Nexaweb RIA Report

Features the latest Gartner Research

nexaweb Rich Internet Application technology update

In This Issue

Gartner®

The Business Value of Rich Internet Applications Today

Source: Nexaweb

Seeking to deliver the best of traditional client/server applications with the global reach of Web-based access, RIAs promise enterprise class performance and productivity along with reduced deployment and maintenance costs. RIAs also mesh well with Service-Oriented Architecture (SOA) initiatives. A growing number of Fortune 1000 companies have already adopted RIAs, or will do so in the near future. According to Gartner, "By 2010, at least 60 percent of new application development projects will include RIA technology, and at least 25 percent of those will rely primarily on RIA (0.7 probability)."[1]

Today's RIA Opportunities

Distributed, browser-based applications with requirements for high reliability and performance, such as financial trading, order management, product configuration and business-to-business self-service, are well-suited for implementation using today's RIA technology.

Organizations that seek competitive advantage or greater operational efficiency are increasingly exploiting RIA technology to re-architect traditional client/server applications, such as those written in Visual Basic or Java Swing. RIAs can offer all the rich features and performance benefits of these "thick client" alternatives, while eliminating the need to install and maintain a custom client on user desktops.

RIA technology is also of great value to companies that wish to improve the performance and user experience of traditional, HTML-based Web applications. RIAs can radically improve the responsiveness of browser-based applications because they enable processing to take place on the client, thus reducing network demands in comparison to HTML's inefficient "click-wait-refresh" model.

RIAs and SOAs

Service-oriented architectures (SOAs) can help organizations integrate disparate back-end resources in a loosely coupled manner, thus making IT more responsive to business drivers.

But in today's increasingly distributed and mobile computing environments, the networks over which users access centralized business services, as well as the

[1]Gartner Research Note Rich Internet Applications Are the Next Evolution of the Web, G00126924, M. Driver, R. Valdes, G. Phifer, May 4, 2005.

range of client types that IT must support, are both becoming more diverse and complex. To realize the full value of SOA-based services, businesses must overcome these significant hurdles on the front-end.

The role of RIAs in this context is to efficiently deliver SOA-based services to users, while at the same time reducing the cost and complexity associated with managing networks and client-side deployments.

As SOAs continue to gain widespread acceptance as the method of choice to deploy both new and existing business services, enterprises will increasingly employ RIA technology to bring those services to their end users. However, to deliver the greatest benefit in relation to SOA initiatives, an RIA solution must confront the dual challenges of deployment and network cost and complexity. As discussed below, most RIA platforms cannot successfully address these problems today.

Today's RIA Marketplace

While a number of options are currently available for developing RIAs, nearly all RIA technologies fall into one of these categories:

- DHTML/JavaScript based approaches, including AJAX
- Traditional Java-based approaches like the JFC Swing/WebStart
- .NET approaches
- Flash-based approaches

While they all share the capability to deliver a richer user interface, these RIA technologies differ in important areas.

DHTML/JavaScript-based RIA Approaches

The Web was originally designed for browsing HTML documents – not running software applications. Traditional Web applications suffer from poor performance, excessive server load and network bandwidth consumption due to their "click-wait-refresh" user interaction paradigm. They also lack a two-way, real-time communication capability because the server cannot update the application context until the client requests it.

DHTML/JavaScript can provide some of the services that HTML alone cannot provide, such as support for a richer user interface. The so-called AJAX (Asynchronous JavaScript and XML) application model can go a step further, by providing partial screen updates and asynchronous client-to-server communications via a lightweight, limited client-side engine.

However, these primarily scripting-based approaches cannot meet enterprise requirements for reliability, server-initiated updates, or offline access. In addition, maintaining scripting-based applications is difficult and expensive. Deployment in heterogeneous environments can introduce further complexity, as DHTML/JavaScript code is often platform- and/or browser-specific.

Traditional Java-based RIA Approaches

Enterprises have long relied on the Java programming language to build robust, scalable, multi-platform applications. Java offers a standards-based development model built on an enterprise class, strongly typed programming language supported by a large developer and vendor community. In this regard, Java offers a fundamental advantage over scripting-based approaches, because scripts are notoriously difficult to maintain. In addition, Java WebStart simplifies Java application deployment by automating application updates.

But while the Java programming language remains central to enterprise software development, traditional Java-based approaches to RIA development are lacking in fundamental ways. Most critically, traditional Java client applications require a specific Java Runtime Environment (JRE) version on client systems, which significantly constrains their reach. They also typically entail a large client footprint – thus creating high deployment cost and complexity similar to traditional GUIs. Finally, traditional Java-based approaches were not specifically intended to create browser-based applications and offer little in the way of built-in capabilities. This tends to make developing RIAs with Java Swing/Webstart time-consuming, expensive and difficult to integrate with normal Web applications.

.NET-based RIA Approaches

Microsoft .NET presents some of the same advantages and disadvantages as traditional Java approaches for delivering RIAs. For example, .NET offers an object-oriented programming model, and can be used to build robust applications. .NET applications likewise entail a large client footprint and require that client systems have an appropriate Common Language Runtime (CLR) installed.

.NET-based RIA approaches have a further disadvantage in that .NET is a proprietary technology that mandates enterprises to standardize on Microsoft server and client operating systems.

Flash-based RIA Approaches

Approaches that utilize Macromedia Flash enable RIA developers to create visually rich user interfaces that feature multimedia content such as transition effects and animation. Flash-based approaches also allow developers to emulate the look-and-feel of desktop applications.

One major limitation of a Flash-based RIA approach is its reliance on ActionScript, a proprietary scripting language. Developing and maintaining the numerous scripts that an enterprise class RIA would require is inherently difficult and expensive. Another major drawback is the limitations of the Flash engine, which is not an "industrial strength" virtual machine and thereby constrains the functionality, reliability and performance of Flash-based applications.

Further, most Flash-based RIAs require the most recent version of Macromedia Flash Player (currently version 7) on user desktops. Because Flash 7 is not pre-bundled with any desktop environment, a separate download and installation process is thus required.

Finally, Flash is a proprietary technology that introduces the risk of vendor lock-in and potentially restricts business flexibility in the future.

An Optimal Approach

Organizations creating RIAs today must find a balance between providing users with rich functionality and reducing the cost and effort associated with application development and maintenance. To maximize the value of their investments, businesses require a standards-based solution that makes the best use of available resources and improves productivity, yet makes it straightforward to develop, deploy and maintain applications.

Only one RIA technology meets all these requirements to provide an optimal solution today: the Nexaweb Platform.

Based on an open foundation of Java, XML and Web standards, The Nexaweb Platform combines the most comprehensive, built-in RIA development support in the industry with a platform-neutral, no-install deployment model. The Nexaweb Platform is the only RIA development platform available today that enables businesses to cost-effectively build and deploy enterprise class applications over the Internet.

See page 6 below for more information on the features and business benefits of the Nexaweb Platform.

Standards for RIAs

As Gartner points out, "Few best practices, and even fewer established standards, are in place for RIAs."[2] A lack of standardization across competing architectures is not surprising given that the RIA marketplace is new and still evolving rapidly.

Moreover, it is, entirely possible to create a robust, full-featured RIA platform based on existing open standards, as Nexaweb has done. Today's industry standards, such as Java, XML and SSL, are adequate to support the development and deployment of enterprise class RIAs. The Nexaweb Platform leverages open standards throughout its architecture, and thus offers the greatest business value with the lowest risk of adoption of any current RIA technology.

Conclusion

RIAs offer the potential to deliver the rich features and robust performance of client/server applications, combined with the global reach and deployment advantages of HTML-based Web applications.

Today's RIA technology can be a viable alternative for new applications of many types. In addition, RIAs offer business a strong alternative for improving existing HTML or client/server applications to gain the benefits of simplified deployment and/or superior performance.

However, to offer businesses strategic and cost benefits that outweigh the expense and risk of embracing new technology, an RIA platform must support applications that require no

[2]Gartner Research Note Rich Internet Applications Are the Next Evolution of the Web, G00126924, M. Driver, R. Valdes, G. Phifer, May 4, 2005.

installation, can run on diverse client systems, and provide a standards-based development model.

Only the Nexaweb Platform provides all the critical capabilities required to deliver the full value of RIAs to businesses today.

For the latest Gartner research on RIAs, see "RIAs are the Next Evolution of the Web" below.

For information on the features and business benefits of the Nexaweb Platform for RIA development and deployment, see "The Nexaweb Platform: Enterprise Class Rich Internet Applications without Compromise" beginning on page 6.

Rich Internet Applications Are the Next Evolution of the Web

Source: Gartner RAS Core Research Note G00126924, Mark Driver, Ray Valdes, Gene Phifer, 4 May 2005.

Rich Internet application technologies combine the best of traditional graphical user interface applications with the wide reach of Web-based access. These solutions will help meet customer demand for a richer customer experience.

Key Issues
How will opportunities in the market be affected by competition, technology, and evolving user requirements?

WHAT YOU NEED TO KNOW

Rich Internet application technology is emerging, but it is young. It has, however, matured to a point where vendors provide compelling toolsets for early adopters. The model will evolve considerably during the next three years, leading to mainstream adoption and critical mass among IT and commercial software projects by 2008. Type A (early technology adopters) companies should consider RIAs a source of competitive advantage today for certain application categories. More-conservative adopters should not discount the role of RIAs, but should balance the risk vs. reward of the emerging model. The major risk is for vendors, due to the highly dynamic nature of the market and the number of competitors. To ensure maximum benefit from RIA, incorporate empirically based usability assessment into the development process.

STRATEGIC PLANNING ASSUMPTION(S)

By 2010, at least 60 percent of new application development projects will include RIA technology, and at least 25 percent of those will rely primarily on RIA (0.7 probability).

ANALYSIS

The graphical user interface (GUI) is the industry standard for virtually all modern distributed IT solutions. The traditional client/server GUI is routinely represented by event-driven "fat client" applications with bitmap addressable interfaces, static and sizable installations. These applications directly build on and have a high affinity with the features and functions of the native operating system.

The advent of the Web further evolved and extended the reach of the distributed computing model to the global Internet. The Web has introduced a new client user interface (UI) as well. The Web's browser-based interface shares a common heritage with traditional GUIs, but it differs in important key areas because it typically places the bulk of business logic in the middle tier (the server). In typical implementations, the client elements of traditional browser-based applications are limited to the interface logic (for example, HTML) with small amounts of script code (such as JavaScript) for minor data validation and control logic. This creates a lightweight "thin client" that can easily be accessed from multiple platforms and devices and constrained network connections.

Modern business-to-business and business-to-consumer IT demands dictate the need for immediate access to real-time information and processes in a manner that has truly revolutionized business models across numerous markets. As developers build increasingly complex and mission-critical applications, some users require the richness of traditional GUI technologies. The increasing need for IT agility also dictates the reach of the browser-based on-demand access mode (for example, access to applications from multiple devices and platforms).

A gap exists between these two models where established technology fails to meet the demands of new business requirements. The traditional fat-client GUI, while supporting a rich user experience look and feel, lacks the ability to support consumer and on-demand corporate deployments. The traditional Web model has extraordinary reach, but lacks the rich user experience needed in many application categories ranging from business intelligence to customer relationship management to e-retail and more. A solution is required that benefits from the best of both of these worlds.

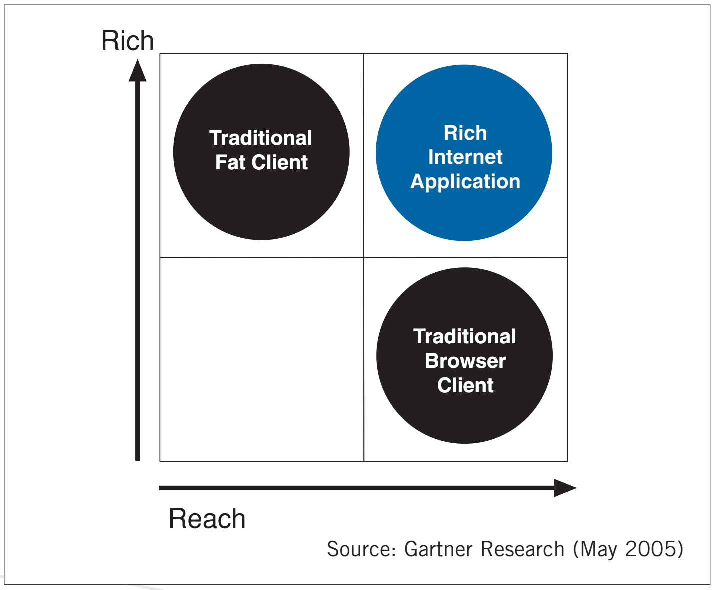
A new application model is emerging that addresses the gap between the fat but rich client/server UI model and the thin but poor Web-based UI model (see Figure 1). The rich Internet application (RIA) addresses the best elements of both, and will enable the Web to evolve beyond the page-based, document-centric metaphor commonly associated with the browser approach.

An RIA provides the look, feel and full user experience of an event-driven GUI with the deployment and management functionality of the Web.

This technology is new and changing, and we do not expect full mainstream maturity and adoption before 2008. However, a number of visionary vendors are delivering solutions that provide strong insight into the model that will – over time – significantly move the Web beyond what it is today. This new model will address usability and customer satisfaction factors for consumer Web sites. It will also address the productivity and performance of corporate line-of-business IT solutions. Furthermore, the RIA model will enhance and support the evolution of the “software as service” movement growing in numerous markets. By 2010, at least 60 percent of new application development projects will include RIA technology, and at least 25 percent of those will rely primarily on RIA (0.7 probability).

Few best practices, and even fewer established standards, are in place for RIAs. Vendors and developers have addressed the RIA challenge using a myriad of approaches, none of which has yet to create a clear base of support. The technology has, however, evolved to the point where we can see value in first-mover early adoption in specific application categories (such as e-retailing, business intelligence or gaming). Although they do not yet support long-term strategic commitments, RIA solutions are now well-suited for tactical opportunities where return on investment can be achieved within 18 months.

Figure 1. The Rich Internet Application


Rich
Traditional Fat Client
Rich Internet Application
Traditional Browser Client
Reach
Source: Gartner Research (May 2005)

Implementations of RIA technology differ significantly, but these approaches share a number of characteristics that define an RIA. These features will certainly evolve during the coming years. Applications will have to support this set of required features to qualify as an RIA. There are also optional features the applications should support where lack of support will have an impact on competitive positioning. Finally, a set of nice-to-have features are not yet high priority, but they may become more important.

- RIAs must be “deployed” or accessed by the user without a separate installation process.
- RIAs must support robust local application processing on the client.
- RIAs must support an event-driven UI model that extends beyond the traditional page request/response HTML model.
- RIAs must support connectivity and synchronization with server-side middle-tier processing.
- RIAs must be usable in bandwidth-constrained network environments (such as dial-up).
- RIAs should provide a “high fidelity” UI that closely matches the look and feel of a native GUI (for example, support vector graphics).
- RIAs should operate efficiently within multiple Web browsers and on multiple operating systems.
- RIAs may run in stand-alone disconnected mode as well as connected.

For developers, RIAs offer a mechanism to take the traditional rich-client GUI model and support it on the infrastructure of the Web. RIAs are not dependent on service-oriented architectures (SOAs), but they fit well into the principles of a modern SOA. RIAs will provide robust user access channels into SOAs, offload some server-side processing, decrease network loads and provide occasionally connected application models for mobile and wireless devices. Most importantly, RIAs will do this without forcing a developer to write multiple application interfaces using widely divergent technologies – as is the case today between traditional GUI and basic HTML technologies.

RIA technology will provide a new form of rich-client application: there are several other models (for example, the Eclipse rich-client platform from IBM) as well. RIAs will provide a new option, but other models will continue to evolve. We expect multiple technology channels to continue for many applications; however, RIAs will reduce the number of channels needed by closing the technology gap between traditional fat-client GUI features and Web-based architectures.

Acronym Key

GUI	graphical user interface
RIA	rich Internet application
SOA	service-oriented application
UI	user interface

The Nexaweb Platform: Enterprise Class Rich Internet Applications without Compromise

Source: Nexaweb

The Nexaweb Platform is an innovative, standards-based solution that significantly lowers the cost of developing, deploying and maintaining enterprise applications. The Nexaweb Platform delivers all the potential cost advantages of the Web, including global reach, no-install deployment, browser-based access and standards support. The Nexaweb Platform also provides a wide range of built-in capabilities that enable businesses to rapidly create high-performance applications with the productivity, reliability and scalability advantages of client/server.

Nexaweb-enabled applications are enterprise class Rich Internet Applications that offer all the advanced capabilities of other RIAs, but are easier and less costly to develop, deploy and maintain.

While the Nexaweb Platform is based on Java, its innovative architecture eliminates the constraints that other Java-based solutions currently impose. In particular, Nexaweb-enabled applications require no client installation, and can run unchanged on all the Web browser and operating system versions an organization needs to support.

Nexaweb Platform Architecture

A Nexaweb-enabled application is a standard J2EE Web application. It consists of a platform-neutral thin client, which communicates bi-directionally with server-side components (standard JSPs and servlets) over a robust and extremely efficient communications layer.

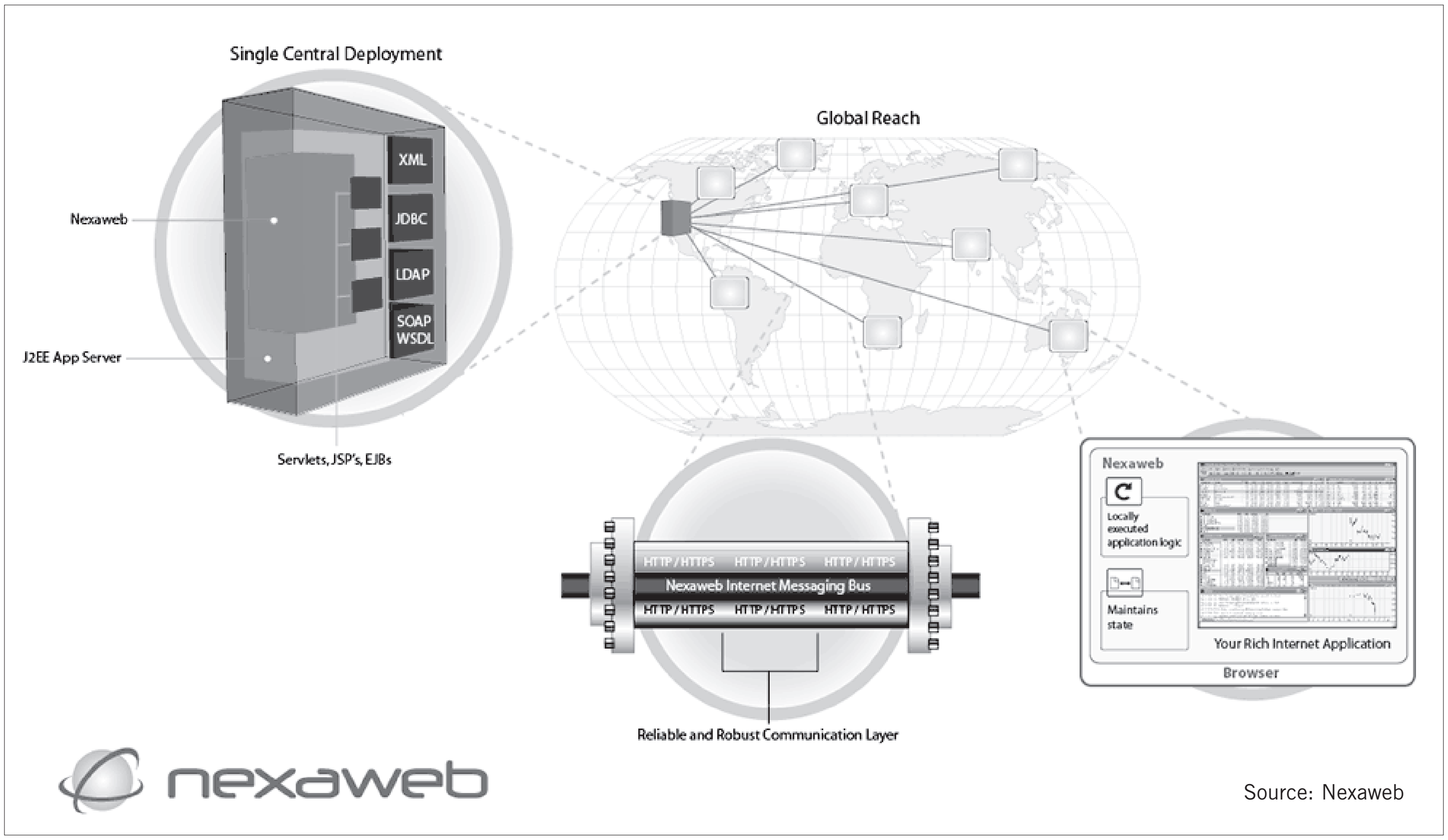
A Nexaweb-enabled application's rich UI is described using XML, and is rendered by the thin client. Application logic, written in Java, can execute on the client-side in response to UI events. Server-side components, in addition to executing business logic, can also dynamically manage the state of the UI.

Updates to UI elements are made incrementally as required: data that is unchanged is not updated, in contrast to HTML applications.

A Nexaweb-enabled application can be deployed as a WAR file among various methods, and has a normal directory structure. It requires only a J2EE Java Servlet container version 2.3 or higher, such as Apache Tomcat 4.x. A Nexaweb-enabled application can also be deployed to a full J2EE application server, such as BEA WebLogic or IBM WebSphere; however, this is not required. In particular EJB programming is not required to use the Nexaweb Platform.

Figure 2 on page 7 illustrates the Nexaweb Platform architecture.

Figure 2: The Nexaweb Platform Architecture.


Single Central Deployment
Nexaweb
J2EE App Server
XML
JDBC
LDAP
SOAP WSDL
Servlets, JSP's, EJBs
Global Reach
HTTP / HTTPS
Nexaweb Internet Messaging Bus
Reliable and Robust Communication Layer
Nexaweb
Locally executed application logic
Maintains state
Your Rich Internet Application
Browser
nexaweb
Source: Nexaweb

Business Benefits of the Nexaweb Platform

The Nexaweb Platform delivers proven business benefits that radically reduce the cost and risk of RIA development and deployment for enterprises today.

Reduced development and maintenance costs

The Nexaweb Platform eliminates the need to install client software or maintain scripts, enabling companies to leverage the Internet more cost-effectively.

Equally important, an organization can deploy the same version of a Nexaweb-enabled application to all its users, across heterogeneous client configurations and network connection types. This eliminates the need to develop and maintain multiple client software versions, the need to standardize client systems and the need to upgrade network infrastructure.

Reduced deployment costs

All clients gain access to new or updated applications immediately upon connecting with a server – no installation is required. Businesses can thus enjoy all the cost advantages of a centralized deployment and management model.

Additionally, companies can minimize retraining costs by leveraging the Nexaweb Platform's unsurpassed rich UI development capabilities to re-architect existing client/server or HTML applications with identical features and look-and-feel.

Improved responsiveness to business drivers

The rapid development and deployment capabilities of the Nexaweb Platform empower development teams to respond more quickly to the changing business needs, and shorten time-to-market for applications. For example, the Nexaweb Platform fully supports the creation of reusable UI, business logic, servlets and JSPs, enabling applications be readily extended to accommodate new requirements in the future.

Reduced technology investment risk

The Nexaweb Platform is built on open standards like Java and XML, for maximum flexibility and compatibility with existing infrastructure, with none of the restrictions associated with vendor lock-in. Organizations need not alter their current development processes or make investments in new tools or skills in order to create and deploy Nexaweb-enabled applications.

Features of the Nexaweb Platform

The Nexaweb Platform is the best technology available for rapidly developing and deploying enterprise class RIAs on existing Web infrastructure.

Key features of the Nexaweb Platform include:

- **Declarative UI definition.** The Nexaweb Platform provides a full complement of UI widgets, which greatly simplify UI development. Screen developers can easily create and modify rich UIs and link UI components to reusable application logic – all without programming – using XML (see Figure 3 below). This frees Java programmers to focus on the application's core logic, and explicitly complements the roles of a typical enterprise development team. The Nexaweb Platform also makes it straightforward to add custom or third-party XML UI widgets to applications with all the capabilities of built-in widgets.

- **Distributed event handling.** The Nexaweb Platform lets application architects decide which events to handle on the client-side (see Figure 4 below), and which to handle on the server-side. Distribution of application logic to maximize performance improves user productivity and satisfaction. At the same time, distributed event handling reduces network utilization by moving appropriate business logic closer to related data sources.

Figure 3: Sample XML source code

```
<nxml>

  <!-- Uses client-side Java to validate field -->

  <dialog title="New Dialog"
          width="240" height="148">
    <freeLayout/>
    <label text="Name:"
           height="20" width="65"
           x="10" y="20"/>
    <textField id="name_field"
               height="20" width="100"
               x="100" y="20"
               focused="true"/>
    <button text="Next &gt;"
            height="25" width="100"
            x="100" y="70"

onCommand="mco://validator.checkName(name_field.text)"/>
  </dialog>
</nxml>
```

Figure 4: Sample Java event handler

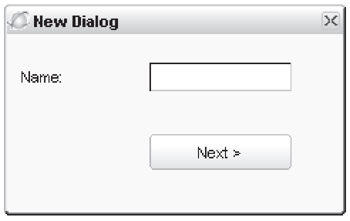
```
package myapp;

public class Validator {
  public void checkName(String name) {
    System.out.println("name entered = " + name);
    if ((name == null) || (name.length() < 1)) {
      // show error alert...
    } else {
      // go to next step...
    }
  }
}
```

The XML code fragment in Figure 3 defines a dialog box in which a user can enter a name. When the user clicks the "Next" button, a client-side Java object processes the event and validates the entry.

Figure 4 shows the corresponding Java event handler, which also runs on the client-side. Figure 5 shows how the dialog box appears to the user.

Figure 5: Sample UI object and the dialog box itself.



- **Build once, deploy everywhere capability.** Built-in Nexaweb client-side components, which are part of every Nexaweb-enabled application, serve to eliminate all version dependencies related to users' browsers, Java Virtual Machines (JVMs) and operating systems. Thus, you can develop and deploy a single version of an application across your entire distributed environment – automatically and with no installation required.

- **Stateful client.** The client-side components of a Nexaweb-enabled application automatically maintain the application's session state across server requests, and can redraw its UI at any time. This "stateful client" model optimizes performance and scalability because UI updates take place dynamically and incrementally, as with client/server applications. In real-world deployments, Nexaweb-enabled applications can reduce network demands on the order of 90% compared to traditional HTML applications.

- **Built-in, enterprise ready communications layer.** The Nexaweb Platform provides a highly robust messaging environment, called the Internet Messaging Bus (IMB), which works reliably across all the unpredictable elements of the commodity Internet – such as firewalls, proxy servers, dial-up connections and satellite networks – in a secure and efficient manner.

  The client- and server-side components of Nexaweb-enabled applications communicate via the IMB. It handles most aspects of messaging in a manner transparent to developers; that is, it eliminates the need to create application-specific protocols, or to write custom code that encodes/decodes messages. This reduces development effort and frees programmers to focus on business logic.

  The IMB further allows Nexaweb-enabled applications to communicate bi-directionally in an any-to-any fashion, an exceptionally advanced and flexible messaging model not found in traditional Web applications. Message delivery and order of delivery are predictable for improved dependability.

  To ensure robust security and maximum compatibility with existing infrastructure, the IMB runs over standard HTTP and HTTPS protocols, and can work across security mechanisms such as SSL.

- **Open architecture throughout.** The Nexaweb Platform is designed from the ground up to be fully compatible with all aspects of the Internet, in order to provide companies with the greatest possible flexibility, freedom of choice and investment protection. A primary design consideration for the Nexaweb Platform has been to make developing, deploying and managing Nexaweb-enabled applications as similar to traditional Web applications as possible. This allows Nexaweb-enabled applications to work with your existing networks, servers, desktops and software.

  The development process for Nexaweb-enabled applications is identical to that of other Java Web applications, so development teams can use their choice of tools. Likewise, Nexaweb-enabled applications can be deployed and managed just like any other enterprise Java Web application; there are no new administrative costs or tasks to perform.

Comparing Nexaweb with Other RIA Platforms

The Nexaweb Platform is the only solution that provides all the built-in features required to cost-effectively develop, deploy and maintain enterprise class RIAs. In addition, Nexaweb-enabled RIAs enable organizations to leverage their current investments in IT skills and infrastructure, development tools, and Web standards, thus reducing the investment risk associated with proprietary RIA solutions.

Table 1 (on page 10) shows how Nexaweb-enabled applications compare with conventional RIAs against Gartner's RIA criteria.[3]

The Nexaweb Platform also incorporates a host of built-in capabilities that would need to be created from scratch with most other RIA platforms, such as enterprise class messaging, client-side security and full support for custom XML widgets.

When to Consider Nexaweb

The Nexaweb Platform is the best available foundation for delivering enterprise class applications via the Internet. It is suitable for developing all kinds of Internet applications, particularly those that require enterprise class reliability, scalability, usability and reach.

However, because of its unique ability to fully meet all enterprise RIA requirements, the Nexaweb Platform is the best solution, far and away, for

[3]Gartner Research Note Rich Internet Applications Are the Next Evolution of the Web, G00126924, M. Driver, R. Valdes, G. Phifer, May 4, 2005.

organizations interested in leveraging the Internet to deliver application that can:

- Support a large number of geographically distributed users over a range of connection types (dial-up, LAN/WAN, etc.)
- Provide users with real-time access to large and/or complex data sets
- Support diverse client environments, including multiple operating systems and Web browsers

The Nexaweb Platform successfully addresses all these critical enterprise challenges. By combining Nexaweb technology with the expertise, tools and infrastructure you already have, you can rapidly develop and deploy high-performance, enterprise class Internet applications that run on your existing clients, servers and networks. This is what we mean by *Enterprise Internet Applications without Compromise*.

Table 1: How RIA technologies meet Gartner's RIA criteria

	DHTML/ JavaScript Approaches	**Traditional Java-based Approaches**	**.NET-based Approaches**	**Flash-based Approaches**	**The Nexaweb Platform**
Gartner "Must Have" RIA Criteria[4]					
No-install deployment	✔	NO	NO	Requires Flash Player installation	✔
Distributed event handling	Limited	✔	✔	Limited	✔
Stateful client	Limited	✔	✔	✔	✔
Connectivity with server-side business logic	✔	✔	✔	✔	✔
Operate efficiently over low-bandwidth connections	✔	NO	NO	Limited	✔
Gartner "Should Have" RIA Criteria					
Rich UI with "look and feel" equivalent to traditional GUIs	Limited	✔	✔	✔	✔
Support for diverse client environments	Browser-dependent	JRE- dependent	Requires .NET CLR	Requires Flash Player installation	✔
Gartner "May Have" RIA Criteria					
Support for disconnected access	NO	✔	✔	NO	✔

[4]Criteria column only attributed to Gartner Research Note Rich Internet Applications Are the Next Evolution of the Web, G00126924, M. Driver, R. Valdes, G. Phifer, May 4, 2005. Balance of table content attributed to Nexaweb Technologies, Inc.

For more information:

For additional information on Nexaweb products and services please visit **www.nexaweb.com**.

The Nexaweb *Concept Guide* **http://www.nexaweb.com/pdf/GC/Nexaweb_Platform_Concept_Guide.pdf** describes the features and business benefits of the Nexaweb Platform, along with a detailed tutorial that illustrates how to create a Nexaweb-enabled application.

Download the Nexaweb Platform Data Sheet at **http://www.nexaweb.com/pdf/GC/nexaweb_datasheet4-05.pdf**

Nexaweb White Papers:

Enterprise Internet Applications without Compromise:
http://www.nexaweb.com/eia.pdf

Enterprise Web Apps: Alternatives to Swing:
http://www.nexaweb.com/nexawebvsswing.pdf

Rich Internet Applications: A Vendor Evaluation Report:
http://www.nexaweb.com/Softcon_final.pdf

Nexaweb Case Study: Best Western Hotels:
http://www.nexaweb.com/pdf/GC/best_western_cs.pdf Nexaweb Demos:

Nexaweb Guided Tours:

System Management Application Guided Tour:
http://stream.nexaweb.com/nxaSanMngr/SanMngrTour.html

Order Management Application Guided Tour:
http://stream.nexaweb.com/nxaOrder/OrderInvTour.html

Articles:

"AJAX: Asychronous Java and XML?" Developer.com, August 11, 2005.
http://www.developer.com/design/article.php/3526681

"XML Rich-Client Technology Brings Zero-Install Rich Client Capabilities to J2EE." JDJ, July 27, 2005.
http://jdj.sys-con.com/read/111208.htm

"Rich Internet Apps to Tie SOA to Desktops." Integration Developer News, February 8, 2005.
http://www.idevnews.com/IntegrationNews.asp?ID=155

"SoftCon Chooses Nexaweb for Siemens Project." ADTMag.com, February 23, 2005.
http://www.adtmag.com/article.asp?id=10677

InfoWorld Product Review: The Nexaweb Platform, December 20, 2004.
http://www.infoworld.com/Nexaweb_3.2/product_50823.html?view=0&curNodeId=0

About Nexaweb

Nexaweb Technologies, Inc. (www.nexaweb.com) provides the leading software platform for building and deploying enterprise-class Rich Internet Applications on existing servers, clients and networks. In real-world deployments, Nexaweb customers have reduced network bandwidth demands by up to 90 percent, improved application response times by over 70 percent, compressed development time and cost by 50 percent, and cut deployment and maintenance costs by over 50 percent. Nexaweb technology enables enterprises to enhance business agility, improve corporate performance, and further leverage current investments to reduce IT costs, increase revenue, and better serve their customers, partners and employees.

Founded in February, 2000, Nexaweb is based in Cambridge, Massachusetts, USA.

Nexaweb Technologies, Inc.
One Charles Park
Cambridge, MA 02142
Phone: (617) 577-8100
Fax: (617) 577-8155
www.nexaweb.com

Nexaweb Japan KK
2-9-10 Sotokanda, Chiyoda-ku
Tokyo 101-0021
Japan
Phone:
Fax:
www.nexaweb.co.jp