



Sybase® EAServer 4.0

The Application Server for Every Enterprise.

open

Table of Contents

Objectives of This Guide	1
Sybase EAServer Facts at a Glance	2
New Features of Sybase EAServer 4.0	3
Overview of Sybase and the e-Business Division	4
The Sybase e-Business Division	4
The New Challenges and Requirements for Distributed Applications	
Development and Deployment	4
The Advantages of Application Servers	5
Application Server Requirements	5
<i>Productivity</i>	6
<i>Flexibility</i>	6
<i>Enterprise Support</i>	6
Introducing Sybase EAServer 4.0: The Cornerstone of Scalable Web	
and Distributed Applications	7
Sybase EAServer Architecture	8
<i>Dynamic Content Page Server</i>	8
<i>Business Component Server</i>	8
<i>EAServer Administrator</i>	9
Clustering	9
<i>Load balancing</i>	9
<i>Fail over</i>	9
Component Interoperability through CORBA	9
Full J2EE 1.3 Support	10
Security	11
OEM Installer	11
Web Server Plug-ins	11
Other Key Features	11
The Different Editions of Sybase EAServer 4.0	11
EAServer 4.0 Advantages	12
Openness and Versatility	12
Time to Market	12
Reliability	12
Security	12
Sybase EAServer in Action: A Scenario	12
High Volume Transaction Support	13
Growth/Staying Power	13
The Top Ten Benefits of Sybase EAServer 4.0	13
Competitive Comparison	14
Criteria	14
Sybase EAServer	14
BEA WebLogic®	14
IBM WebSphere®	14
BEA WebLogic Strengths and Weaknesses	15
Sun iPlanet™ Strengths and Weaknesses	15
IBM WebSphere Strengths and Weaknesses	16
Technical Support	17
Appendix A: Features at a Glance	18

Objectives of This Guide

Thank you for your interest in reviewing Sybase EAServer. In the following pages, we hope to acquaint you with the product—including the many new features for building, deploying, and managing n-tier distributed applications—and provide you with background to facilitate your evaluation.

In this Guide you will learn about many key aspects of Sybase EAServer, including:

- Market positioning
- Differentiators
- Key components
- Competitive positioning

While this document covers the product in some detail, it is not intended to be a replacement for the technical documentation. Careful reading and use of the technical documentation is an integral part of the product and should become a tool for your evaluation. This Guide assumes that you have some familiarity with the following technologies:

- Component and object models
- Application servers
- Database terminology, structures, and usage
- Internet and intranet concepts, including World Wide Web servers and browsers

Before you begin your evaluation, be sure that your test system meets the following recommended minimum requirements:

Platform	Min. RAM	Min. Disk
Microsoft® Windows NT™ 4.0 Service Pack 5	256 MB	1 GB
Microsoft Windows 2000 Service Pack 2	256 MB	1 GB
Sun® Solaris™ 2.6 (with patches) Sun Solaris 2.7 (with patches) Sun Solaris 2.8 (no patch required)	256 MB	1 GB
HP-UX® B11.0 (with required patches)	256 MB	1 GB
IBM AIX® 4.3.3 (with 4.3.3 Fixe Package)	256 MB	1 GB
Red Hat® Linux 6.2	256 MB	1 GB

Sybase EAServer Facts at a Glance

Languages	Java™, PowerBuilder®, C, C++, Microsoft Visual Basic®, other languages supporting COM IDispatch
Component Models	C, CORBA®, C++, CORBA Java, Enterprise JavaBeans™, COM/ActiveX®, <i>Native PowerBuilder</i>
Supported Clients	CORBA C++, CORBA Java, Enterprise JavaBeans, COM/ActiveX, <i>Native PowerBuilder</i> , database clients using patented MASP technology (Methods As Stored Procedures).
Web Servers	Built-in Web Server supports static pages, J2EE™ Web Applications and PowerDynamo™ pages. <i>Other Web Servers:</i> Load balancing proxy plug-in provided for forwarding Web Application requests into EAServer from Netscape 3.6, iPlanet™ 4.1, Apache 1.3, and IIS. Also Web Servers can act as C++, COM or Java clients to EAServer (e.g. NSAPI/ISAPI plug-ins or ASP/LiveWire scripts).
Distributed Protocols	<i>Objects:</i> IIOP, IIOP/SSL <i>Web:</i> HTTP, HTTPS <i>Database:</i> TDS, TDS
Platforms/Processors	Microsoft Windows NT and Windows 2000® Sun Solaris HP-UX IBM AIX Linux <i>All platforms support symmetrical multiprocessor implementations.</i>
Data Access	JDBC ODBC Native Sybase Oracle® OCI 7, 8 EJB 1.1/2.0 Container Managed Persistence
Security	SSL 3.0 X.509 v3 RSA Entrust <ul style="list-style-type: none">• Automatic HTTP tunneling for IIOP clients enabling the use of IIOP over the Internet, even in the presence of proxies and firewalls.• Integrated OS security• Authentication and authorization can be delegated to user-provided plug-in components for effective integration with enterprise security systems Authentication and authorization caching for performance
Naming	<i>CORBA:</i> CosNaming <i>Java:</i> JNDI <i>Storage:</i> LDAP

New Features of Sybase EAServer 4.0

If you're familiar with Sybase EAServer, you might want to know what's different in version 4.0. Sybase EAServer introduces several new major features in release 4.0, including:

- **J2EE 1.3 Support** – Sybase EAServer is a complete Java application server with support for the latest version of Java 2 Platform, Enterprise Edition (J2EE).
- **Web Server Plug-Ins** – Web server plug-ins for Apache, Netscape, iPlanet, and Microsoft® Internet Information Server reside on the Web server and pass JSP and servlet requests without requiring a third-party engine or Web server.
- **Unified Installer** – A “silent installer,” designed for OEMs who want to bundle/embed EAServer with their unique products, automatically installs and configures the server and allows for transparent deployment of components from the command line.
- **Transaction Server Enhancements** – Several manageability enhancements, including repository versioning with full backup and restore capabilities, the ability to run C++ components in an external thread for stability, and full clustering support for heterogeneous platforms
- **Performance Tuning** – Administrators can tune their systems with object and page caching.
- **Hot Spot VM Support** – JDK 1.3 includes Hotspot VM support to improve memory management through better garbage collection and to improve performance in many CPU-intensive applications.

Overview of Sybase and the e-Business Division

The universal access and low-cost connectivity of the Internet is driving a fundamental transformation in business. Today, enterprises around the world are looking to reduce costs and increase the appeal and effectiveness of their business models by moving to a system of universal self-service for customers, partners, and even employees. These self-service environments, or enterprise portals, are electronically linking customers, suppliers, and employees around the world creating, in effect, a new “Connected Economy.”

To participate in this new economy every part of your business—and the associated applications—need to be accessible from the Internet. Delivering this universal accessibility, while maintaining near-absolute levels of performance and reliability, is the major challenge facing the IT industry today. Not only must you provide new functionality over the Internet, but you must also provide access to existing functionality to provide a complete self-service environment.

The Sybase e-Business Division

In response, Sybase’s e-Business Division helps companies build enterprise-Web solutions to transact business and communicate efficiently with customers, partners and employees. The Division provides enterprise-portal, application server, integration server, and development-tool software that connects Web operations to disparate, back-office information systems.

The New Challenges and Requirements for Distributed Applications Development and Deployment

For enterprise and commercial software developers today, distributed computing architectures pose several significant challenges. In just a few years, Web technologies have rapidly evolved from simple static delivery of hypertext pages to more interactive two-tier applications using CGI, DHTML, applets and servlets to today’s robust distributed Web applications that require a broad set of tools and services to deliver sophisticated business solutions over the Web.

But this increased sophistication also brings greater complexity and significant challenges. Distributed architectures are much more complex to build. There are more and varied integration and architectural issues to take into account, each with significant implications on the outcomes.

A survey of large enterprises showed that distributed systems spurred a dramatic improvement in development productivity—as much as 80 percent—through object-oriented technology and component reuse. The catch: it took anywhere from three to five years to build up a repository of reusable code.

Distributed applications continue to be very popular with companies that want to Web-enable their applications to make them more accessible and easier to manage and maintain. The Web is not conducive to traditional client/server “fat clients.” The entire Web paradigm is predicated on thin-client (or even zero-footprint) browser clients. As a result, traditional client/server “fat” clients had to lose weight—fast. That fat—or business logic—has to go somewhere else: an application server.

The Web client should have only presentation logic and form validation. The core business logic should be located in the application server as reusable components.

All application servers provide a set of services that simplify the complexity of distributed computing. These services generally include transaction management, result set handling, multi-threading, security, and more. Some of these services are driven by standards organizations.

The Advantages of Application Servers

Stateless Component Lifecycle – One of the major features and advantages of an application server is instance pooling. When a component is created or instantiated, it is a very expensive process—you want to do it only once. After the component is created, it is activated, the method requested is invoked, deactivated and placed into the server’s instance pool. The next time the component is requested, it is activated from the pool, not re-created, saving considerable time. This tremendous performance enhancement is known as a stateless component. When a client creates a stateful component, it holds onto the component while making multiple method requests. This not a scaleable solution because the stateful component holds on to server resources for longer periods of time. The goal is to create stateless components as often as possible, minimizing the use to statefull components.

Connection Caches – Similarly, establishing a database connection is a very expensive operation for the database. Application servers can use connection caches—persistent connections to a database—to manage this challenge. You can create and maintain as many caches as you need and each cache can be configured for the number of connections needed and type of driver used. When a component needs a connection, it requests a free connection from the cache, makes its request, and releases the connection back to the cache. With the Web, developers must support hundreds or even thousands of concurrent users. The connection cache helps to service all equally and quickly.

An important note: connections from a cache use a connection user not the login user. If you are authorized to access the server and a component, you are implicitly authorized to access the connection cache. However, sometimes a user needs to “become the user on the database side.” Sybase facilitates this through support for ANSI-standard proxy authorization. This allows a database user to be tunneled through the connection cache and become that user on the database side, enabling the user to take advantage of database privileges and audit logging.

Transaction Support – Like databases, application servers use the concept of component transactions. A component can call another component which can call another component—all submitting queries to a database as a single unit of work. But what happens if a component fails? Rolling back transactions in different components can be a difficult challenge. Fortunately, application servers provide a transaction manager. A component transaction can either be successful or failed. In either case the transaction manager handles committing or rolling back the component transaction unit of work.

Security – Since the client does not talk directly to the database in an distributed architecture, security must be handled at the application server level. Most application servers use SSL, digital certificates, and operating system authentication. Components are granted execute access to roles or individuals and the application server becomes the gatekeeper.

Application Server Requirements

Sybase believes the best development tool and application server technologies must merge the strongest features of two IT domains: the Internet and traditional business computing technologies. This includes the reliability, availability, and scalability of two-tiered client/server computing as well as the universal access and rapid development of the Web. At the same time, the integrated and open environment maximizes investments in both people and systems while building for the future. These requirements can be described as follows:

Productivity

You need to reduce cost of development, maintenance, and time to market through a high-productivity development paradigm. This implies:

- The ability to use your existing skills/code—including PowerBuilder and Java support to both maintain your existing applications and develop new ones
- RAD development, deployment, and management of sophisticated Web business applications
- The ability to reuse components among various Internet and client/server applications

Performance and Reliability

You need to achieve excellent performance at low cost and to meet the unpredictable demands of the Internet with enterprise-class reliability. This translates into:

- Great performance with features such as native multi-thread support, tuning, caching, and connection pooling
- Unlimited scalability from CPUs to machine clusters
- Automatic deployment and management of applications across multiple application servers
- Ability to add C/C++ components for ultimate performance of mission-critical components
- In-process, high performance between components of different types
- Built-in reliability, scalability

Security

Protecting your business and your customers in the wide-open environment of the Internet is a non-negotiable requirement. It encompasses:

- Integration with security industry standards, like Entrust, Verisign, RSA
- End-to-end security—client to integration with your back-end
- Declarative security; roles
- Authentication, authorization, encryption, and international flexibility

Flexibility

Heterogeneous systems need flexibility at all levels, including:

- Flexibility of client support—Web, Java, Windows, Mobile
- Open, native database and back-office application support
- The ability to wrap multiple language/component types in a CORBA interface
- Support for key standards: operating systems (NT, Unix), databases (native interfaces, wide range of back-end integration with EnterpriseConnect™), languages (Java, PowerBuilder, C/C++ and more), Web standards (Netscape Navigator/Internet Explorer, HTML/XML, Java), enterprise standards (CORBA - IIOP and OMG IDL 2.2), directory standards (CORBA CosNaming, Java Naming and Directory Interface [JNDI], and LDAP), and security standards (SSL, x.509 digital certificates, 128 bit cipher suites)
- The ability to integrate and extend your existing infrastructure and provide a solid foundation to deal with future changes and additions

Enterprise Support

- Experience with servers, distributed applications, development, heterogeneous environments and connectivity
- Enterprise support (7x24), education, consulting services
- Strong partnerships with worldwide availability.

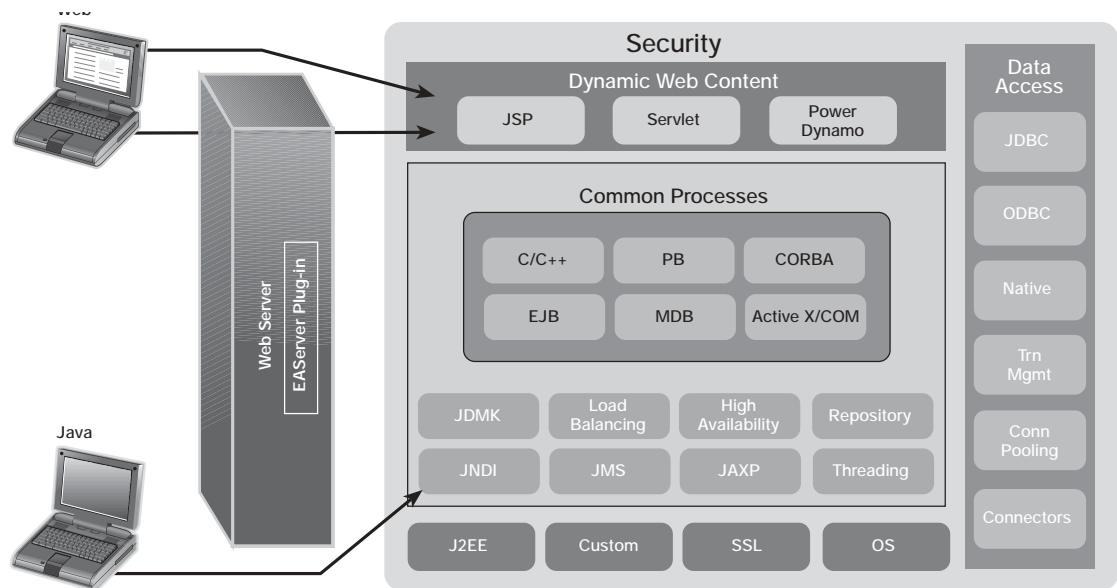
Introducing Sybase EAServer 4.0: The Cornerstone for Scalable Web and Distributed Applications

Sybase EAServer is the highly scalable, robust, fully supported J2EE application server for e-portal and Internet business solutions. It provides industrial-strength services for deploying Web and distributed applications using core J2EE standards, including Enterprise JavaBeans (EJB). EAServer also offers cross-client and cross-component support for almost any type of application, including those based on CORBA, XML, HTML, DHTML, COM/ActiveX, PowerBuilder, C, and C++. Sybase EAServer provides a single point of integration for heterogeneous back-office systems and extends customers' businesses to the Web. It provides best-of-class transaction and security management and built-in load balancing, failover, and high availability, enabling customers to support highly trafficked Web sites.

- **The Target Market** – EAServer is ideal for mid-to-large size enterprises requiring an enterprise-class open application server that ties together existing as well as new heterogeneous Web components and applications. These enterprises recognize the critical market and ROI requirement to extend their current method of conducting business to a comprehensive and open e-Business Web model.
- **The Value Proposition** – With Sybase EAServer, companies can quickly develop, deploy, and integrate a secure, scalable, and available application environment that takes advantage of all component models, including J2EE, COM, and CORBA.
- **The Positioning** – For mid-to-large size companies, Sybase EAServer is the only comprehensive, open, enterprise-class e-Business application server that provides standards-based solutions and integrates application development and deployment of complex transactions processing—all in one seamless environment.
- **The Key Messages** – Sybase EAServer is the open platform for conducting business communication with complete, open application support, deployment, development, transaction processing, and integration of critical applications inside and outside the enterprise. This next generation EAServer seamlessly provides:
 - J2EE, CORBA, COM, and C/++ heterogeneous development and deployment within a single application server environment.
 - **J2EE leadership** – EAServer is the first to market with J2EE 1.3 support.
 - Scalable, reliable, and high performance for always-available applications providing communication within and between businesses environments.
 - Seamless application integration within and beyond the enterprise to leverage the value of existing information assets.
 - Automatic deployment foundation for partner applications, including built-in international/globalization support for OEMs and IT organizations.
 - The most comprehensive support for open industry standards for A2A, B2B and B2C.

Sybase EAServer Architecture

EAServer is composed of several major pieces.



Dynamic Content Page Server

EAServer includes a built-in Web server that can be used as your primary Web server or in conjunction with other Web servers such as Microsoft Internet Information Server, Apache, or Netscape. The Web server is also needed to support the Web 'plug-ins' that support HTML dynamic content.

There are three ways you can support dynamic content: using server-side JavaScript[™], JavaServer Pages[™], and Java Servlets. Each of these techniques can access databases directly or through the component server. Each of these technologies request information and process it into HTML. Today, the most widely used technique is using JSP and Servlets.

Business Component Server

This is where you place your business logic as components. EAServer supports a multi-component model, including PowerBuilder, EJB, Java, C, C++, and COM/ActiveX objects. This allows shops with a mix of developers to build in the language they understand and have all the pieces interoperate. CORBA provides for language-neutral components in the server.

For example, a PowerBuilder component can talk to Java component and vice versa. As a developer, you do not need to know what the component language is. All components in the server are just that, components that can be used to handle the business logic. When you look at all the components in the server, you don't need to concern yourself with what language they are written in. The J2EE services provides the EJB container and services to support EJBs. EAServer is a J2EE fully supported server. The component server engine provides all the services for transaction management, security, connection and session management, and more.

By putting your business logic in EAServer, you can achieve considerable component reuse. Any client that can access EAServer can use the business components it contains, directly or indirectly. The central location of business logic in EAServer means you have one place to enter or change business logic that affects any number of applications that use those components, dramatically simplifying application updates and maintenance.

In addition, EAServer is database neutral and can connect to any data store that supports ODBC, JDBC, native Sybase, and native Oracle.

EAServer Administrator

This provides for the management of all aspects of EAServer. It enables administrators to configure server and component properties, manage server security, and monitor server performance. You can also use it to import and configure components built with other tools. Most management operations can be changed without restarting the server.

Clustering

After components are created and tested for functionality, they are typically moved to a test platform for unit testing with the system being created. Once the testing is completed and signed off, the application/components are moved into production, often on multiple servers. EAServer clustering enables multiple servers, located anywhere on the network, to be logically grouped together. This allow for two things to occur: load balancing and failover.

Load balancing

This feature enables user sessions to be distributed over the cluster for better resource utilization, performance, and scalability in high-volume environments. All the servers' components and Web applications in the cluster are synchronized from the primary server. This component replication synchronizes all of the servers and ensures all are running the same components at the same version. Sybase provides algorithms for dynamic load balancing:

- Random (even distribution)
- Round Robin (even distribution)
- Manual Weighted (random, proportionate to weights)
- Load Weighted (random proportion to [manual weights] * [Load Metrics])

Failover

If a server goes offline, the other servers in the cluster will pick up the load. User sessions that are in process will transparently fail over to one of the other servers.

Component Interoperability Through CORBA

Since EAServer supports multiple component models, it is important that all the various components—in different languages—be able to communicate and interoperate with one another seamlessly. CORBA 2.3 provides this language neutrality through CORBA wrappers (Stubs and Skeletons) and a common Interface Definition Language (IDL).

When you deploy a component, it gets placed in a CORBA wrapper that lets the component function in a language-independent manner. EAServer will create IDL wrappers either automatically at deployment or through a wizard.

Stubs (client-side proxies) and skeletons (server-side proxies) are the remote controls or proxies for a client to communicate with a server component. For example, the stub knows how to take parameters of a PowerBuilder client and call and pass the parameters to the server components in Java. This process is also known as marshalling. All components in the server are simply components—without regard to language. The stubs and skeletons take care of the 'communications' between the two objects.

Full J2EE 1.3 Support

Sybase EAServer fully supports the Java 2 Platform, Enterprise Edition (J2EE) 1.2 standard and will support J2EE 1.3 when it is finalized. As a J2EE application server, Sybase EAServer must pass a 5,000-test compliance text suite. J2EE dramatically enhances enterprise and Web application development, deployment, and management in several important ways.

By supporting J2EE, EAServer insulates application developers from the complexities inherent in writing large-scale applications—such as transaction management, lifecycle management, resource pooling—by supplying those services in the application server, rather than in the application.

J2EE also effectively partitions system administration from development, allowing for maximum system configurability and application portability. EJB containers associate platform-provided services with specific components, allowing services to be configured at deployment time rather than be hard-coded into the application.

J2EE enhances developer and application performance by tailoring components to the types of functions they implement.

- **Enterprise JavaBeans (EJBs)** – These contain the business logic of the middle-tier transaction servers. They consist of 1) session beans that control client sessions (such as a user purchase transaction) and 2) entity beans that represent collections of data, such as rows in a relational database, and encapsulate operations on the data they represent. EAServer 4.0 supports the EJB 2.0 specification, supporting message-driven beans and enhancements to container-managed persistence.
- **Servlets** – Java programs that produce dynamic Web (HTML, XML) pages. Employed in middle-tier Web servers, they are best suited for advanced Java developers wishing to implement more complex functionality. Servlets can call EJB methods.
- **JavaServer Pages (JSPs)** – Static HTML and XML pages containing snippets of Java code, JSPs also reside in Web servers, are easier to program than servlets, yet have most of the capabilities of servlets.

J2EE provides portable connectors to back-end databases and vendor-specific enterprise applications, thereby giving J2EE applications the flexibility to “plug in” to a variety of services. J2EE supports a variety of client-side implementations, including HTML, XML, Java applets, and Java applications.

In addition, J2EE provides “one button” deployment of complete applications as multi-component packages with built-in deployment descriptors that specify how components are to be configured and assembled across the network. Other key Java support in EAServer 4.0 includes:

- **JAXP** – Java API for XML Parsing
- **JAAS** – A Java-based security interface allowing for pluggable authentication components
- **JMS 1.0.2** – EAServer supports the standard Java messaging interface as a wrapper on top of existing service.
- **Connectors (JCA)** – This common access methodology lets you pool connections into legacy systems such as SAP and PeopleSoft®. It's similar to cache connections.
- **JTS** – Java Transaction Service
- **JDMK API** – This provides access to Java applications through standard SNMP monitoring systems.

Security

EAServer's Security Manager handles the extensive security features. You can use native operating system username/password, x.509 digital certificates, or if you have a security mechanism in place, you can use the user-defined hook to access it. EAServer also supports SSL from HTTP requests.

EAServer supports role-based security. You can assign logins to roles and roles to objects in the server. You can set roles at the package, component, or method within a component. You can explicitly exclude logins as well.

Unified Installer

Version 4.0 of EAServer introduces a "silent installer" designed for OEMs who want to bundle/embed EAServer with their unique products. Through this feature, EAServer automatically installs and configures the server to match an existing file and allows for the transparent deployment of components from the command line.

Web Server Plug-Ins

EAServer has Web server plug-ins for Apache, Netscape, and Microsoft Internet Information Server. These plug-ins reside on the Web server and pass JSP requests through to another machine. This enables the Web server to run on one machine while EAServer runs on a second machine without requiring a JSP engine on the Web server.

Other Key Features

- Declarative page caching for JSPs/servlets
- Declarative object caching for entity components
- Declarative method (result) caching for all component types

The Different Editions of Sybase EAServer 4.0

- **Enterprise Edition** – This is ideal for very large organizations requiring robust transaction-processing and high-end capabilities such as access to CICS applications, two-phase commit, fault tolerance, and 24x7 operation.
- **Advanced Edition** – Similar to Enterprise Edition, this features unlimited connections and clustering with scalability for high performance. It does not include access to CICS applications, two-phase commit, or fault tolerance.
- **Developer Edition** – Designed for developers seeking to build and test EAServer applications, this edition is full featured, but restricted to 5 IIOP connections.
- **Small Business Edition** – This edition is targeted at information-publishing applications. A maximum of 10 active transaction-server connections are supported. It's appropriate for organizations not requiring clustering, mainframe database connections, two-phase commit, or fault tolerance.

Sybase EAServer in Action: A Scenario

A large financial institution has just acquired another top 10 financial services firm and plans on integrating and deploying its combined applications on the Web. The acquiring financial institution has implemented its critical applications running in a Java J2EE environment and the acquired institution has a Microsoft.NET environment.

The success of the acquisition strongly depends on a six-month integration timeline that was imposed on the acquiring institution's IT department. Traditionally, successful integration requires the organization to possess skillsets across platform, components, and technology. In this scenario, the acquiring institution did not have the .NET skillsets and through the acquisition and attrition, the acquired company no longer has the supporting .NET skills. The acquiring IT company faces a major challenge and the integration timeline is severely jeopardized.

Sybase EAServer—enabled with cross component development and deployment—addresses this dilemma. Existing IT staff can use their current J2EE skills along with the acquired organization's .NET talent in one seamless application server. The existing tools are still utilized for effective development and deployment, and with effective modeling capabilities, designs are put in place to utilize “Best of Breed” processes and create maximum productivity in the integration effort.

Sybase EAServer offers a range of choices:

- Utilize one application server to deploy all components while making only minimum changes,
- Migrate to new corporate initiatives for a new component over a more cost effective timeline,
- Develop all components with existing or new initiatives on one common application server platform.

The new combined financial institution decides to choose and schedule its own direction; the EAServer can provide an application deployment environment; and cross-component and platform development allows timelines to be met with minimal added cost, minimizing reduction downtime and achieving maximum ROI.

EAServer 4.0 Advantages

Openness and Versatility

Sybase EAServer is the industry's most open and flexible Web application server. It supports an exceptionally broad range of standards including:

- J2EE 1.3
- Major programming languages, including PowerBuilder, Java, C/C++, and Visual Basic
- Major component models, including CORBA, ActiveX/COM, Enterprise JavaBeans, Java Servlets, and PowerBuilder non-visual objects (NVOs)
- More than 30 DBMSs for complete integration of back-end and legacy systems

As a result, a development team can implement EAServer without having to learn new languages or adopt new component models. What's more, this enables the team to build a new generation of more sophisticated distributed applications quickly through code/component reuse. Collectively, these benefits translate into a lower cost of ownership, lower investments in IT infrastructure, faster time-to-market, and greater application reliability.

Time to Market

One of the most important and relevant metrics for the success and value of a development paradigm is the speed with which developers can create and deploy their applications. EAServer fosters an environment for very rapid development and deployment of Web applications through minimal coding, reuse of existing code, and a short, low learning curve.

What's more, EAServer promotes rapid and simplified application maintenance through centralized management of business logic. That means enterprise IT can support new business initiatives faster, help the enterprise reach its market faster, and create and sustain competitive advantage.

Reliability

Today's Web and distributed applications are no longer optional, non-strategic systems. Today, these mission-critical applications drive the enterprise. To respond to this mandate, Sybase EAServer provides the reliability characteristics and services that enterprises require:

- High-performance clustering
- Load balancing
- Failover
- Caching

As a result, EAServer protects the enterprise from the risks of costly downtime while supporting high-volume Web traffic 24 x 7.

Security

Another key advantage of EAServer is its comprehensive security features. Through its support of HTTPS, Secure Sockets Layer (SSL) 40- and 128-bit encryption, and x.509 digital certificates, EAServer ensures the privacy of sensitive data.

High-Volume Transaction Support

EAServer provides an exceptional level of support for high-volume transaction applications through robust transaction management support and two-phase commit. Enterprises can leverage existing back-office systems with a single integration broker for excellent integration and interoperability. EAServer includes built-in support for industry-standard transaction monitoring through DTC, XA, OTS, JTS, and EJB. In addition, EAServer provides support for multiple database commits through a single unit of work. Now, enterprises can leverage and extend existing enterprise systems through a single point of integration and control.

Growth/Staying Power

Sybase EAServer provides a core infrastructure for Web and distributed applications that enables you to extend the power of your IT portfolio to a broader range of internal and external customers. Sybase ensures and preserves investments with an open platform that is enhanced and extended through a network of solutions from Sybase and its certified partners.

The Top 10 Benefits of EAServer

- 1. Achieve Rapid Time-to-Market** – Sybase EAServer cuts deployment time and simplifies e-Business by managing the complexity of multi-tier computing.
- 2. Lower Development Risk** – Sybase EAServer's unique support of industry standards ensures that applications can be extended and enhanced over the long term to stay in step with your business and with contemporary computing environments.
- 3. Leverage IT Investments in Skills and Resources** – Sybase's open component support enables developers to get their e-Business applications to the Web using the skills they already have.
- 4. Increase Developer Productivity Through J2EE Support** – Java developers spend less time coding infrastructure and more time building differentiating functionality because EAServer fully supports J2EE.
- 5. Accelerate Development Cycles** – Built-in wizards add drag-and-drop functionality, simplifying the development, deployment, and debugging processes.
- 6. Secure and Protect Digital Assets** – Sybase EAServer provides a comprehensive set of security features to control user and data access.
- 7. Leverage Investments in Legacy Applications and Platforms** — Enterprises can easily integrate data from legacy and back-end systems, making information available to new generations of e-Business applications.
- 8. Integrate with Emerging Technology** – As the recognized leader in wireless computing and enterprise portals, Sybase provides a breadth of products that extend EAServer to a complete platform, giving developers an added advantage.
- 9. Minimize Total Cost of Ownership** – EAServer is very competitively priced and packaged in several bundled configurations.
- 10. Deploy with Confidence** – Sybase brings more than 6 years of proven experience in developing enterprise servers for mission-critical environments. In addition, Sybase backs its solutions with one of the industry's most experienced and respected service organizations.

Competitive Comparison

Criteria	Sybase EAServer	BEA WebLogic®	IBM® WebSphere™
J2EE Leadership?	• Yes	• Yes	• No
Open Standards Adherence?	<ul style="list-style-type: none"> • Fully J2EE supported • All major component models: EJBs, COM, CORBA, PowerBuilder • All major languages, plus HTML, XML, and JavaScript for Web 	<ul style="list-style-type: none"> • J2EE supported • EJB is <i>only</i> component model supported 	• Only selective support of J2EE
All major component models supported?	• Yes	• No	• No
Support mixed component models in a single application?	• Yes. For example a PowerBuilder NVO can interoperate with an EJB or COM component	• No	• Limited
Full Database Transaction Support?	<ul style="list-style-type: none"> • Supports mixed component models in a single application • Multiple databases supported; connection pooling for Sybase, Oracle, and any JDBC or ODBC database 	<ul style="list-style-type: none"> • EJB-centric transactions, no native CORBA support, COM support invoked only indirectly • Fewer databases supported 	<ul style="list-style-type: none"> • Only DB2 and Oracle support for storage of object and configuration data. • Connection pooling for DB2 and Oracle only
Integrated Development Tools	• Yes	• No	• No
Ease of Deployment?	• Wizard-based approach enables you to deploy, debug, and configure components inside EAServer without leaving IDE	<ul style="list-style-type: none"> • Requires third-party development environments • But Visual Café is for Java only. Visual Café and WebLogic are on separate commercial cycles – impossible to integrate 	• Must use at least two separate tools: VisualAge for Java and WebSphere Studio.
Execute existing PowerBuilder NVOs?	• Yes	• No	• No
Leverage Existing Development Skills?	• Re-use PowerBuilder components. Run them side-by-side with Java, C++, COM components	• No re-code in Java	• No
Built on 6 years of middleware success?	• Yes	• Yes	• Yes

BEA WebLogic Strengths and Weaknesses

Strengths:

- Mindshare
- Established user base
- Time-to-market advantage of WebLogic Server with pre-J2EE technologies
- Ability to grow through acquisition

Weaknesses

- **Limited Tools Integration** – With WebLogic, you cannot redeploy EJBs and the server requires constant restarts. The application unit, NOT the EJB, is a deployable module and must be assembled by the developer through hand-written scripts without any assistance from WebLogic.
- **Not Scalable** – BEA's performance literature states that "there are no database I/O costs included in each of the above results, and hence they should not be used for capacity planning." However, an application server must perform database transactions. WebLogic requires and assumes clustering to prevent poor-performance. All WebLogic servers must be on the same single network segment.
- **Lack of Multiple Component Support** – Developers need more than EJBs in their arsenal. They need to extend their existing skills and reuse their existing non-EJB components.
- **Separate Codelines** – WebLogic Server and WebLogic Enterprise are two entirely separate codelines. Extending WebLogic Server with WebLogic Enterprise is extremely difficult.

Sun iPlanet Strengths and Weaknesses

Strengths

- Created by Sun Microsystems, creator of J2EE
- Reputed to be highly scalable
- Includes IBM's Encina transaction monitor
- Includes run-time LDAP-enabled iPlanet Directory Server and uses a C++ data engine for boosting performance

Weaknesses

- **Poor Integration** – Uses different administrative consoles to manage the Web and LDAP servers, and another to manage J2EE services
- **No Development Tool Integration** – iPlanet is strictly a run-time environment with limited integration through plug-ins to development tools.
- **Difficult Installation** – Especially under Microsoft Windows
- **High Price** – iPlanet is not a good value.
- **No Support of Multiple Components** – No COM/ActiveX or C/C++
- **One Edition Only** – No flexibility of bundles
- **No Debugger**
- **Poor Alliances**

IBM WebSphere Strengths and Weaknesses

Strengths

- Huge marketing budget and mindshare
- Existing, loyal customer base
- Excellent reputation for service
- Integration capability with mainframe applications through TXSeries transaction processing
- Deep discounting to buy market share

Weaknesses

- **Confusing and Conflicting Product/Platform Messages**
- **Limited Tools Integration** – You need three different products: VisualAge, VisualAge Rules for Java, and WebSphere Studio
- **Lack of J2EE Branding**
- **Mainframe Heritage** – IBM has had limited success with most previous non-mainframe offerings. New telco ventures, start-ups, and dot-coms are unlikely to become S/390 users and rarely require IBM implementation services for e-Business infrastructure.

Technical Support

Sybase offers an unmatched set of technical support options to provide thorough, professional, accessible assistance to EAServer users:

- Telephone support is available from Sybase offices. Hours are 8:00 a.m. to 8:00 p.m. Monday through Friday. Telephone support is available in a variety of plans, including pay-per-incident, annual support contracts, and 7x24 coverage.
- Sybase offers a 24-hour fax line for submitting questions and receiving answers.
- Sybase maintains its own support site on www.sybase.com/support. In addition, customers may dial directly into the firm's own BBS.
- A variety of third-party books are available on Sybase products. In addition, independent magazines also provide specific coverage on Sybase issues.
- Sybase User Groups, including the Sybase User Conference, offer the chance to meet with other users, preview new releases, and learn more about the products.
- Technical support is provided by Sybase through local offices worldwide, assuring you of local support near your location.

Appendix A: Features at a Glance

Enterprise Class Performance and Scalability

Scalability and Speed	
Multi-threaded execution engine	<p>EAServer is a highly scalable application engine, featuring:</p> <ul style="list-style-type: none">• A multi-threaded execution engine, with full SMP support• Support for Windows NT, Unix, and Linux for genuine multi-platform scalability• A technology foundation whose history spans more than a decade in distributed computing
Load balancing, failover, high availability	<p>EAServer's built-in load balancing, failover, and high availability features enable customers to support the most highly trafficked Web sites in the world:</p> <ul style="list-style-type: none">• Configure JDBC connections to a Highly Available Systems (failover) feature in Adaptive Server® Enterprise (ASE) 12.0• Built-in software solution – Configure a cluster of EAServers that share replicated applications, files, and configuration information:• Load balancing – Evenly distribute incoming I/O requests• Failover – Provide access to EAServer services, session information, and selected components marked to support transparent automatic failover even when the primary server is unavailable• High availability – Provides access to business components even if server is unavailable• Seamless integration – With industry leading hardware solutions:<ul style="list-style-type: none">• Cisco LocalDirector• HydraWeb HydraSeries
Pool all resources	<p>Sophisticated, high-performance tuning features allow EAServer to handle the most demanding applications the Web has to offer. Some of these features include:</p> <ul style="list-style-type: none">• Instance pooling that enables EAServer to reuse component instances across client connections• Thread pooling, where EAServer maintains a pool of threads for component execution• Efficient HTTP session management that EAServer uses to provide high-performance interaction with Web servers.
Database connection pooling	<p>Database connection pooling helps control the impact of Web applications on existing operational systems, by eliminating the time-consuming and resource-intensive task of creating and destroying new database connections. Some additional features include:</p> <ul style="list-style-type: none">• Pooling for Sybase, Oracle, JDBC, and ODBC• New connections created if the cache is full• Direct database connections still permitted• Proxy connection support in Adaptive Server Enterprise that allows uniquely identified users to use pooled connections• Fine levels of control over the connection pool that enable setting minimum/maximum number of connections and setting connection timeouts to eliminate unused database connections.

HTML page caching & scheduling	<p>EAServer provides HTML page caching and scheduling that eliminates constant reprocessing of frequently used dynamic pages and allows for scheduling of dynamic pages during non-peak hours:</p> <ul style="list-style-type: none"> • HTML output can remain in cache for any specified period of time—from a few minutes to several days. • Dynamic pages can be processed at times when the server load is typically low, such as the middle of the night.
--------------------------------	--

Enterprise-class Services

Enhanced naming services	<p>EAServer enhances object location transparency and interoperability among components:</p> <ul style="list-style-type: none"> • Java Servlets, JSPs, and EJB components now use aliased JNDI names to refer to EJB home instances, database connections, JavaMail servers, and environmental properties (e.g., to tune cache usage or specify the name of a logging file). • Both transient and persistent storage are supported (persistent storage is implemented via LDAP)—simply a configuration issue. • CORBA CosNaming Service is supported.
--------------------------	--

Asynchronous Message Service	<p>EAServer enables clients and components to send and receive messages. The message service provides a pull-style mechanism for client notification and a push-style mechanism for component notification. Clients can check their queues for new messages. Components can also check for new messages and they can register to be notified when messages arrive in their queue:</p> <ul style="list-style-type: none"> • Message service is implemented as an EAServer component with interfaces specified in standard CORBA IDL—consequently it can be used by all types of clients and components. • You can administer the message service using either the service's graphical user interface or the service API. • Server clustering can provide high availability and load balancing of message services. • Role-based security is implemented for message queues and message topics. • Messages can be published to receivers who subscribe to messages with those topics. • Messages can be durable (persistent) or non-durable
------------------------------	---

JavaMail	<p>In EAServer, you can use the JavaMail API to send email from a Java component, JSP, or servlet using any general mail system. For example, you may wish to send confirmation email from an online ordering application.</p>
----------	--

Thread Manager	<p>Use the Thread Manager to implement server-side processing to run asynchronously with user interactions.</p> <ul style="list-style-type: none"> • Taking thread control out of the application avoids system calls that can impact portability. • Combined with Message Service's scheduling feature, Thread Manager can start or stop processes at any time. • PowerBuilder developers can develop services that can be stopped or refreshed, using Thread Manager.
----------------	--

Run-time management and monitoring	<p>EAServer provides easy runtime management, with the ability to:</p> <ul style="list-style-type: none"> • Manage the server • Manage database connections • Monitor runtime activity • Monitor server clusters • View log files
Windows NT Service Support	<p>EAServer can run as a Microsoft Windows NT service, so that your transaction server starts automatically whenever Windows NT restarts.</p>
Highly productive development and deployment	<p>EAServer simplifies the creation and administration of distributed, component-based applications. It supports integrated development for both Sybase tools and today's other popular 4GL, Java and Web tools such as Visual Basic, JBuilder, Symantec, and others.</p> <ul style="list-style-type: none"> • The combination of EAServer and PowerJ® or PowerBuilder provide a comprehensive Integrated Application Environment for building, testing, deploying, and debugging Web, distributed and client/server applications—without ever leaving the development environment.
Easy cross-platform deployment of enterprise applications with EAR files	<p>The J2EE Enterprise Archive (EAR) is the standard for servers that support J2EE, providing an easy way to treat components as a unit. This format allows portability to other vendors' J2EE servers. EAR files include:</p> <ul style="list-style-type: none"> • XML-formatted deployment descriptor • EJBs
Easy cross-platform deployment of Web applications with WAR files	<p>The J2EE Web Archive (WAR) is the standard for servers that support J2EE, providing an easy way to treat a collection of Web resources as a unit. This format allows portability to other vendors' J2EE servers. WAR files include:</p> <ul style="list-style-type: none"> • XML-formatted deployment descriptor • EJBs, JSPs, servlets, applets, and HTML
Enhanced application deployment between EAServer servers with JAR files.	<p>The JAR format preserves all information in enterprise and Web applications, including information not preserved in the WAR:</p> <ul style="list-style-type: none"> • EJB references • Resource references • Role mappings
Integration with PowerJ and PowerBuilder wizards	<p>PowerJ and PowerBuilder provide a wizard-based approach to deploy, debug, and configure components (EJBs, PB NVOs) inside of EAServer without ever leaving the IDE. Point and click your way to distributed computing:</p> <ul style="list-style-type: none"> • Create and deploy EAR, WAR, and JAR files—formats that make it easy to treat collections of diverse application resources as a single unit. • Set properties at the application or component level: role mappings, resource references, EJB references, and environment properties.

Easy-to-use visual management tool to administer and monitor all aspects of EAServer in conjunction with your tool of choice—Visual Basic, JBuilder, Delphi, Symantec

EAServer provides point-and-click administration and monitoring capabilities, freeing developers to focus on the business logic rather than specialized system-level programming. This makes it easy to import, configure, and deploy components from your tool of choice into EAServer.

- Create and deploy EAR, WAR, and JAR files—formats that make it easy to treat collections of diverse application resources as a single unit.
- Set properties at the application or component level— role mappings, resource references, EJB references, and environmental properties.
- Create and configure connection caches within EAServer.
- Mark selected components to support transparent failover. If a client references a component on a server that is a member of a cluster, the client's object reference will provide transparent failover, unless all servers in the cluster fail.
- Import and configure components with the Jaguar Manager EJB, servlets, JSPs, C/C++ DLL's, ActiveX-type libraries—using OMG IDL 2.2 definitions.
- Specify package and component name.
- Specify the class type and interface to use for your component.
- Set implicit transaction management—Not Supported, Supported, Requires Transaction, Requires New Transaction.
- Manage the component's lifecycle—automatic thread and process management—multi-threaded, single-threaded, and private threading models.
- Set component instance and transaction timeout information.
- Set security for packages and components—authorization and authentication.
- Assign access privileges to roles and assign roles to components and methods.
- Automatically generate and compile stubs and skeletons for any component.
- Dynamically refresh servers, packages, components, applications and Web applications directly from the Jaguar Manager (i.e., without shutting down the server).
- The Jaguar manager enables the same functionality that you would find using PowerJ and PowerBuilder with the following exceptions:
 - Deployment into EAServer requires moving between the development IDE and Jaguar Manager.
 - No distributed debugging is possible
 - Jaguar Manager—standalone—does not provide a tool to develop the client- or server-side code (it does generate template implementation class files for Java and C++).

Unified control of objects and other resources in a single environment

EAServer offers a single unified environment in which to deploy, configure, manage, and execute all J2EE objects. Choose the best of both worlds: page computing for ease-of-development and universal access, or component computing for complex reusable business logic integrated with operational systems.

- EAServer now implements version 2.0 of the Enterprise Java Bean specification with support of features such as enhanced JNDI naming services, property references, and XML deployment descriptors.
- EAServer now implements version 2.3 of the Java Servlet Specification. Servlets are Java classes that extend the functionality of a Web server. They respond to HTTP requests from Web browsers or other clients that connect to EAServer using the HTTP protocol for example, to create dynamic HTML page content or to act as a gateway between HTML forms and EAServer Components.
 - EAServer extends the Servlet API so that servlets can use EAServer services such as inter-server component invocations.
 - Request dispatching allows one servlet to invoke another (target) servlet and either forward a request or include the target servlet's response with its own.
 - Response buffering allows the servlet to control how the servlet container buffers responses to a client and when to send a response to a client.
 - Servlets in EAServer can instantiate component instances using the same techniques used with Java/CORBA—or EJB components (where portability to other J2EE servers is required.)
 - Servlets can log error messages or other text to the EAServer servlet log file, using the standard servlet log methods.
- EAServer now implements version 1.2 of the JavaServer Pages (JSP) Specification. Where servlets are Java code containing embedded static HTML and XML, JSPs are static pages of HTML or XML containing snippets of Java code. They require less effort to learn and use than servlets and offer the same portability, performance, and scalability.
 - JSPs inherit the concepts of applications, ServletContexts, Sessions, and requests and responses from the Java Servlets API.
 - JSPs make it easier to partition and maintain an application on multiple servers. The JSP runs on the Web server and can be updated whenever the page designer needs to change elements of the presentation. The components called by the JSP run on the transaction server, or on a cluster of transaction servers, and can be updated whenever the business logic needs to be changed.
 - For maximum flexibility, portability, and performance, JSPs can be compiled at server startup or when the JSP is first called. You can also use a command-line utility to compile JSPs independently of the EAServer server to facilitate, debug and test.
 - JSPs allows separation of application logic from page format and design, making it easy for the Web-page developer to add dynamic content to a Web page without writing Java code.
 - Servlets can log error messages or other text to the EAServer servlet log file, using standard servlet log methods.

- EAServer now supports the creation and manipulation of XML.
 - Use a template wizard to create queries that generate output in either HTML or XML.
 - XML Document Object Model may be used to manipulate XML documents.
 - EAServer seamlessly integrates HTML with transactional business logic. EAServer extends HTML generation with access to component methods, including processing of result sets from method calls.
- PowerDynamo™ templates can call any EAServer CTS component- Java, C++, PowerBuilder NVO, and ActiveX
- PowerDynamo templates can directly call ActiveX components, Java Servlets, C++ objects, or any other executable from a PowerDynamo template.

Synchronization

EAServer synchronization capabilities automate the process of maintaining your applications by minimizing the time-consuming task of deploying your application from testing or development to production. Some of the features include:

- Automated synchronization of application files and configuration information between EAServers
- Automated synchronization at the cluster, server, package, and component levels
- Automated synchronization at the servlet, JSP, application, and Web application levels

Powerful scripting language

The EAServer page server features a scripting language that is a superset of JavaScript—which combines the simplicity and efficiency of a scripting language, yet provides the benefits of an object-based programming language. Because this script is based on an industry standard, it guarantees the longevity and reusability of your business logic, and leverages existing JavaScript development skills.

Benefits of PowerDynamo scripts:

- Scripts can call business components through the following:
 - ActiveX objects (including EAServer proxies)
 - Executable programs
 - Java classes accessible from within scripts including EAServer proxies
- Support for class inheritance
- A Java-like import statement
- A number of predefined objects that make the script a very powerful language for building sophisticated, database-driven Web applications.
 - Connection object for managing database connections, including the database connection pool and temporary database connections
 - Session object for managing client sessions and client specific information
 - Query object for working with SQL query result sets
 - Email objects for sending, receiving, and manipulating email within an EAServer Web site
 - FTP objects to provide FTP functionality to EAServer Web sites.
 - Other predefined objects such as: the site object for managing Web site information, the file object for manipulating files through the file system, the document object for representing a document in your Web site, and the system object for interacting with the server operating system
- Is now ECMAScript compatible. (ECMAScript is the new standard for JavaScript and JScript like languages.)

<p>Easy and powerful database publishing</p>	<p>The EAServer page server offers the quickest and easiest way to deploy Web applications integrated with corporate data. It facilitates database access in conjunction with a Web server and provides an easy way to access databases via the Internet/intranet Web browser environment.</p> <ul style="list-style-type: none"> • Wizards quickly generate HTML templates from SQL queries. • At run-time, EAServer processes these templates against corporate data, returning pure HTML to the client. • Templates embed both business logic and database connectivity within HTML. • Template-based applications are the easiest to design, develop, and maintain.
<p>Sharing data across components</p>	<p>EAServer allows Shared Components for effective sharing of data between components.</p> <ul style="list-style-type: none"> • Multiple concurrent users easily share global resources. • Concurrent access to shared components is fully synchronized.
<p>Service components</p>	<p>Service components are long-lived components that start when EAServer does, and can be shared by any other component. They are used to:</p> <ul style="list-style-type: none"> • Capture live data feeds • Perform data caching • Monitor messaging queues • Provide utility services such as mail, ftp, etc.
<p>End-to-end security</p>	<p>Sybase EAServer includes a comprehensive high-end package of security functions that allow IT organizations to move past their security concerns when developing n-tier applications in a networked environment.</p>
<p>Declarative security</p>	<p>EAServer integrates all major aspects of network security with authentication, access control, encryption, and ease of administration features. By providing these capabilities in a declarative manner, EAServer eliminates the need for the application developer to build these capabilities into the application. Point-and-click your way to end-to-end security:</p> <ul style="list-style-type: none"> • Visual-based Jaguar Manager configures user authentication and authorization, which enables the EAServer kernel—at run time—to authenticate the user and verify authorization to execute components • Visual-based Jaguar Manager defines security profiles, e.g., protocol used (SSL version 3), authentication requirements (server only vs. client and server), and encryption strength (40-bit or 128-bit) • Visual-based Security Manager enables the user to manage certificates, map digital IDs to EAServer roles, and use Entrust certificates within EAServer. This ensures that access to components is authorized, based on client SSL certificates.

Security models

EAServer provides a comprehensive industry standards-based security architecture encompassing all aspects of security—authentication, authorization, and encryption.

- Authentication verifies user identity—EAServer utilizes user name and password pairs, X.509 digital certificates, or Entrust digital IDs to identify users and then leverages SSL, native operating system authentication or custom event handlers to actually authenticate the users.
- EAServer enables caching of session authentication information for a configurable amount of time, which eliminates the overhead of passing the user name/password combination with every request. This provides a significant performance boost over most other implementations.
- Authorization verifies how much or what type of data can be accessed—EAServer uses access control lists based on user defined roles that can be applied at the package and/or component level.
- Encryption of communication is assured on a network using SSL over IIOP and HTTP

Secure Sockets Layer (SSL)

EAServer supports industry-standard mechanisms to secure applications. Security is provided through configuration of the application environment with minimal impact to code. This gives applications the ability to work in a state-of-the-art, secure, Web environment.

- C++ SSL Support means SSL connections are supported from C++ clients.
- PowerBuilder SSL Support means SSL connections are supported from PowerBuilder clients
- ActiveX SSL Support means SSL connections are supported from ActiveX clients.
- Native SSL is provided for browser clients.
- Client and server authentication uses X.509 v. 3 digital certificates
 - RSA PKCS #11 based implementation
 - Integrated SSL protocol driver
 - SSL 2.0 & 3.0 dynamic negotiation
 - Negotiation of export (40-bit) and non-export (128-bit) cipher suites
 - Resumable cached sessions (reduces session re-establishment overhead)

Entrust integration

EAServer supports the award-winning Entrust Technologies public-key infrastructure. By being “Entrust Ready,” EAServer enables corporations to use Entrust public key certificates to authenticate EAServer servers and client connections

Flexible development and deployment

- Open CORBA APIs for Management, Monitoring, and Repository access.
- Command-line tool (jagtool) and integration with Jakarta Ant (jagant) for automated deployment.

Open Server migration

EAServer provides a migration path for existing Open Server applications by providing native threading and ability to run in multiprocessor CPUs. This functionality will improve the scalability of single-threaded Open Server applications. Open Server is installed in tens of thousands of the most demanding sites in financial services, telecom, healthcare, and government industries.

Universal Component Model

Broad server-side component support

EAServer seamlessly and simultaneously supports multiple component models. Components allow maximum reusability for critical business logic and EAServer enables an application developer to easily combine various components in the same application. EAServer supports the following component models:

- Enterprise JavaBeans
- JavaServer Pages
- Servlets
- PowerBuilder non-visual objects
- Java and JavaBeans
- COM/ActiveX including “UNION” datatype
- CORBA
- C++
- C

Universal Repository (using CORBA IDL)

All component interfaces are defined for you with the full power of OMG IDL. EAServer automatically generates IDL by importing existing code or by filling out simple property sheets. EAServer also supports some critical IDL extensions:

- Full datatype support
- User-defined data types
- User-defined exceptions
- Pass object-by-value parameters (for Java components)

Object model transparency through automatic proxy generation

By leveraging OMG IDL and the CORBA Stub/Skeleton architecture to rapidly generate proxies for server-side components, EAServer eliminates object model boundaries allowing for greater interoperability.

- Efficiently separates implementation object model from client-side deployment decision.
- Any EAServer component is callable from any other type of client component. For example, a Java applet could instantiate and invoke methods on a server-side ActiveX component.
- Transparent inter-component calls across object models (no proprietary extensions required)
- Automatically generates slim Internet friendly client proxies for server-side components
- All components run in the same process on native threads. No need to struggle with different servers and multiple process to host business logic implemented in different object models.
- ActiveX clients can now use CORBA-style (ORB< CosNaming, etc.) interfaces to instantiate proxies for EAServer components
- Configure properties for EAServer packages, components, and servlets to associate application files with these entities. For example, you might associate a Java applet and HTML files with a Java component.

Supports virtually any client	<p>EAServer provides support for virtually any type of client, including:</p> <ul style="list-style-type: none"> • Java applets and applications • PowerBuilder • C and C++ applications • Any ActiveX client (no additional software required) • HTML and DHTML • XML • CORBA (e.g., Visigenic, IONA)
Supports ultra-thin (browser-independent) clients	<p>The EAServer page server processes HTML templates on the server so applications are completely browser-independent. These applications can be 100 percent thin-client, meaning only HTML needs to be passed to the client</p>
Integrate the back office	
Any database	<p>Sybase provides broad and powerful connectivity to enterprise databases. EAServer supports:</p> <ul style="list-style-type: none"> • JDBC 2.0, including pooling of connections of JDBC 2.0 data sources • Two-phase commits • Adaptive Server Enterprise natively through CT-Library • Oracle natively through Oracle Call Interface (OCI) enables connection caching for OCI connections to be used by components coded in C or C++ • Any ODBC target • Access to more than 25 different data sources (with full location transparency and support for distributed joins) through OmniConnect,[™] Sybase's industry-leading data access product • High-speed connectivity to mainframes, including CICS, IMS/TM, and MVS-native environments
Multiple communication protocols	<p>EAServer supports multiple industry-standard protocols:</p> <ul style="list-style-type: none"> • IIOP/IIOPS for distributed component invocation and CORBA interoperability • HTTP/HTTPS for built-in Web access • TDS (Sybase Tabular Data Stream) for optimal communication with other Sybase products • All standard protocols secured with SSL
Universal Web server support	<p>EAServer supports any type of Web server in your environment:</p> <ul style="list-style-type: none"> • ISAPI and NSAPI provide in-process support for both Netscape and Microsoft Web servers. • CGI provides for interoperability with all other Web servers. • Built-in HTTP support is included for development and/or mobile deployment. • EAServer now implements version 2.3 of the Java Servlet Specification. Servlets overcome many of the platform dependency and performance issues of CGI, ISAPI, and NSAPI. They do not have to be recompiled for different platforms (like CGI scripts do) and offer better performance for large scale applications because each new CGI request requires a new server process. • EAServer now implements version 1.2 of the JavaServer Pages (JSP) Specification. JSPs offer the same portability, performance, and scalability as servlets.

Multiple platforms	EAServer is available on Microsoft Windows NT, Sun Solaris, HP-UX, IBM AIX, and RedHat Linux. The EAServer dynamic page server is also available on Windows 95/98 and Win32s.
International environment support	Single- and multi-byte character support
Supports highly-distributed deployments	<p>In instances where multiple servers are needed to support the required volume, database replication facilities can be used to automatically keep the entire Web solution (both application and data) up-to-date across any number of servers.</p> <ul style="list-style-type: none"> • The EAServer page server can store the entire application in a database, which is easily replicated, to multiple servers using Sybase's industry-leading replication facilities. • The EAServer component transaction server's package import/export feature supports easy deployment of applications across multiple servers. • Sybase Central provides centralized administration for both the application and database with one interface.
Supports mobile deployments	Where multiple workgroups need constant, reliable access to a Web application, the EAServer page server provides a unique offline solution. With EAServer's page server and Sybase's database and replication facilities, you can replicate the entire solution (both application and data) to each offline workgroup. This guarantees constant access to the application by eliminating the dependency on an Internet connection to the central server. It also provides the best application performance because users can run at LAN speed versus WAN or modem speed.
Application Integrators	<p>Application Integrators provide rapid enterprise access from EAServer to key back-end legacy systems such as:</p> <ul style="list-style-type: none"> • CORBA • SAP • Stored Procedures in Sybase and Oracle • CICS (MVS/CICS) • Tuxedo • IBM MQSeries®

International Contacts

Argentina +54 11 4 313 4488	Malaysia +603 2141 4218
Australia +612 9936 8800	Mexico +52 5 282 8000
Belgium +32 2 716 8311	Netherlands +31 34 658 2999
Brazil +55 11214 4044	New Zealand +64 4473 3661
Canada +1 905 273 8500	Norway +47 2310 5500
China +8610 6856 8488	Philippines +632 750 2550
Denmark +45 3927 7913	Singapore +65 370 5100
France +33 1 41 90 41 90	Spain +349 1 302 0900
Germany +49 211 59760	South Korea +82 2 3451 5200
Hong Kong +852 2506 6000	Sweden +46 8 587 11000
India +91 22 651 1555	Switzerland +41 1 308 63 63
Indonesia +62 21 526 7690	Taiwan +886 2 2715 6000
Italy +39 2 483 241	United Kingdom +44 1628 597100
Japan +81 3 5210 6000	

For other Europe, Middle East, or
Africa inquiries:
+31 34 658 2999

For other Asia Pacific inquiries:
+852 2506 6000

For other Latin America inquiries:
+1 305 671 1020



Sybase, Inc. Worldwide Headquarters

6475 Christie Avenue
Emeryville, CA 94608 USA
Tel: 1-800-8-SYBASE
Fax: 1-510-922-3210
www.sybase.com