Hello OpenROAD!
An Introduction to Rapid Application Development for Ingres

David Tondreau, Architect
Agenda

- Brief Recap from Last Week’s Webinar
- An Introduction to OpenROAD
- Overview of OpenROAD Workbench 2006
- Writing A Simple Database Application
- Developing Applications with Templates
- Debugging, Building and Deploying Applications
- Rich Enterprise Web and Mobile Applications
Getting Started with .NET

- Create a C# project
- Add the Ingres .NET Data Provider to the project
- Add an entry field to the form
- Add the code to:
  - Connect to Ingres
  - Set up and run a query
  - Put the data on the form
  - Disconnect from Ingres
`.NET Code Snippet`

```csharp
using Ingres.Client;
...
string location =
    "Host=localhost;" +
    "User Id=ingres;PWD=password;" +
    "Database=testdb";
IngresConnection dbConn = new IngresConnection(location);
dbConn.Open();
string sql = "select text from greeting";
IngresCommand ingCmd = new IngresCommand(sql, dbConn);
IngresDataReader dataReader = ingCmd.ExecuteReader();
dataReader.Read();
text.Text = dataReader.GetString(0);
dataReader.Close();
dbConn.Close();
```
Hello OpenROAD!
Getting Started with Java

• Create a Java project

• Add the Ingres JDBC Driver to the project

• Add the code to:
  • Connect to Ingres
  • Set up and run a query
  • Write out the value on the console
  • Disconnect from Ingres
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import com.ingres.jdbc.*;

IngresDataSource dataSource = new IngresDataSource();
Connection dbConn = null;
PreparedStatement statement = null;
ResultSet results = null;
String text = "";

dataSource.setServerName("localhost");
dataSource.setDatabaseName("testdb");
dataSource.setPortName("II7");
dataSource.setUser("ingres");
dataSource.setPassword("password");

try {
    dbConn = dataSource.getConnection();
    statement = dbConn.prepareStatement("SELECT text FROM greeting");
    results = statement.executeQuery();
    results.next();
    text = (String) results.getObject(1);
    System.out.println(text);
    dbConn.close();
} catch (SQLException e) {
    e.printStackTrace();
}
```java
import java.sql.Connection;

public final class Application {

/**
 * @param args
 */

public static void main(String[] args) {
    IngresDataSource dataSource = new IngresDataSource();
    Connection dbConn = null;
    PreparedStatement statement = null;
    ResultSet results = null;
    String text = "";

    dataSource.setServerName("localhost");
    dataSource.setDatabaseName("testdb");
    dataSource.setPortName("II7");
    dataSource.setUser("ingres");
    dataSource.setPassword("password");

    try {
        dbConn = dataSource.getConnection();
        statement = dbConn.prepareStatement("SELECT text FROM");
        results = statement.executeQuery();
        results.next();
    }
```
Getting Started with OpenROAD

• Connect to the Database
• Create an Application
• Create a Frame
• Create an Entry Field
• Write the Code
OpenROAD Code Snippet

```sql
SELECT :text = text FROM greeting;
COMMIT;
```
initialize() =
declare
{
  select :text = text from greeting;
  commit;
}
# Reduce Complexity, Increase Productivity

## Lines of Code Per Function Point by Programming Language

<table>
<thead>
<tr>
<th>Language</th>
<th>SLOC per Function Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cobol default</td>
<td>107</td>
</tr>
<tr>
<td>C++ default</td>
<td>53</td>
</tr>
<tr>
<td>Java 2 default</td>
<td>46</td>
</tr>
<tr>
<td>Visual Basic 6</td>
<td>24</td>
</tr>
<tr>
<td>Delphi 5</td>
<td>18</td>
</tr>
<tr>
<td>HTML 4</td>
<td>14</td>
</tr>
<tr>
<td>SQL default</td>
<td>13</td>
</tr>
</tbody>
</table>

William H. Roetzheim, CEO, Cost Xpert Group

http://www.ddj.com/dept/architect/184414658
What is OpenROAD?

- A powerful Rapid Application Development (RAD) solution for building data-centric software applications
- Enables quick construction and deployment of client/server, n-tiered, web & mobile applications
  - Scalable server-side business logic management
  - Browser deployable, rich thin client interfaces
- Targeted at businesses with rapidly evolving application development needs
OpenROAD Components

- OpenROAD Development
- OpenROAD Runtimes (VMs)
- OpenROAD Server
- Database Connectivity
OpenROAD Development

- **Graphical interactive development environment**
  - Interactive and template based form design
  - Wizards and component editors that simplify development

- **Object-oriented 4GL programming language**
  - Supports inheritance, polymorphism, encapsulation, etc.
  - SQL is part of the language dialect
  - Tight bindings (form to class, class to database)

- **All the tools needed to develop, test and deploy desktop, web, mobile and server applications**
OpenROAD Desktop Client Runtime

- Traditional, desktop runtime
  - Platform independent Virtual Machine (VM)
  - Write once, run anywhere applications
  - Includes Ingres networking

- Supports multiple application architectures
  - Single-tier (host based)
  - 2-tier (fat client)
  - 3-tier (thin client)
  - n-tier (distributed)
Sample Partner Applications
OpenROAD Browser Client (eClient)

- Complete packaging of the OpenROAD Runtime as a plug-in to a web browser
  - Think “Acrobat Player” or “Flash”

- User applications are also browser plug-ins
  - Delivered to clients running only a browser

- Build rich enterprise applications
  - With the features of desktop applications
  - Without the complexity of other REA technologies
OpenROAD Mobile Client (mClient)

• Complete port of the OpenROAD Runtime to handheld devices and mobile phones
  • Supports Windows Mobile 5+ and Windows CE 4.2+

• Features
  • Runtime installer and application delivery (< 7 Mb)
  • Exploits native Windows Mobile features
  • Server connectivity via HTTP
  • Binary application compatibility with other runtimes
Mobile Client Example
OpenROAD Server

- Supports deployment of server-side 4GL business logic
- **Makes OpenROAD 4GL business logic available to:**
  - OpenROAD desktop, browser and mobile clients
  - J2EE (Java applets, servlets, & beans)
  - .NET (C#, ASP.Net, VB.Net)
  - Browsers (HTML, ASP, JSP)
  - Other clients (Visual Basic, C++, etc.)
- Supports publishing of OpenROAD 4GL object classes as Web services
- Visual server farm administration utility
The OpenROAD 2006 4GL Server

• **Business Logic**
  • Stateful/Stateless Applications
  • Call Level Security
  • Business Process Management

• **Application Scalability**
  • Connection Pooling
  • Load Balancing
  • Fail-Over
Platform Independent, Client Agnostic

- HTML
- WAP
- 4GL Client
- .NET
- Java
- XMLHttpRequest
- SQL99
- XML
- 4GL

Web Services

Multi-threaded

4GL SQL99 XML

Server Pooler

HTTP Interface XML Interface

.NET Interface COM Interface Java Interface
OpenROAD Connectivity

• Seamless access to a wide variety of relational and mainframe data stores (Client or Server)
  • Ingres/Net
  • Enterprise Access (Unix, Windows RDBMSs)
  • EDBC (Mainframe databases)

• Thin client applications access the OpenROAD Server via HTTP or DCOM
The Ingres “Stack”

**Clients**
- ABF Client
- Embedded C Client
- OpenROAD Client
- Windows/Linux/Unix Fat Clients
- C++, VB... Java, C#

**Desktop**
- Ingres/Net
- HTTP
- DCOM
- Ingres ODBC

**Web**
- Browser
- HTML, VBScript...
- HTTP
- DCOM
- HTTP

**Mobile**
- Mobile Phone
- WML
- 4GL

**Middleware & Application Servers**
- Apache Tomcat
  - JSP, PHP, Perl, Python...
- JBOSS
  - JSP, Java Servlets/Beans
- WebLogic
  - JSP, Java Servlets/Beans

**OpenROAD Server Pooler**
- Servlets/Beans
- 4GL

**Enterprise Data Stores**
- EDBC
- IMS
- VSAM
- DB2
- IDMS
- Datacom
- Ingres DBMS
- Oracle
- SQL Server
- DB2 UDB
- RDB

**Integration Points**
- Ingres/Net
- HTTP
- DCOM
- 4GL
More Questions? See You In The Forums!