



**Introducing the IBM
WebSphere Application Server
Version 4.0 for z/OS and OS/390**



Contents

2	<i>Market Perspective: Application Servers</i>
3	<i>Technology Perspective: Enterprise Java Technology</i>
4	<i>Reality Perspective: Not All Applications Servers Are Alike</i>
7	<i>Introducing The WebSphere Application Server for z/OS and OS/390®</i>
7	<i>WebSphere Application Server Version 3, Standard Edition for z/OS and OS/390</i>
8	<i>Java Specification Levels</i>
8	<i>Prerequisites</i>
9	<i>Pricing</i>
9	<i>WebSphere Application Server Version 4, for z/OS and OS/390</i>
9	<i>WAS V4 for z/OS and OS/390 Supporting The J2EE Architecture</i>
10	<i>COBRA Support</i>
11	<i>Prerequisites</i>
11	<i>Pricing</i>
11	<i>WebSphere Application Server for z/OS and OS/390 Packaging</i>
12	<i>Development Tools</i>
13	<i>In Conclusion</i>
14	<i>For More Information</i>

Market Perspective: Application Servers

In the span of less than three years, the “application server” has become one of the hottest commodities in the computer industry. Although the specific definition of the term varies widely depending on the source, the general consensus describes an application server as software or middleware that facilitates the rapid deployment of e-business applications. (This use of the term “application server” is different than the popular IT meaning; namely, a server box that is dedicated to running business applications.) As we will discuss later, no two application servers are alike; specific hardware, operating system and middleware choices will produce a wide variety of results in such key areas as performance, scale and security. For now, let’s take a generic look at the appeal of this class of middleware from the perspective of the application developer.

Why did the application server become important so fast? Since all e-business applications share a common need for a base set of system services (such as persistence, security and transactions), it became readily apparent that the productivity of e-business application developers would be greatly improved if these services could be standardized and presented to applications in a transparent fashion. You will often see such services referred to as “plumbing” by business programmers, because they are programming activities that are necessary yet have no direct correlation to business processes. The connotation is deliberately negative.

The purpose of an application server, therefore, is to hide this plumbing by transparently extending a set of essential system services to e-business applications so that business programmers can easily assemble them from compact pieces of business logic without requiring a deep understanding of platform-specific infrastructure. Business programmers could no longer afford the time it took to figure out how to use system-level services to achieve business objectives, and application servers were seen as one way to address this problem.

The primary value proposition of application servers is to facilitate the development of e-business applications by allowing business programmers to focus solely on writing business logic; the application server handles all of the difficult system-related chores transparently. This is achieved through a programming model that partitions the task of developing applications



Highlights

Businesses looking to become e-businesses desperately need to reduce the amount of system-level skill needed in order to build their e-business applications.

Java technology has grown up in a hurry, and enterprise Java now provides the most pragmatic approach to e-business application development.

(which does not require knowledge of platform infrastructure) from the task of deploying applications, (which does require platform skills). Application servers are not necessarily aimed at easing the deployment of these applications; the major appeal of this class of middleware is to the people responsible for writing business applications.

Businesses looking to become e-businesses desperately need to reduce the amount of system-level skill needed in order to build their e-business applications. The application server class of middleware plays a key role in the hosting of new Web-oriented applications because it simplifies application development.

Technology Perspective: Enterprise Java Technology

A steadily increasing number of application servers supports a programming model based on enterprise Java® technologies. Therefore, it is important to understand what the enterprise Java programming model is, and why it is so appealing to e-business application developers.

Java's early popularity centered around the "Write Once, Run Anywhere" philosophy of application development. An application written in Java could run on any platform that supported the Java Virtual Machine (JVM). JVMs quickly appeared on all major platforms, and true application portability became a reality for perhaps the first time ever. However, in these early days Java technology was used almost exclusively for visual, client-side applications because performance was lackluster and Java technology did not support the basic system-level services required by enterprise-class applications. Perhaps most importantly, Java technology did little to address the complexity of writing multi-tier e-business applications.

Java technology has grown up in a hurry, and enterprise Java now provides the most pragmatic approach to e-business application development in the industry. The key to the enterprise Java programming model is the concept of container-based components.

A container is a software run time environment that provides common services on behalf of the Java components that run in the container. The container-based component model allows the developer to defer the execution of critical computing tasks – such as maintaining a secure environment, executing transactions and persisting data to a resource manager – to the



Enterprise Java technology provides an environment in which business programmers with Web-oriented skills can produce meaningful, advanced enterprise applications without having platform-specific coding expertise.

run time containers. A business programmer merely specifies what services are required by an enterprise Java component, leaving the implementation of these services to the container. No plumbing required! Enterprise Java technology provides an environment in which business programmers with Web-oriented skills can produce meaningful, advanced enterprise applications without having platform-specific coding expertise.

The enterprise Java programming model defines three container-driven Java components: Java Server Pages (JSPs) to handle server-side presentation logic; Java Servlets, which are ideal for controlling the flow of execution; and Enterprise JavaBeans™ (EJBs), the powerful object-oriented technology for representing business logic and facilitating the access to non-Java systems and data. Combined with a set of useful services and connectors for direct access to back-end systems, these container-driven Java components make up the backbone of Java 2 Enterprise Edition (J2EE), the industry's premiere model for building e-business applications.

The J2EE programming model is flexible enough to allow any combination of containers, services and connectors to be used in building an e-business application, on one, two, or "n" physical tiers. The use of J2EE containers very clearly separates the roles of business application developer, container provider and application installer.

Reality Perspective: Not All Application Servers Are Alike!

Although all Java-based application servers should, in theory, provide a uniform programming model to the developer, they will vary greatly in their ability to provide enterprise-class qualities of service on behalf of the applications that they host. Hardware, such as IBM® @server zSeries 900, S/390®, Intel, and RISC, and operating systems such as z/OS, OS/390, Microsoft® Windows®, UNIX®, and LINUX®, have always been important factors in choosing where to host an application. With the J2EE container-based model for developing applications, selecting the application server that provides these containers and the right hardware platform become the predominant decision points.

Consider that in more traditional methods of developing applications, the programming model required that access to system services be explicitly coded into the application. The developer was responsible for invoking the transaction service; establishing security protocols; utilizing efficient threading



With J2EE, the responsibility for efficient use of system resources shifts from the developer of the application to the provider of the container.

techniques; and much more. Once a hardware platform and operating system was chosen, a skilled programmer could create highly efficient code that took full advantage of the qualities of service offered by the chosen platform environment.

With J2EE, the responsibility for efficient use of system resources shifts from the developer of the application to the provider of the container. Business logic is the same on all platforms; what is different from one platform to another are things such as data access, transactions and how one provides security. All of these capabilities are provided by the container. The degree to which an application server vendor can provide J2EE containers that exhibit the required qualities of service can make or break an e-business application. Some new factors that become important decision points include:

- *Availability on multiple platforms.*

Even the most efficient application server is limited to providing qualities of service that can be supported by the underlying platform. To meet the most diverse set of requirements, an application server should be available on a wide variety of platforms and, of course, fully optimized to each.

- *Ability to leverage existing applications and data.*

In the large enterprise space, the ability to incorporate existing resources into new J2EE applications is critical. Since connections to these resources are the responsibility of the application server containers, not the business programmer, optimization of automatically generated attachment code is essential.

- *Tools for installation.*

Although application servers present a neutral platform view to the developer, personnel responsible for application deployment will likely need to learn new skills to install applications in a particular server environment. A key point of differentiation will be the quality of the tooling that is provided to ease the task of deploying applications.

In order to successfully offer a suite of enterprise Java application servers that meet these requirements, a vendor must have in-depth knowledge and experience with all of today's top hardware and operating system environments. Furthermore they must have an equally strong background in the major

A key point of differentiation will be the quality of the tooling that is provided to ease the task of deploying applications.



transaction monitors, data managers and messaging systems, and an integrated suite of development and management tools to leverage this environment. IBM excels in all these areas, and is putting that expertise to work in its WebSphere® family of application servers.

• *The IBM J2EE Foundation: WebSphere Application Servers*

IBM was an early adopter of and contributor to Java technology and has kept pace with Java's advancement through the current J2EE specifications. IBM was also an early supporter of the trend toward application servers, making the concept of the application server an integral part of our Application Framework for e-business.

IBM was an early adopter of and contributor to Java technology and has kept pace with Java's advancement through the current J2EE specifications.

The IBM vision has been to make J2EE technology available on all IBM, and non-IBM, platforms, thus leveraging our cross-system experience in helping customers build heterogeneous e-business applications. In 1999 IBM introduced WebSphere, a Web infrastructure software that helps companies at each stage of e-business development: from startup, to integrating and exploiting business processes, to handling high volume Web transactions. The WebSphere family consists of a variety of products designed to meet a wide range of needs:

WebSphere Foundation: Provides the essential e-business functions of handling transactions and extending back-end business data and applications to the Web.

Foundation Extensions: Integrated services and tools that enable rapid application development, improve the presentation of information and applications, and help meet performance and manageability needs through advanced deployment services.

Application Accelerators: Modular and extensible business services that enable companies to anticipate and react swiftly to business demands and opportunities, while differentiating their business practices and deliverables.

All of WebSphere rests on the foundation, a family of WebSphere Application Servers that implement the J2EE architecture across a wide variety of platforms—including z/OS and OS/390.



New applications can now be integrated with the extensive inventory of existing zSeries and S/390 applications.

Introducing The WebSphere Application Server for z/OS and OS/390

WebSphere Application Servers (WAS) for z/OS and OS/390 extend the S/390 architecture as a premier platform for the next generation of e-business and exploit its enterprise class qualities of service such as scalability, performance, security and availability. Through the incorporation of open industry standards (such as HTML, HTTP, IIOP, J2EE - servlets, Java Server Pages, and Enterprise Java Beans - and CORBA), new e-business applications and transactions can now be scaled up for global deployment on the server of choice regardless of the development platform. In October 2000, the IBM @server zSeries 900 was introduced to address the requirements for next generation e-business solutions. The zSeries 900 inherits and continues to extend the rock solid features of its predecessor, the S/390.

New applications can now be integrated with the extensive inventory of existing zSeries and S/390 applications. The result is the rapid creation of production-ready e-business applications, along with the ability to evolve to the Enterprise Java Beans or CORBA-based deployment model of choice.

Being J2EE application servers, the primary guiding principle of the WebSphere application server family is to simplify the task of the person developing business applications. This principle mandates that the view of the WebSphere programming model from a developer's perspective is completely consistent platform to platform.

Keeping this in mind, the most important element of the WAS for z/OS and OS/390 product plan is the synchronization of the capabilities of the J2EE containers on z/OS and OS/390 with the rest of the family: with Version 4.0, supported specification levels for JSPs, servlets, and EJBs have been made consistent across the family.

WebSphere Application Server Version 3, Standard Edition for OS/390

As we will discuss shortly, the complete WAS V4.0 for z/OS and OS/390 server is a powerful package of J2EE technology. As a result, it exploits many advanced features of the zSeries and S/390 environments (e.g. WLM goal mode, RRS, DB2® V7). At some point the use of WAS V4.0 for z/OS and OS/390 will dictate upgrading of software and policies, which takes preparation.

In order to accommodate customers who do not yet need some of the more advanced features of the WAS V4.0 for z/OS and OS/390 product, we offer



WebSphere Application Server V3.5, SE for OS/390 is the Java 2-based product that increases spec levels to Servlet 2.2 and JSP 1.1.

WAS Standard Edition for OS/390 (SE) – also supported on z/OS. The SE product provides the support needed to host Java Web components (JSPs and servlets) without the WAS V4.0 for z/OS and OS/390 software prerequisites.

WAS for z/OS and OS/390 SE supports Web components developed in accordance with the WebSphere Family Version 3 programming model. Servlets and JSPs developed and tested using WebSphere Version 3 run-times on distributed platforms can be redeployed unchanged into the WAS for z/OS and OS/390 SE run time. SE supports Java-based connector access to CICS®, IMS™, and DB2.

For customers who have already developed applications using the JSP/servlet engine that used to be shipped with OS/390 (WebSphere Application Server Version 1.2), this engine will remain in service through 3/2002 to ease migrations. However, this engine will not be functionally enhanced.

Java Specification Levels

There are two WAS for z/OS and OS/390 SE Version 3 deliverables available. WebSphere Application Server V3.02, SE for OS/390 is the Java 1-based product that provides support for deployment of Java Servlets written to the JavaSoft version 2.1 specification level, and support for the deployment of Java Server Pages written to either the JavaSoft version .91 or version 1.0 specification level. WebSphere Application Server V3.5, SE for OS/390 is the Java 2-based product that increases spec levels to Servlet 2.2 and JSP 1.1. The Version 3 WebSphere application servers are currently available for the distributed platforms which also support these specification levels.

The pricing for WAS SE for OS/390 SE is based on an OTC (One Time Charge) model and not on an MLC (Monthly License Charge) model.

Prerequisites

WebSphere Application Server V3, Standard Edition for OS/390 requires:

- V3.02: OS/390 V2.7 or higher; V3.5: OS/390 V2.8 or higher
- DB2 V5 or higher
- IBM HTTP Server 5.1 or higher
- RRS enabled (for database connection management)

Pricing

The pricing for WAS SE for OS/390 is based on an OTC (One Time Charge) model and not, as OS/390 customers normally expect, on an MLC (Monthly License Charge) model. This pricing model was adopted to ensure consistency with the WebSphere family pricing model on other platforms.



As the cornerstone of WebSphere application offerings and services on z/OS and OS/390 it supports servlets, Java ServerPages (JSPs), and Enterprise JavaBeans (EJBs), in compliance with J2EE specifications.

WAS V4 for z/OS and OS/390 supports the deployment of applications developed to the J2EE architecture.

The OTC price of WAS SE for OS/390 V3.02 or V3.5 is \$795 per S/390 Parallel Enterprise Server™ (i.e. the price is \$795 regardless of whether the S/390 Parallel Enterprise Server is a 1-Way or 12-Way processor).

WebSphere Application Server Version 4.0 for z/OS and OS/390

The WAS V4 for z/OS and OS/390 product completes the task, which began with last year's delivery of WAS V3.02, Enterprise Edition for OS/390, (also supported on z/OS) of bringing a production-ready, J2EE-compliant runtime environment to z/OS and OS/390. As the cornerstone of WebSphere application offerings and services on z/OS and OS/390 it supports servlets, Java ServerPages (JSPs), and Enterprise JavaBeans (EJBs), in compliance with J2EE specifications.

WAS V4 for z/OS and OS/390 is, in effect, a cross-platform, distributed, object-oriented transaction server. It supports the development and deployment of large scale object oriented business solutions and incorporates a robust persistent object framework that facilitates the integration of existing CICS, IMS, and DB2 assets into a new object world. WAS V4 for z/OS and OS/390 specifically addresses the needs of:

- Programmers responsible for building mission critical, enterprise-level, transactional application solutions using object technology
- Enterprises wishing to take advantage of the benefits of an object-oriented transaction server that is solidly grounded on the industrial-strength qualities of service OS/390 provides

WAS V4 for z/OS and OS/390 Supporting The J2EE Architecture

WAS V4 for z/OS and OS/390 supports the deployment of applications developed to the J2EE architecture, including:

- Support for the deployment of Enterprise Java Beans (EJBs) written to the JavaSoft version 1.1 specification level; Java Servlets written to the JavaSoft version 2.2 specification level; and Java Server Pages written to the JavaSoft version 1.1 specification level
- Remote method invocation (RMI) using Internet inter-ORB protocol (IIOP)



WAS V4 for z/OS and OS/390 is an enterprise Java server that exploits the enterprise-class services available with z/OS and OS/390.

- Java Transaction API (JTA)
- New IMS APPC and CICS EXCI connectors which support the J2EE Connector architecture and implement the Common Client Interface (CCI)
- Access to J2EE services from Component Broker Java Business Objects (Java BOs)

WAS V4 for z/OS and OS/390 is designed as an integrated runtime, providing for the execution of both Web components (servlets and JSPs) and EJBs in a single container. This provides for tighter servlet-EJB interaction and simplifies the deployment and management of applications. The z/OS and OS/390 Parallel Sysplex[®] technology and Workload Manager are fully exploited to provide an extremely high degree of scale and availability.

WAS V4 for z/OS and OS/390 provides a fully secure Web deployment environment with Kerberos as the backbone and SSL as the endpoints. Java RAS services are integrated with existing RAS services and quality of services for z/OS and OS/390.

Through full support of J2EE components, WAS V4 for z/OS and OS/390 supports interoperability with IBM WAS servers on other platforms as well as non-IBM J2EE servers. Java Developer's Kit (JDK) 1.3.0 is shipped with WAS V4 for z/OS and OS/390. This JDK provides the base support for applications at the Java 2 API level.

WAS V4 for z/OS and OS/390 is an enterprise Java server that exploits the enterprise-class services available with z/OS and OS/390. For example, Version 4 actively exploits OS/390 Workload Manager and is fully Parallel Sysplex enabled. The zSeries 900 extends these capabilities with enhanced Workload Manager functions that support LPAR clustering technologies. LPAR clustering will further allow the handling of peak workloads that can be generated by pervasive computing sources. This allows for a highly scalable and reliable environment, unsurpassed in the industry.

CORBA Support

In addition to the support for J2EE applications just described, WAS V4 for z/OS and OS/390 also includes support for a distributed application deployment platform based on the Object Management Group's Common



Object Request Broker Architecture (CORBA), 2.1 specification level. CORBA objects participate in a robust management object framework that supports container management of the object life cycle.

Prerequisites

WebSphere Application Server V4.0 for z/OS and OS/390 requires:

- OS/390 V2.8 or higher
- DB2 V7
- WLM with Goal Mode enabled
- RRS enabled
- LDAP
- CICS TS 1.3 (if CICS used)
- IMS V6 (if IMS used)
- IBM HTTP Server 5.1 or higher

Pricing

As with WAS for z/OS and OS/390 SE, the pricing model for Version 4.0 remains OTC. The price for WebSphere Application Server V4.0 for z/OS and OS/390 is \$35,000 per Central Processor (CP) used to support it in an S/390 Parallel Enterprise Server (e.g. if WAS for z/OS and OS/390 V4.0 runs in an LPAR supported by 2 CPs, the cost would be 2 x \$35,000 or \$70,000).

WAS V4.0 for z/OS and OS/390 also introduces IPLA licensing, under which service/subscription is an additional cost.

Although the developer's view of WAS for z/OS and OS/390 will be indistinguishable from other WebSphere platforms.

WebSphere Application Server for z/OS and OS/390 Packaging

Although the developer's view of WAS for z/OS and OS/390 will be indistinguishable from other WebSphere platforms, packaging will not be exactly the same. The WebSphere application servers on other platforms incorporate three major functional areas that will not be included in the WAS for z/OS and OS/390 Version 4 package, but will be available on z/OS and OS/390 via other means:

Subsystem code, in particular CICS TS and DB2, is not integrated into WAS for z/OS and OS/390. Customers will continue to use these subsystems outside of the scope of WebSphere. However most customers typically might not want to see the products integrated, which would have created additional copies to manage. Subsystems are pre-reqs, but will be ordered and installed through normal channels.



Since subsystems are not incorporated into WAS for z/OS and OS/390, connectors to these subsystems do not ship in the WAS for z/OS and OS/390 package but are available through other means (e.g. with the subsystems or with tool packages, such as VisualAge® for Java). For detailed descriptions of the different connectors see the z/OS and OS/390 Guide to e-business Connectors: Second and Third Editions and the z/OS and OS/390 Web Enablement Overview White Papers.

The bottom line is that the WAS for z/OS and OS/390 plan addresses the key requirements to bring OS/390 and the IBM new high-end operating system, z/OS, into the WebSphere family.

WAS for z/OS and OS/390 does not incorporate an HTTP server. For customers looking to develop two-tier solutions by hosting the HTTP server on z/OS or OS/390, the IBM HTTP Server for OS/390 (once known as Domino™ Go WebServer) is already a part of z/OS and OS/390, and will remain available for e-business applications. The bottom line is that the WAS for z/OS and OS/390 plan addresses the key requirements to bring OS/390 and the IBM new high-end operating system, z/OS, into the family. Most notably is the alignment of enterprise Java programming capabilities at the latest spec levels, even though it does not precisely match the packaging of the WebSphere application servers on other platforms.

1. The WebSphere brand defines two complementary containers for EJBs: one provided by WAS and one provided by CICS TS. Even though CICS TS is not packaged as part of WAS for z/OS and OS/390, it is still very much a part of the overall WebSphere programming model on z/OS and OS/390.
2. z/OS and OS/390 Guide to e-business Connectors: Second and Third Editions, GF22-5124-01 and GF22-5124-02 are available on the IBM internet site s390.ibm.com/e-business/connectors/.
3. z/OS and OS/390 Web Enablement Overview, GF22-5138-02.

Development Tools

Developing applications on workstations is not synonymous with developing applications for workstations, but you can use powerful technologies – such as visual programming, testing and debugging during the development phase – and then move the generated or created code to z/OS or OS/390. IBM offers an integrated application development environment to develop applications on Microsoft Windows NT® which will be deployed on top of WAS for z/OS and OS/390:



We fully expect that transactional enterprise Java programming will eventually become mainstream technology for e-business applications.

- IBM WebSphere Studio is a comprehensive set of tools for Web site content creation and management including site publishing to different stages and servers. It provides an easy-to-use toolset that can reduce time and effort when creating, managing, and debugging multiplatform Web applications. It is the industry's first visual layout tool for dynamic Web pages using JSPs, full HTML, JavaScript and DHTML support. Automatic update and correction of links when content changes make Studio a best-of-breed tool.
- IBM VisualAge for Java is the IBM award-winning Java application development environment for building Java applications, applets, servlets, and JavaBean and Enterprise JavaBean components. It allows the visual development of Web applications by a team based on a shared source code repository and provides an integrated test and debug environment for the different components (e.g. JSP, servlets, JavaBeans and Enterprise JavaBeans).

In Conclusion

This paper has focused attention on the industry trend towards application servers, in particular those that fully support the enterprise Java programming model. WebSphere Application Server for z/OS and OS/390, as an element of the overall WebSphere brand, is a critical vehicle for capitalizing on this major market trend. This is not a one-time investment by IBM or our zSeries customers. Rather, it is an evolution of J2EE technology that will unfold across a number of releases.

One of the great things about enterprise Java technology is that it is not an all-or-nothing technology; enterprise Java programming can be added to the overall application mix in an incremental fashion.

One of the great things about enterprise Java technology is that it is not an all-or-nothing technology; enterprise Java programming can be added to the overall application mix in an incremental fashion. This is important to z/OS customers, many of whom are still in the entry-intermediate stages of enabling their systems for e-business. Servlets and connector technology, already available on the platform, are sufficient for many. There is no requirement for customers to advance their technology base until there is a business need to do so.

But that need will inevitably come; we fully expect that transactional enterprise Java programming will eventually become mainstream technology for e-business applications. Many of our customers have already investigated



this technology in anticipation of beginning serious projects to move to more advanced stages of e-business deployment. WAS V4 for z/OS and OS/390 completes the WebSphere picture for both zSeries 900 and S/390 and paves the way for a new millennium of exciting e-business applications.

For More Information

For more information, contact your IBM representative or visit:

ibm.com/eserver/zseries/a14

ibm.com/software/websphere

ibm.com/eserver/zwebenable



© IBM Corporation
Integrated Marketing Communications
Server Group
Route 100
Somers, NY 10589

Produced in the United States of America
03-01
All Rights Reserved

References in this publication to IBM products or services do not imply that IBM intends to make them available in every country in which IBM operates. Consult your local IBM business contact for information on the products, features, and services available in your area.

IBM, The e-business Logo, IBM Logo, CICS, DB2, IMS, OS/390, Parallel Sysplex, S/390, S/390 Parallel Enterprise Server, VisualAge, WebSphere, z/OS and zSeries are trademarks or registered trademarks of IBM Corporation in the United States, other countries or both.

Lotus and Domino are trademarks or registered trademarks of Lotus Development Corporation.

Intel is a registered trademark of Intel Corporation.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademarks of Linus Torvalds.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through The Open Group.

Windows NT is a registered trademark of Microsoft Corporation.

Other trademarks and registered trademarks are the properties of their respective companies.

IBM hardware products are manufactured from new parts, or new and used parts. Regardless, our warranty terms apply.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

This equipment is subject to all applicable FCC rules and will comply with them upon delivery.

Information concerning non-IBM products was obtained from the suppliers of those products. Questions concerning those products should be directed to those suppliers.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

All copyrighted and trademarked names and terms used in this document are the property of the respective owners and are used in this document under the doctrine of "fair use" for evaluative or comparative discussion. Sine Nomine Associates does not endorse or disclaim any claims made by the copyright or trademark owners as to usability or business value for any product or service referenced in this document

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

GF22-5146-03