



macromedia® white paper

## Choosing a J2EE application server for your commercial software application

by Charlie Arehart  
Founder/CTO SystemManage, and JRun Consultant

May 2002

Copyright © 2002 Macromedia, Inc. All rights reserved.

The information contained in this document represents the current view of Macromedia on the issue discussed as of the date of publication. Because Macromedia must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Macromedia, and Macromedia cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for information purposes only. MACROMEDIA MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Macromedia may have patents, patent applications, trademark, copyright or other intellectual property rights covering the subject matter of this document. Except as expressly provided in any written license agreement from Macromedia, the furnishing of this document does not give you any license to these patents, trademarks, copyrights or other intellectual property.

Add Life to the Web™, Afterburner™, Aftershock™, Andromedia®, Allaire®, Animation PowerPack™, Aria®, Attain™, Authorware®, Authorware Star®, Backstage™, Bright Tiger®, Clustercats®, ColdFusion®, Design In Motion™, Director®, Dream Templates™, Dreamweaver®, Drumbeat 2000™, EDJE®, EJIPT®, Extreme 3D®, Fireworks®, Flash™, Fontographer®, FreeHand®, Generator™, HomeSite™, JFusion™, JRun™, Kawa™, Know Your Site®, Knowledge Objects™, Knowledge Stream™, Knowledge Track™, LikeMinds™, Lingo™, Live Effects™, MacRecorder Logo and Design®, Macromedia®, Macromedia Action!®, Macromedia Flash™, Macromedia M Logo and Design™, Macromedia Spectra™, Macromedia xRes Logo and Design®, MacroModel®, Made with Macromedia™, Made with Macromedia Logo and Design®, MAGIC Logo and Design™, Mediamaker®, Movie Critic®, Open Sesame!™, Roundtrip®, Roundtrip HTML™, Shockwave®, Sitespring™, SoundEdit™, Titlemaker™, UltraDev™, Web Design 101™, what the web can be™, Xtra™ are either trademarks or registered trademarks of Macromedia, Inc. in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Macromedia, Inc.  
600 Townsend Street  
San Francisco, CA 94103  
415-252-2000

## Contents

Introduction.....	1
About the J2EE platform.....	2
Enhancements in J2EE 1.3.....	3
About Macromedia JRun 4.....	3
JRun: the ISV solution.....	3
J2EE: opportunities, challenges, and solutions.....	4
Support for the entire J2EE API and its components.....	4
Performance, scalability, and availability.....	4
Performance.....	5
Clustering for EJBs.....	5
JMX implementation and availability.....	6
Extensibility and flexibility.....	6
Affordability and ROI.....	6
Enterprise integration.....	6
Integrating enterprise systems with JCA.....	7
Greater XML integration with JAXP.....	7
Integration with web services.....	7
Developer productivity.....	7
Automatic and hot deployment.....	8
XDoclet support: automatic EJB interfaces and deployment descriptors.....	8
Stubless EJB development.....	8
Dynamic creation of database tables for entity beans.....	8
Customizable EJB container-managed persistence.....	8
Enterprise Deployment Wizard.....	9
Technical support.....	9
Flash integration.....	9
Breadth of platforms supported.....	9
Operating system support.....	9
Integration with existing databases.....	10
Integration with existing web servers.....	10
Recommending an application server to your customer.....	10
Considering an embedded application server.....	11
Benefits to the application customer.....	12
Small footprint and flexible customization.....	12
Ease of installation and administration by the customer.....	12
Benefits to the ISV and their developers.....	12
Maintaining your own brand.....	12
Ease of building your application and bundling the embedded server.....	13
Ease of ongoing maintenance by you and your customer.....	13

Obtaining JRun trial edition or Developer Edition and additional product information.....	13
Conclusion.....	13

Whether you are deciding which J2EE server you want to support, helping your customer choose one, or contemplating embedding a J2EE server with your solution, Macromedia JRun 4 offers a fast, affordable, and reliable platform for implementing your product into a customer's production environment. With the available Software Development Kit (SDK), JRun provides an excellent solution for Independent Software Vendors (ISVs) looking to embed a Java application server.

## Introduction

As a developer of products and services based on the J2EE (Java 2 Platform, Enterprise Edition) platform, you face an important challenge when deciding which J2EE application server to use (or support).

You might have customers demanding that your application run on their existing J2EE architecture, or they might be asking you to recommend an architecture for implementing your solution. In either case, choosing the right application server for you and one that meets your customers' needs is a business-critical decision.

You could also provide your application on a J2EE application server that is deployed with your application in such a way that the server is transparent to your customer. This way they won't need to install and manage a separate server. In addition, they will be able to keep any current application server and even run the one you've decided to "embed" on the same physical server, transparent to, and having no conflict with, any other server in their architecture.

This paper describes the challenges you face as an ISV choosing the right J2EE platform for your customers or the implementation of your application. The following topics are included:

- About the J2EE platform
- About Macromedia JRun 4
- J2EE: opportunities, challenges, and solutions
- Recommending an application server to your customer
- Considering an embedded application server
- Obtaining JRun trial edition or Developer Edition and additional product information

After you decide that JRun makes the most sense for your application development needs, or if you find you still need more information, visit the award-winning Macromedia website at: <http://www.macromedia.com/software/jrun>. Besides ample product information and complete product documentation, it provides many specific ISV resources and free product downloads.

## About the J2EE platform

The J2EE platform is a powerful solution for today's enterprise web application development requirements. Among its strengths, J2EE enables you to fully leverage Java Servlets, JavaServer Pages (JSPs), Enterprise JavaBeans (EJBs), and Java Message Service (JMS).

**Table 1:** *The following table describes some benefits of using the J2EE platform:*

<b>Benefit</b>	<b>Description</b>
Reusability	The Java platform comprises a robust and extremely broad set of libraries—literally hundreds of them—providing a vast array of commonly required services and solutions. With Java and J2EE, the promise of reusability and the benefits of standards-based development are clearly realized.
Productivity	Besides being an object-oriented programming language, Java also provides built-in support for many features that result in programming challenges in other platforms, including multithreading support and automatic garbage collection. J2EE also solves common web application challenges in stateless HTTP request processing, as well as messaging and distributed components (supporting persistence, security, and transactions).
Portability	Since the J2EE platform is a well-defined standard, many architectural alternatives are available for deployment of your application. From a broad range of operating systems, databases, and web servers, to choices among application servers, developing for the J2EE platform leaves you free to choose the best alternative for you, while still enabling you to move to another platform supporting J2EE when your needs change.
Flexibility	With support for thin and fat clients in the web programming model, J2EE applications benefit from centrally located configuration, administration, and maintenance, leading to savings in deployment and support costs. Choices about the best architectural implementation (clustered machines, alternative databases, and so on) are abstracted, further increasing your flexibility in selecting among alternatives.
Scalability	When your application expects very large workloads, it might require use of multiple servers for load-balancing and failover. Not all application servers are the same. Some support complete clustering of not just the web server tier, but also the EJB tier.
Adaptability	Vendor implementations, driven by market forces, will need to implement the new Java standards as they evolve. Your applications will benefit from those new Java features as they are added.  The J2EE architecture includes support for features that reflect the need for greater integration in today's enterprises, both within their own architecture and among those of other enterprises and service providers. These features include support for the following: <b>XML</b> The common language for enterprise data interchange, XML is becoming even more important, because more platforms now support it, it is easier to use, and the challenges of inter-process communication are being resolved. <b>Web services</b> A natural outgrowth of the growing ubiquity of XML, web services define a standard means to share information and services. (For more information, see <a href="#">“Integration with web services”</a> .) <b>Messaging</b> Asynchronous processing brings a new dimension to application development, and the Java Message Service (JMS) brings this capability to J2EE applications and servers.

## Enhancements in J2EE 1.3

As a developer, you should be aware of a few changes in the 1.3 API that bring additional benefit for developer productivity:

- With support for the Servlets 2.3 API, developers can now leverage filters, a powerful new feature that allows pre- and post-processing of servlet requests. For example, filters can provide a more effective means to perform authentication of requests and compression of responses. Application and session events are useful for application and session initialization and cleanup code that can replace preloaded servlets in some cases and provide enhanced management of session processing.
- Message-driven beans (MDBs) combine the asynchronous benefits of the JMS with the EJB component architecture. With MDBs, operations can be re-architected so that long-running processes can be requested and queued for later execution.
- Support for EJB query language (EJBQL), a standard, database-independent mechanism for database interaction, brings improved order to the specification of container-managed persistence.
- In terms of performance, EJB 2.0 introduces support for local interfaces to efficiently invoke an EJB component residing on the same computer as the calling application. For many applications, this should have a dramatic impact on performance.

## About Macromedia JRun 4

Macromedia JRun 4 is a J2EE 1.3 compatible application server (supporting servlets, JSPs, EJBs, JMS, and all other J2EE services) that is powerful, fast, and scalable, while also being affordable, easy-to-use, and highly customizable for ISV needs. JRun supports the benefits listed in the preceding table and the J2EE 1.3 API specification. It has all this while providing a cost-effective solution that is supported by a leading software company.

### JRun: the ISV solution

JRun 4 empowers leading ISVs to build reliable J2EE-based applications more quickly and affordably. The available SDK lets software vendors who choose to embed JRun customize the installation and configuration of JRun, and smoothly integrate JRun with their application. Additionally, the small footprint of JRun allows ISVs to deliver a powerful J2EE 1.3 compatible application server with their product, without adding a large amount of incremental overhead.

New features in JRun 4, such as the Java Management Extensions (JMX) service-based architecture, enterprise-class clustering, hot deployment, and improvements to the JRun Management Console (JMC), help ISVs speed the development and deployment of their applications.

Macromedia provides high-quality technical support for JRun, which many of the open source alternatives do not offer for their products. Priced at a fraction of the leading J2EE application servers, JRun delivers an affordable J2EE-compatible solution that meets the cost objectives of leading ISVs and other developers.

Indeed, beyond the overall promise of Java and J2EE, unique opportunities and challenges face ISV application developers. JRun 4 has been designed to help you take advantage of those opportunities and meet those challenges.

## **J2EE: opportunities, challenges, and solutions**

If you are the vendor of a J2EE application that you want to offer customers for deployment on their servers, you face several unique challenges. Choosing a full-featured application server like JRun, designed especially for ISVs, might give you an advantage.

Some of the issues faced by the typical developer who has to choose a J2EE application server might be even more important to an ISV that has to decide which application server to support, recommend, or embed. The critical attributes you need to look for include the following:

- Support for the entire J2EE API and its components
- [Performance, scalability and availability](#)
- [Extensibility and flexibility](#)
- [Affordability and ROI](#)
- [Enterprise integration](#)
- [Developer productivity](#)
- [Breadth of platforms supported](#)

### **Support for the entire J2EE API and its components**

JRun was one of the first application servers to become certified J2EE 1.3 compatible. This demonstrates the Macromedia commitment to J2EE. JRun supports the following facets of J2EE application development:

- Servlets and JSPs
- EJBs—Enterprise JavaBeans (including message-driven beans)
- JDBC—Java Database Connectivity
- JMS—Java Messaging Service
- JTA and JTS—Java Transaction API and Service
- JCA—Java Connectivity API
- JAXP—Java APIs for XML Processing
- JMX—Java Management Extensions
- JNDI—Java Naming and Directory Interface
- JAAS—Java Authentication and Authorization Service
- JavaMail
- JavaBeans Activation Framework

### **Performance, scalability, and availability**

JRun is an advanced J2EE application server that not only meets the J2EE specifications, but also exceeds expectations by providing the critical functionality needed to assure high-quality deployment of your systems. With a new and innovative approach to its architecture, server clustering, and runtime performance, JRun enhances performance, scalability, and availability.

## **Performance**

Whether creating applications for the enterprise, e-commerce, or other business needs, performance is critical. You need to know that the application server you choose is capable of performing effectively and efficiently.

Macromedia JRun 4 provides superior performance, as measured by servlet and JSP page response time, throughput, scalability, and reliability. Customers can develop and deploy applications on JRun 4 with the knowledge that their applications can handle a high volume of user activity with predictable performance and stability.

Macromedia recognizes the importance of performance to JRun customers and has developed a separate [white paper](#) detailing performance issues and metrics. It is available at the Macromedia website, at <http://www.macromedia.com/software/jrun/whitepapers>.

While the licensing restrictions of other commercial application servers preclude the publishing of their performance results, comparisons with Apache Tomcat version 4 show that JRun performs from two to four times better on a variety of typical web application processing tasks.

Another important factor to consider is the price-performance ratio. JRun provides J2EE enterprise-level performance at a price among the lowest of commercial J2EE application servers. It's able to provide this performance on a variety of platforms and configurations, including relatively lightweight configurations that often stress other application server alternatives.

## **Clustering for EJBs**

JRun 4 provides enterprise-class server clustering for maximum reliability and scalability. Clustering is built into both the web server connectors and the JRun kernel, including EJBs and message queues, as follows:

- **Standards-based** JRun leverages a Jini™ technology-based architecture to transparently deliver clustering at all levels to support load-balancing and failover.
- **Connector clustering** Configure your environment so that requests received through your web server can be processed by any one of a cluster of JRun servers.
- **Web server clustering** Create a cluster of web servers so that requests sent from the web browser can be processed by any of the web servers in the cluster.
- **Object clustering** Create a cluster of JRun servers so that requests received by a web container can be routed to and processed by any of the EJB containers in the cluster.

At the web server connector level, clustering enables load-balancing and automatic failover between the web server and the web containers. In-memory session replication ensures that state information is preserved if the web server fails over to another web container in the cluster.

At the kernel level, clustering enables load-balancing and automatic failover between clients (for example, JSPs and servlets) and server objects, such as EJBs, JMS queues and topics, JNDI trees, or any custom service. The object's state (for example, the state of a stateful EJB) is automatically replicated in an optimized manner to provide the highest level of reliability, while maintaining performance.

JRun servers use the Jini technology to dynamically locate their peers in the cluster.

### ***JMX implementation and availability***

All JRun features are implemented as Java Management Extensions (JMX) services. JMX is an emerging Java standard for management and customization that Macromedia has thoroughly embraced. All JRun features (EJB container, web container, logging, and so on) are implemented as JMX services (called MBeans) plugged into the JRun kernel. These services can be managed by the JMX-enabled JRun Management Console (JMC), or other third-party JMX-enabled management tools.

Because services are independent from each other and can be restarted individually, the JRun service-based architecture ensures high application availability. It also provides a highly customizable and easily extensible platform. As a result, unneeded services can be unplugged to avoid the overhead of unused functionality. Administrators, advanced developers, or ISVs can also create their own custom services (MBeans) and plug them into the JRun kernel.

All standard JRun and custom services can take advantage of the clustering capabilities built into the JRun kernel, as described previously.

### **Extensibility and flexibility**

JRun is a highly configurable solution. As described later, in “[Considering an embedded application server](#),” the SDK offers you more flexibility to decide which components you want to include.

While complete support for the entire J2EE API is a powerful basis for your application, you might not require all these components. A significant advantage of JRun for ISVs is that you can choose to implement as few, or as many, of the J2EE APIs as you want. By leaving out those that you do not need, you can reduce the memory footprint and management overhead of your installed or embedded JRun server.

### **Affordability and ROI**

Also important, especially for ISVs that deploy applications on the servers of many different customers, are matters of affordability, including low total cost of ownership (TCO) and return on investment (ROI).

Besides being more affordable than other commercial application servers, JRun is also more cost-effective over time when compared to open source alternatives because of all the additional developer productivity features JRun provides. These are described later in this paper.

Both the SDK and the standard configuration of JRun provide a cost-effective J2EE application server platform that saves you time and money, which, as an ISV, you can either pass on to your customers or use to increase profitability.

### **Enterprise integration**

As you develop applications for your customers, you will often require integration with their existing enterprise information systems (EIS), including Enterprise Resource Planning (ERP) systems, mainframe transaction processing systems, and legacy database systems.

Whether by means of direct integration using JCA or XML-based integration with JAXP or web services, the J2EE architecture—and JRun—provides alternatives.

### ***Integrating enterprise systems with JCA***

With its support for the J2EE Connector Architecture (JCA) 1.0, JRun provides a standard way for back-end applications (for example, enterprise resource planning (ERP), customer relationship management (CRM), and legacy systems such as SAP, PeopleSoft, Siebel, Oracle or third-party connector developers) to plug into the J2EE platform.

### ***Greater XML integration with JAXP***

With support for Java APIs for XML Processing (JAXP) 1.1, JRun enables the latest XML development features. JAXP enables applications to parse and transform XML documents. As an additional measure of flexibility, the implementation of JAXP support with JRun is independent of a particular XML processing implementation, uses a pluggable architecture for adding your choice of XML parser or XSL processor into the server, and overrides the default components that ship with JRun. JAXP supports processing of XML documents that use either Document Object Model (DOM), Simple API for XML Parsing (SAX), and XML Stylesheet Language Transformations (XSLT).

### ***Integration with web services***

JRun lets you publish and consume web services. Web services provide platform- and location-independent computing using XML and standard Internet protocols such as HTTP. By allowing previously incompatible applications to interoperate over the web, regardless of language, platform, or operating system, web services create new business opportunities and help companies adapt to changing business relationships. For example, a Microsoft .NET component can communicate with a J2EE component, such as an Enterprise JavaBean (EJB).

Using JRun, you can reuse existing Java code as a web service, or write new code specifically to publish as a web service. Also, you can create object- and tag-based clients that can invoke methods on remote web services, even when those services reside on non-Java platforms.

Building web services in JRun is as simple as editing a deployment descriptor to expose your regular Java classes or Enterprise JavaBeans as web services. Web Service Description Language (WSDL) is automatically generated upon request. JRun also provides different options to consume web services. A WSDL2Java tool allows you to automatically generate Java proxies based on a WSDL document. The JRun web services tag library can also be used to dynamically invoke web services from your application more easily.

Macromedia recognizes the importance of web services integration for its customers and has developed a white paper, *Using Web Services in Macromedia JRun 4*. It is available at: [www.macromedia.com/support/jrun/extend/using\\_web\\_services/](http://www.macromedia.com/support/jrun/extend/using_web_services/).

### **Developer productivity**

Whether developing an application for yourself or your customers, ease of development, rapid deployment, and time-to-market are important. There are many ways in which JRun makes development and deployment easier. These can significantly benefit developers, especially those working for an ISV that needs to distribute updated code to customers.

Vendor support is important, too. The following facets of JRun contribute to your developer productivity:

- [Automatic and hot deployment](#)
- [XDoclet support: automatic EJB interfaces and deployment descriptors](#)

- [Stubless EJB development](#)
- [Dynamic creation of database tables for entity beans](#)
- [Customizable EJB container-managed persistence](#)
- [EJB Deployment Wizard](#)
- [Technical support](#)
- Flash integration

### ***Automatic and hot deployment***

JRun sets a new standard in J2EE application deployment. With JRun, deploying an application is as simple as copying an EAR, JAR, WAR, or RAR file to the server's deployment directory. JRun automatically senses the new file, deploys it, and makes it available in the target server. This allows ISVs to test and update their application, without needing to stop and restart the server.

Also, during development, you can hot modify any application component (servlets, JavaServer pages, Enterprise JavaBeans, Java classes, tag libraries, and deployment descriptors). Your changes take effect immediately, with no need to restart the server. This makes your application development process go faster, and improves your time-to-market.

### ***XDoclet support: automatic EJB interfaces and deployment descriptors***

XDoclet is a popular open source Java utility that can greatly simplify the development of J2EE components (such as Enterprise JavaBeans, servlets, and custom tags) by allowing you to maintain your component metadata using Javadoc-style tags in a single source file.

For example, to create an EJB, you traditionally have to maintain four different files: the home interface, the remote interface, the implementation class, and the deployment descriptor. Using XDoclet, you only maintain a single source file (the implementation class) that you annotate with Javadoc-style tags that indicate how the auxiliary files should be built.

XDoclet has been completely integrated into JRun 4. JRun senses modifications to the source file, and automatically builds the auxiliary files (such as interfaces and deployment descriptors).

### ***Stubless EJB development***

Even without leveraging XDoclet, JRun makes EJB development easier because there is no need to generate EJB stubs to deploy EJBs in JRun. The server handles this for you dynamically.

### ***Dynamic creation of database tables for entity beans***

When you deploy an entity bean and its required database tables do not yet exist, JRun generates them for you when you configure the appropriate Java Database Connectivity (JDBC) data source in JRun.

### ***Customizable EJB container-managed persistence***

JRun provides a JRun-specific EJB deployment descriptor, `jrun-ejb-jar.xml`, that lets you customize container-managed persistence (CMP) for CMP beans.

### ***Enterprise Deployment Wizard***

The JRun Enterprise Deployment Wizard streamlines the process of developing and deploying Enterprise JavaBeans. Using its Swing-based graphical user interface, you can create any type of EJB from scratch, edit the deployment descriptor of existing EJBs, package them into JAR files, and deploy them to JRun.

With the Enterprise Deployment Wizard new object-relational mapping capabilities for CMP 1.1 and CMP 2.0, you can generate and deploy a fully functioning entity bean in minutes. The Enterprise Deployment Wizard runs as a stand-alone tool, or as a plug-in on top of Borland JBuilder, Sun Forte, or IntelliJ IDEA.

### ***Technical support***

JRun is backed by one of the world's leading software companies, Macromedia. Besides top-quality technical support representatives and a very active development community, JRun customers can also leverage the JRun Support website at <http://www.macromedia.com/support/jrun>. The site provides many resources for JRun developers, including all the specifications, documentation, white papers, downloadable trial versions, and much more.

It also offers many resources for obtaining self-help, including a rich knowledge base, very active developer forums (including an ISV-specific forum), and a Designer and Developer Center that offers articles and tips that let you continue your education. Instructor-led training is also available.

### ***Flash integration***

Out of the box, JRun provides substantial integration with Macromedia Flash, the leading tool for creating highly interactive customer interfaces.

With built-in Macromedia Flash connectivity, you can provide J2EE applications with a high-impact user interface. Flash movies can connect to EJBs, servlets and other Java classes, and JMX MBeans deployed in JRun in a simple, fast, and secure manner. For more information, see the Flash and JRun integration site on the Macromedia website that includes a tutorial at <http://www.macromedia.com/software/jrun/resources/integration/flash/tutorials>

### ***Breadth of platforms supported***

When deciding among J2EE application servers, consider the server's ability to support the platforms on which you will be deploying.

### ***Operating system support***

JRun requires the following minimum operating system versions:

- Windows 98/ME/NT/2000/XP
- Solaris 7, 8
- Red Hat Linux 6.2, 7.X
- IBM AIX 4.3, 5L v5.1
- HP-UX 11.0, 11i
- Compaq Tru64 Unix 5.1

- Turbolinux 6.5
- SuSE Linux 7.2, 7.3

JRun also supports the following virtual machines:

- Sun Java Virtual Machine 1.3.1 or higher
- IBM Java Virtual Machine 1.3.1 or higher
- Appeal JRockit

### ***Integration with existing databases***

JRun supports any database for which a JDBC driver is available. In addition, JRun comes with type 4 JDBC drivers for the following databases:

- Oracle
- Microsoft SQL Server
- Sybase
- DB2
- Informix

JRun also comes with a restricted version of the Pointbase Server, a small footprint Java database. The Sun JDBC:ODBC bridge provides access to any ODBC-compatible data source, although Sun does not recommend it for production use.

### ***Integration with existing web servers***

JRun comes with its own integrated, production-quality web server. High performance and clustering-enabled web server connectors also let you use JRun with every major web server, including:

- Apache HTTP Server
- Microsoft IIS
- Netscape Enterprise Server
- Netscape iPlanet
- Zeus Web Server

The JRun SDK lets you access the JRun installer scripts to modify how JRun connects to the web server(s).

## **Recommending an application server to your customer**

You might be inclined to simply recommend one or a variety of application servers to your customer and then develop your application to support those server(s). But what if your customer already has a specific application server on which they wish to deploy your application? Will your application work on that application server? What if they have concerns over whether the server you recommend will support any future development they might want to do?

The choice of platforms that an ISV supports is based on the needs and requirements of their customers, as well as their own needs. Naturally, an ISV cannot support every application server immediately. Despite the promise of Java and J2EE portability, in some cases code developed for one application server is not readily ported to another. Each application server vendor offers options for improving some aspect of the environment in which applications run, and indeed some aspects of J2EE implementation are left to application server vendors to resolve.

If you have already chosen some other application server as your standard, consider JRun as yet another alternative for recommendation to your customers. This not only gives you an advantage since JRun offers ISVs many benefits, it also benefits your customers because they will have more choices in the application servers you offer for your application. JRun can also coexist quite well with other J2EE application servers that might be installed in your customer's site.

There are costs to consider, especially compared to choosing to recommend only one application server or to simply embed the application server under your application. By supporting multiple servers (if you have already chosen and developed for another), you must test your application and indeed your development process against multiple servers. You also might have to deal with some incompatibilities (since application servers are given some leeway in implementation of some aspects of the J2EE specification). Also, consider that unlike JRun 4, some of the more popular application servers might not be fully compliant with the J2EE 1.3 specification, which could lead to challenges in bringing your existing application up-to-date.

If these challenges become too great, you do have another option. As the next section describes, JRun can be configured as an embedded application server. This offers many benefits. Indeed, many of the benefits described will give you more reasons to use JRun, regardless of whether you prefer to embed it.

## **Considering an embedded application server**

One of the most powerful ways to leverage JRun is to embed it as a transparent application server that supports your application. As a result, your customer won't need to worry about understanding the underlying server implementation. This section describes embedding JRun as your application server.

If your customer already has an installed application server, you might be hesitant to recommend that they implement another one. Then again, their existing one might not be able to support your application (perhaps the server is not compliant with modern J2EE standards). On the other hand, if they don't currently have an application server installed, they might not have the skills to support installing one themselves. In either of those cases, embedding an application server within your product is a highly effective solution.

An ISV benefits from this approach because you may not have to consider supporting multiple application servers. You could support only the one that you choose to embed in your distribution. You can also better tune your application for optimal performance with that chosen embedded application server. If you do choose to support multiple application servers, the embedded approach could be yet another alternative.

A benefit for your customers is that they do not have to conduct another evaluation or request for proposal (RFP) process to select the application server. Also, the customer, typically someone in a Line of Business (LOB) Unit, can focus on their business requirements and how your application can meet them, rather than worry about the details of choosing and implementing an application server.

The following sections describe the benefits of an embedded server, and the SDK, in terms of how it benefits your ISV application customer and you, the ISV developer. Some of these benefits overlap and benefit both.

## **Benefits to the application customer**

The benefits that will assist the customer primarily deal with the installation of the application server and how to make that easier for the customer.

### ***Small footprint and flexible customization***

You can modify the JRun installer to exclude unnecessary components of JRun. This reduces the footprint in terms of memory required to run the application server (or, from your customer's perspective, your application).

In addition, you can use the SDK to remove some or all of the default JRun applications to make the installation files smaller or hide these applications from your customer. This lets you run JRun on even modestly configured servers. JRun is a great solution for satisfying a wide range of customer resource configurations.

### ***Ease of installation and administration by the customer***

If the customer is not very skilled in application server installation and maintenance, you can use the SDK to configure JRun to perform a "silent install" that is transparent to your customers. This lets them avoid the complication of setting up such a server. Also, if they already have another application server installed, this embedded one can be implemented so as not to conflict with any existing one.

Additionally, you can include your web applications in the JRun installation scripts so that your applications are deployed at installation time.

JRun services can be added and removed as needed, giving you the ability to only install the services that are needed to run your application. Also, there is no need to restart JRun when other services are added or removed.

## **Benefits to the ISV and their developers**

The benefits in this section relate to the needs of ISVs and the developers of ISV applications.

### ***Maintaining your own brand***

The existence, installation, and maintenance of the application server can be made transparent to your customers, so as not to interfere with the branding of your application. You can install JRun and connectors transparently, hide the JRun Management Console (JMC), hide JRun on the desktop, set the JRun license key, and more.

### ***Ease of building your application and bundling the embedded server***

The earlier section, "[J2EE: opportunities, challenges, and solutions](#)," described the many benefits JRun offers developers, including hot and automatic deployment, XDoclet support, and stubless EJB development. These are equally valuable to the ISV developer, if not more so, in that they especially facilitate changes you might need to make and deploy over multiple customer installations.

Another benefit when building your application for an embedded server is that your engineering and QA teams can test the application against this single application server. The JRun instrumentation feature lets you identify and resolve performance bottlenecks in your code by tracing and timing each method invocation in your application. Instrumentation is supported for JSPs, servlets, and EJBs.

Also, the SDK provides integration with InstallShield on Windows and GNU Make on Unix platforms.

### ***Ease of ongoing maintenance by you and your customer***

The hot and automatic deployment features of JRun also benefit you and your customers during ongoing maintenance of the application (if they're given access to the code and management console).

As mentioned previously, you can hide the JRun Management Console (JMC), or you can expose it. If it's exposed, JRun allows you to add custom links to the JRun navigation tree. This lets you add a link to your website or a link to technical support information within your product installation.

An embedded server shields your customer from complexity.

## **Obtaining JRun trial edition or Developer Edition and additional product information**

When you are ready to try JRun, go to the Macromedia website for the 30-day evaluation and a free Developer Edition at <http://www.macromedia.com/software/jrun/download/>.

The fully functioning 30-day trial of JRun offers all the benefits of a J2EE 1.3 compatible application server, and supports Servlets 1.2, JSP 2.3, EJB 2.0, JMS, and all other J2EE services.

For developers, the available Developer Edition is a non-expiring edition that is limited to the developer's local machine plus one additional IP address. This is an excellent alternative for both demonstration, extended trials, and development.

The available SDK is your key to making JRun your embedded J2EE application server of choice. See the Macromedia website for details on how to obtain the SDK.

## **Conclusion**

Designed from the ground up to provide a pure implementation of J2EE standards, JRun 4 is the fast, affordable, and reliable J2EE application server.

ISVs can customize the installation and configuration of JRun in order to smoothly integrate JRun with their applications. Additionally, the small footprint and great degree of configurability of JRun enables ISVs to integrate a powerful J2EE 1.3 compatible application server into their product installation without a large amount of incremental overhead. With an elegant JMX service-based architecture, JRun offers a highly customizable and easily extensible deployment platform. For these reasons and more, JRun is also the best solution for ISVs that want to embed a J2EE application server.

With the new Jini-based clustering capabilities, JRun lets you scale your applications in a reliable, high-performance environment. New developer productivity features, such as automatic deployment, hot modification, the Enterprise Deployment Wizard, and XDoclet integration, help you speed up development and deployment of J2EE applications with the most intuitive, easy-to-use J2EE application server on the market.

Finally, Macromedia provides high-quality technical support for JRun, which many of the open source alternatives do not offer. Priced at a fraction of the leading J2EE application servers, JRun delivers an affordable J2EE-compliant solution that meets the cost, performance, and productivity objectives of leading ISVs.

For additional information on how you can leverage JRun as an ISV, see the *JRun SDK Guide*, available at: [http://livedocs.macromedia.com/jrun4docs/JRun\\_SDK\\_Guide/contents.htm](http://livedocs.macromedia.com/jrun4docs/JRun_SDK_Guide/contents.htm)