



IBM Software Group

Building Modern Web 2.0 User Interfaces for CICS and IMS with EGL

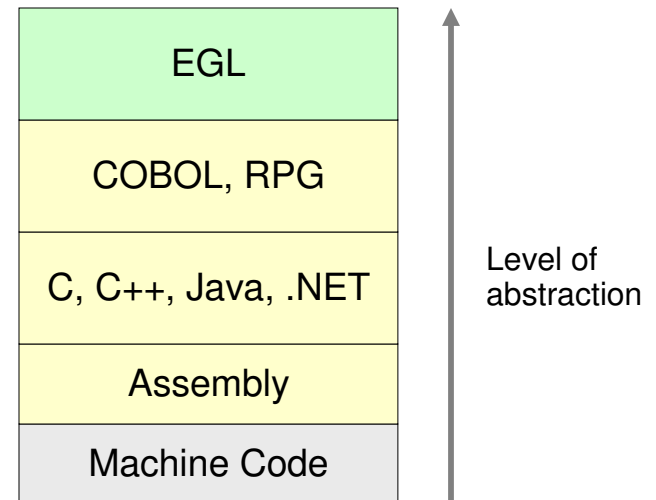
Wednesday, September 2, 2009

Will Smythe
Rational Business Developer / EGL Product Manager



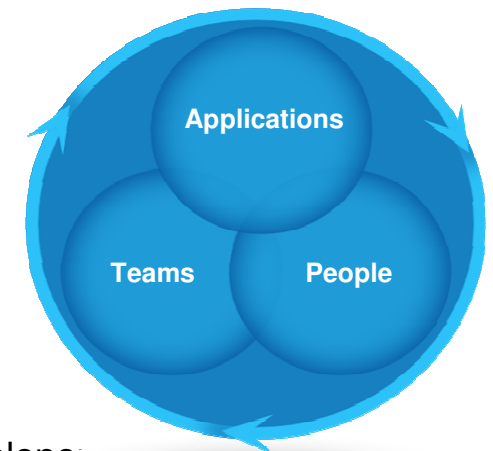
EGL – IBM’s Newest Business Language

- EGL is a higher-level programming language designed for developing business applications
 - ▶ The goal is to shield developers from complexities typically associated with enterprise application development
- A true cross-platform, cross-tier language targeted at all types of developers
 - ▶ Traditional, procedural, object-oriented
- Provides flexible deployment options
 - ▶ Compiles to Java, COBOL, or JavaScript
- Web 2.0 and SOA built-in
- EGL is also an excellent target language for migration of existing, traditional applications.



Benefits of EGL

- **Flexibility:** Affords maximum platform independence and architecture support
- **Rich user interfaces:** Enables business developers (COBOL, RPG, VB, 4GL programmers) to create extremely rich, Web 2.0 user interfaces, along with Service-Oriented, multi-platform applications with a very short learning curve
- **Integration:** Enables developers to easily connect to, wrapper and extend trusted, valuable assets
- **Productivity:** Encourages developers focus on business problems, not technology problems
- **Adaptability:** Delivers a modern language that adapts more easily to changing technologies
- **Migration:** EGL is ideally suited for migration of COBOL, RPG, apps and development

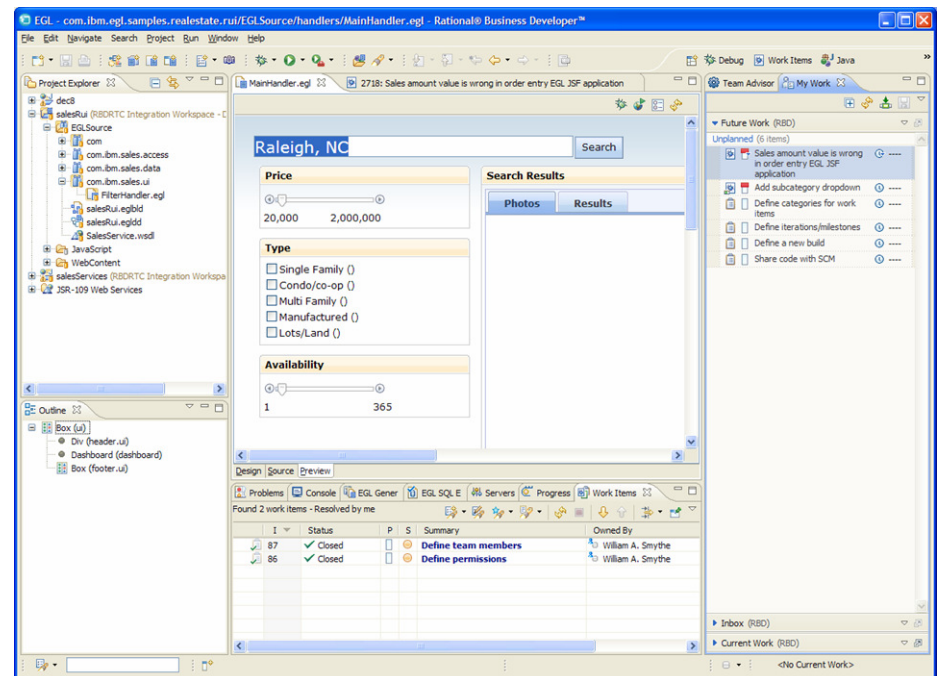


EGL Tools from IBM Rational

- EGL tooling is available in the following Rational offerings:
 - ▶ Rational Business Developer
 - ▶ Rational Developer for System z with EGL
 - IDE for System z development; includes COBOL, PL/I editing tools, and more
 - ▶ Rational Developer for i for SOA Construction
 - IDE for IBM i development; includes RPG and COBOL editing tools, and more

■ Features

- ▶ Visual and source editors
- ▶ Code completion, templates, and snippets
- ▶ Service generation for database tables
- ▶ SQL visualization and editing
- ▶ References and declarations
- ▶ Open on selection
- ▶ Refactoring
- ▶ Cheat sheets and dynamic help



Rational Developer for System z with EGL

Build traditional and composite applications with Rational Developer for System z with EGL

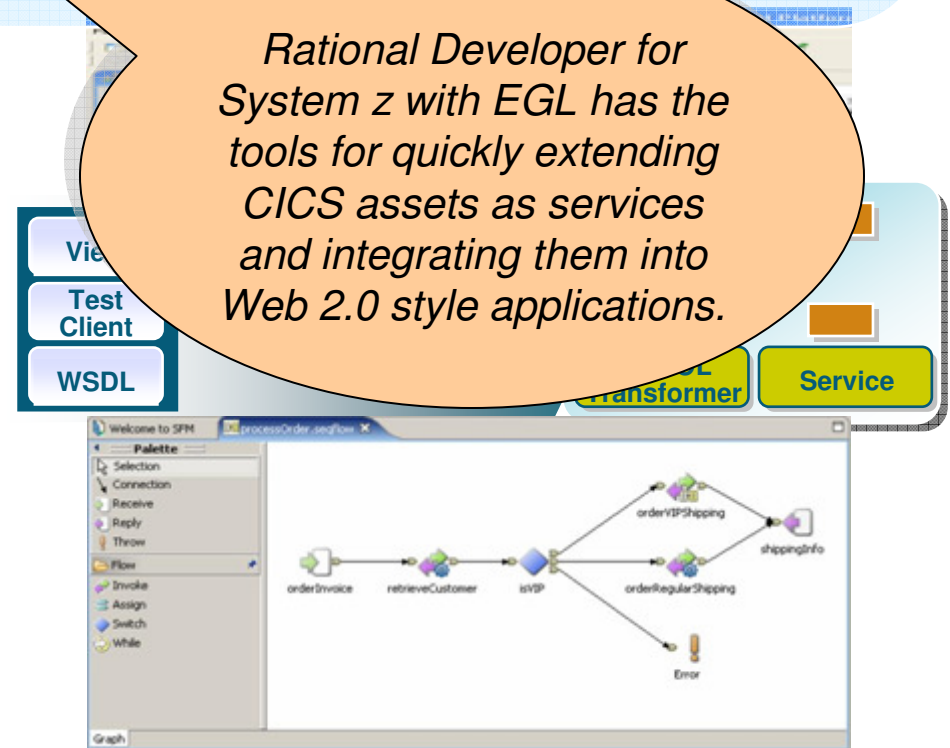
IBM Rational Developer for System (RDz) provides powerful eclipse-based tools for building COBOL, PL/I, C/C++, EGL, HLASM CICS, IMS and batch applications. RDz is designed to optimize and simplify application development for traditional processes, and for the SOA you're building to address your needs today and in the future—through best practices, visual tools, templates, code generation and a comprehensive development environment.

RDz with EGL 7.5

- ▶ Leverages service flow modeler to automate development of CICS services
- ▶ Develop z/OS applications and CICS and IMS Web services more productively than ever
- ▶ Manage core and composite artifacts from z/OS
- ▶ Speed software delivery with integrated, automated end-to-end processes

- Enables traditional; and modern UI's (via RBD)
- Supports local and remote development
- Creates connectivity via web services, CICS SFM, CTG, IMS Connect
- UML to COBOL speeds Model driven development
- Delivers distributed debugging support for SOA applications via Debug Tool
- Integrates to File Manager, Fault Analyzer

Rational Developer for System z with EGL has the tools for quickly extending CICS assets as services and integrating them into Web 2.0 style applications.



IBM Rational Developer for System z



2 EGL code is generated as COBOL, Java, JavaScript, services, etc. based on target environment and deployed as native services, Web applications, hosted Web 2.0 applications, Text UI applications, etc.

Developer Workbench (RBD/RDz EGL)

1 Developers use the EGL tools to develop Web, Web 2.0, SOA, and batch applications.

Windows, Linux, Unix

System z

IBM i

Java and JavaScript generation

- WebSphere Application Server
- Apache Tomcat
- Java Runtime Environment

Java generation

- WebSphere Application Server
- USS
- z/Linux

JavaScript generation

- WebSphere Application Server

COBOL generation

- Batch
- CICS
- IMS

Java generation

- WebSphere Application Server
- Native IBM I

JavaScript generation

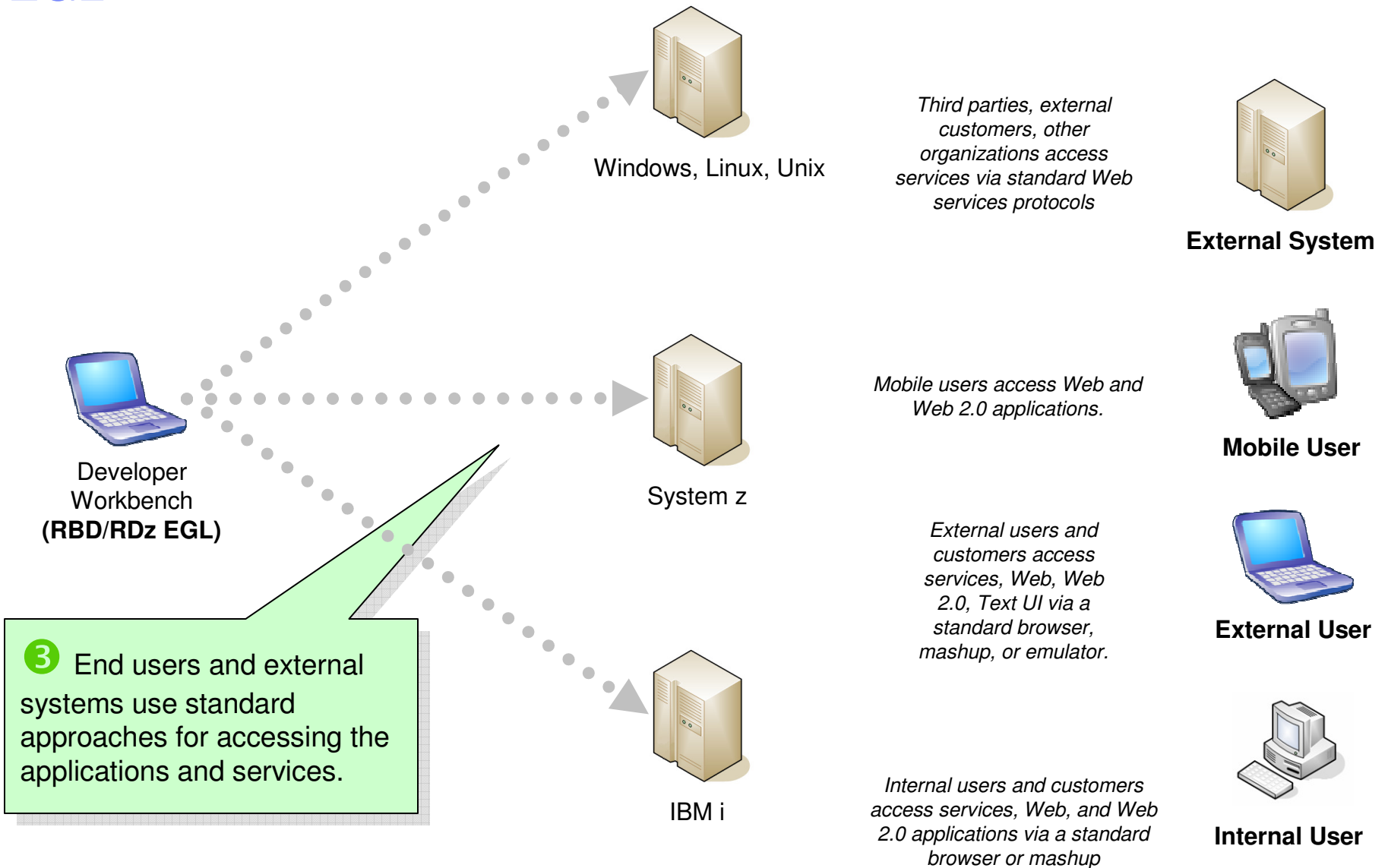
- WebSphere Application Server

COBOL generation

- IBM i



EGL



Team Development with Rational Team Concert

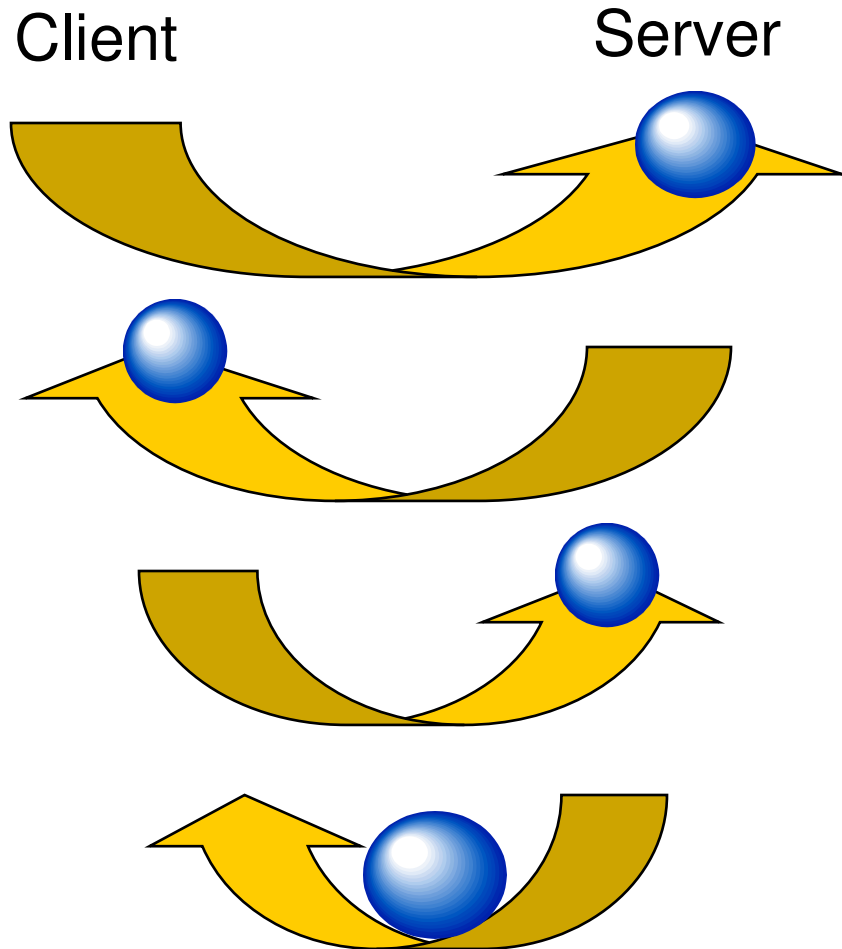
- EGL development teams can take advantages of the capabilities provided by Jazz/Rational Team Concert to manage EGL development projects.
- **Features and benefits:**
 - ▶ A single, integrated development workbench
 - ▶ Source code for all EGL project types can be managed in the RTC repository, along with other asset types (e.g. RPG, Java, JSF, HTML, CSS, COBOL)
 - ▶ New requirements and defects can be created and managed from within the tool, or from a Web-based client
 - ▶ Development process rules can be customized to enable organizations to enforce and maintain their development processes
 - ▶ Team and project status can be easily viewed via Eclipse or web UIs



Summary: Rational Business Developer and Rational Team Concert provide an integrated, collaborative environment for managing EGL development projects.



Evolution of Computing



Mainframe computing

“Dumb” green screen clients
Omnipotent big mainframe servers

Client-server computing

“Smart” Personal Computer clients
Simple file and database servers

Web (1.0) computing

Light Web Browser clients
Rich application and database servers

Web 2.0 computing

Rich Internet Application clients
Lighter application and database servers



Web Applications Today

- Web applications are no longer static, server-generated collections of pages
- Web 2.0 and Rich Internet Applications represent the next generation of Web applications
 - ▶ Provides capabilities of a desktop application, but with the manageability of a Web application
 - ▶ Enabled by technology like JavaScript and Ajax
 - ▶ Lightweight and built on open standards
- Provides a richer user experience, compared to traditional Web 1.0 applications
 - ▶ Simplified, but powerful user interface
 - ▶ Integrates relevant data from multiple sources (i.e. mashups)
 - ▶ Dynamic content
 - ▶ Client-side validation
 - ▶ Collaborative
- Provides a new vehicle for delivering richer, more **powerful business applications**
 - ▶ Web 2.0 and RIAs are not just for college kids



Mashups

Combine content from more than one source into an integrated experience

Google Map

Hotel information — separate database

Directions — come from somewhere else

Send to a phone — Additional functionality

Why?

- Rapid application development
- Reuse existing services
- Avoid reinventing the wheel
- Empowers users

Disney's Yacht & Beach Club Resorts Yacht Club + ×

★★★★☆ 430 reviews - [more info](#) »

1800 Esprit Blvd.
Lake Buena Vista, FL 32830
(407) 934-8000
[go.com](#)

[Get directions](#) - [Search nearby](#)
[Save to My Maps](#) - [Send to phone](#)



Challenges

- Why is it difficult to build Web 2.0 Rich Internet Applications today?
 - ▶ Currently domain of “tech heads”
 - ▶ Developer must learn multiple complex technologies
 - JavaScript, Ajax, JSON, SOAP
 - ▶ Compound the skill/tool silos and fragmentation
 - ▶ Most solutions are either front-end or back-end focused, but not both
 - Results in code duplication and manual efforts to keep code in sync
 - ▶ Most solutions are built on Web 1.0 style architectures
 - Not an ideal programming model for building RIAs

- RIA creation typically required lots of time, tools, and languages ... until now.

```

<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas
<head>
  <meta http-equiv="content-type" content="text/html; charset=
  <title>Google Maps API Example: Simple Geocoding</title>
  <script src="http://maps.google.com/maps?file=api&v=2.x
  <script type="text/javascript">

    var map = null;
    var geocoder = null;

    function initialize() {
      if (GBrowserIsCompatible()) {
        map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(37.4419, -122.1419), 13);
        geocoder = new GClientGeocoder();
      }
    }

    function showAddress(address) {
      if (geocoder) {
        geocoder.getLatLng(
          address,
          function(point) {
            map.setCenter(point, 13);
            var marker = new GMarker(point);
            map.addOverlay(marker);
            marker.openInfoWindowHtml(address);
          }
        );
      }
    }
  </script>
</head>

<body onload="initialize()" onunload="GUnload()">
  <form action="#" onsubmit="showAddress(this.address.value); return false
  <p>
    <input type="text" size="60" name="address" value="1600 Pennsylvania
    Washington DC" />
    <input type="submit" value="Go!" />
  </p>
  <div id="map_canvas" style="width: 500px; height: 300px"></div>
</form>
</body>
</html>

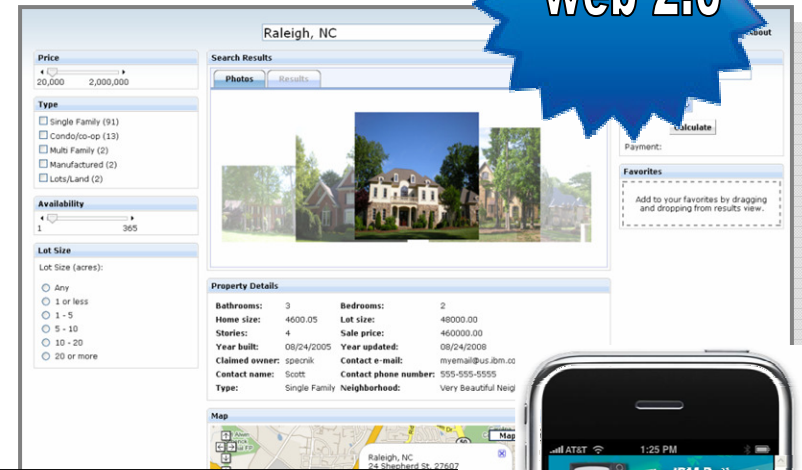
```



Rich User Interfaces with EGL



- Simplify creation of Rich Internet Applications
 - ▶ Deliver end-to-end Web 2.0 quickly in a single language
 - ▶ Build rich user interfaces to modernize existing applications
- Generates standard JavaScript and Ajax
- **Does NOT replace HTML or JavaScript**
- Easy-to-learn language
- Fully open and extensible
- Use a rich, extensible widget library
- Eclipse-based development, testing, and debugging
- Consume all types of Web services

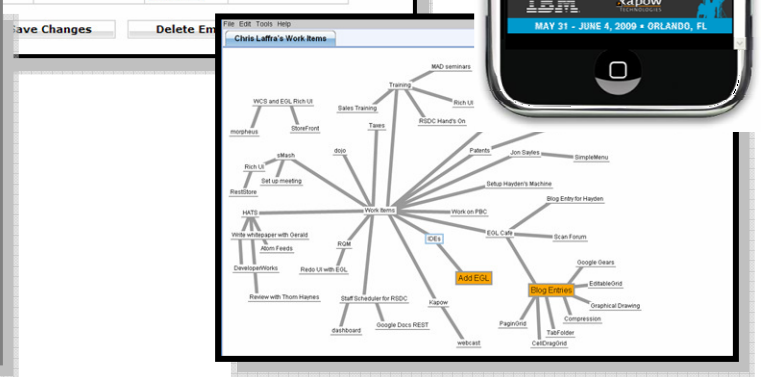


Welcome to the sMash Employee demo with EGL Rich UI front end.

The grid below contains a list of employees served up using REST services written in Groovy and hosted by sMash. You can edit each cell by clicking in it. You can save the changes, create a new employee, or delete the currently selected employee using the buttons that are provided below.

The entire UI is written in EGL Rich UI, and no JavaScript, HTML, or dojo is needed.

Username	First name	Last name	Location	Phone Number
Chris	Chris	Laffra	East	555-3345
Saul	Ben	Margolis	West	555-9932
hdjkhasjkd	jhadjkashk	dhajkdhaksj	West	555-3321
Terry	Theresa	Ramsey	West	555-3321

EGL in Action (Side-by-Side Comparison)

EGL Rich UI

```

handler MyRuiHandler type RuiHandler { initialUI = [ addressForm,
map ] }

addressField TextField { text = "1600 Pennsylvania Ave, Washington
DC", width = 250 };

goButton Button { text = "Go!", onClick ::= goButton_clicked };
addressForm Box { children = [ addressField, goButton ] };

map GoogleMap { width = "500px", height = "300px" };

function goButton_clicked (e Event in)
addresses String[] = [ addressField.text ];
map.showAddresses(addresses, addresses);
end
end
    
```

HTML and JavaScript

```

<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:com:vml">
<head>
<meta http-
<title>Goog
<script src=
"urn:schemas-microsoft-
html; charset=UTF-8"/>
</title>
api&amp;v=2.x

var map = new
var geocoder = new

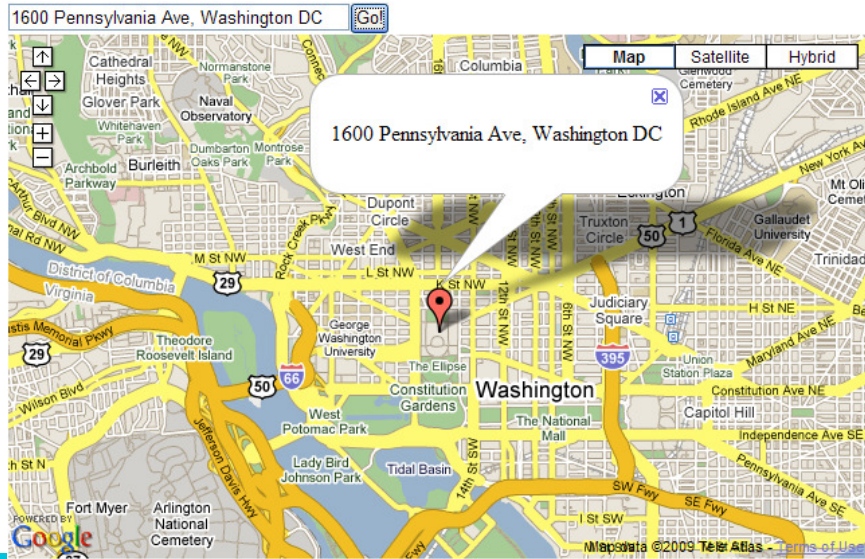
function initialize() {
if (GBrowserIsCompatible()) {
map = new GM
map.setCenter
}
}

function showAddress() {
if (geocoder)
geocoder.get
address
function
map.s
var m
map.a
marke
}
}
}
</script>
</head>

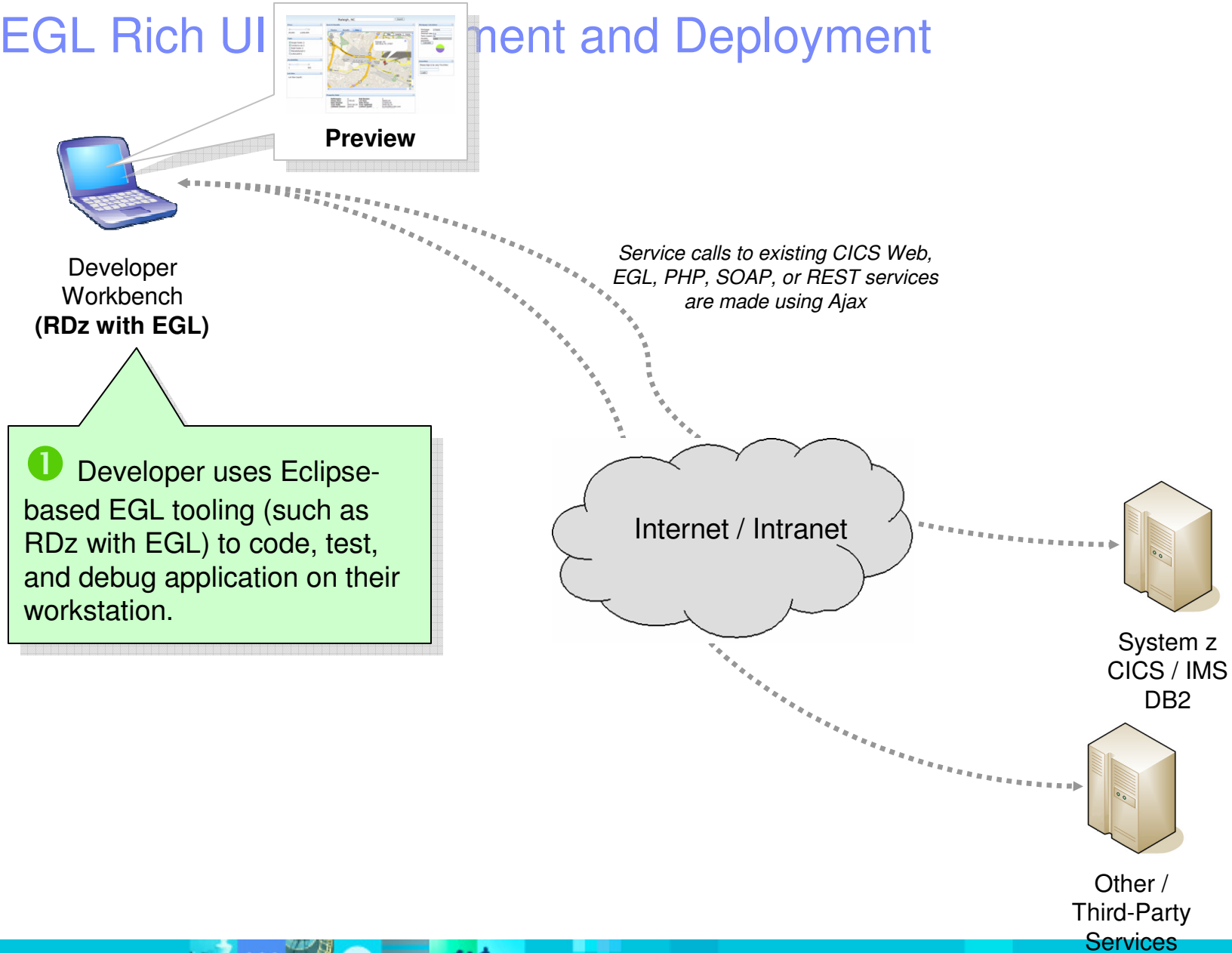
<body onload="initialize()" onunload="GUnload()">
<form action="#" onsubmit="showAddress(this.address.value); return
false">
<p>
<input type="text" size="60" name="address" value="1600 Pennsylvania
Ave, Washington DC" />
<input type="submit" value="Go!" />
</p>
<div id="map_canvas" style="width: 500px; height: 300px"></div>
</form>
</body>
</html>
    
```

All code, including UI and controller logic, is written completely in EGL.

The complexity of the Google Map APIs are hidden from the developer, so the developer can focus on the actual business requirement and not technical complexities.



EGL Rich UI Development and Deployment



EGL Rich UI Development and Deployment



Developer Workbench (RDz with EGL)

2 Developer runs Deployment wizard to create a Web application containing HTML and JavaScript (generated from EGL code). Application is deployed to an application server.

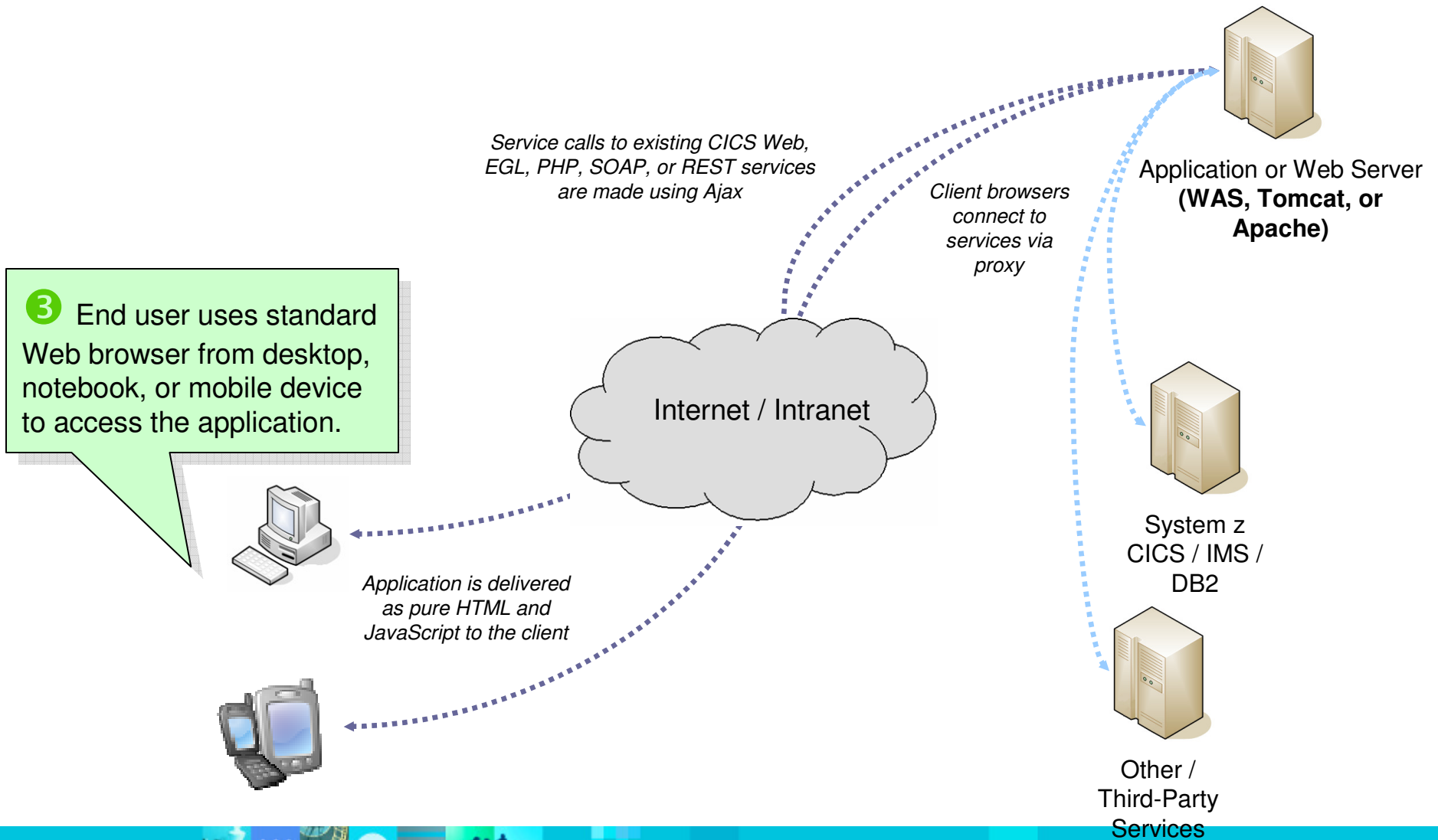
JavaScript, HTML, CSS, images, etc. packaged as a Web application and deployed on the server



Application or Web Server (WAS, Tomcat, or Apache)



EGL Rich UI Development and Deployment



UI Modernization with EGL Rich UI

- Existing applications are among the most valuable assets a company owns
 - ▶ Run the business
 - ▶ Significant investments over the years
 - ▶ Fine-tuned for dependable optimal performance
- Delivered via text based terminals or Web browsers
 - ▶ Non-intuitive, difficult to navigate user interfaces
 - ▶ Difficult to integrate in modern business processes
- Business is demanding modern, Web-based user interfaces
- 5X less expensive to reuse existing application logic than to write new applications from scratch
 - ▶ Reusing proven, time-tested business logic can result in significantly lower risk and faster time to market
- **EGL Rich UI is ideal for UI modernization!**



EGL and CICS

- Rational Developer for z with EGL provides a single workbench for building new user interfaces for CICS with EGL
- EGL Rich UI supports a number of approaches for interacting with CICS/COBOL resources on System z
 - ▶ Option 1: Call CICS Web service directly
 - ▶ Option 2: Call COBOL program via EGL-generated CICS Web service
 - ▶ Option 3: Call COBOL program via CICSECI or CICS J2C with EGL service running on a mid tier (e.g. WebSphere Application Server).



EGL and IMS

- Integrate Web services from IMS SOAP Gateway to build rich Web 2.0 user interfaces that incorporate popular widget libraries like Dojo.
- Generated COBOL code can run in:
 - ▶ Message Processing regions
 - ▶ Batch Message Processing (BMP) regions
 - ▶ IMS Fast Path (IFP) regions
- Quickly build new Text UIs (TUIs)
- Use EGL data access keywords such as get, add, replace, delete to easily work with data.
 - ▶ DL/I support with tooling to customize DL/I statements
- Full transaction support
- Invokes IMS programs remotely
 - ▶ EGL handles transaction invocation and data conversion
- Work with serial and print files
- Use a single IDE (Rational Developer for System z with EGL) to develop EGL and IMS solutions.

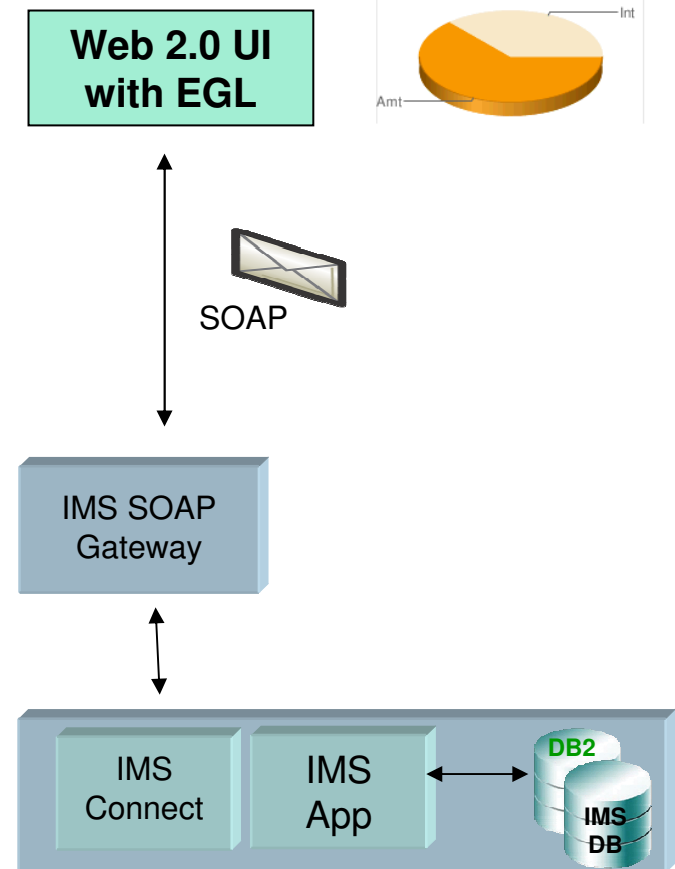
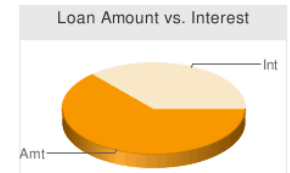
Mortgage Calculator

Amount:

Rate:

Term:

Payment: 1552.18



EGL Community Edition



Introducing an EGL for Everyone!

*Community Edition
Coming Summer 2009*

- Simple to install, **free** to use
- Work smarter and deliver modern Web 2.0 applications faster and without the pain of JavaScript™ or Ajax
- Extend application value with richer user interfaces using Dojo and flexible deployment with Java™
- **Start small, do REAL stuff**



Also, check out the **RSC Scheduler** written with EGL!
rsc2009.eqlplanner.com



Register now to get notified when it is available for download!

www.ibm.com/rational/eglcafe

EGL Community Edition Overview



- EGL Community Edition (CE) is a free, Eclipse-based tool that simplifies the development of Web 2.0 applications
 - ▶ Simplifies development of open standards-based Web user interfaces
 - Utilize the popular Dojo widget toolkit without learning HTML or JavaScript
 - ▶ Supports simplified development of Java-based Web services
 - Build services without learning Java, XML, JSON, etc.
 - Easily interact with existing programs or databases using EGL's simplified APIs
 - Supports DB2, MySQL, and Derby
 - ▶ Package and deploy as a standard JEE application
- Easy to consume, free to use
 - ▶ Small download, simple install – get up and running quickly!
- EGL takes a more open approach to Web 2.0 development
 - EGL does not try to replace HTML or JavaScript (unlike Flash/Flex and Silverlight)
 - ▶ EGL is completely extensible, allowing for unlimited extension and flexibility
 - ▶ Build rich, modern Web applications without writing a single line of HTML or JavaScript
- **Available Tuesday, September 8!** (<http://ibm.com/rational/eglcafe>)



RBD vs. EGL CE Feature Matrix

Feature / capability	RBD	EGL CE
COBOL generation	✓	
Java generation	✓	✓
JavaScript generation (Rich UI)	✓	✓
JavaServer Faces (JSF)	✓	
Tools for editing Text/Console UIs	✓	
Creation and consumption of SOAP and REST Web services	✓	✓
External program call support	✓	✓
BIRT / EGL text reporting	✓	
Install/updates via IBM Installation Manager	✓	
IDE translation (Group1 + others)	✓	
Support from IBM	✓	



Next Steps



EGL Distance Learning 2009

- Learn EGL in 2 weeks with IBM's no charge (**free**) remote course.
- Explore the emerging cross-platform rapid development environment from IBM that enables you to build Web, Web 2.0, and SOA solutions using EGL.
- Discover how quickly you can learn EGL and build state-of-the-art applications, all without getting bogged down in the technical complexities of middleware and runtime platforms.
- **Week 1:**
 - ▶ Foundations of EGL technical workshop
- **Week 2:**
 - ▶ Web 2.0 development with EGL Rich UI technical workshop



Dates

- May 4
- June 15
- July 27 (part 2 only)
- September 14
- November 30

2 hours of instruction per day

3 - 5 hours of independent hands-on per day

Register now!

<http://www.ibm.com/software/rational/cafe/docs/DOC-3004>



EGL Café

- Online community for EGL developers, partners, and clients
- Discussion forums
- Gallery of sample applications and widgets
- Presentations, videos, and articles
- Blogs by IBMers and partners
- Success stories
- Become part of the community today!

<http://ibm.com/rational/eglcfe>



Resources: Download, Learn, Presentations, Video/viewlet, Sample Code

Community: Clients, Partners, Influencers, Press, News and Events

Collaboration: Blogs, Forums, Tips and Techniques Comments, Ratings

Testimonials: Case Studies, Celebrations!



Thank You

