
- Mobile Server
- HTTP Listener
- Oracle Database 10g
- Client
- AS
- DB

© 2003 Intel Corporation. All rights reserved.
© 2003 Oracle Corporation. All rights reserved.
*Third party brands and names are the properties of their respective owners
Secret of Building OCC Application

Design the Application by Factoring in Mobile Device and Network Characteristics!

- Device CPU, Disk, Display, Peripherals
- Mobile Database Size
- Data Synchronization Load
- Network Bandwidth and Latency
Five Steps of Building an OCC Application

1. Create Application Code
2. Package Application
3. Publish & Provision Application
4. Deploy Application and Mobile Database
5. Run OCC Application
Step 1: Create Application cont’d

- Application (API)
- Mobile Infrastructure (API)
- Database (API)
- Synchronization (API)
Step 1: Application API

- Programming Languages options
  - C++, Java, .Net*, Java Servlet/JSP/HTML

- Database API options
  - ODBC, JDBC, ADO(CE), ADO.Net

- Synchronization API options
  - Mobile Sync C++
  - Mobile Sync Java
  - Mobile Sync COM
  - Mobile Sync .Net

*Third party brands and names are the properties of their respective owners
Step 1: Application API cont’d

- We will convert an existing online Web Application into a Occasionally Connected Application
- Step 1: Install Oracle JDeveloper
- Step 2: Load an existing Web Application using Java Servlet 2.2, JSP 1.1, Oracle BC4J
- Step 3: Modify the Java Application Code
  1. User Profile
  2. JDBC Connection
Step 1: Application API

**User Profile**

- This object can be obtained from the `oracle.lite.web.servlet.OraHttpServletRequest`.
- *Servlets* can typecast the request parameter to the `OraHttpServletRequest` object and call the `getUserProfile` method to obtain the user profile object

```java
public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // Retrieve the User Profile,
    java.security.Principal user = request.getUserPrincipal();
```
Step 1: Application API

cont’d

- **JDBC Connection**
  - JDBC connection can be retrieved from User Profile

```java
public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // Retrieve the JDBC Connection,
    java.security.Principal user = request.getUserPrincipal();
    Connection conn = ((OraUserProfile) user).getConnection();
}
```
Adding Mobility the Easy Way

- Application Designer is fully aware of Mobility...
- ...but Application Developer only applies minor modifications to Application Code and Database Schema
- **Mobility** aspects are added after programming is complete using **declarative** techniques ➔ Step 2: Packaging
Step 2: Package Application cont’d

APPLICATION specific Information:

1. Name
2. Application Files
3. Database Name
4. Snapshot Definition
5. Sequence Definition
6. Create JAR/WAR
Step 2: Application Name

cont’d

<table>
<thead>
<tr>
<th>Function</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Name</td>
<td>Sample3</td>
</tr>
<tr>
<td>Virtual Path</td>
<td>/Sample3</td>
</tr>
<tr>
<td>Description</td>
<td>Sample 3 - The Recording Tracker demonstration</td>
</tr>
<tr>
<td>Application Classpath</td>
<td>/sample3/servlets</td>
</tr>
<tr>
<td>Default Page</td>
<td>sample3.html</td>
</tr>
<tr>
<td>Local Application Directory</td>
<td></td>
</tr>
<tr>
<td>Icon</td>
<td>sample3.gif</td>
</tr>
</tbody>
</table>
Step 2: Application Files cont’d

Use the Files panel to specify which files you want to include in this mobile application.

File Name:
- oracle_oem-mobile-serversamplesample3src/strDeleteRecord.java
- oracle_oem-mobile-serversamplesample3src/strDisplayMaster.java
- oracle_oem-mobile-serversamplesample3src/strDisplayRecord.java
- oracle_oem-mobile-serversamplesample3src/strListSearchResult.java
- oracle_oem-mobile-serversamplesample3src/strProgramList.java
- oracle_oem-mobile-serversamplesample3src/strSampleResource.java
- oracle_oem-mobile-serversamplesample3src/simplelist.java
- oracle_oem-mobile-serversamplesample3sql/op.sql
- oracle_oem-mobile-serversamplesample3sql/insert.sql
- oracle_oem-mobile-serversamplesample3sql/sample3.sql
- oracle_oem-mobile-serversamplesample3sql/table.sql
- oracle_oem-mobile-serversamplesample3sql/sample3.war

Controls:
- New
- Delete
- Load
- Compile JSP
- Import WAR file

Help
OK
Cancel
Step 2: Application
Database cont’d

Use the Database panel to specify how the mobile application interacts with the Oracle database.

Server side
- Database Username: master
- Number of Connections: 10
- Share Connections: [ ]

Client side
- Database DSN Name: Sample:
Step 2: Snapshot Definition cont’d

Use the Snapshots panel to list the database tables for this mobile application. You can enter the definition to create the tables on the Mobile Server and specify the snapshots definitions to create database snapshots on the Mobile Clients.
Step 2: Sequence Definition cont’d

Use the Sequences panel to list the database sequences for this Web-to-Go application. You can enter the definition to create the sequences on the Web-to-go Server and specify how Web-to-Go should create the sequences on the Web-to-Go clients.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Start Value</th>
<th>Increment</th>
<th>Window Size</th>
<th>Thread</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUDIO03 WINDOW</td>
<td>1002</td>
<td>2</td>
<td>500</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Help | OK | Cancel
Step 2: Generate JAR cont’d

You have completed the application definition. From this panel, you can create the application related files, or publish this application to the Mobile server.

Please select one of the following actions:

- Create files
  - Package application into a JAR file
- Generate SQL scripts for database objects
- Publish the current application
- Restart wizard

Back OK Exit
Step 3: Publish & Provision cont’d

• Step 3.1 Publish
  – Publish JAR/WAR file on the Server

• Step 3.2 Provision
  – Create User
  – Define ACL
    – → Bind user to application & data
  – Define User Profile
    – → Specify user specific data subset (next slide)
Step 3: Define User Profile

• Temporary Snapshot Definition (from Packaging Wizard)
  
  ```sql
  select * from scott.emp where code = :code
  ```

  DBA defines a value for each user using a GUI Tool

Final Snapshot Definition
  
  ```sql
  select * from scott.emp where code = 1111
  ```
Step 4: Deploy Application and Mobile Database

- Step 4.1 Install Mobile Client on the Device
  - Installs and register Mobile Client libraries on the Intel® Centrino™ Mobile Technology Laptop

- Step 4.2 Synchronize Application and Data
  - Deploys and install Web Application and mobile Database on Intel Centrino Mobile Technology Laptop
Step 5: Run OCC Application

- **Step 5.1 Run OCC Application from Workspace**
  - The Web application updates the mobile database using JDBC

- **Step 5.2 Synchronize**
  - Today: Manually by the mobile user
  - Future: Automatically when roaming into HotSpot
Overcoming Application Limitations

- **Offline Data Management**
  - Data caching and Synchronization
  - Security Protection
  - Manageability

- **Seamless Application Connectivity**
  - Detect changes in network state & take action

- **Multiple Platform Support**
  - Identification of Platform Attributes

- **Power Reduction and Performance Management**
OCC Technical Development Kit

- Release mid-Q4
- Reference Architecture Guides
  - Database, Portal, Distributed Document architectures
- Tools
  - Communication management/simulation
  - Power management/simulation
- APIs
  - ITI – Intel Transport Interface
    - Information and notification about network status
    - QoS**, Bandwidth**, Reliable messaging**
  - Power APIs**

**Will be included in OCC TDK 2.0
Conclusion

- Mobility is happening now
  - Large Mobile Computing Base Today
  - Ever Increasing Connectivity
- Oracle provides solution to convert existing DB applications to an OCC apps using 9iLite
- Join Intel and Oracle in pioneering user friendly mobile applications

Join us at the Intel Sponsored Wireless Pavilion for a demonstration of our ISV’s mobile solutions
Reminder – please complete the OracleWorld online session survey

Thank you.