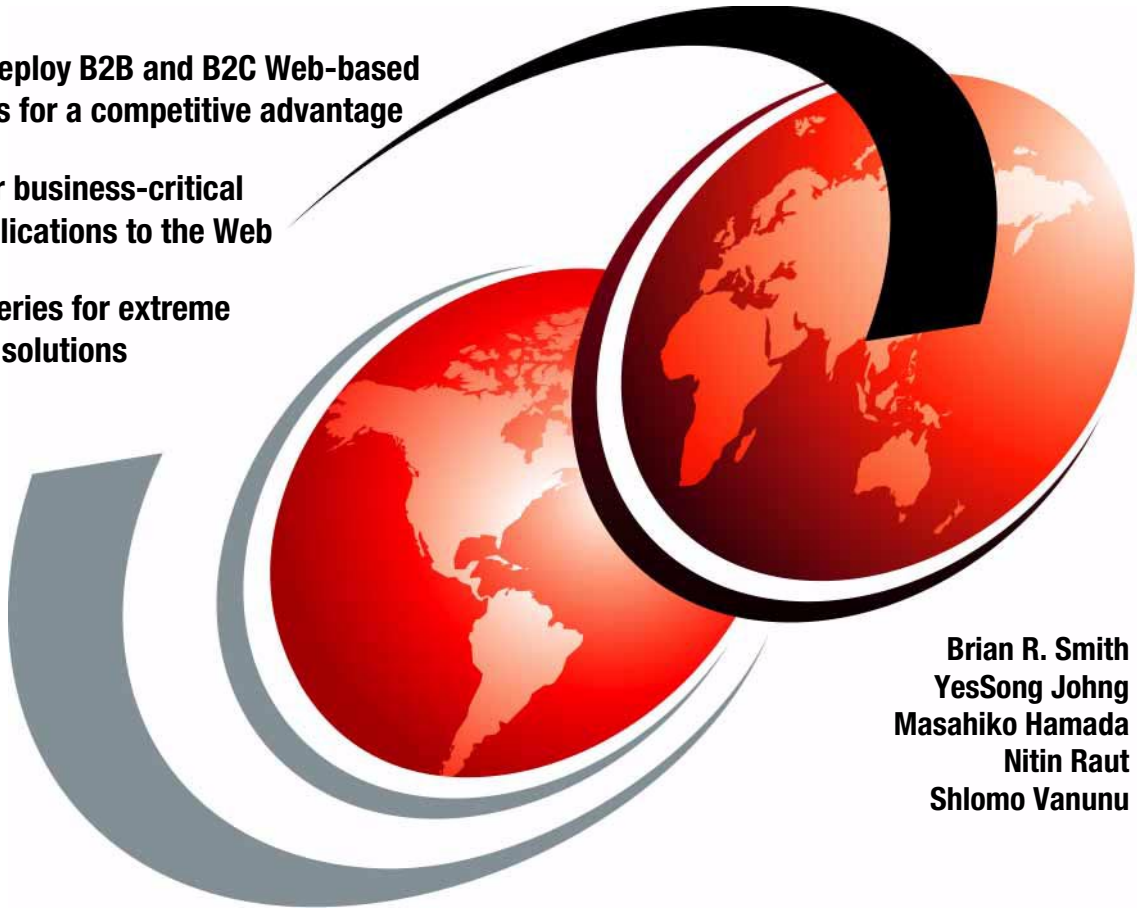
IBM

iSeries e-business Handbook

A Technology and Product Reference

- Build and deploy B2B and B2C Web-based applications for a competitive advantage
- Extend your business-critical OS/400 applications to the Web
- Discover iSeries for extreme e-business solutions



Brian R. Smith
YesSong Johng
Masahiko Hamada
Nitin Raut
Shlomo Vanunu

Redbooks

ibm.com/redbooks

SG24-5694-01

International Technical Support Organization

iSeries e-business Handbook: A Technology and Product Reference

March 2001

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix E, “Special notices” on page 301.

Second Edition (March 2001)

This edition applies to Version 4, Release Number 5 of OS/400, Program Number 5769-SS1.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JLU Building 107-2
3605 Highway 52N
Rochester, Minnesota 55901-7829

Contents

Preface

This technology and product reference guide targets IBM marketing personnel, Business Partners, and AS/400e and iSeries customers who are looking to extend and expand their information server into e-business. It positions e-business on the IBM @server iSeries server against other solutions in the marketplace. Plus, it describes how the iSeries competes in the business-to-business (B2B) and business-to-consumer (B2C) marketplace.

This book shows the solid potential of the iSeries in each facet of e-business solutions. By reading this handbook, you'll gain a broad understanding of how the iSeries serves in an e-business environment. In addition, you'll learn how to sell the iSeries for e-business solutions, including Web presence, dynamic data, and as a transactional site.

This handbook also addresses the needs of both the technical Chief Information Officer (CIO) and the Chief Executive Officer (CEO) of businesses run by iSeries servers.

Note: This redbook reflects the IBM @server iSeries 400 server name. Throughout this redbook, we use the shortened version "iSeries" to refer to both AS/400e and iSeries servers.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Rochester Center.

Brian R. Smith is a Senior Consulting I/T Specialist in the IBM International Technical Support Organization (ITSO) Rochester Center. The first half of his career was spent in design, coding and testing on the System/38 and AS/400 in the area of communications. He then "jumped the wall" into technical marketing support in 1990 to pursue the life of teaching and writing about the iSeries. Brian is the e-business team leader at the ITSO Rochester Center. You can reach Brian at: brsmith@us.ibm.com

YesSong Johng is a Consulting IT Specialist at the iSeries Technology Center, IBM Rochester. He started his IT career 19 years ago as a S/38 Systems Engineer in 1982 and since then, has worked with the S/38, AS/400, AS/400e, and now the iSeries. He has authored a number of books and extensively develops and teaches the technical workshops for iSeries. His

current major responsibility is e-business, which includes Java and WebSphere products on the iSeries server. His other responsibilities include VPN implementation on iSeries, ERP, performance, database, and SQL. He can be reached at: yessong@us.ibm.com

Masahiko Hamada is an Advisory International Technical Support Specialist for the iSeries and AS/400e servers at the International Technical Support Organization, Rochester Center. He writes extensively and teaches IBM classes worldwide on all areas of iSeries e-business. Before joining the ITSO in 2000, he worked in the AS/400 field support center in Japan as an AS/400 System Specialist.

Nitin Raut is a Consultant at CDI Information Services. He has 12 years of experience within IT Industry, specializing in the AS/400e and iSeries servers. He started his career as an application developer and has worked with various ERP packages like BPCS (SSA), Mapics XA, and SAP. He is currently assigned to the iSeries Technology Center (iTC) at IBM Rochester to provide technical consulting and education for various technologies on the iSeries. His responsibilities include e-business, Connect for iSeries and Enterprise Resource Planning (ERP).

Shlomo Vanunu is a Magic/400 Team Leader at Magic Software Enterprises. Has 15 years experience in systems and programming assignments for the IBM System/38 and AS/400e. Shlomo works for Magic in the International Technical Services department, providing Application and System Support for the Magic/400 product. He is an instructor and a writer for the Magic University International. His responsibilities include: Magic/400 application development, legacy application integration, porting, product support, OS/400 system operations, communications, and work management.

A special thanks to the previous authors of this IBM Redbook:

YesSong Johng
Toru Kawashima
Thierry Petit
Susan Powers
Stephen Quan
Suehiro Sakai
Matthias Schmidt
Kristin Zibell

A special word of thanks to YesSong Johng for his leadership and knowledge of e-business. His consultation during and after the project helped reshape this book to its present form.

Thanks to the following people for their invaluable contributions to this project:

Marcela Adan
Robert Boutault
Deb Landon
Bob Maatta
ITSO Rochester

Jon Tate
ITSO San Jose

Nadir Amra
Doug Fulmer
George Gaylord
Al Grega
Steve Gruber
Dick Kiscaden
Terry O'Brien
Tony Perkins
Joe Peterson
John Quarantello
Bill Rapp
Jon Rush
Keith Rutledge
Kelly Schmotzer
Rick Stevens
Marketing and Technical Marketing/Sales

Steven Tri
iSeries Services Network

Jennifer Bigus
Mark Even
Rochester Custom Technology Center

Mike Franks
Pam Sclafani
Technical Education Requirements

James Cioffi
Rochester Domino for AS/400 Competency Center

Steve Gruber
Don Kerlin
Tracy Smith
Rochester Center of Competency for e-business

Dick Salz
Deb Smith
PartnerWorld for Developers

Jim Beck
Jim Herring
Greg Hoffa
Mike Koranda
Doug Luebbe
Rochester Development Laboratory

Bill Benjamin
Lansa, Inc.

Janette Wong
IBM Toronto Laboratory

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please send us your comments about this or other Redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 325 to the fax number shown on the form.
- Use the online evaluation form found at `http://www.redbooks.ibm.com/`
- Send your comments in an Internet note to `redbook@us.ibm.com`

Part 1. What is e-business and why with the iSeries server

This part is written mainly for CEOs or business decision makers from a customer perspective. It is also beneficial to the marketing or the sales force from the IBM or Business Partner perspective. It explains:

- What e-business is
- What the IBM approach is to meet the clients' needs to build and maintain their e-business Web sites
- What users should and can expect from their e-business implementation
- Why the iSeries server can be the product of choice in many cases

While this part addresses both the business-to-business (B2B) and business-to-customer (B2C) impact upon e-business, Part 4, "B2B: Business transform through applications transform" on page 207, focuses on the B2B strength that is particularly well suited for the iSeries server.

Chapter 1. e-business on the iSeries server

Since its inception as a U.S. Defense department special project in the late 1960s, the Internet has become an economic and social phenomenon. It does not overstate the case to declare that companies today must, in some manner, conduct business on the Internet or they will go out of business. However, at a minimum, without e-business participation, they will miss a large opportunity and be at a disadvantage to competitors who are Web enabled.

At the highest level, your business will survive and grow if you can increase revenue or decrease expense at rates faster than your competitors. Business-to-customer (B2C) transactions can increase your revenue as a new channel to market. Business-to-business (B2B) has the power to both decrease expenses and increase revenue depending on how you can align your business as a buyer or seller (or both) in the online marketplaces of today. If you are interested in a focus on iSeries B2B, skip to Part 4, “B2B: Business transform through applications transform” on page 207, as soon as you are done with this chapter.

This redbook reviews the role of the iSeries server in this new technology area and the various tools and methods available for implementing e-business solutions on the iSeries.

1.1 The transition to e-business

There are many factors involved in the success of a business and industry, in general, that support today’s emphasis of electronic information. This section highlights Internet (business) growth and models for developing a successful e-business.

1.1.1 Internet: Tremendous growth

The Internet fire is fueled by the nature of the technology (anytime and anywhere) and by a real growth in usage. Online shopping sales in the 1999 Holiday Season totalled $7 billion. 25 million shoppers spent an average of $200 online while Christmas shopping. In addition, online shoppers were very satisfied with the experience.

In a recent January 17, 2001, report by Jupiter Research (the worldwide authority on Internet commerce), U.S. consumers spent $10.8 billion shopping online during the 2000 holiday season – a 54 percent increase over the $7 billion spent last year – despite concerns of a poor holiday season.

The latest Jupiter Post-Holiday 2000 Consumer Survey reveals that during the holiday season, approximately 36 million consumers purchased online and spent an average of $304 worth of merchandise.

Growth also occurs geographically as more people surf and shop online. In fact, a June 8, 2000, IDC study found that the Western European Internet access market continues to grow rapidly, driven by sustained growth in new users, the success of subscription-free and unmetered access services, the emergence of broadband access technologies, and continuing price erosion. Total user spending on Internet access services is forecasted to increase at a 28% compound annual growth rate from 1999 to 2004, with great variation between countries and access technologies.

This foreshadows a similar explosion to what has happened in the U.S.

More information

For more information on the studies cited here, refer to the following Web sites:

- **Jupiter Communications**: Online Sales Increased by 54 Percent this Holiday Season, Despite Dot Com Closures and Soft Offline Purchases (17 January 2001): `http://www.jup.com`
- **IDC Research**: Western European Internet Access Industry Continues to Reinvent Itself (08 June 2000)
  `http://www.idcresearch.com/Press/default.htm`

1.1.2 New business models

e-business is much more than buying and selling over the Web. It is a new business model where the traditional business processes merge with Internet technologies in business-to-business (B2B) and business-to-consumer (B2C) applications. e-business is about business change and evolution, not just technology, even though the technology makes much of it possible.

By harnessing Internet technologies to extend the reach and range of your business, you can respond more quickly to market shifts, cut product development cycles, enhance teaming within your organization, reach new markets, and serve existing customers better. The iSeries is designed to help you gain a competitive advantage by moving quickly and efficiently into e-business.

1.1.3 e-business development

Almost all organizations and businesses follow a similar process to build their e-business. This process is on-going. It begins with a Web presence, which moves to a dynamic site and finally to a transactional site. This redbook follows this process as a structure to present the various tools, technologies, processes, framework, and models. The phases are discussed in Chapter 2, “Building e-business sites: Phased approach” on page 49.

1.1.4 Chapter objectives

The first objective of this chapter is to review key e-business concepts and terminology so that you have a clear and concise understanding of the various terms and concepts. It is common place in the media and for technology companies to use e-dropping: what do all the “e”s stand for anyway? There are key concepts fundamental to understand these new technologies and to enable a discussion of their relative importance or impact on your business.

The second objective is to provide an overview of key IBM framework and analysis models to understand and explain the components of an e-business solution to customers and potential clients, as well as to design solutions for them. This helps identify places to apply these technologies to modernize and improve various business processes and transactions. It also helps identify key requirements for a successful application of the technology and supports the case for the iSeries server as a major e-business player.

This chapter also looks at where iSeries customers are today and some of the challenges and opportunities they have when implementing these new technologies. This final portion of the chapter highlights and sets the stage for the rest of this redbook, by introducing the topics covered in the remaining chapters.

1.1.5 Target audience

The target audience of this e-business handbook is iSeries customers, IBM marketing representatives, and IBM Business Partners. This chapter can help these audiences understand e-business so they can discuss it with people in their organizations and with prospects. It offers the fundamental information needed to understand e-business in general terms.

IBM and Business Partner marketing and sales representatives should use this chapter and handbook as a sales aid that can be left at the customer site, to remind them that the iSeries server plays a major part in e-business and Internet-based applications. Customers can use this chapter and handbook

as a guide to e-business, with details specifically on the iSeries tools, technologies, and processes available to put your iSeries server and applications on the Web.

1.2 Understanding e-business: Key concepts to know

This section reviews e-business terms and concepts that are often confusing because of their broad use to describe such a broad topic.

1.2.1 Definition of e-business and e-commerce

What is *e-business*? e-business is a business process transformed to leverage WWW (Internet, intranet, and extranet) technology for business benefit. It is about using the Internet infrastructure and related technologies to enable business anywhere and anytime.

e-business is not a technical issue, but rather a business issue that leverages the Internet infrastructure that exists as the delivery vehicle for a variety of goods and services. This includes typical business transactions such as providing goods and services for sale, access to product and service information, marketing and sales, and communications with customers and suppliers.

What is e-commerce? e-commerce is the act of selling products and services on the Internet. It is *one* element of e-business, the primary element. It concerns itself with business-to-business (B2B) and business-to-consumer (B2C) selling of products and services. e-commerce is typically implemented as some form of an electronic store (e-store).

1.2.2 The electronic impact

As for any major change, society needs time to get used to Internet capabilities. However, e-business is already changing the way society works. We are seeing a shift from a vision to something increasingly pervasive. For example, consumers expect to find a Web site for a particular organization. Otherwise, they think the company is not legitimate. A customer or a supplier can refuse to do business with you because you are not “connected”. At first, it was just by e-mail, but now this includes self-service Web sites and online ordering.

Global reach of Internet technology

The reach of the Internet is getting broader. Have you noticed that your mobile phone is ready to handle Internet communications? Do you know that your future refrigerator will be a communications platform to help you

maintain the refrigerator itself and to help you to shop when you need it, so you know at anytime what you have at your disposal in the deep freezer? This is reality, no longer just a vision. This is called *pervasive computing*. It is the idea of putting powerful computer chips and functions into everyday things such as cars or household appliances.

As the Internet becomes increasingly common, the technology and commerce, and social uses of the technology, are racing forward. Huge investments are being made to support the increasing Web traffic as current Internet resources are being stretched to the limit. For example, IBM is involved in several next-generation Internet projects, including Internet2, a project which could create main arteries for the Internet that will be 1,000 times faster than today. This will make entirely new ways of using the Internet possible.

A good example is the cinema industry. Today, you can check out a movie through the Web. However, because of the limitation of Internet resources (the bandwidth), you can't download it instantly. With Internet2, a new channel will be open for distribution of movies – the Web.

1.2.3 The characteristics of e-business

In the late 1960s, the U.S. Department of Defense Advanced Research Projects Agency began funding an experimental wide area computer network that connected important research organizations in the U.S., called the ASRPAnet. The original goal of this network was to provide better collaboration and communication between research sites, share scarce computer resources, and serve as a backup communication channel in case of a national emergency. Since that time, the Internet, as it is now known, has transformed itself in many ways from the hardware and protocols used to communicate to the type of work being done over it.

Transition: Community to commercial

The language used to discuss the Internet, now and in the past, helps tell the story of transition that has happened. Articles and discussion groups were filled with words like community, sharing, education, information, democracy, and people. These words have slowly changed and been replaced by such words as e-commerce, consumer, e-retailer, anytime, anywhere, information database, target audience, and subscriber base.

Availability: 24 x 7 operations

Like it or not, the easy going days of the Internet are gone. Today's Internet is becoming the backbone for commerce and communication in the 21st century. The Internet is now open for business non-stop 24 hours a day, 7

days a week (24x7). It is accessible from almost anywhere. Security concerns are being addressed rapidly. Customer-driven products and services are offered at excellent prices. This electronic culture spans across languages and borders.

Security

Recent denial of service attacks on leading Internet companies, such as Amazon.com, e*Trade, Datek, eBay, Yahoo, CNN, and Buy.com, has brought security issues to the front pages of the popular media. Standards and technology, such as SSL encryption, digital certificates, SET, encrypted e-mail, and firewall, provide protection against such attacks. Additionally, server security including vulnerable features and access points need to be managed by the system administrator. The iSeries server provides the full set of communication and transaction security methods, as well as industry leading server security. The server security is based on an object model that prevents viruses and closes loopholes found on other servers. The government C2 security rating validates iSeries server security.

1.2.3.1 e-business security checklist

With the highly publicized hacking incidents in February 2000, a meeting was held of government and IT industry leaders to determine proactive measures to reduce the likelihood of security-related incidents. Senior vice president and group executive for the IBM Technology Group, Nicholas Donofrio, stated, “The recent denial of service attacks are strong reminders that security needs to be the priority of every online business.”

IBM security experts advise that companies use the following checklist to evaluate their online security practices:

- Implement a thorough and aggressive security policy that is reflected through your business, including firewall configuration access controls and employee communications.
- Conduct a security awareness campaign to regularly remind employees of their security responsibilities (using Web-based certification or regular e-mails, for example).
- Install a firewall on outside and internal borders (between Human Resources and engineering departments, for example). Be sure to change the default settings, which can be easily defeated.
- Use intrusion detection software. This is like having burglar alarms and motion detectors, but for your network. Just as with the firewall, it’s important to have intrusion detection on external and internal networks.

- Distribute anti-virus software. The best anti-virus systems have an easy, effective update mechanism to ensure thorough coverage.
- Establish rules for password selection. Determine very clear guidelines for passwords (such as “six characters with at least one numeral”) and an easy way to verify whether a password is acceptable. Passwords should also be changed periodically.
- Perform security audits on a regular basis. These should be unannounced and random, some electronic, some physical, some stealthy, and others blatant. The ultimate goals of these audits are to enter into the target system, access valuable data if possible, and determine if the intrusion was even noticed.
- Designate someone as the main network security contact and determine clear procedures for reporting and responding to security issues. Employees should clearly understand who to report incidents to and should report all incidents that seem to breach the security policy.
- Ensure that the system administrator stays abreast of security advisories and makes security-related changes in a timely manner. These are the people on the front line, so they need to be as proactive as possible and in a position to react quickly to security issues.
- Have a clear policy for action when an employee leaves for any reason. Actions to take quickly include disabling the former employee’s building and computer access, deleting or redistributing their computer accounts, and changing all passwords and access codes they may have known.

1.2.4 e-business is hot

The business environment, as a whole, has changed. Globalization, deregulation, and competition are now common terms used by almost every business. The business world is now global and is highly competitive with companies competing across national boundaries. The term “global” includes: global markets, global customers, global suppliers, global shareholders, and global opportunities. Consequently, customers are becoming more sophisticated, have more options than ever before, and are more demanding of businesses. Figure 1 on page 10 illustrates the forces that have fragmented national markets and changed the business environment.

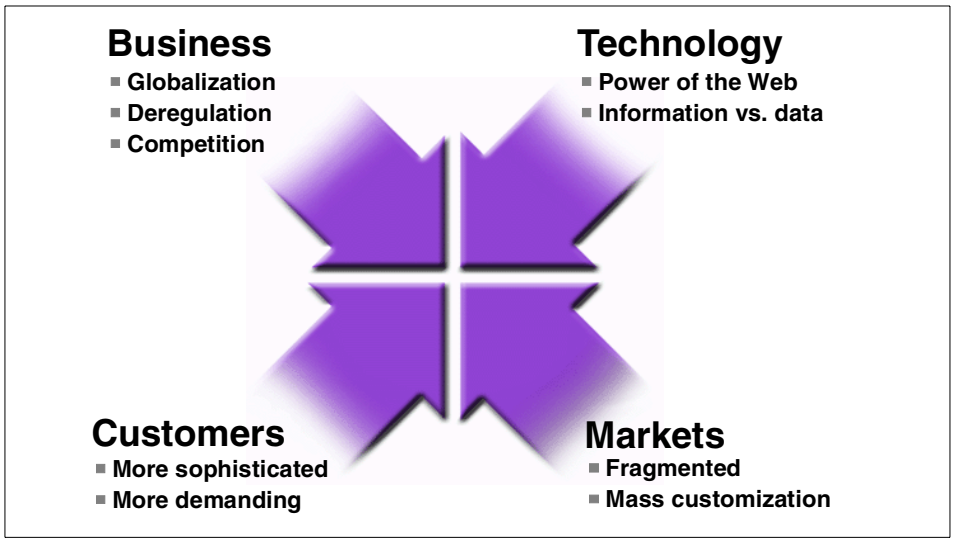


Figure 1. Why e-business? The changing business environment

Four years ago, Amazon.com did not exist. Today this online book shopping site exists only in cyberspace, with over three million titles, expanding their activities into music, video, and gifts. They are open all day, every day, all year, shipping to 160 countries and serving over 1 million customers. Today, we cannot avoid references to the .coms (*Dot Coms* as they are referred to on Wall Street). It is a part of everyday life.

1.2.4.1 e-business is not a license to print money

Their have also been a number of well publicized failures in the .com business recently. For an example, go to Living.com's Web site at `http://www.living.com` to read all about their bankruptcy filings.

The simple point is that e-business is about leveraging all the good old-fashioned and pragmatic business values and strength that you have now with your iSeries server. Business leaders that can both manage the core value-add that your company provides, while at the same time, use e-business to leverage new opportunities for revenue growth and reduced expense will win.

Those that think that e-business is a license to print money will lose.

1.2.4.2 Fast adoption rate: Consumer and business

The Internet has had the fastest adoption rate of any new medium in history. It has taken less than five years to connect 50 million people worldwide. It took radio 38 years and television 13 years to reach the same audience.

A 1999 IDC study entitled *Lotus Notes Agent of Change: The Financial Impact of Lotus Notes on Business* (`http://www.idc.com`) found that almost 25% of suppliers are already online. 67% have started commerce initiatives to build a lead in market share. Figure 2 shows us that 95% of retailers will sell via the Web, with about 50% in the next year.

Note

The IDC study mentioned in this section is available on the Web from the IDC Research home page at: `http://www.idc.com`

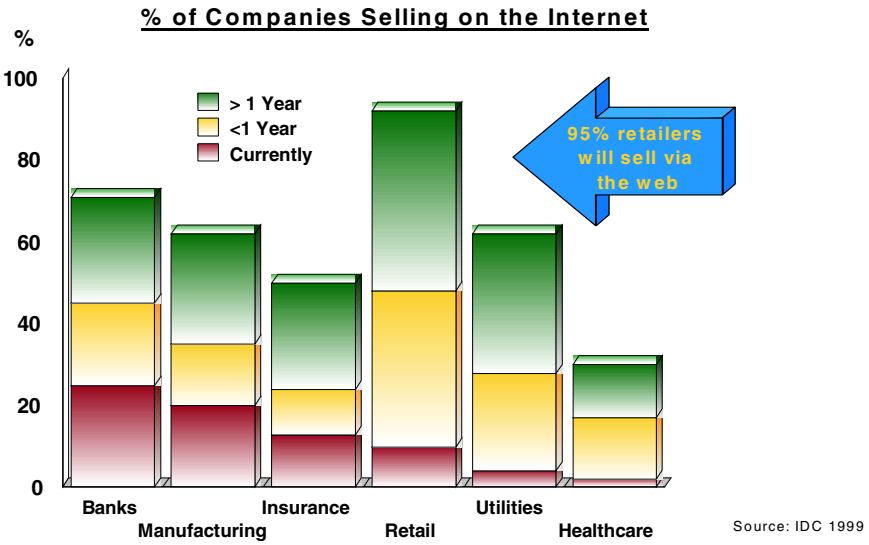


Figure 2. Selling through the Web is transforming industries

In 1999, an IDC paper reported that the B2B model for e-commerce will reach 9% of the total B2B trade (Figure 3 on page 12).

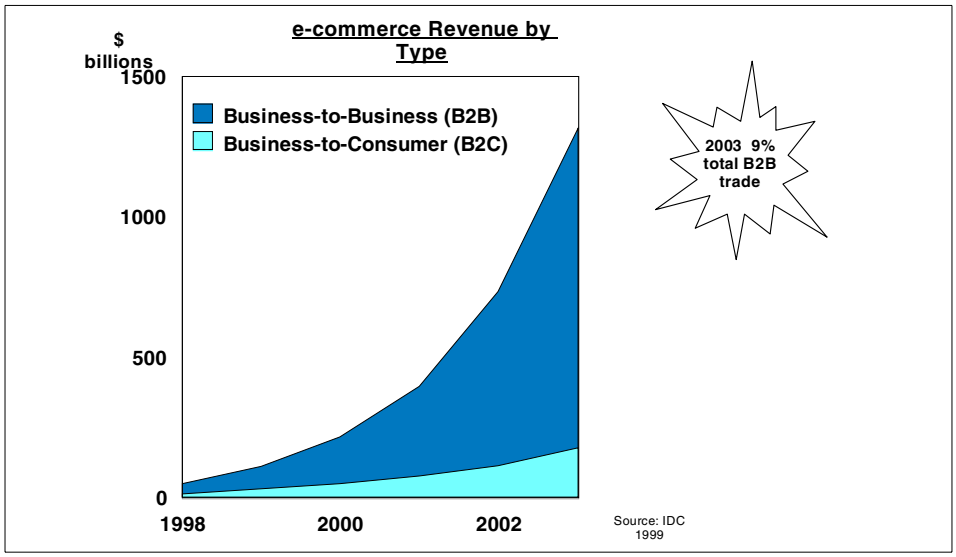


Figure 3. e-commerce revenue business-to-business and business-to-consumer trend

1.2.4.3 Rapid return on investment

There are many references to show you how putting an “e” on your business means rapid Return On Investment (ROI), according to the McKenna Group and IBM:

- Manufacturing
  - Increased order volume by 50% and improved dealer satisfaction
  - Projected 75% reduction in transaction costs and $60 million savings in better-negotiated procurement details
- Retail
  - Quadrupled previous year’s revenue in three months
  - Estimated savings of $28 million over four years
  - Improved franchise satisfaction
- Telecommunications
  - Increased customer satisfaction as response time rose by 20%
  - Deflected calls to the Web site and offered paperless billing, saving about $2.5 million at current levels of adoption
- Insurance
  - Doubled the number of policies sold and increased commissions by 150%, while reducing the turnaround time for policy approval from weeks to days

- Saved $600,000 annually in call center expenses and acts as a platform for wide a range of agent communication needs

- Travel
  - Saved $4 million in revenue from an online reservation system within the first three months
  - Reduced average ticket prices by 15% and cut travel agency fees in half, resulting in $1.5 million to $4.25 million in annual ticket price savings

Coupled with the low cost of ownership, which was rated lowest in the industry (IDC study in November 1998) (hardware, software, staffing, maintenance, and repair), the iSeries server is the right choice to handle your business needs, now and in the future.

1.2.4.4 Summary: Business benefits of e-business

Much has been written about the benefits of e-business. Depending on your organization, its value proposition, and organization, different benefits can be reaped. Some common benefits associated with e-business are:

- **Higher customer satisfaction**: The reasons are extensive, but some examples are customers having personalized 24x7 access to your business and information is better, faster, and easier to access.
- **Increased revenue**: The Internet allows companies to access new markets and customers without having to physically be there.
- **Decreased costs**: Online order status, shipping status, account inquiries, and other information lookups deflect calls from call centers, reducing support costs.
- **Lower prices for consumers**: Cutting out non-value-added distributors and shortening the supply chain pushes prices down.

If you are not convinced that your business has to be transformed to benefit from the Internet opportunity, somebody else will, such as one of your competitors or even a new and unexpected participant.

The question to ask is no longer “Should I go to the Internet?”, but rather “How?”

1.3 e-business development

Companies typically follow a similar process when building their Web presence. Figure 4 on page 14 graphically presents three phases starting

with a Web presence, then a dynamic site, and finally a transactional site. The graph is very steep to indicate the speed at which companies now move through these phases and the benefits and impact the transactional and dynamic sites can have.

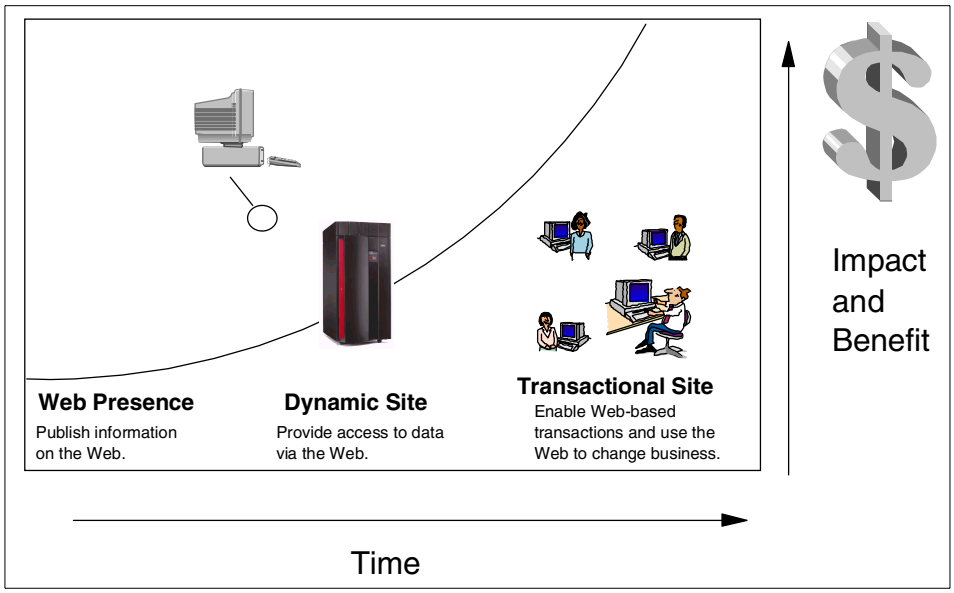


Figure 4. e-business development overview

The first phase in developing your business as an e-business is to establish a Web presence. A Web presence involves presenting marketing and company information on a relatively static HTML home page or site. As business forces change information access requirements, the Web presence typically moves toward providing dynamic data.

Dynamic data sites usually contain user-defined information that is extracted from business systems per the user's request. The user, however, does not add, delete, or edit data or complete a transaction. Customers are satisfied with this improved way to access data that traditionally was provided by call centers or 1-800 telephony applications.

The final phase in this process is to have a transactional site. A transactional site allows users to add, delete, or edit data on business systems and complete transactions (e-commerce). In a B2B environment, the users would purchase agents for other businesses that you supply.

Note that, in Figure 4, the vertical axis has a big dollar sign. You might first think this is cost of implementing a solution. In part, you would be correct. But a better way to look at the vertical axis is Return on Investment (ROI). It is the ROI on the investment you are making into your business that gives you the impact and benefit from your Web presence.

This redbook is organized around these phases so you can understand where your business is in this process, read what tools and technologies are available and how they are used, and review the transition to the next phase. The phases are discussed in Chapter 2, “Building e-business sites: Phased approach” on page 49.

Note

The Rochester Opportunity Center serves as an e-business resource and solution provider. Experts can be reached at (888) 426-9851 in the United States or (507) 253-7056 worldwide. Contact them to explore the offerings and services available or to initiate an e-business analysis.

1.4 IBM Application Framework for e-business

The IBM Application Framework for e-business (Figure 5 on page 16) is an architecture and methodology for building e-business applications that uses industry standards and leading products. The framework for e-business is also the IBM view on how to create a successful foundation and architecture for applications being built. The iSeries fully supports and participates in this framework.

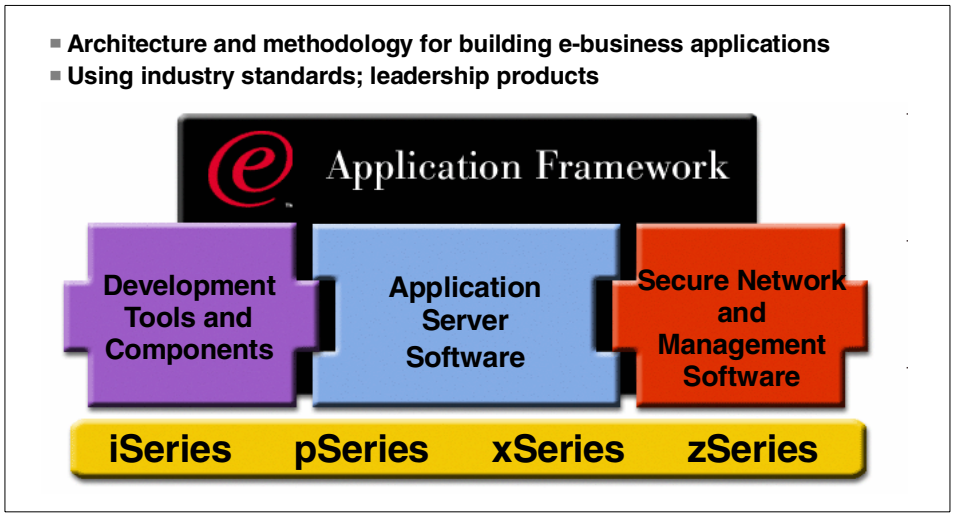


Figure 5. IBM Application Framework for e-business

The application framework provides the right supporting foundation for your new or existing iSeries applications (Web-enabled) working with the Web. That means that this foundation is standards-based (in particular Java), easy-to-understand, and is a prescriptive approach to developing applications that are specially tuned to run on the Internet. This is an entire solution approach, rather than a single product or point solution. This means consistency in application development, faster speed for development, faster speed to deploy. Plus, it gives companies who “build fast and grow fast” the ability to quickly take advantage of the extensive opportunity that the Internet represents.

The IBM Application Framework for e-business helps to build applications for the heterogeneous multi-vendor world and shortens development cycle times by providing cross-platform tools and standards-based software that leverage the existing infrastructure and applications. It provides the fastest, safest way to capitalize on e-business. The iSeries server is strongly committed to participate in this “vision”.

1.4.1 Framework overview

The IBM Application Framework provides a methodology, including the recommended architecture, programming model, ideal standards and techniques, as well as the supporting software portfolio (application server software, development tools and components, secure network, and

management software). These elements are shown in Figure 6 and are explained in the following list.

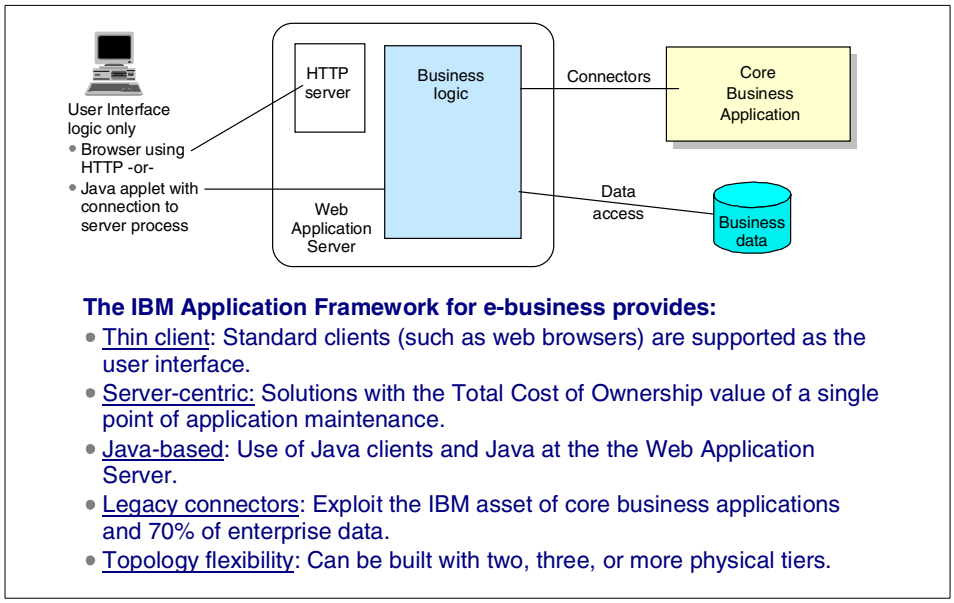


Figure 6. The e-business application model

- **Programming model**: A single unifying Java-based programming model for building Web applications that can be written once and run anywhere.
- **Architecture**: Based on a “Web-able” style of network computing (object oriented design) and providing universal connectivity, rapid development and deployment, software reuse, and connections to “external services” where existing applications and data reside.
- **Ideal standards**: TCP/IP, HTML, XML, Java, servlets, JavaBeans, etc.

  Note: These standards are defined later in this handbook.

Note

The Enterprise acceptance of Java is strong. There are 70 million Java-enabled seats, with more than 1,000 shipping applications. Today, there are 700,000 serious Java developers, 40,000 universities providing Java courses, more than 1000 books available on Java, and more than 2 million downloads of Java Development Kit 1.1.

1.4.2 Development tools and components

IBM offers a rich set of development tools and reusable application components. These are complemented by leading application server software:

- VisualAge Family (includes Java)
- Domino Designer and Lotus Tools
- WebSphere Studio
- IBM San Francisco Application Framework

The software portfolio contains state-of-the-art products and a set of business-tested software to help customers develop, serve, integrate, secure, and manage their e-business applications.

1.4.3 Application server software

The heart of the IBM Application Framework is a set of application servers for building, running, and managing advanced e-business applications:

- IBM HTTP Server for iSeries (Original and Powered by Apache)
- Lotus Domino
- IBM WebSphere
- WebSphere Commerce Suite (previously called Net.Commerce)
- DB2 Universal Database (UDB)
- MQSeries
- CICS Transaction Server for iSeries

Each of these servers is mature and secure, feature-rich, and field-tested, the product of years of experience.

1.4.4 Secure network and management software

Security and manageability are keys within an e-business environment. The SecureWay Family of products simplifies the challenge of locating, connecting, and securing all the parties and resources involved in an e-business transaction or interaction. In this family of products, we cover:

- Host On-Demand
- Host Publisher
- Tivoli (for more information, see `http:\\www.tivoli.com`)

Those products could be seen as “Web-enabler” tools for existing applications. Host Publisher can also be classified as an application development tool.

1.5 The e-business cycle

Experience shows that companies typically go through well-defined steps when going through business transformations. IBM has developed a four-phase model to summarize these steps. This model is shown in Figure 7 entitled the *IBM e-business cycle*. The IBM e-business cycle provides a basis for the IBM Application Framework for e-business. Companies use the e-business cycle repeatedly for each business transformation project they undertake.

The e-business cycle provides companies a blueprint of how to move through each phase when creating and deploying applications and providing support for e-business initiatives. In practice, it can be a powerful tool that allows businesses to act more quickly and decisively when faced with business threats or opportunities.

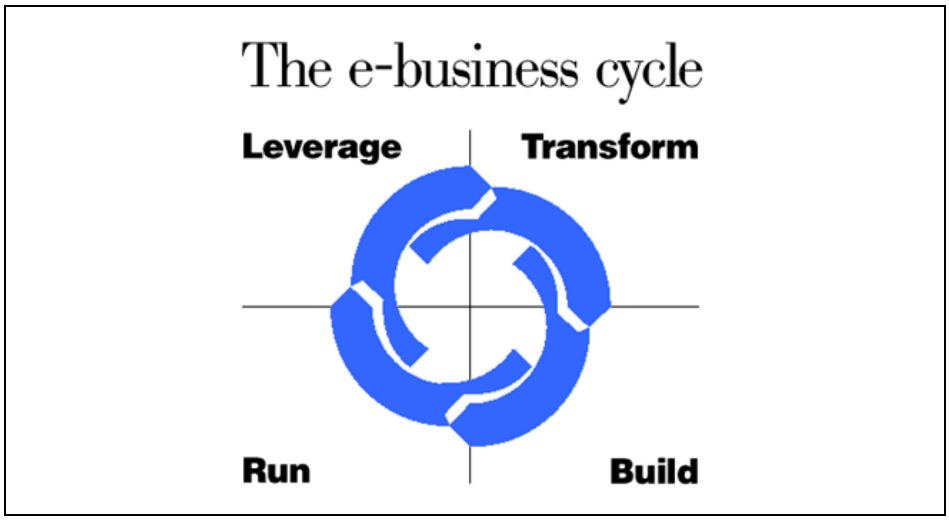


Figure 7. The e-business cycle

The four phases of the e-business cycle are further explained here:

- **Transform core business processes**: This stage is about doing business in new ways by applying Internet technologies to create maximum value for your business. It's about “business, not just technology”. e-business changes the way you actually do Customer Relationship Management (CRM), Supply Chain Management (SCM), and electronic commerce. This means developing an electronic means to do business that provides benefits to the business and the customer.

- **Build new applications**: Transforming core business processes requires a new generation of applications. They run on servers, leverage existing applications and data, and scale to meet user demands. The IBM Application Framework for e-business is designed to help you build and deploy a new generation of applications that are open, flexible, and easy to change. This allows businesses to “start simple and grow fast”.
- **Run a scalable, available, safe environment**: The infrastructure that provides these new applications is under considerable pressure. Businesses are looking for a better return on investment. Users want systems that are easy to use, yet always responsive. The solution is to provide an environment with scalable servers, flexible clients, and advanced storage devices, which are all handled in a secure, manageable way. The iSeries provides the availability, security, and scalability benefits required.
- **Leverage knowledge and information**: e-business is about creating a responsive organization that makes intelligent use of all types of data and organizational knowledge. It allows you to use data as a competitive advantage (for example, profiling, personalization, and product offering customizing), and businesses can quickly customize product and service offerings to the customers requirements.

In its entirety, the e-business cycle builds the basis for the IBM Application Framework for e-business to provide a supporting structure for businesses and provide organizations with a common, unified programming environment.

1.6 e-business value chain: End-to-end solution model

The end-to-end solution model (Figure 8) shows the variety of business applications that e-business can have if applied from suppliers to customers. This model also focuses on the back-office applications in a company that are typically Enterprise Resource Planning (ERP) style business applications that house much of a company’s valuable data. This data is not only valuable for the actual business, but also for suppliers (for example, production or inventory information) and customers (for example, order status or support).

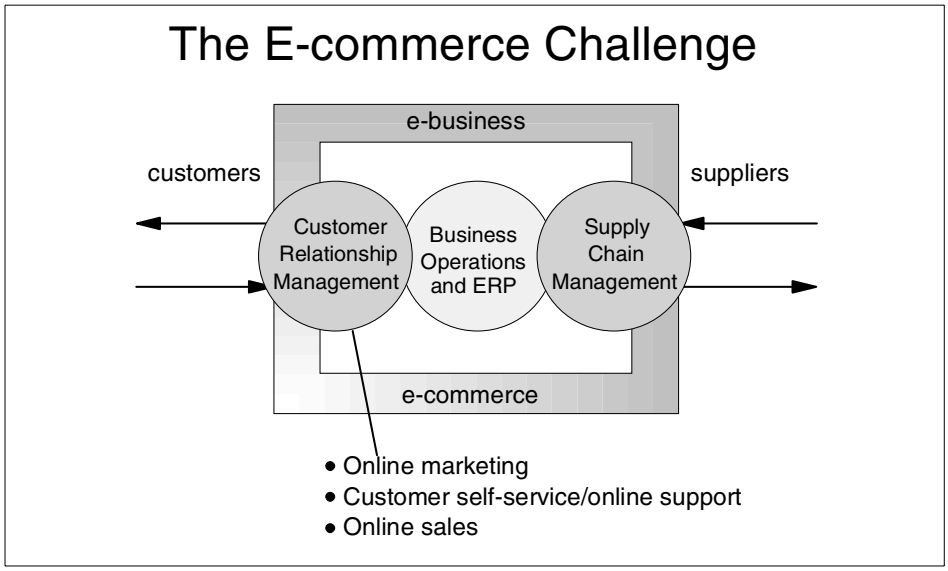


Figure 8. End-to-end solution model

For iSeries customers, this data is typically almost entirely housed on an iSeries server, and, by its nature, is a very central part of most e-business applications. The e-data is not only valuable for the company, but also its suppliers and customers. End-to-end e-business solutions leverage core business data and are the basis for most e-business applications today. This is where to start when looking for potential e-business opportunities.

1.6.1 ERP: Core business applications

ERP packaged software covers the primary back-office functions of a business such as financial systems (General Ledger, Accounts Payable, Accounts Receivable, ans so on), inventory management, human resources, planning, procurement management, supply chain management, customer relationship management, and sales force automation. The business data that ERP applications contain is the raw material for many kinds of e-business applications that leverage into Customer Relationship Management, supply chain management solutions, e-commerce, and other related solutions.

1.6.2 Customer Relationship Management

Customer Relationship Management (CRM) involves supporting, developing, and retaining profitable customers. CRM is central to e-business. Analyzing customer behavior enables businesses to personalize their offerings and to

anticipate their customers' wants and needs. Doing this successfully means organizations can maintain good customer relationships. This is key to retaining customers, which is something every organization is working hard to do. Gaining a new customer is six times more costly than retaining an existing one. The cost of customer defections is also well documented. A company with a 90% customer retention rate, which most businesses would consider exemplary, in fact loses almost half of its customer base every five years.

Creating an e-business application involves exploring the central theme of CRM: developing solutions that serve customers better. The development of an e-business application for CRM begins by answering some key questions:

- Which back-end support processes (for example, inventory management, transportation management, product design) are integral to serving customers better? This involves inventorying available data and going through a process to assess the value of that data to customers and then determine which information to supply in which manner.
- How can a Customer Relationship Management initiative to involve or integrate these processes in a manner that results in true performance improvements for customers be deployed? Some examples are to provide better information (for example, more accurate, faster, and easier to access) or better service (for example, 24x7 support, product access, and lower prices).
- Which business processes can be significantly improved by effectively implementing them as e-business solutions? Typically these involve customer interaction points (for example, price lookups, account status, and reports), large distributions of paper (for example, promotions, announcements, and newsletters), or data inputs (for example, account information, such as an address change).
- What is the relative business value of each independent process and which application will customers value most? There can be many.
- How capable is the IT infrastructure of supporting Customer Relationship Management initiatives to integrate customer-facing or back-end operational processes? Back office systems are opened to customers without a customer service representative as a middle person. This creates its own unique availability, performance, and usability issues.

An effective Customer Relationship Management system helps businesses be more successful because they can identify and sustain profitable and enduring relationships with their customers. This happens when customers obtain the right information, at the right time, in the right format. It provides value!

1.6.3 Supply Chain Management

A supply chain is the way an enterprise ties together the people, process, and related information, both internal and external, associated with its flow of products. Supply Chain Management (SCM) is a strategy for linking enterprises with their suppliers, distributors, and customers to facilitate information exchange and to unite all steps in the product cycle. This includes initial product design and procurement of raw materials, production, shipping, distribution, and warehousing, until a finished product is delivered to a customer. Companies deploy SCM technology to help improve communications, planning, and collaboration with trading partners to gain efficiencies and cost savings that lead to a competitive advantage.

SCM applications are developed to leverage better communication; access information, funds transfers, and others provided by e-business; and reap benefits like lower costs, faster cash cycles, lower inventories, more satisfied customers, and so on. These applications have been successfully deployed across many industries and are major areas for the deployment of e-business applications.

Some examples of supply chain solutions include:

- Forecasting and demand planning, for example, supplying production schedules to suppliers to minimize inventory and build it before peak periods
- Strategic sourcing and procurement, for example, selecting qualified suppliers and sourcing from the appropriate one when necessary
- Production logistics, for example, just-in-time (JIT) systems
- Transportation and shipment management, for example, timely distribution and delivery of products and services
- Distribution and warehouse operations, for example, stock replenishment

As with CRM, SCM leverages existing business data and applications by using the Internet and e-business applications as an access and distribution point for suppliers, distributors, and partners.

1.6.4 E-commerce and e-business

As defined earlier, e-commerce is about actually exchanging products or services for money with the end-consumer via the Internet. When you make it possible for customers, distributors, and suppliers to conduct actual business with you over the Internet, you open an entirely new sales channel. This channel is unique because it can sell products and services in nearly every country in the world, 24 hours a day, for a fraction of the cost of traditional

channels. It is similar to hiring a global sales force for the cost of a single representative.

In addition to the sales and financial potential, there is also a large amount of strategic information about your Web customers that is hard to access through other sales channels. This information can help sell additional products and services and keep customers loyal.

In relation to core business systems, e-commerce brings them closer to the actual point of sale. Customers actually interact with inventory and financial systems (in a controlled and secure manner, of course). Web orders flow directly into your information systems, thereby reducing intermediate steps and error rates, which speeds up product deliveries and builds customer satisfaction levels. This makes the availability, security, and usability of these options very important.

In summary, the IBM Application Framework for e-business gives you the foundation on which you can build successful e-business applications.

1.7 Requirements for success: The realities of e-business

Business pressure is increasing. Executives, including the CEO and Board of Directors (BOD), demand responsiveness and flexibility in their IT systems. Users demand reliability and functionality. The CFO demands better cost control. Existing and new customers demand new ways to access information and world class customer service.

All of these demands challenge and put tremendous pressure on systems to accommodate these varying requirements. Downtime is feared because it impacts more than employee productivity. It affects the bottom line in the e-business world. And on the top, there is more pressure than ever before on development cycles, and the speed to deploy them is critical.

Given these tremendous pressures, what are the characteristics of a successful e-business application?

1.7.1 Attributes of successful e-business applications

Here are strong indicators of successful e-business applications:

- **Standard-based**: Support all clients equally.
- **Server-centric**: Data, application logic, and business rules are managed centrally and can update all in one place, which means quick and cost-effective deployment.

- **Leverage core systems**: Extend existing investments in legacy systems that still run your business.
- **Scalable**: Applications that grow with your business and meet unpredictable demands, day and night.
- Quick to deploy and easy to use.
- **Manageable**: Ensure system continuity and availability; downtime is costly.

The environment in which these applications run also needs to have specific attributes such as scalability, availability, and security.

1.7.2 Additional attributes for success

Beyond scalability, availability, and security, successful e-businesses incorporate these additional attributes:

- **Scalability for capacity on demand**: Investment protection for applications, snap-in upgrades for hardware, ability to add capacity on the fly, an so on.
- **Interoperability among systems**: Synchronization of updates, real-time currency, sharing of data, common business rules, and so on.
- **Availability around the clock**: Continuous access to data, workload management, redundancy, backup and restore, and clustering.
- **Security of data and transactions**: Built-in security, firewall, cryptography, access control, global sign-on, network security, and secure gateways and servers.
- **Manageability of multiple resources**: For example, networks components, operating systems, databases, applications, and servers.
- **Integrated system services**: For example, database services, transaction services, basic HTTP services, Java services, messaging services, components services, and Enterprise JavaBeans.

1.7.3 How the iSeries server fares in the e-business game

To put it simply, it fares very well! Integration, server centric model integration, availability, reliability, security of data and transactions, serviceability, ease of use and manageability, and scalability all sound familiar. It is because they are the core value propositions that have made the iSeries server what it is today. They continue to be the value propositions that make it such a good e-business server.

It makes sense that if your business data and applications are on your iSeries, “Web-enabling” those applications and providing access to that information should be done directly from your iSeries.

The IBM Applications Framework for e-business provides the iSeries server with a strong architecture and set of tools to develop with, especially if you are Java-minded.

What about other benefits of running your e-business application on the iSeries server?

Let us explain the top advantages that the iSeries platform delivers today:

- **Security and integrity:** Both are quite important on the Internet. Security provides access to core business applications, secures data internally, and secures transactions. There are several types of products customers use to solve network security issues, such as stand-alone firewalls, security appliances, and security options in both hardware (routers and hubs) and software.

  The iSeries server offers a C2 security rating (delivered by the U.S. Government, which means ready to work securely for the Pentagon where security requirements are very high) and supports Secure Sockets Layer (SSL) and Virtual Private Network (VPN) natively. SET Secure Electronic Transactions support is included since there is an integrated firewall. And last but not least, there has never been a known virus on the iSeries server.

Note

On 15 February 2000, IBM announced that the IBM Firewall for AS/400 product (5769-FW1) and Integration Services for FSIOP (5769-SA2) will be withdrawn from marketing effective 31 December 2000. Customers running either of these products on the Integrated Netfinity Server (now referred to as Integrated xSeries Server for iSeries) will be supported with their current capabilities until 31 May 2001. However, these products will not be enhanced.

We recommend that customers running Integration Services for FSIOP or IBM Firewall for AS/400 plan to implement an alternative solution. For planning information, refer to the Web site:
`http://www.iseries.ibm.com/products/firewall/index.htm`

In addition, see the redbook *All You Need to Know When Migrating from IBM Firewall for AS/400*, SG24-6152.

There are many alternatives available for customers concerned with network security. Customers are likely to find alternative products available from third-party vendors that have equivalent or more functions, and are more extendable compared to the AS/400 Firewall product.

- **Reliability and availability:** These benefits are key because e-business solutions are 24x7 operations and downtime equals lost money and customers.
- **Scalability:** This means how easily a computer system can grow and how much total growth is possible. Typically e-business solutions start small and grow quickly. This type of growth requires a system that can easily and quickly grow as demand grows. This is an iSeries strength because of its modern architecture. The iSeries server delivers over 18.9 TB of disk capacity for data on a single machine, up to 96 GB of “memory” on a single machine, and over 330 times the growth for processor capacity with the same operating system.
- **Ease of management:** Central administration is another essential point when working with the Web. Through easy graphical configuration wizards, with features like Management Central, part of Operations Navigator, you can easily manage multiple iSeries servers centrally. The iSeries server can also integrate Windows NT applications through the Integrated xSeries Server for iSeries, and Domino running natively. Having central management of system resources makes the operator function much simpler.
- **64-bit architecture:** 64-bits means speed, and speed is valuable for Internet users. Often seen as an old system, the iSeries server is everything except an old system. The iSeries server is completely 64-bit hardware, operating system, applications, and database. As new hardware technologies are introduced, there is no need to re-compile or re-write existing applications, applications that are going to benefit immediately from the technology. Thanks to this architecture, the iSeries server is now ready for the next step – 128-bits, without disruption.
- **Logical Partitioning (LPAR):** Recently released in OS/400 V4R4, LPAR allows one iSeries server to have multiple server functions running at the same time (requiring at least processor one for each partition). For example, this allows you to have the Web server, multiple OS/400 versions, languages, e-commerce, database access, and e-mail systems on the same machine.
- **Java leadership:** The iSeries server offers a broad range of capabilities, sometimes underestimated or, even worse, unknown. Through extensive development, the iSeries server has optimized OS/400 for running complex, multi-threaded Java applications that are coming to market. The

iSeries server also supports the WebSphere Application Server (WAS) that lets ordinary Web servers run Java servlets on the iSeries server with links to DB2 Universal Database for AS/400 (DB2 UDB for AS/400).

- **Cost of ownership**: With scalability, security, and reliability built in from the ground up, the AS/400e platform is designed to enable you to conduct business-to-business and business-to-consumer Internet transactions. The iSeries delivers all those things for the lowest total cost of ownership (see *Server Selection: Reversing the trend of rising IT costs*, IDC (December 1998), or *AS/400 equals cost-effective ERP*, Meta Group (April 1999)).

Clearly, the iSeries server provides a path to e-business. By supporting transformation of business applications to an e-business model while minimizing disruption, this platform has a unique position. It has business proven values (reliability, security, scalability, low cost of ownership, logical partitioning, service, and support). Plus, the iSeries server supports the latest enabling technologies for e-business. In combination, these qualities make the iSeries server a good choice for not only extending existing applications, but also for deploying new solutions that require attributes that are iSeries standard features.

Java and the iSeries server

The iSeries server offers improved performance, scalability, and reliability through the optimized implementation of the Java Virtual Machine (JVM) below its Technology Independent Machine Interface (TIMI) and because the iSeries is a true 64-bit Java server. The iSeries server provides an object-based architecture that closely matches Java's own object-oriented architecture. Finally, the iSeries server offers an AS/400 Java Transformer, which improves performance by creating optimized 64-bit AS/400 execution objects for the iSeries server.

Because of the above, the iSeries performs very well in Java-related benchmarks. Please see `http://www.iseries.ibm.com/whpapr/jbob400.htm` for a white paper on the subject.

1.8 The key products: An overview

As we have discussed in this chapter, e-business is and will continue to be an important part of almost any businesses strategy. Not every business is suited to have an online store. However, e-business technologies can be

used in business-to-business and supply chain applications, meaning creativity is the only limitation.

The iSeries server plays an important role because of its inherent operational strengths and also because of the data typically stored on it. The operational strengths allow it to be a powerful Web and application server. The data stored on the iSeries server is the raw material and back-office operational system for many e-business applications.

The iSeries server also provides many tools, features, and benefits for customers as they move through e-business phases. The following section highlights products featured in this redbook with a reference to read more.

To complement IBM e-business products, many third-party solutions are available. Some are outlined in Chapter 12, “B2B: Application solutions” on page 261.

1.8.1 OS/400 (5769-SS1)

The iSeries operating system, OS/400, is conceived as a *single entity*. This means that facilities, such as relational database, communications and networking capabilities, online help, and much more, are fully integrated into the operating system and the machine. The user communicates with all components of OS/400 using a single command language – Control Language (CL).

OS/400 provides tools to handle two different computing environments for the iSeries: servers and systems. The iSeries client/server dimension combines an open system environment with iSeries server price/performance and the integration of system solutions to extend a complete product package for the server environment. For legacy systems, the iSeries server provides integrated functions based on the traditional commercial computing environment.

As the computing industry moves rapidly toward a network-centric world made up of global networks, iSeries software grows with significant enhancements to make the iSeries server a key player in this vibrant and vital area.

OS/400 is designed to be comprehensive and scalable. Some of its features are described in the following list. They help to make OS/400 the most complete operating system on the market today.

- Ease of installation and ease of use: Includes system-supplied menus and fast path commands, automatic configuration of local devices, online help

text with index search and context sensitivity, copy screen images, operational assistance, a built-in communications line to contact service, enable remote service personnel to diagnose the system online, analyzing symptoms, and downloading program temporary fixes (PTFs) to resolve errors.

- Security is selected to meet business needs, ranging from minimal security (where no passwords are used and any user can perform any function), to resource security (where passwords are required and users and the use of objects can be restricted to specific functions), on to operating system integrity (which prevents unsupported interfaces to interfere with the system). Security violations are logged in a security journal.

  The highest level of security (known as Level 50) enables the iSeries server to operate at the C2 level of trust as defined by the U.S. government.

  Within communications, further security is possible by implementing LU6.2 Session Level Encryption (SLE) for iSeries applications, which use LU6.2 communications.

1.8.1.1 Connectivity

The iSeries server offers a wide range of communication capabilities and functions to enable the iSeries server to communicate with most IBM and non-IBM systems.

The iSeries server supports many protocols and networks, including:

- ISDN Data Link Control (IDLC)
- IBM Token-Ring Network (IEEE 802.5 and 802.2) (token-ring up to 100 Mbps)
- T1/E1/J1 and Fractional T1 Networks (high bandwidth)
- Synchronous Data Link Control (SDLC)
- Ethernet Version 2 or IEEE 802.3 (Ethernet up to 1 Gbps)
- ATM LANs

OS/400 has the following communication facilities (and more), with each facility as part of OS/400:

- TCP/IP Support
- Remote Work Station Support
- Advanced Peer-to-Peer Networking (APPN)
- Dependent Logical Unit Requester (DLUR)
- SNA Distribution Services (SNADS)
- Distributed Data Management (DDM)

- ISDN Support
- File Transfer Support
- IPX/SPX Communications
- ATM LAN Emulation

1.8.1.2 Euro currency

iSeries support includes updates to input, display, print, and process the euro currency sign for both the host and PC client computing environments. This includes an addition of euro keyboard types, including device configuration and device controller changes, the euro font and glyph support, and printer support.

1.8.1.3 Integrated file system

The integrated file system (IFS) is a part of OS/400 that supports stream data, including images, audio, and video; storage management similar to a personal computer; and UNIX operating systems. The IFS also provides structure for all information stored in the iSeries server with consistent use of object names and associated object information across national languages.

Key features include:

- Support for storing information in stream files that can contain long continuous strings of data
- A hierarchical directory structure
- A common interface that allows users and applications to access not only the stream files, but also database files, documents, and other objects that are stored in the iSeries server
- A common view of stream files that are stored locally on the iSeries server, an Integrated xSeries Server for iSeries, or a remote Windows NT server

1.8.1.4 Built-in system management facilities

A variety of tools and functions are available to provide system availability and management. These include:

- System Managed Access Path Protection (SMAPP) to automate the process of selecting which access paths should be protected.
- Expert Cache as a disk cache tuner. It dynamically responds to system jobs to cache pages of data in main storage to reduce the time to process disk I/O.
- Integrated Hardware Disk Compression supported by OS/400. Data is dynamically compressed and uncompressed by the DASD controller as data is written to and read from disk. Disk compression does not effect the

main CPU utilization since this function is performed by the DASD controller IOP. Compression is a method to increase disk storage beyond capacity limits, while maintaining access and reasonable performance levels.

- Hierarchical Storage Management (HSM) APIs used by Backup and Recovery Media Services (BRMS) (5769-BR1) to provide HSM functions. These APIs can also be used to develop custom HSM applications.
- Internet PTFs: iSeries customers can download PTFs over the Internet. The client hardware needed is a PC with Windows 95 or Windows NT, a TCP connection to the iSeries server over a LAN, and access to the Internet. The functionality is the same as the Electronic Customer Support (ECS) method of transport. The user selects the PTFs and options using a Web browser and submits the order. The user can search on PTF cover letters and read them before the order is even placed.

1.8.1.5 Server availability: Over 99% and climbing

The iSeries server boasts a 99.94% availability rating, second best in the industry behind the IBM S/390, 43 times more available than Windows NT, and 4.5 times more available than UNIX (from the Gartner Group, “Platform Availability Data: Can You Spare a Minute?” October 1999).

Various functions are available to help maintain this high-level availability of an iSeries server. These include:

- All iSeries servers support an optional uninterruptable power supply (UPS) to maintain power to the iSeries server during a site power loss.
- Disk mirroring for the entire system or one individual auxiliary storage pool.
- Journaling provides the capability to record all changes to records in a file as they occur. These journaled changes are applied to the file if the system is lost.
- Commitment Control ensures that if a transaction requires multiple database changes, all of them (or none of them) are made.
- Auxiliary storage pools (ASPs) isolate objects to assist in their recovery.
- The Save-while-active function allows continued use of applications while objects are backed up. This reduces the time the objects are unavailable and the time to save.
- Concurrent maintenance of I/O cards, power, and other components reduces down time due to the failure of components in the system.

- RAID-5 disk protection with the use of a disk controller and at least four disks to make up an array.

1.8.1.6 Database support

The integrated database, DB2 Universal Database for AS/400 (DB2 UDB for AS/400), provides stability and compatibility of previous releases of the iSeries database with a standards-based technology required for a heterogeneous computing environment. DB2 UDB for AS/400 provides compliance in the area of standards compliance, coupled with advanced function, distributed capabilities, and performance.

These and more functions are further described in the *iSeries Handbook,* GA19-5486.

Other functions available for the operating system include:

- **Client Access Express for Windows (5769-XEI)**: Includes the client software needed to connect Windows 95 and 98 and Windows NT workstations to an iSeries server.
- **Operations Navigator for iSeries** (part of OS/400): Provides a graphical interface to systems administration functions. It is designed to be highly integrated with Windows. Windows NT Server (Version 4.0) is packaged, priced, and supported by Microsoft and must be purchased through a Microsoft dealer.

Refer to the online version of the *iSeries Handbook*, GA19-5486, for further descriptions of OS/400, its features, and functions. A softcopy update of the *iSeries Handbook* is kept online at: `http://www.redbooks.ibm.com`

1.8.2 IBM HTTP Server for iSeries (5769-DG1)

The IBM HTTP Server for iSeries (Original) is the follow on to the IBM Internet Connection Server (ICS). It is a scalable, high-performance, Web server that has been available since OS/400 V4R3 and other IBM and non-IBM platforms.

Two distinctly different Web servers are offered with the iSeries server. The IBM HTTP Server for iSeries (Original) as mentioned above and the IBM HTTP Server for iSeries (Powered by Apache). The iSeries Powered by Apache server is based on the open-source server code provided by the Apache Organization. This version is based on the “alpha” code for Apache Version 2.0. It will be updated as future Apache versions are made available. While the iSeries source code will not be published, IBM will offer the enhancements it develops to the Apache Organization in an open-source

form for inclusion in the Apache server. As with any supported product, IBM will provide defect support for the iSeries Powered by Apache server. Group PTF SF99035 is required to enable the Apache server running on OS/400 at V4R5. The Apache server will be integrated into a future release of OS/400.

IBM HTTP Server is a complete Web server product with advanced security and application development features. With IBM HTTP Server for iSeries, you have everything you need to quickly and easily establish a Web presence and start on your road to working the Web for business.

Gartner Group says

The Gartner Group (find them at `http://www.gartner.com`) has provided a good holistic view of the strengths of the AS/400 as a Web server in today's business environment. In fact, the title of the research report is "AS/400 as a Web Server: Perspective" updated 28 December 2000. You can subscribe directly to Gartner to read the report.

You can find more information on the IBM HTTP server in Chapter 5, "HTTP Server for iSeries" on page 89, or see the Web site at:
`http://www.as400.ibm.com/products/`

1.8.3 IBM AS/400 Developer Kit for Java (5769-SS1, 5769-JV1)

The AS/400 Developer Kit for Java is optimized for use in an iSeries server environment. It uses the compatibility of Java programming and user interfaces, so you can develop your own applications for the iSeries server.

The AS/400 Developer Kit for Java allows you to create and run Java programs on the iSeries server. The AS/400 Developer Kit for Java is a compatible implementation of the Sun Microsystems, Inc. Java Technology. Unique aspects of the AS/400 Developer Kit for Java design include:

- The integration of critical Java Virtual Machine components below the Technology Independent Machine Interface (TIMI)
- A Java transformer for the conversion of Java bytecodes to RISC machine instructions
- Advanced, scalable implementations of garbage collection, object allocation, and synchronization
- The Remote Abstract Window Toolkit implementation that supports the java.awt APIs

The AS/400 Developer Kit for Java is not an Integrated Language Environment (ILE) language. It introduces a new language environment on the iSeries server that is built on Java, American National Standard Code for Information Interchange (ASCII), integrated file system, and other industry standards.

The AS/400 Developer Kit for Java supports the common JDK tools, such as `javac`, `javadoc`, and `jar`. There are also CL commands and Operations Navigator interfaces to the AS/400 Developer Kit for Java.

1.8.3.1 Supported Java versions on the iSeries

For V4R5 (and V4R4 equally), the following Java versions are supported:

JVM Version	Install option and comments
Java 1.1.6	5769-JV1 option 1
Java 1.1.7	5769-JV1 option 2
Java 1.1.8	5769-JV1 option 4
Java 1.2	5769-JV1 option 3 (also known as Java 2)
Java 1.3	5769-JV1 option 5 (also known as Java 2 Version 1.3)

1.8.4 AS/400 Toolbox for Java, 5769-SS1, 5769-JC1

The AS/400 Toolbox for Java is a set of Java classes delivered as a Java package. The classes can be used by Java applets and applications to easily access iSeries data and resources and require no additional support over the inherent OS/400 support of Java Virtual Machine and the AS/400 Developer Kit for Java.

The Toolbox for Java provides support for similar functions to those provided by Client Access APIs. It uses the OS/400 host servers (part of OS/400) to access the iSeries data and resources. Each of these servers run in a separate job on the iSeries server, communicating with a Java client program using architected data streams on a socket connection. The socket interfaces are hidden from the Java programmer by the Toolbox classes. JavaBeans are provided for most public interfaces. They provide access to these iSeries resources:

- Remote commands
- Distributed program calls
- Data queues
- Data areas
- System values
- Integrated file system data (extended in V4R4)
- Print
- Record-level access

- Database access using a JDBC on the client
- User spaces
- Digital certificates
- Jobs
- Message queues
- Message files
- Users
- Object authority
- System status

Additional classes provide the infrastructure needed to manage sign-on information, create and maintain sockets connections to the iSeries services, and send and receive data. Data description classes for numeric and character data are provided to allow the Java program to describe the record format of a buffer of data with an object.

The Toolbox also provides a set of Graphical Access classes. These classes use the access classes described above to retrieve data and then present the data to the user. The classes use Java's Swing 1.1 framework. Graphical APIs are available to access various iSeries resources, such as the database, integrated file system, command call, and data queues. The iSeries data is then displayed in various pane formats.

1.8.5 WebSphere Application Server for AS/400

WebSphere Application Server (WAS) is the IBM premier Web application server product that runs on various IBM server platforms including the iSeries server. The IBM WebSphere Application Server provides a framework for consistent, strategic linkage between the HTTP requests and business data and logic. IBM WebSphere Application Server is intended for organizations that want to take advantage of the productivity, performance advantage, and portability that Java provides for dynamic Web sites. It includes:

- Java run-time support for server-side Java servlets.
- JavaServer Pages (JSP). A relatively new technology that provides a server-side scripting technique for generating Web pages. Where Java servlets typically require the skills of a Java programmer, JavaServer Pages can be created by authors who have some basic HTML programming skills.
- High-performance connectors to many common back-end databases to reduce the coding effort required to link dynamic Web pages to real line-of-business data.
- Application services for session and state management.

The WebSphere Application Server is available in three different editions: Standard, Advanced, and Enterprise. On the AS/400e and iSeries servers, we support only Standard and Advanced Editions. Table 1 helps you identify the editions and versions of the WebSphere Application Server that are available for OS/400 V4R3, V4R4, and V4R5.

Table 1. Supported editions and versions of WAS for a given version of OS/400

OS/400 Version	Edition of WAS	WAS Versions
V4R3	Standard	1.1 (only)
V4R4 and V4R5	Standard	2.03, 3.02.2, and 3.5
	Advanced	3.02.2 and 3.5

To find out more about WebSphere, visit the IBM Web site at:
`http://www.as400.ibm.com/products/websphere/`

1.8.6 Lotus Domino for AS/400

Lotus Domino is the world's leading workflow, messaging, groupware, and Web software. Lotus Domino enables you to communicate with colleagues, collaborate in teams, and coordinate strategic business processes on and off the Web.

Powerful, flexible communications
Lotus Domino gives you the power you need to communicate within and beyond your organization. If you need to communicate with suppliers, customers, and partners at other companies that use different e-mail systems, or reach them using the Internet, Lotus Domino makes it easy. Mobile Notes users can take their desktop along with them, transforming airports, hotels, and cars into work spaces complete with up-to-the-minute information. The Lotus Domino family also includes sophisticated client server e-mail, based on the market leading cc:Mail user interface. Lotus Domino applications can be accessed from any Web browser that extends the openness and flexibility of your network.

World-class collaboration and coordination
Lotus Domino goes beyond traditional e-mail and groupware. With Lotus Domino, you can collaborate with team members using a local area network, wide area network, or the Internet. With the unique ability of Lotus Domino to integrate structured and unstructured information into coherent databases, you can organize and coordinate the most complex business processes.

Rapid application development
Lotus Domino allows you to create custom business applications that coordinate everyday business processes from start to finish to achieve results such as improved customer service, improved sales force productivity, and faster time-to-market for products. Lotus Domino customers consistently find significant payback on their Lotus Notes investment, regardless of whether they enable their Lotus Domino applications for the Web. According to the 1999 IDC study entitled *Lotus Notes Agent of Change: The Financial Impact of Lotus Notes on Business* (`http://www.idc.com`), Lotus Notes users achieve an average of 179% annual return on their investment.

Portability and interoperability
Lotus Domino is a server product that runs on a variety of platforms and provides easy-to-manage interoperability in a heterogeneous network. With the sophisticated replication capability of Domino, applications are easily distributed to multiple Domino servers in your enterprise, and just as easily deployed to end users. Replication also simplifies the job of deploying application changes. Lotus Domino applications are also available to any Notes client (such as Windows 95, Windows 3.1, OS/2, Windows NT, and Macintosh). Lotus Domino Version 4.5 and later releases are fully Internet-ready. You can access Lotus Domino server functions from either a Lotus Notes client on your workstation or a browser (including a browser on a Network Station).

1.8.6.1 Domino for AS/400
Domino for AS/400 is the Lotus Domino server product running on a 64-bit AS/400e RISC processor. It requires OS/400 V4R2 or later. Domino for AS/400 provides all the functionality of the Lotus Domino server that runs on other platforms and more.

Domino for AS/400 is an application that is packaged and distributed by Lotus Development Corporation. You may purchase Domino for AS/400 from a Lotus distributor, just like you buy the Domino server product for any other platform. Beginning 20 August 1999, you may also purchase the Lotus Domino Enterprise Server for AS/400 (5769-LNT) as a licensed program from IBM. At the same time, the Lotus Enterprise Integrator (5769-LNP), which was formerly called NotesPump, was also made available as a licensed program for purchase from IBM. The iSeries server continues to be purchased through IBM iSeries channels.

With V4R4, the OV/400 Migration to Domino for AS/400 licensed program allows the migration of users, groups, mail, calendars, and folders to Domino from OffiveVision/400 (OV/400). The Lotus Calendar Connector for

OfficeVision (LCCOV) allows free-time search and the distribution of meeting notices between Domino and OfficeVision/400.

Unmatched scalability
Within a single architecture, the iSeries server spans a vast performance spectrum. The smallest Domino for AS/400 server may have less than a dozen users. The largest iSeries server is capable of accommodating more than 75,000 mail users on a single footprint.

World-class reliability and availability
With more than 700,000 systems shipped to over 150 countries worldwide, the iSeries server has earned a reputation as a reliable, undemanding workhorse. iSeries users expect their system to be consistently available, night and day, and the iSeries server does not disappoint. Domino for AS/400 takes advantage of the reliability and availability features of the iSeries server, such as RAID-5, mirrored disk units, and integrated backup capability. Each Lotus Domino server runs as an OS/400 application in its own subsystem. The unique architecture of OS/400 makes it safe to run your Lotus Domino server and your mission-critical business applications on the same iSeries server.

Powerful integration
Domino for AS/400 includes integration between Lotus Domino databases and DB2 UDB for AS/400 databases. Both real-time and scheduled integration of databases are available to meet a variety of application needs.

Automatic synchronization between the Domino Directory and the iSeries System Distribution Directory provides a powerful, integrated mail server for organizations with multiple e-mail products, including OfficeVision/400, POP3, and Internet mail.

The Lotus Enterprise Integrator option to synchronize authorizations between DB2 UDB for AS/400 databases and Domino databases is platform exclusive.

Proven security
Integrated, flexible security is a long-standing strength of both Domino and the iSeries server. The iSeries server maintains a high reputation for security.

1.8.7 Other IBM products

In addition to the operating system, licensed program solutions are available to take the iSeries server beyond base requirements. The products discussed in this chapter represent those that are most beneficial to an e-business environment. Additional products are offered, but not covered within this

e-business handbook. Contact your IBM marketing representative for further information, or refer to the *iSeries Handbook*, GA19-5486.

1.8.7.1 IBM Payment Manager for AS/400 V2.1, 5733-PY2

The IBM Payment end-to-end suite of products helps enable more-secure commerce over the Internet.

The IBM WebSphere Payment Manager Version 2.1 (formerly IBM Payment Server Version 1.2, which was originally IBM CommercePoint eTill Version 1.0) is a secure electronic cash register for merchants. This is an offering for Service Providers to host payments for multiple remote merchants. It allows merchants to receive payments from consumers on the Internet and to process those payments with banks and financial institutions. It provides functions similar to the physical cash register in traditional stores. It calculates and stores payment information, including split payments, and interfaces with financial institutions to get payment authorizations, refunds, deposits, and other credit card payment functions. IBM Payment Manager can be extended through payment cassettes to support multiple payment protocols and multiple currencies.

The strength and flexibility of the Payment Manager program is complemented with the level of security with which the transactions are conducted. The product architecture, which is designed to accept emerging payment modules representing additional payment systems, also offers strength and flexibility in the Payment Manager program. The Payment Manager program implementation includes the SET protocol, which can provide enhanced security for these systems.

The SET protocol, created and maintained by MasterCard and Visa, with help from IBM and others, defines the actions and security of the card holder, the merchant, and the acquirer when functioning across the Internet.

In addition to the SET protocol, a cassette (a software plug-in to an architected interface used to expand the number of payment types) for CyberCash is also available.

For additional information about IBM Payment Manager for AS/400, refer to the Web site: `http://www-4.ibm.com/software/webservers/commerce/payment/`

1.8.8 Third-party products

IBM recognizes the benefits and need to supplement iSeries e-business solutions with non-IBM software. As the iSeries server remains the leading application development system in the world (with over 700,000 shipments

worldwide), many of these systems are installed in our business partner locations. Truly, the iSeries server is what it is today because of the relationship with our third-party solution providers.

The iSeries server is flourishing with a multitude of software offerings. As the interest, growth, and value of e-business have increased, the offerings from our business partners have also increased.

It is beyond the scope of this handbook to describe all of the options that are available and is too exhaustive to “simply” list all of the offerings. Therefore, we chose to highlight some of the available offerings, with a strong reminder to you that the solutions represented are just a few of the total available.

Refer to Chapter 12, “B2B: Application solutions” on page 261, for a description of some non-IBM solutions.

1.9 Testimonials and quotes

Testimonials and quotes give a practical view of how technologies can be implemented. The objective of those described in this section is to profile some typical implementations by iSeries customers who have benefitted from the tools and technologies in this redbook. Several chapters in this redbook also have their own product-specific references.

1.9.1 Case study 1: J. Leleux & Cie S.A. (Belgium)

“The Internet has enabled us to survive and prosper in a highly competitive market. We could not have taken advantage of these new technologies without IBM.” *Olivier Leleux, Director, J. Leleux & Cie S.A.*

In Belgium, the privilege of buying and selling stock on the Brussels Stock Exchange has always been reserved for licensed stockbrokers.

However, in the last five years, legislation has changed to allow banks and other financial institutions to trade on the market. Consequently, independent stockbrokers have lost their biggest customers. What’s more, they have watched their biggest customers become their fiercest competitors by offering investors lower brokerage fees with the convenience of banking and investing through the same institution.

To stay competitive, J. Leleux & Cie S.A., one of the last surviving independent investment firms in Belgium, has refocused its business from buying and selling stocks to portfolio management and financial consulting. The 70-year-old, $5 million company already had a loyal customer base of

5,000 customers. Even so, J. Leleux knew that it needed to reach a wider audience and offer more convenient services at a competitive cost. It viewed the Internet as a powerful means of achieving this goal.

Although J. Leleux already had a Web site, it offered only static pages of financial analyses and other reference information. With the help of IBM, J. Leleux developed and launched the first online brokerage service in Belgium, JLC OnLine (`http://www.leleux.be`). This service allows investors to view real-time stock performance information and submit orders to buy or sell stock on the Brussels Stock Exchange.

“With JLC OnLine, we're adding about 20 new clients every day,” says Olivier Leleux, director of the company and grandson of the founder. “Considering that it's taken us nearly 12 years to build up a base of 5,000 clients, attracting 500 new clients a month, an increase of 10 percent, is very significant growth.”

1.9.1.1 Leveraging investment in hardware and software

Leleux says that even though his competition would also like to offer online brokerage, many have neither the hardware that can scale up to meet the demands of such an application, nor the finances to abandon their existing systems and start from scratch. J. Leleux overcame these obstacles by leveraging its existing IBM hardware and software.

Leleux explains, “Because our brokers were already doing real-time transactions with the Brussels Exchange on our IBM AS/400e server, we simply needed to provide an Internet window to our existing platform. In addition, by using Lotus Domino for AS/400, we were able to build JLC OnLine as a Web shell right on top of our existing site. This allowed us to develop the online service in just three months and saved us about $10,000, which is the cost of building a whole new site.”

Financial developments, corporate background, and stock analyses are stored in an IBM DB2 UDB database on an IBM AS/400e Model 620 system, which is linked to the Brussels Stock Exchange. Lotus Domino was used to establish direct e-mail links to the company's stockbrokers. Domino also runs on an IBM AS/400e Model 170 server and makes it easy for the company to update its Web site. IBM MQSeries is used to enable the AS/400 database to communicate with the Domino Web server. A third AS/400e server, a Model 600, runs IBM eNetwork Firewall to protect the company's data assets from unauthorized users.

1.9.1.2 Internet transactions 100 percent accurate, efficient, secure

According to Leleux, JLC OnLine virtually eliminates the transaction errors that can occur when orders are communicated verbally. In fact, of the 3,000 orders that have been processed through the Internet so far, the company has not made a single error.

“Greater accuracy gives our brokers more time to provide financial counseling. This personal service is a definite competitive advantage over the large banks,” says Leleux.

Despite these advantages, secure transmission of data was a concern. Many of the transactions that investors perform on JLC OnLine involve information that, for obvious reasons, must remain private. Because 128-bit encryption technology cannot legally be exported from the U.S. for security reasons, IBM Denmark has developed software that provides equivalent protection. As an extra level of security, clients are issued a password so that e-mail messages can be authenticated.

Having successfully addressed all these issues, J. Leleux & Cie S.A. feels it is ready to take on the global market and is looking forward to adding to JLC OnLine real-time quote information from the Paris Stock Exchange and other major stock exchanges. “The Internet has enabled us to survive and prosper in a highly competitive market,” Leleux concludes. “We could not have taken advantage of these new technologies without IBM.”

This testimonial delivers the following key messages:

- e-business is real, secure, effective and could be seen as a necessity to survive in extremely competitive and changing markets.
- e-business is not an opportunity only for large companies.
- e-business benefits from the AS/400e and iSeries servers’ strengths.
- e-business gives value.
- IBM delivers on e-business with leading servers and software.
- The integrated architecture of the AS/400e and iSeries servers is key for security, scalability, and investment protection.

1.9.2 Case study 2: Massachusetts Institute of Technology (USA)

The Massachusetts Institute of Technology (MIT) is one of the world's outstanding universities. Education and research, with relevance to the practical world as a guiding principle, continue to be its primary purpose. MIT is independent, co-educational, and privately endowed. It is organized into five schools that contain twenty-one academic departments, as well as many

interdepartmental programs, laboratories, and centers whose work extends beyond traditional departmental boundaries.

MIT is one of the leading research universities in the United States. Between 1997 and 1998, approximately 2,700 researchers worked with faculty and students on projects funded by government, foundations, and industry. MIT employs 2,415 research assistants and 698 teaching assistants. A recent study concluded that university inventions add more than $20 billion and 150,000 jobs to the US economy every year. MIT routinely leads all U.S. universities in patents granted, and in 1999, it signed 99 license agreements with private companies.

MIT wanted to develop an e-commerce portal site to give students and customers Internet access to MIT's public materials. The University also wanted to include a feature that would allow specified audiences to view non-public materials.

Using the IBM Application Framework for e-business methodology, MIT created an e-commerce portal. The site provides students and customers with a simple, quick way to access public MIT materials. The core application includes a repository of searchable research documents. A messaging system alerts registered users when a document in their interest area has been posted to the site. The site also includes a private section that can only be accessed by select users via a registration page. Data on the select users is stored and analyzed in a database. For the solution pilot, MIT gave students Web access to their registration and billing records so they could review their bills and schedules for accuracy.

The site is housed on an AS/400e Model 170 server running the OS/400 operating system. WebSphere Studio, IBM's team-enabled Web project workbench environment, was used to develop the Web pages. WebSphere Application Server is used for the site's transactional functions and connects to the database that houses the documents. Lotus Domino is used to run the site's messaging and registration systems and was customized using Lotus Domino Designer. The Domino server runs on a Windows NT platform and enables students and alumni to send e-mail from their Web browser.

IBM partnered with MIT to implement the solution, and as part of the package, included instructional classes to help the customer use the products and create the solution.

The solution benefits have not been quantified, but MIT can now offer expanded information access to its customers and students. The customer

anticipates significant service and maintenance cost savings from the solution, while providing services to a wider audience.

This example delivers the following messages:

- The IBM e-business vision of the e-business Applications Framework and that it works on the AS/400e and iSeries servers
- The cross-platform solution works with an “open” architecture
- The AS/400e and iSeries servers are the platform of choice for developing a Web site

The December 1999 IDC white paper *Total Cost of Ownership (TCO)* analyzes the IBM AS/400 Dedicated Server for Domino versus PC Servers and shows that the AS/400e for Domino TCO was 51% lower than PC servers. This white paper is available on the Web at: `http://www.idc.com`

The following example shows the synergy between Net.Commerce and the AS/400e server.

1.9.3 Case study 3: ICON Health and Fitness

ICON Health and Fitness wanted to enhance its business by creating a fully functional e-commerce site to make its operations more efficient. More specifically, ICON wanted to develop and implement a site where customers could order products, accessories and parts; check and track the status of present orders; and contact customer service with questions or comments. Security was a major concern for ICON since the primary method of payment would be through their credit card. In addition, ICON desired a solution that was not only competitively priced, but also one that would complement its existing technology.

ICON chose to work with the IBM Business Partner, Assist Cornerstone International, because it was able to provide the most efficient and cost-effective solution that would run on ICON's already established AS/400e environment. The ICON solution was made possible by enabling software such as IBM Internet Secure Server, IBM Firewall, and Net.Commerce.

The first phase of the solution included implementing the Internet Secure Server (HTTP Server for AS/400) and the firewall. At this stage in the solution, ICON has been taking orders (since 09 May 1998) for fitness equipment through a static Web site. However, this site only allows customers to place orders through an online customer service form. Once the second stage of development is complete (implementing Net.Commerce, for example), customers can access all of the fitness Web sites that ICON

represents, peruse the online catalogs, and order and purchase hundreds of fitness and outdoor products and parts. Plus, users of the site can track the status and location of present orders and review the history of orders made over the past several months. Once Net.Commerce is up and running, the customer service area of the sites will remain to provide users with an alternative avenue so they can voice their questions and comments.

ICON plans to continually develop the site, making it more dynamic, to give customers an incentive to return. Plans for future additions include up-to-date price changes, fitness/health tips and advice, and a Q&A section.

AS/400e details

ICON is currently running Net.Commerce on the AS400e server Model 9406-S20. The customer also has AS/400e server Models 9406-640 and 9406-620 installed.

Applications that run on the AS/400e server include: Acacia for Manufacturing software, Assist for Financial software, and various other third-party packages. There are 500 users. ICON has a topology that uses both token-ring and Ethernet.

Teaming up with IBM, ICON has been thoroughly pleased with the results of the initial static ordering site. In fact, since the site's inception in May of 1998, ICON has seen an increase in sales with one order made per 250 hits. This is quite impressive since ICON has not yet launched an official advertising campaign. Once the application is up and running and a proper advertising campaign is in effect, all of the previously mentioned sites promise to boost revenue and sales at ICON and the other companies that ICON represents. As the site improves over the next few months, ICON is excited at the prospect of adding more buyers to its already large customer base.

This customer testimonial reminds us of these key elements:

- AS/400e and iSeries Business Partners play an important role and deliver valuable solutions.
- Net.Commerce on the AS/400e server, or WebSphere Commerce Suite on the iSeries server, can be quickly deployed.
- The line-of-business applications running on the existing AS/400e or iSeries server can be “Web-enabled”.
- e-business is a moving environment that needs a flexible platform for deployment and development.

Part 2. A practical guide for building e-business sites

Part 2, which is written primarily for technical people, covers the major technology components and standards of the e-business world, detailed products descriptions, and implementation road maps. It consists of the following chapters:

- Chapter 2, “Building e-business sites: Phased approach” on page 49
- Chapter 3, “Standards and technologies for e-business products” on page 59

Chapter 2. Building e-business sites: Phased approach

There is no written rule that you have to follow certain steps to develop an e-business site and add functionality to it. However, there is a common or typical pattern in growing an e-business Web site. The pattern is a change, not just in terms of size, but in terms of functionality. In other words, an e-business site evolves through phases:

- Phase 1: Web presence
- Phase 2: Dynamic site
- Phase 3: Transactional site
  - Business to customer (B2C)
  - Business to business (B2B)

This development model is not fixed. Usually customers move through the Web presence phase quickly. Earlier in the e-business wave, having a Web presence meant being “online” on the Internet. Today this is not nearly enough. Most companies skip right past Web presence. For example, you can start into e-business by building a transaction site, such as an online shopping store for the classic B2C, and then move on to a dynamic site.

At any rate, the phased approach is a methodology we adopted in writing this book. It is the same phase used in many other documents, whether hardcopy or online. It is our belief that you will see this pattern in most companies.

Figure 9 on page 50 provides a high-level overview of the three phases, including the typical functionality for each phase.

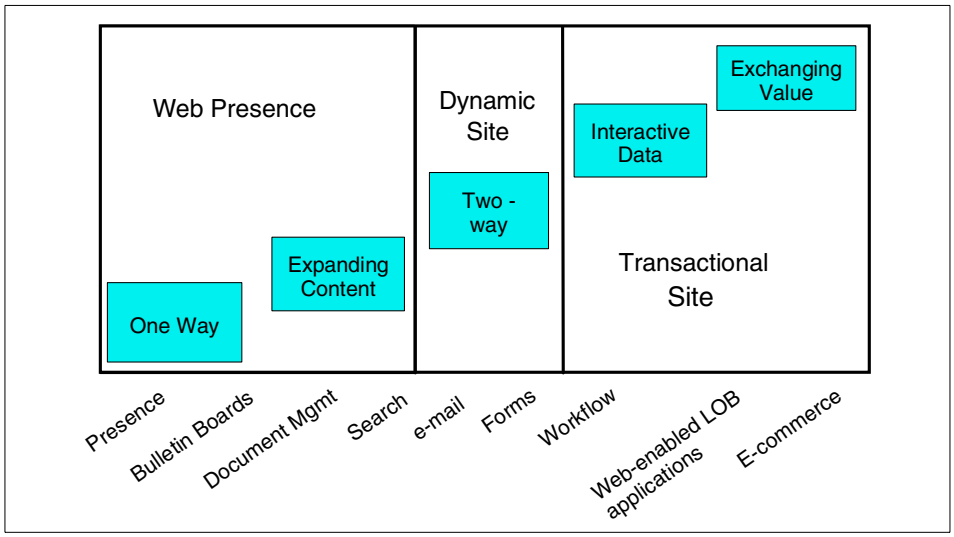


Figure 9. Overview of the e-business evolution phases

2.1 The e-business evolution phases: Description

This section describes each phase of the e-business evolution in more detail. To understand the difference between the different phases in more practical terms, we use the imaginary company of *ITSO-Rochester Investment Firm* for an example. We also include some tips on why the iSeries server should be considered as the first choice in each stage and overall.

2.1.1 Phase 1: Web presence

As the Web continually grows and quickly becomes a place where companies must have presence to survive in today's competitive world, your Web site gains importance as a company information resource.

The first step, or phase, is Web presence. This involves presenting marketing, product, and company information on a relatively static HTML home page or Web site.

Web presence is an easy and low-cost way to distribute information about your business. Establish a Web presence and millions of people around the world are your potential audience. In addition, the Internet offers:

- **Speed**: Instantly deliver information anywhere in the world.
- **Availability**: The Web is open for business 24 hours a day, 7 days a week.

- **Scope**: Send and receive information to and from anyone, anywhere, and anytime so long as a connection exists.

The iSeries server is necessary for establishing a Web presence because of its history of minimal downtime and powerful Web serving capacity. Also, all of the software necessary to publish a Web presence is packaged with the iSeries operating system (OS/400) at the time of purchase.

As the Internet has evolved and forces shape individual businesses, information access requirements also evolve. When this happens, Web presence typically moves toward providing dynamic data.

An example: ITSO Rochester Investment Firm
An investment firm publishes brochures, contact information, services offered, and other marketing related information. Without the Internet, the customer interacts with the company using the telephone or meetings. Account information is mailed or accessed by a telephony application and trades are done by telephone.

The IT manager of the ITSO Rochester Investment Firm started this exercise by asking, “Where is my data that I want to make available to my customers?” The answer was, “on the iSeries”, which made the iSeries the best platform to serve that data.

2.1.2 Phase 2: Dynamic site

Dynamic data sites contain user-defined information that is extracted from business systems, as per the users request. The user, however, does not add, delete, or edit the business data. Nor does the user complete a transaction. Customers are satisfied with this improved method to access data that traditionally was provided by call centers or 1-800 telephony applications. These solutions are often seen in business-to-business applications and consumer self-service Web sites.

An example: ITSO Rochester Investment Firm
The firm can start to offer account information on a Web site where customers can login and view the current balance based on that particular day’s trading. Customers still place orders by phone.

2.1.3 Phase 3: Transactional site

The final phase in this e-business process is to have a transactional site. A transactional site allows users to add, delete, or edit data on business systems and to complete transactions. This is known as *e-commerce*. This type of site is growing very quickly for business-to-business and

business-to-consumer solutions as the demand for online commerce by businesses and consumers increases.

An example: ITSO Rochester Investment Firm
The firm sets up an application to allow customers to trade various financial instruments through their firm. This can replace placing orders by telephone.

2.2 The e-business evolution phases: Technology view

Figure 10 illustrates a high-level view of the technology presented at each of the three phases of e-business.

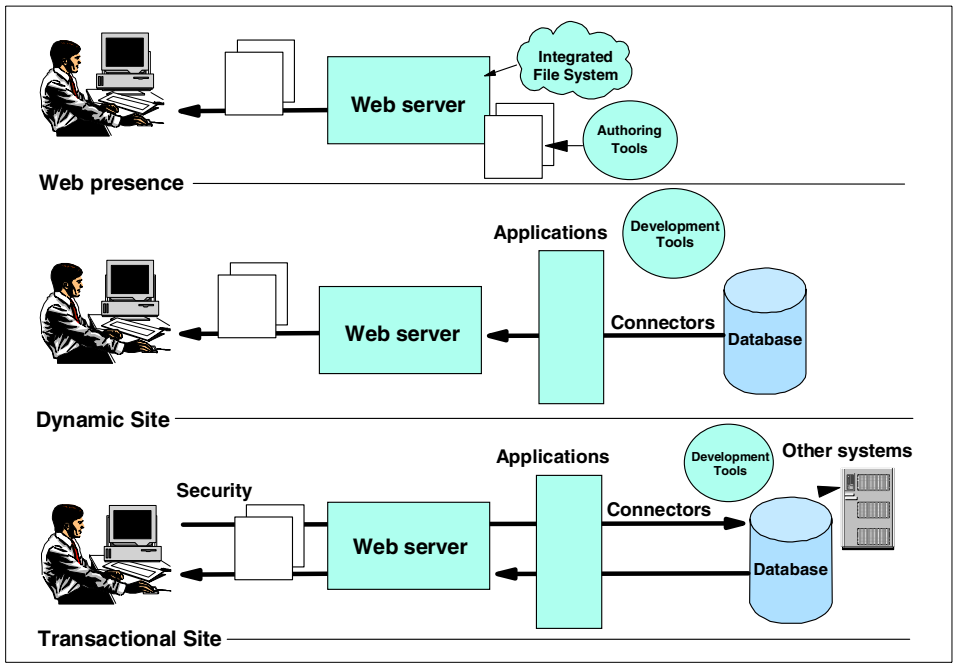


Figure 10. Technology overview of the phases of e-business adoption

Figure 10 is further explained here:

- **Web presence**: Users interact only with a Web server, and the information presented is static. Technically this requires the lowest level of security and no integration with host systems. Any information published is developed and deployed manually.
- **Dynamic site:** Users interact with databases, and the Web server and the users can select and determine the information presented to them. This

solution requires greater security, especially in communications, usually encryption or digital certificates. The business must now manage access to host data and the application through which the customer accesses it.

- **Transactional site:** Users interact with host business applications and processes to add, delete, or update information and to complete transactions (buying and selling). Security requirements are highest with communication and transaction security required (SET and so on). Transactions now occur, and the business needs to manage a variety of systems management and integration issues.

2.3 The e-business evolution phases: Business view

This section portrays the business view of the e-business phase for each technology component described in this redbook.

2.3.1 Web presence

The first phase, Web presence, involves your company publishing static Web pages onto the World Wide Web. As a result, your company has a Web site and a presence. Figure 11 shows how a business can use Web presence technology.

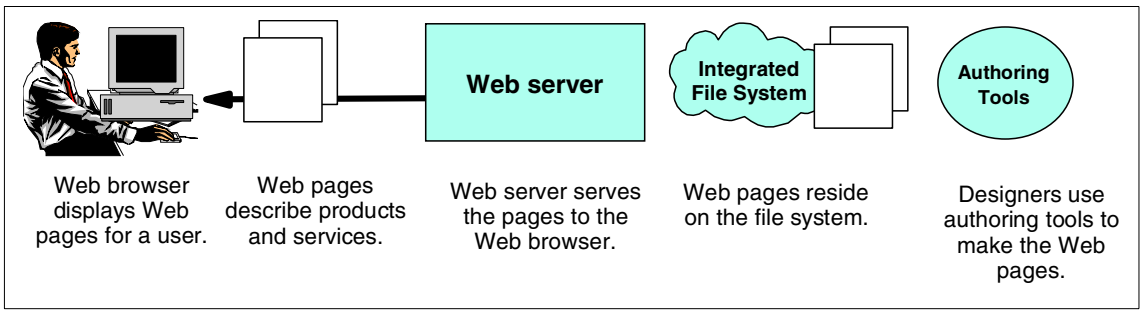


Figure 11. Business view of the Web presence phase of e-business adoption

2.3.2 Dynamic site

The second phase, dynamic site, turns the static information on your site into dynamic information. The Web server retrieves information from databases to display on a client machine. Figure 12 on page 54 shows how a business can use dynamic site technology.

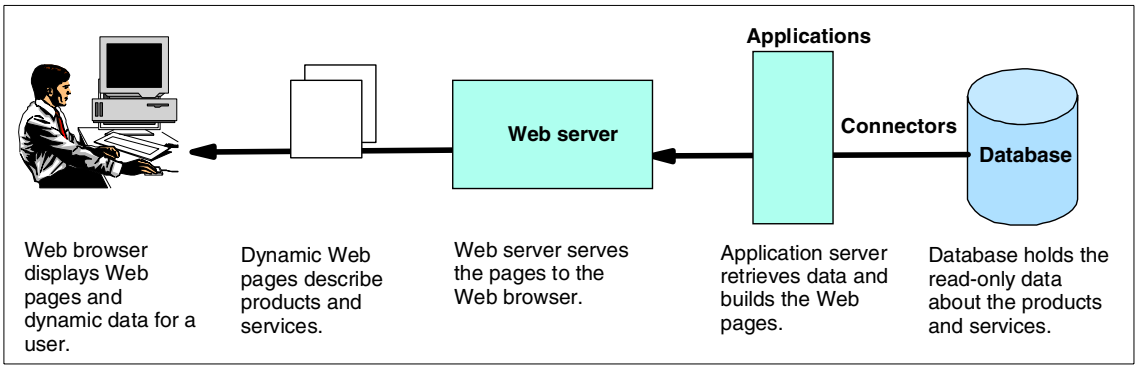


Figure 12. Business view of the dynamic site phase of e-business adoption

2.3.3 Transactional site

The third phase, transactional site, allows people to interact with the information presented on the Web pages. Therefore, data on your inhouse system is changed to reflect their interaction. Figure 13 shows how a business can use the transactional site technology.

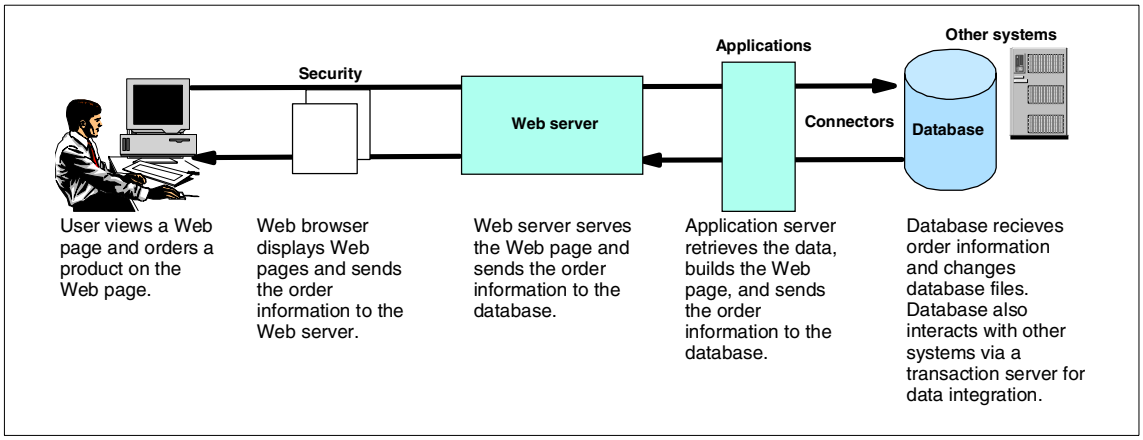


Figure 13. Business overview of the transactional site phase of e-business adoption

2.4 The e-business evolution phases: Products view

The three views shown in this section combine the technology with the IBM products that support it. The remainder of this redbook reviews the various technologies highlighted here.

2.4.1 Web presence

The Web presence stage is perhaps the simplest in regard to products. To establish a presence on the Web, your company needs a Web server that can retrieve files from an integrated file system. Your users have the client software, a Web browser, to view the pages you place on the Web server. Figure 14 shows which products support Web presence technology.

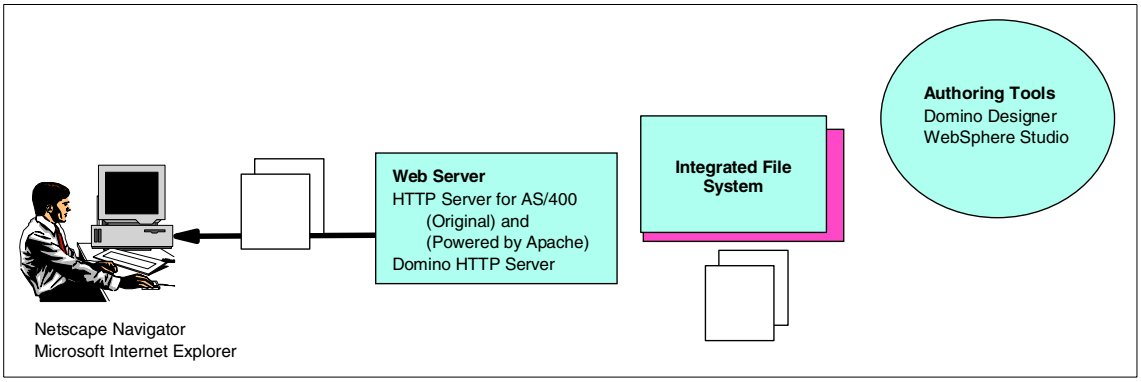


Figure 14. Product view of the Web presence phase of e-business adoption

At its simplest, a Web site serves static Web pages. “Static” does not imply that the information on the Web site is unchanging. You must constantly refresh your Web content to keep your visitors coming back. Rather, “static” means that the content of what the visitor sees is not changed by user interaction. It's like reading a newspaper. The page that you see is the same as the page that you see a few minutes later. But tomorrow, brand new newspaper pages are available.

In simple terms, the role of the HTTP server is to receive requests from browsers for Web pages, locate the pages, and send them to the requesters. The browser communicates with the HTTP server using URLs that contain the location of the pages that the user wants. To process that URL, the HTTP server uses a set of directives that the site administrator has created. The directives may have some mapping to tie a URL to the real directory where the page is located (to isolate your Web site design and the user from your underlying storage structure). The directives may also have protection information that requires user ID and password for certain information-sensitive Web pages.

2.4.2 Dynamic site

The nature of a dynamic site is that it changes. Over time, organizations have seen the benefit of interacting with browser users by sending and receiving data. These interactions range from the simplest transaction, such as collecting the name and address of a browser user who wants to receive a catalog, to displaying order status and actually processing orders. In general, these interactions start with a form: a Web page that contains input-capable fields and push buttons (like function keys). The information that a user views through their Web server changes based on the information that lies in the database. Figure 15 shows which products support a dynamic site technology.

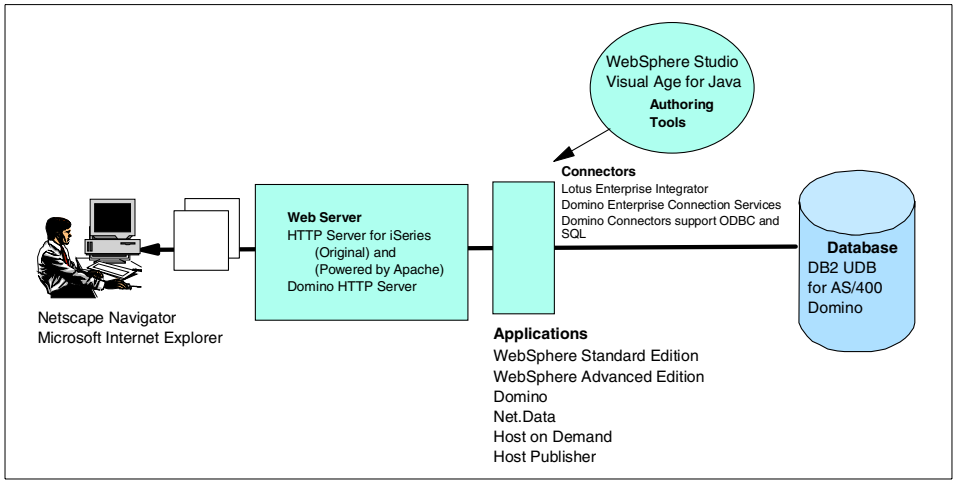


Figure 15. Product view of the dynamic site phase of e-business adoption

2.4.3 Transactional site

With a transactional site, the user can interact with information, for example, order a product or fill out a form. Applications and connectors ensure that database information reflect these changes and the user sees the correct information over the Web browser. Figure 16 shows the products that support a transactional site technology.

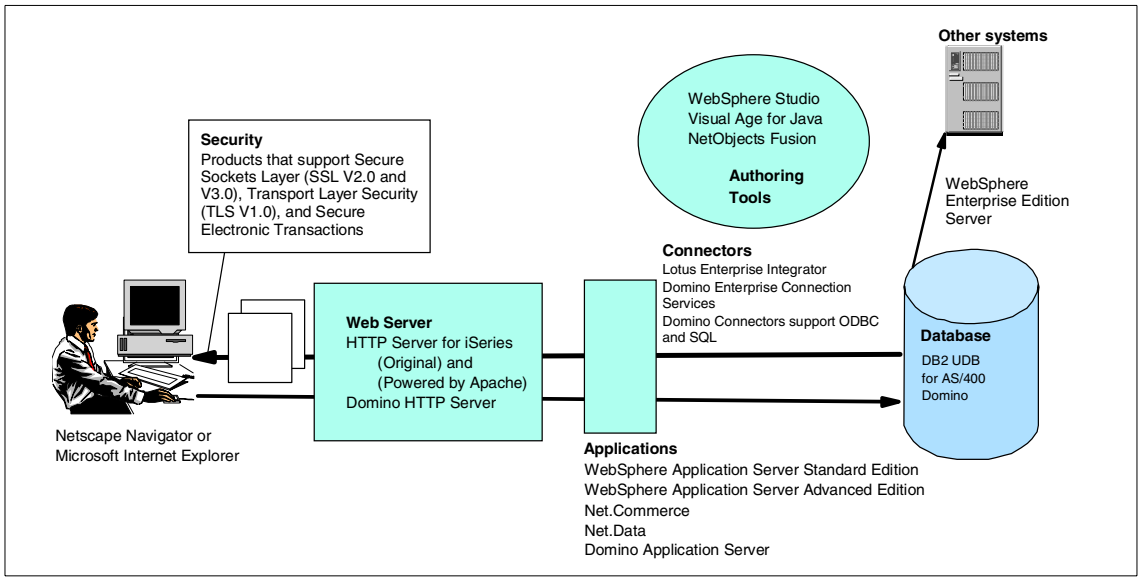


Figure 16. Product overview of the transactional site phase of e-business adoption

IBM provides numerous e-business products for iSeries customers to build, run, leverage, and transform their businesses. Depending on which phase in the e-business development model you are in, different tools and technologies are appropriate. The remainder of this redbook discusses these technologies and products in detail. It also gives you in-depth positioning, best-use scenarios, case studies, and product information so you can understand the IBM suite of offerings and apply the appropriate technology to your situation.

Chapter 3. Standards and technologies for e-business products

This chapter discusses the major standards and technologies of e-business that the industry has or is adopting. We focus on those standards related to applications and database serving that are the essential components in linking Web browsers and the server system. By definition, they are platform independent, but we include iSeries-specific information wherever appropriate.

3.1 Applications and database serving standards: Non-Java based

This section discusses two non-Java based standards for applications and database serving in the e-business environment: CGI and Net.Data.

3.1.1 Common Gateway Interface (CGI)

On most non-iSeries Web serving platforms, CGI programming implies working with scripts written in the PERL language. Although there is an unsupported version of PERL available for the iSeries server, most iSeries CGI programs are created using ILE RPG, ILE COBOL, ILE C, or Java.

3.1.1.1 Understanding CGI processing

Figure 17 shows CGI program processing and how CGI programs run on the iSeries server.

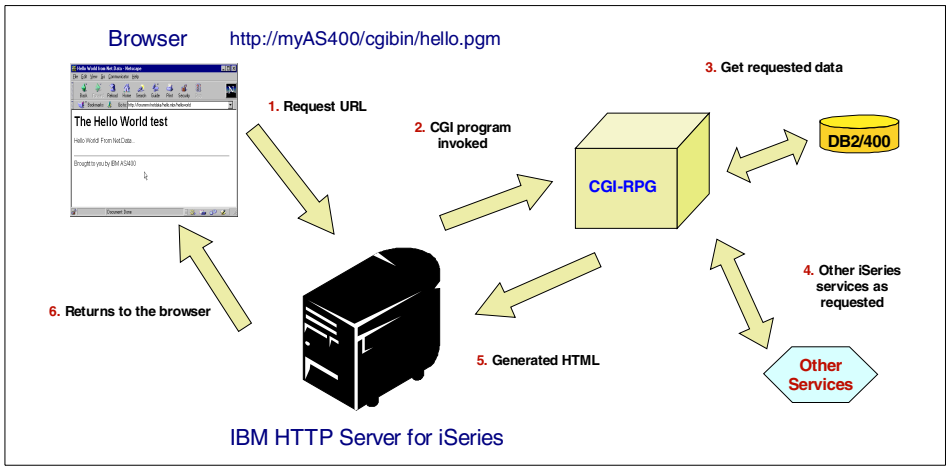


Figure 17. How CGI processing works

The process shown in Figure 17 on page 59 is explained here:

1. A CGI program is requested on an incoming URL. When a browser user enters a URL containing a request in a Web page, the request is sent to the HTTP Server for iSeries with data that was entered on the Web page.
2. The CGI program is invoked. The HTTP Server for iSeries finds the CGI program. The program name is specified in a MAP, PASS, or EXEC directive of the HTTP configuration file.
3. The CGI program retrieves the data that was entered by the browser user. Since the CGI program is now invoked, it runs as the traditional iSeries program to open files, read and write records with the DB2 UDB for AS/400 databases, run SQL statements, or call other user programs, commands, or APIs.

   At this point, the CGI program is conceptually similar to one of iSeries traditional programs. It receives input from the browser form, works with DB2 UDB for AS/400 database files, and prepares a response to the browser.
4. The resulting HTML is returned to the HTTP Server for iSeries program. When the response HTML to be returned to the browser is completed, the CGI program uses API calls to send the resulting HTML to the HTTP Server for iSeries program.
5. The resulting HTML is sent back to the browser. HTTP Server for iSeries sends the completed HTML page to the browser. The process is complete.

3.1.1.2 Benefits and considerations of using CGI

This section discusses the benefits and considerations of using CGI programs.

Benefits of using CGI

The primary reason to use CGI programing on the iSeries server is that you or your staff may already be familiar with one of the iSeries server programming languages such as ILE RPG, ILE COBOL, and ILE C. For example, if you already know RPG, it is relatively simple to learn how to incorporate CGI processing techniques into an RPG program, compared with learning Net.Data or Java.

When you write a CGI program, you have access to all of the iSeries server programming tools. For example, you can use native database operations in your CGI programs. You can also use string handling operations in the language to create the exact HTML statements that you need. Finally, you can use the same debugging tools to put your CGI program into production.

Considerations of using CGI

A major concern of using CGI compared to other options (predominantly compared to Java-based technologies) is that it is somewhat labeled as legacy technology with reasons.

Generally there is considerably more code required for an RPG CGI program as compared to Net.Data. Once you create a working RPG CGI program, you can easily copy code for further programming.

The most difficult part of the RPG CGI program is the code that parses the buffer returned from the QzhbCgiParse API. Again, once you develop several programs using this API, you can extract the routines easily.

Positioning of CGI

CGI programming can be your optimal choice to Web-enable your iSeries server applications if your primary goal is to use existing programming skills.

3.1.1.3 Location of the CGI product on the iSeries server

CGI is supported on the iSeries server using IBM HTTP Server for iSeries (5769-DG1). It is shipped with OS/400 at no additional charge.

3.1.1.4 References

There are many users who have already started their dynamic sites using CGI and Net.Data.

Turn to the following sources for more information:

- *Unleashing AS/400 Applications on the Internet*, SG24-4935, gives instructions enabling an AS/400 application to the Internet including CGI and Net.Data.
- *Cool Title About the AS/400 and Internet,* SG24-4815 (Chapter 9), discusses the application development interfaces including CGI and Net.Data programming.

3.1.2 Net.Data

Net.Data is a server-side scripting language that is packaged as part of the IBM HTTP Server. Net.Data extends Web servers by enabling the dynamic generation of Web pages using data from a variety of data sources. The data sources can include relational and non-relational database management systems such as DB2, DRDA-enabled databases, and flat file data. Net.Data applications can be rapidly built using a macro language that is simple, yet powerful. Net.Data allows the reuse of existing business logic by supporting calls to applications written in a variety of programming languages, including

Java, C/C++, RPG, COBOL, CL, REXX, and others. Net.Data provides several features for high performance, including persistent connections to databases.

Using macros that you develop as input to the CGI program, Net.Data allows users to achieve the same tasks as CGI programs do without writing their own CGI programs. The system provided Net.Data CGI program, Net.Data, uses the macro to:

- Send HTML to your browser
- Run SQL commands
- Call system services such as programs compiled in other languages

3.1.2.1 Understanding Net.Data processing

Figure 18 shows how Net.Data processing works.

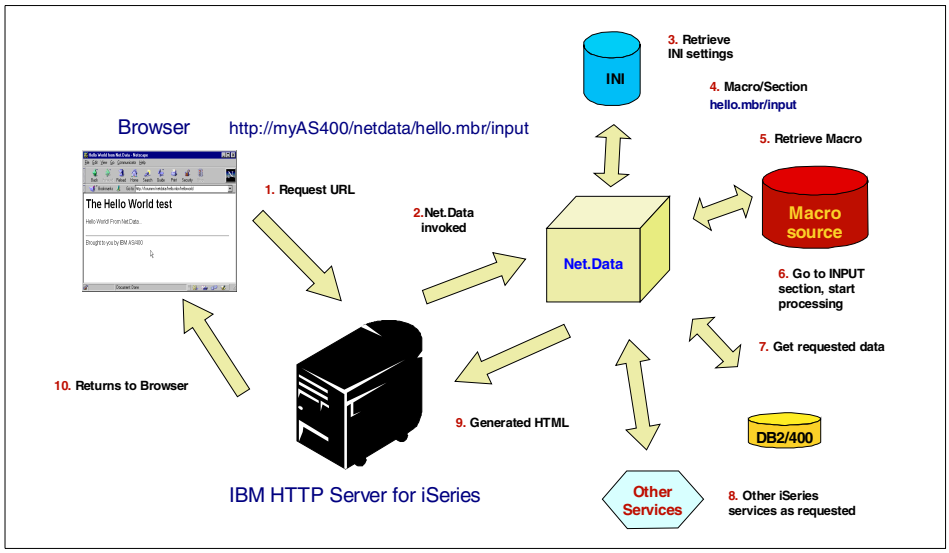


Figure 18. How Net.Data processes a macro and generates a response

This process is explained here:

1. A Net.Data macro is requested on an incoming URL.

   The request is sent through the URL to the IBM HTTP Server for iSeries program.

2. Net.Data is invoked.

   The IBM HTTP Server for iSeries program recognizes that the request is for Net.Data. Then, IBM HTTP Server for iSeries invokes Net.Data.

3. Net.Data configuration options are retrieved from the INI file.

   Upon starting, the Net.Data program retrieves initialization options from the optional INI file. Although it is optional, if you do not have an INI file, the URLs are considerably more complicated.

4. The macro and start-at section within the macro are identified.

   Net.Data determines where on the system the macro source file is located.

5. Net.Data now retrieves the macro.

   All Net.Data macros are stored in text format.

6. Net.Data starts the execution of the macro at the start-at section.

   Net.Data parses the macro, and any global function calls are processed. The start-at section is typically an HTML block that contains statements describing the initial page to be sent to the browser.

7. DB2 UDB for AS/400 data is processed with SQL statements.

   If there are any SQL statements or Net.Data function calls to other iSeries services, those statements or function calls are now processed.

8. Other system services are invoked.

   In a typical Net.Data macro, you embed SQL statements or function calls within HTML statements. Net.Data runs the SQL statement or function call at the point where it is encountered. The resulting HTML sent to the browser can include your headings and footings with the merged output of an SQL statement or function call.

9. The resulting HTML is returned from Net.Data to the IBM HTTP Server for iSeries program.

10. The resulting HTML is sent back to the browser.

    IBM HTTP Server for iSeries sends the complete HTML page back to the browser. The process is completed at this point.

3.1.2.2 Benefits and considerations of using Net.Data

This section discusses the benefits and considerations of using Net.Data.

Benefits of using Net.Data

Net.Data is a superb tool for creating simple HTML forms that act as front ends to database queries. In addition to the default behavior of Net.Data (display SQL results in an HTML table), you can customize the resulting table or use Net.Data functions to add other HTML options to the table, such as listbox fields and checkbox fields.

If you do not have iSeries programming skills (usually RPG, COBOL, or C) or if you are uncomfortable with using CGI APIs required to get, parse, and return data to the browser, you may find it easier to work with Net.Data. You can concentrate on the application, rather than the mechanics of communicating with the browser.

Net.Data also provides built-in support for working with the results of SQL queries. It takes care of getting and parsing requests from the browser and preparing output to return to the browser.

Net.Data is available on a variety of platforms, including AIX, OS/2, OS/390, and Windows NT.

Considerations of using Net.Data

A major concern of using Net.Data, especially when you expect a heavy transaction at any given time, can be a performance issue. Net.Data is interpreted, not compiled. This can be a benefit on one hand. For example, you can develop or make changes more quickly than the equivalent compiled program. On the other hand, it can cause some performance concerns.

Positioning of Net.Data

Net.Data can be your optimal choice to Web-enable your iSeries server applications if your primary goal is completing the project without serious investment on internal or external programming skills acquisition.

3.1.2.3 Location of Net.Data product on the iSeries server

Net.Data is supported on the iSeries server using IBM HTTP Server for iSeries (5769-DG1). It is shipped with OS/400 at no additional charge.

3.1.2.4 References

There are many users who have already started dynamic sites using CGI and Net.Data. Refer to the following publications for more information:

- *Unleashing AS/400 Applications on the Internet*, SG24-4935, gives instructions for enabling an AS/400 application to the Internet including CGI and Net.Data.
- *Cool Title About the AS/400 and Internet,* SG24-4815 (Chapter 9), discusses the application development interfaces including CGI and Net.Data programming.

3.2 Applications and database serving standards: Java based

This section provides a brief review of the Java environment. Then, it discusses Java-based applications and database serving standards including:

- Java applets
- Java servlets
- JavaServer Pages (JSP)
- Enterprise JavaBeans (EJB)

Java is both a programming language and a platform. As a high-level programming language, Java is unique in its architecture-neutral, object-oriented configuration.

The Java language is unusual in that each Java program is both compiled and interpreted. With a compiler, you can translate the Java program into an intermediate language known as *Java bytecodes*. These are platform-independent codes interpreted by the Java interpreter. With this interpreter, each Java bytecode instruction is parsed and run on the computer. Compilation happens once. Interpretation occurs each time the program is executed.

Think of Java bytecodes as the machine code instructions for the Java Virtual Machine (JVM, the base for the Java platform). Every Java interpreter (for example, a Web browser for running applets or a Java development tool) is an implementation of the JVM. The JVM can also be implemented in hardware. The basic structure of the Java environment is shown in Figure 19.

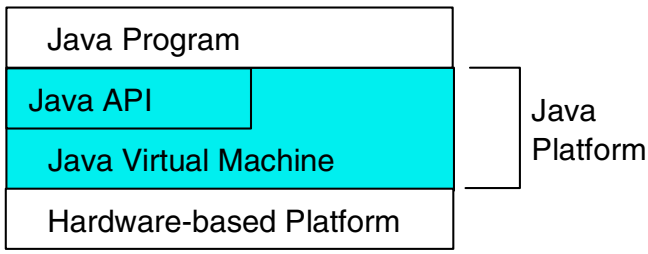


Figure 19. Basic structure of the Java environment

Note how the JVM allows for platform independence from the processor.

The Java bytecodes help make Java's famous "write once, run anywhere" capability possible. Once the Java program is compiled into bytecodes on any platform having a Java compiler, these bytecodes can run on any

implementation of the Java Virtual Machine. The same program, for example, can run on the iSeries server, Windows NT, or Sun Solaris.

The Java *platform* refers to the environment in which a program runs. While most platforms consist of both hardware or software (for example, Intel-based Windows NT), the Java platform differs in that it is a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

The JVM has already been mentioned. The Java API is a large collection of ready-made software components that provide many capabilities, such as GUI widgets. The Java API is grouped into libraries, or packages, of related components.

Java manifests itself in the e-business arena in several forms, including applets, servlets, JavaServer Pages, and Enterprise JavaBeans.

3.2.1 Java applets, servlets, JavaServer Pages, and Enterprise JavaBeans

Java applets are dynamic and interactive programs that can run inside a Web page displayed by a browser enabled for Java, such as Microsoft Internet Explorer or Netscape Navigator. Applets were the first application of Java to gain widespread notice. They first gained popularity in the mid-1990s as a tool to add animation and other effects to Web pages.

Browsers that provide support for applets include:

- Netscape Navigator Version 4.04 or later
- Microsoft Internet Explorer Version 4.01
- Sun Microsystems HotJava

Java applets are delivered to the browser in HTML files that contain a reference to the applet. Upon arriving in the browser, the applet begins execution. The applet usually displays what appears to the user as a Web page. However, all of the user interface elements are contained within the applet itself and are not rendered by HTML.

Most importantly, the applet can communicate directly back to the server. When the server responds, it is up to the applet to display the response. Because the applet is in control of its user interface elements, it can update them to display the server response, again without requiring HTML.

3.2.1.1 Java applet processing on the iSeries server

One of the most important features of applets for iSeries developers is the ability of the applet to interact with the iSeries server. When you create an applet, you can include Java classes from the AS/400 Toolbox for Java that let you access objects on the iSeries server.

For example, you can:

- Access records from the iSeries database using SQL statements or record-level access techniques
- Call programs or invoke commands on the iSeries server
- Send and receive entries from data queues on the iSeries server

Figure 20 shows an overview of applet processing.

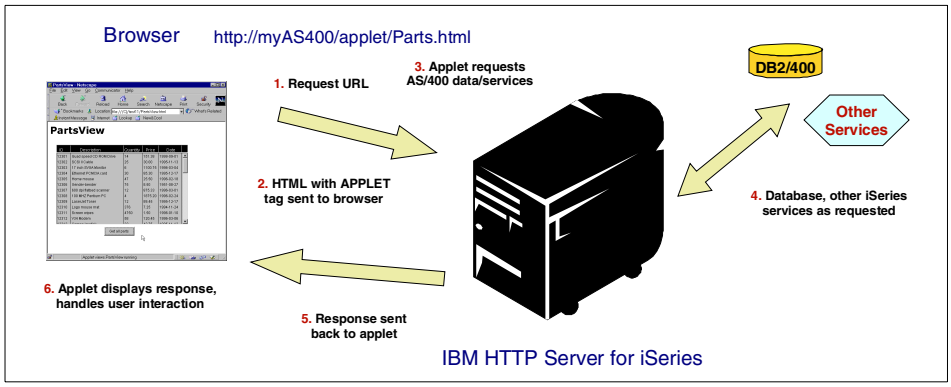


Figure 20. Applet processing

Each step of applet process is explained here:

1. An HTML page that contains an `APPLET` tag is requested on an incoming URL.

   You start an applet by requesting an HTML file that contains an `APPLET` tag.

2. A requested HTML page is returned to the browser.

   As the page is sent to your browser, the Java classes used in the applet are also sent to the browser. After receiving the applet, the browser starts it and passes control to the applet.

3. A request is sent from the applet to the iSeries server.

   Once the applet is started, you interact with it as you would with other forms displayed in the browser. The applet may require that you enter data or make selections. You usually have one or more buttons in the applet

that you can click. The buttons are not associated with a FORM statement in an HTML form, but rather are used to invoke methods in the applet's Java code. It is up to the Java code in the applet to use methods to send requests to the iSeries server.

4. The iSeries server services the request from the applet.

   The request from the applet is serviced the same as other types of program requests running on the iSeries server. For example, if the request was to create an SQL result set, the SQL processor is invoked to query the iSeries database.

5. The response is sent back to the applet.

   After the iSeries processes the request, the results of servicing the request are sent back to the applet. The results are in the format that pertains to the request. For example, an SQL request generates a result set. A data queue read operation generates a packet of bytes that contains the data queue entry. The applet receives the results using the Java class and methods appropriate to the request.

6. The applet displays the response data.

   Once the results are available to the applet, it displays those results using any of the user interface components included in the applet when it was designed. The display is not limited to HTML only. After formatting and displaying the results, the applet is available for additional user interaction, which can include additional requests to the applet for more data from the iSeries server.

3.2.1.2 Benefits and considerations of using applets

This section discusses benefits and considerations of using applets.

Benefits of using applets

Applets are useful when you need to work with iSeries resources and you do not want to create Net.Data or CGI programs on the iSeries server. You can also choose to work with applets to have greater control to design the user interface and interact with the user at run-time.

Another reason to use applets is to use the industry standard Java language, rather than the proprietary AS/400 Net.Data or CGI programming languages. Although the Java classes to access the iSeries server are proprietary, the classes are readily usable by any Java programmer with basic knowledge of the iSeries server.

Considerations of using applets

Applets depend on the capabilities of the browser under which they run. Browsers are available from numerous vendors. Each of these browsers use a different security model. You may have to program an applet differently based on the browser under which you are running. In other words, there is a chance that your application may not run as it was designed depending on your clients' browser environment.

Another point of consideration is the size of the applets and communications link speed. Applets and their associated classes should be stored at a central location and downloaded as needed. This results in downloading the classes as needed by the browser. Depending on the size and number of classes, and the speed of the communications line, the download time can be an expensive use of resource and time.

Location the Java applet product on the iSeries server

Java applet is supported on the iSeries using IBM HTTP Server for iSeries (5769-DG1). It is shipped with OS/400 at no additional charge.

3.2.2 Java servlets

Java servlets have many advantages over their alternatives, such as Net.Data or CGI programming, or Java applets for a number of reasons. Two of the main advantages are in regard to:

- Performance
- Ready access to the iSeries database

Here, the advantages are because Java servlets run entirely on the iSeries server as part of the IBM HTTP Server for iSeries. Therefore, the servlets can access the iSeries database and other system resources with much less interactions over the network link.

3.2.2.1 Java servlet processing on the iSeries server

If you are already familiar to CGI programming, covered earlier in 3.1.1, "Common Gateway Interface (CGI)" on page 59, servlet processing is similar. The primary difference is that the servlet can be pre-started in the multi-threaded job pool so that there is no start-up overhead when it is invoked.

If a servlet is not currently active when it is invoked, you incur the start-up overhead on its first usage. After that point, the servlet is available for subsequent invocations.

Figure 21 on page 70 shows an overview of servlet processing.

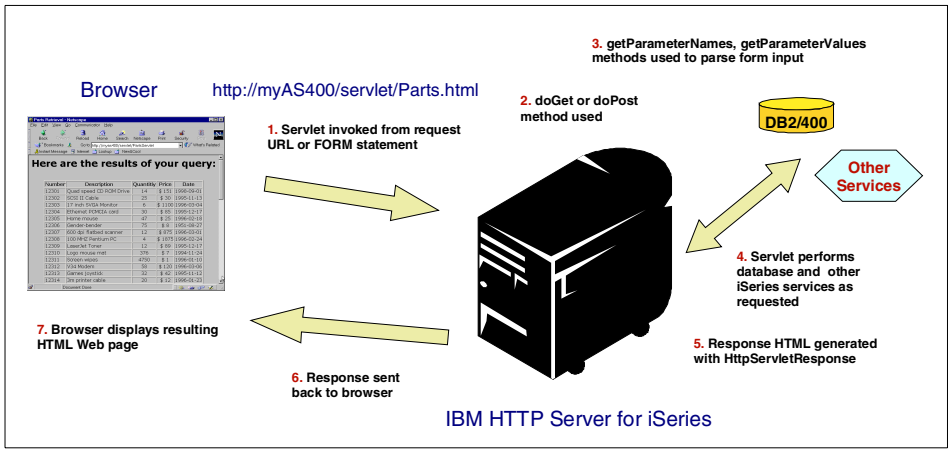


Figure 21. Servlet processing

Each step in the servlet process is explained here:

1. The servlet is invoked from a URL or `FORM` statement.

   The servlet is identified either in a URL that you type into the browser or click as a link. Or, it can be specified on an HTML `FORM` statement that is used when a Submit button is clicked. The request is sent using the HTTP protocol to the HTTP Server for iSeries program, which identifies the servlet to invoke.

2. The servlet uses the `doGet` or `doPost` method to read form data.

   The Java servlet API includes the `doGet` and `doPost` methods that correspond to the HTML `METHOD="GET"` and `METHOD="POST"` techniques of sending data from the browser to HTTP Server for iSeries.

   The input data is available to the servlet in the HttpServletRequest input stream.

3. The `getParameter` methods are used to parse field name/value pairs.

   Now that the form data is available in the HttpServletRequest stream, it can be parsed into field name/value pairs that correspond to the data fields used on the HTML form. The Java servlets API includes the `getParameterNames` and `getParameterValues` methods to retrieve the list of field names and values from the input stream. After retrieving the name and value pairs, the values are available in enumerations within the servlet.

4. The servlet processes database and other requests.

   At this point, all of the data from the form is available to the servlet. The servlet can now run the functions that are required to service the request. For example, the servlet can run an SQL query against the iSeries database or use other Java classes in the AS/400 Toolkit for Java to work with other iSeries resources.

5. The response HTML is generated.

   The servlet can start generating response HTML to send to the browser at any point. Typically, the servlet generates HTML headers, followed by the actual form heading, then one or more lines of data, and finally a page footer.

   The Java servlets API provides the HttpServletResponse output stream to transport generated HTML statements from the servlet back to the browser. You create well-formed HTML statements as simple strings, using concatenation as necessary to build a string of HTML tags and the response data. To actually send the HTML, you simply use the `print` or `println` methods on the HttpServletResponse stream object.

6. The HTTP Server for iSeries program sends the response HTML to the browser.

   Because the HTML is written in the servlet, it is sent from HTTP Server for iSeries to the browser. The STDOUT file is used, which is common for all server-to-browser communication.

7. The resulting Web page is displayed in the browser.

   Because the page is composed of standard HTML elements, there are no special requirements or security considerations for the browser.

3.2.2.2 Benefits and considerations of using servlets

This section discusses the benefits and considerations of using servlets.

Benefits of using servlets

If you have not yet started creating iSeries Web serving applications, seriously consider adopting Java servlets as the technique to use, even if you do not yet use Java. In fact, learning Java by working with servlets is ideal, since servlets are basically batch processes that do not deal with user interface issues found in client-side programs.

A Java servlet is based on industry standard Java. Although using Net.Data and CGI programming techniques for Web serving is fine for iSeries server users who have those skills, there was little to attract Java programmers to

the iSeries server as a Web serving platform. With Java servlet support, the iSeries server now runs as a powerful Web serving platform.

Java servlets provide an alternative to Net.Data, CGI programming, and applets. With servlets, you can parse requests from HTML forms and use simple `print` and `println` methods to send response HTML back to the browser. There are few browser dependencies when you use the servlet, while there are some differences between browsers when you run applets.

Java servlets are often faster than equivalent CGI programs. CGI programs have a reputation of being slow and unable to scale. The primary reason for this is the way they are loaded. CGI programs are loaded at each invocation in a process separate from the Web server. This requires a relatively large amount of resource, both from the Web server and the operating system. Also, many CGI programs are written in the interpreted PERL scripting language. This interpretation adds significant overhead.

Java servlets were designed with performance and scaling in mind. They are loaded only once, either when the Web server is initialized or at first invocation. They are not unloaded unless this is explicitly done by the Web server or system administrator. Also, servlets run in the same process as the Web server by taking advantage of Java's built-in multi-threading model. This results in faster communication between tasks and threads.

Figure 22 shows this relationship.

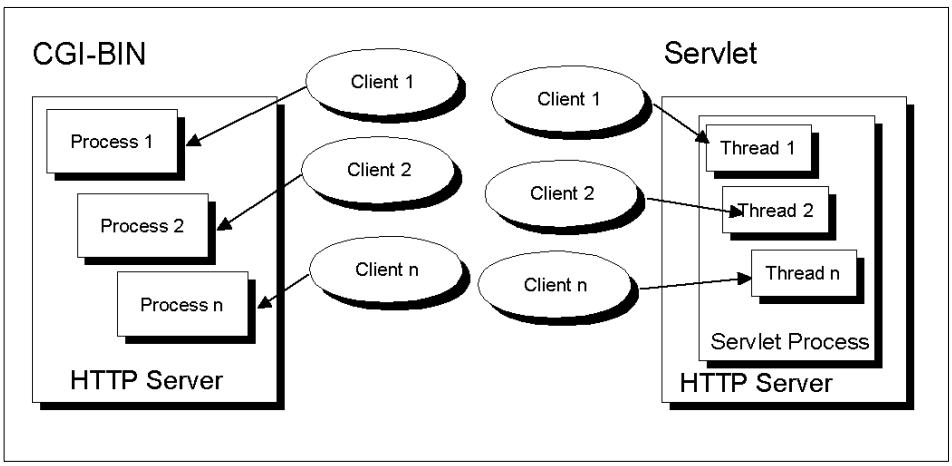


Figure 22. Servlets versus cgi-bin

Your servlets can optionally be started when the IBM HTTP Server for iSeries program is started, so that they are available and waiting for incoming requests. On the other hand, Net.Data and CGI programs are typically short-lived; there is little space to optimize their performance. Servlets also do not necessarily end when they are done serving a request. They can remain active for further requests. If you create an equivalent Net.Data or CGI application and a Java servlet, you typically see much better response time with the Java servlet.

Java servlets provide two simple input and output stream objects to get the data and to write the response, and two simple methods to parse field name and value pairs. Most of the work in CGI programming is concerned with getting the input data from the Web form, parsing the data into discrete field and value pairs, and writing response HTML back to the STDOUT file.

Considerations of using servlets

There are practically no considerations or potential disadvantages for using the servlets. In the e-business arena, the Java servlet is a far more useful business tool than the applet. A servlet is essentially the opposite of an applet. Think of it as a server-side applet in that servlets run inside the Web server the way applets run inside the Web browser.

3.2.3 Location of the Java servlet product on the iSeries server

A Java servlet is supported on the iSeries server using IBM WebSphere Application Server Standard Edition (5769-AS1). It is shipped with OS/400 at no additional charge.

3.2.4 JavaServer Pages (JSP)

JavaServer Pages (JSPs) allow you to embed Java code into HTML pages to execute directly by the server when the page is served. JSP technology is similar to Microsoft's Active Server Pages (ASP) technology, but JSP is standardized and portable. By using JSPs with servlets, you can separate business logic from its presentation and reuse the business logic (using JavaBeans) more practically.

In today's environment, most Web sites display dynamic content based on a user and session. And most content, such as images, text, and banner ads, is easiest to build with HTML editors. You need to mix the "static" content of HTML files with "directives" for accessing or generating dynamic content.

Such technologies as Java servlets, for example, make it easier to write server-based code using the Java programming language for interactive applications. Developers can write servlets that take an HTTP request from

the Web browser, generate the response dynamically (possibly querying databases to fulfill the request), and then send a response containing an HTML or XML document to the browser.

The problem with using this approach is that the entire page must be composed in the Java servlet. If a developer or Web master wants to tune the appearance of the page, they *edit and recompile* the Java servlet, even if the logic is already working. With this approach, generating pages with dynamic content still requires application development expertise.

An answer is an industry-wide solution for creating pages with dynamically-generated content. This solution addresses the limitations of current alternatives by *separating* the application logic from the appearance of the page (in other words, separating the business logic from the presentation logic). In addition, the solution should result in faster and more simple development and testing of interactive Web-based applications than in previous methods.

JSPs address this need. They provide server-side scripting support for generating Web pages with *combined* static and dynamic content, allowing easy access to server-side components from Web pages. A JavaServer Page is an extended HTML page with Java code inserted directly into the HTML source. At run time, the JSP is compiled into a servlet class and activated for use servicing client requests. If the JSP changes between client requests, it is automatically recompiled before the servlet is executed again.

JSPs provide a way to combine the worlds of HTML and Java servlet programming. JSPs are text files (identified by the .jsp extension) that look much like HTML pages. In fact, they are a new type of file that combines standard HTML and new *tags* for scripting in the Java language.

These tags specify the programming of a servlet to control the generation of dynamic content. One of the new tags gives JSPs the ability to call *reusable* components called JavaBeans.

JSPs look like HTML, but they are code that gets compiled into Java servlets. The responsibility of the resulting servlet is to generate a user page. That page is a combination of the HTML from the JSP file with embedded dynamic content specified by the new tags. This way, a JSP describes a representative HTML response. The dynamic content is specific to a given HTTP request.

In Model 1 shown in Figure 23, a user working in a browser makes a request that is sent to a JSP file.

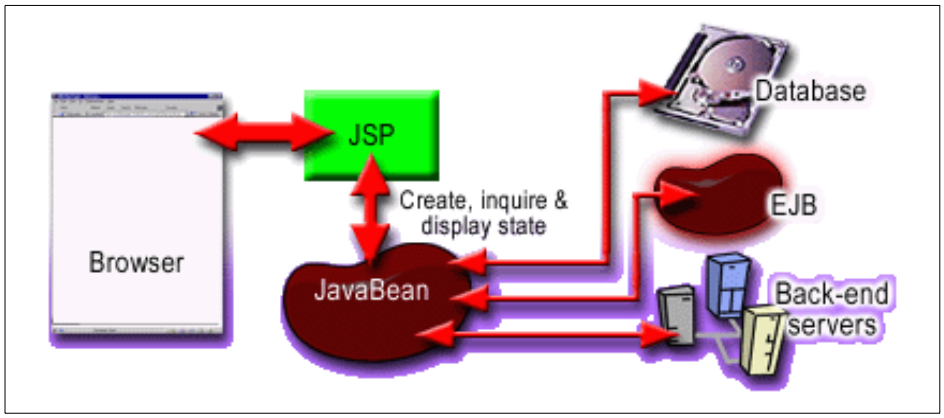


Figure 23. JSP model 1

After receiving the client request, the servlet that is compiled from the JSP file requests information from a JavaBean. The JavaBean can, in turn, request information from an Enterprise JavaBean, database, or other back-end service. Once the JavaBean accesses the requested content, the JSP servlet can query and display the information as HTML in the user response.

Model 2, shown in Figure 24, shows another way to use JSPs, by sending a request to a Java servlet.

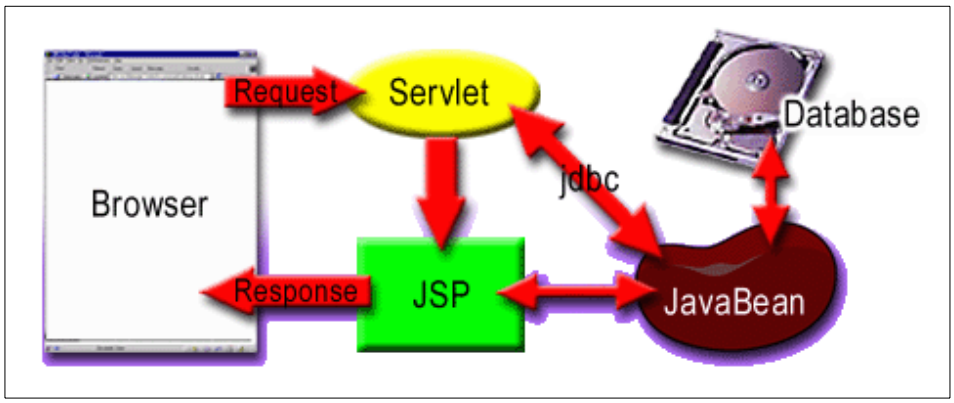


Figure 24. JSP model 2

In this example, the client makes a request that is handled by a Java servlet. The servlet, like the JSP in the model 1 case, requests dynamic content from a JavaBean or Enterprise JavaBean. The resulting dynamic content is then wrapped in a JavaBean. The servlet invokes the JSPs servlet, which

accesses the dynamic content from the bean and delivers the content to the client Web browser.

3.2.5 Enterprise JavaBeans (EJB)

Enterprise JavaBeans (EJB) is the key technology to deal with distributed transaction and complex persistence issues in a Web application. In the past few years, it is self-evident that the WWW has transformed the way in which a business interacts with its customers. Progressing from maintaining a simple Web presence using a home page, then onto a dynamic presence with an active Web site allowing ordering of products and services, contemporary businesses now face the next evolution of Web enablement. This is the need to *integrate* their Web-based systems with their other business systems. This linkage is the basis of the *transactional* stage of a company's Web enablement.

EJBs enable users of such application servers as WebSphere to achieve this integration. EJBs are critical to the operation of a transactional server.

EJBs extend Java's "Write once, run anywhere" model of hardware and operating system independence to middleware and application server independence. It achieves this by separating the business application from the system services. Therefore, developers are not locked into using a particular vendor's middleware servers.

EJB technology complements and extends the existing Java architecture by providing an application server environment to handle system services that would otherwise be handled by the application code.

Enterprise JavaBeans, based on the Sun Microsystems Enterprise JavaBeans (EJB) specification, are *reusable* software components to build distributed, *transactional* business applications using Java. EJBs contain business logic functions, which simplify the development of database applications.

Some notable features of EJBs include:

- They make it possible to build distributed applications by combining components developed by you or by different software vendors. Defining business objects as components promotes reusability and development efficiency. An application developer can create new components, reuse existing components, or purchase components. By using existing components or purchasing components, the developer assembles the application rather than taking further time to develop and test new code.

- EJBs make it easy to write applications. Application developers do *not* have to deal with low-level details of transaction and state management, security, persistence, multi-threading, resource pooling, and other complex low-level application programming interfaces (APIs). A programmer gains direct access to the low-level APIs. The majority of these details are managed by the EJB server. Therefore, developers can concentrate on producing business logic (the coding of their business rules, logic, and knowledge) rather than spending time on “plumbing”.
- EJBs are developed once and then deployed on multiple platforms without recompiling or modifying source code.
- They allow interoperability with other Java and non-Java applications.

Enterprise JavaBeans and the iSeries server are a perfect match. The Enterprise JavaBeans component model logically extends the JavaBeans concept. It is targeted at server tier business logic development. It provides interfaces that insulate the programmer from the complexities and dependencies unique to a platform.

EJB technology provides a component model for server applications. It allows you to easily *separate* user interfaces from business logic.

The server-side business logic is packaged as Enterprise JavaBean components. Once they are written and deployed on a server such as the iSeries server, client programmers can use them with very little knowledge of how the beans actually work. The client programmer only has to know what methods the Enterprise JavaBeans support and how to call them.

Another key advantage is that regardless of whether you are writing a Java application, that is a Java servlet, it always works the same. You only need to call the methods provided by the Enterprise JavaBeans to handle the application processing.

EJB technology also leverages *existing* applications currently running business environments. EJB technology allows the extension of existing applications to provide new and additional functions of the business.

Even without CORBA, EJB objects are used to wrap existing application logic. For example, you can wrap Microsoft’s Component Object Model (COM) objects as EJB components and use them as part of a Java application.

Figure 25 on page 78 illustrates the architecture of the Enterprise JavaBean environment.

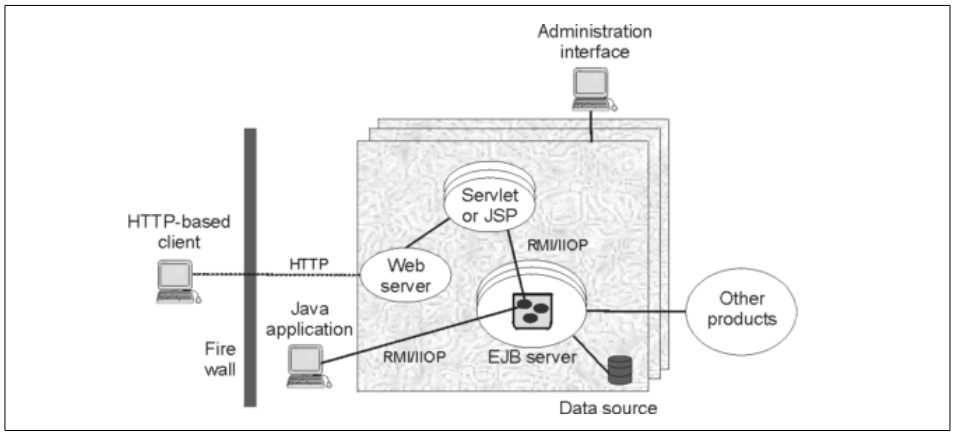


Figure 25. Enterprise JavaBeans (EJB) architecture

The EJB specification consists of two major units, reflecting the critical separation between the business application and system services:

- Components: Entity beans and session beans
- Services: EJB server and EJB container

3.2.5.1 EJB server

At the center of the environment is the *EJB server*. The EJB server is the application server tier of the WebSphere Application Server's three-tier architecture. It connects the client tier (Java servlets, applets, applications, and JSP) with the resource management tier (the data source).

This server contains and runs one or more *enterprise beans*, which *encapsulate* the business logic and data used and shared by EJB clients. The enterprise beans installed in an EJB server do *not* communicate directly with the server, which allows developers to focus on the business problem and lets the EJB server handle system services. Instead, an *EJB container* provides an interface between the enterprise beans and the EJB server to provide many low-level services such as threading, support for transactions, and the management of data storage and retrieval.

The EJB server allows the application developer to obtain the system services required by the application without directly dealing with lower level APIs. The EJB server carries the majority of the burden of dealing with the system environment by managing and coordinating the allocation of resources to the application.

Key system services (APIs) that the EJB server includes are:

- **Security service**: The security server handles authentication and authorization for principals that need to access resources in an EJB server environment. In the EJB component model, security is both granular an flexible. It is granular because it is configured at the component level or at the method level. It is flexible because it is configured outside the application by using utilities provided by the application service provider.
- **Transaction service**: In the EJB component model, transactions take two basic forms: database and business.

  Note: The EJB specification does not specifically distinguish between these two forms. This breakdown is to simplify the explanation.
- **Workload management service**: The workload management service ensures that resources are used efficiently by distributing the workload to other servers.
- **Persistence service**: The persistence service handles interaction between an entity bean and its data source to ensure that persistent data is properly managed. The EJB architecture eliminates the need for the components to deal with the persistence mechanism. Using JDBC as the database management technology, the application server interacts with a wide variety of data stores on the market today.

EJB server support is currently available on the iSeries server with WebSphere Application Server Advanced Edition 3.02.

3.2.5.2 EJB containers

The container is defined as a separate mechanism from the EJB server. The clarity of this separation is at the discretion of the tool vendor.

EJB containers serve as the means to insulate the enterprise bean developer from the specifics of the EJB server services, such as transaction management, security, and object distribution. Containers provide a simple interface for the enterprise bean and access system services for it. This interface is referred to as the *component contract* for the Enterprise JavaBean.

The container manages Enterprise JavaBean objects. It manages the life cycle of the object (creation, maintenance, and deletion), implements the security for the object, and coordinates distributed transactions involving the object. By performing these activities, the container eliminates the need for bean developers to concern themselves with these issues.

There are two types of enterprise beans or components. The first type is *session beans*, which encapsulate short-lived, client-specific tasks and objects (transient persistence). The second type is *entity beans*, which encapsulate permanent or persistent data (persistent persistence). The EJB server stores and retrieves this persistent data in a data source (for example, database, other application, or file). A description of the two types is given in the following sections.

3.2.5.3 Entity beans

These components represent business objects and contain business data. Because an entity bean contains business data, its contents are persisted for later use. Entity beans often reflect a row within an application database. An entity bean has methods to manage its data (get and set methods) and can support business logic pertaining to its business data. Entity beans are either:

- **Container managed**: The reading and writing of the bean attributes to the persistent datastore are delegated to the container that holds them. This allows mapping from the bean attributes to the database columns outside of the entity bean itself. It also allows the entity bean developer to keep the persistence details separate from the business object (the entity bean).
- **Bean managed**: Bean managed entity beans allow the bean provider to control the reading and writing of the bean attributes to the database. This gives the provider greater flexibility in providing their own persistence strategy.

3.2.5.4 Session beans

Session beans perform business processes or tasks within a business process. A client uses a session bean to complete a particular task. Session beans are transient (their data is not persisted) and only exist for the life of the transaction. Session beans usually perform activities like obtaining or storing business data by using the entity beans or performing business logic that is maintained separately from the business data. Session beans are either:

- **Stateless**: Perform activities for the client, but do not maintain any data.
- **Stateful**: Have data and maintain that data for the life of a transaction. If this data must be persisted, it must be forwarded to entity beans. Stateful session beans minimize the amount of interaction between the client and the server, making the application more efficient.

Part 3. Practical guide for building e-business: iSeries products

Part 3 discusses the iSeries products that are available to build an e-business site. This part is for both business decision makers and technical people.

Part 3 consists of the following chapters:

Refer to Chapter 12, “B2B: Application solutions” on page 261, for a description of some of the complementary tools and middleware products available from non-IBM companies.

Chapter 4. iSeries e-business products overview

Version 4 Release 5 of OS/400 provides a variety of tools that help you to start and grow with e-business implementation. Figure 26 provides an overview of OS/400 V4R5 packaging for e-business. The components that are new with OS/400 V4R5 are marked as such.

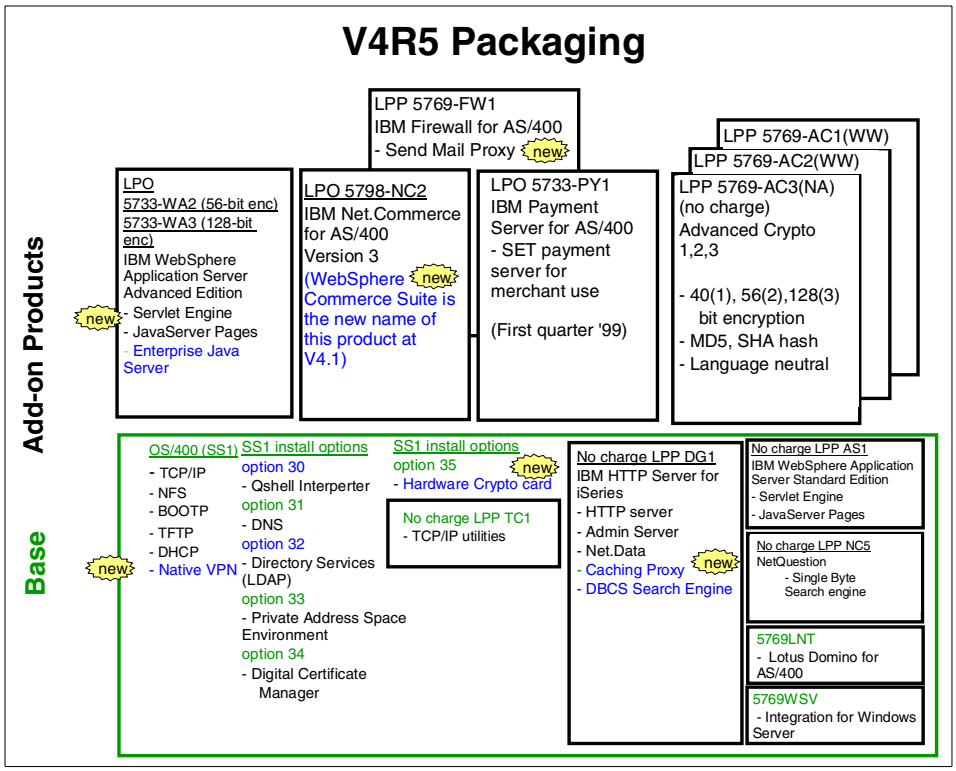


Figure 26. OS/400 V4R5 overview

This chapter provides an overview of the products offered to iSeries users from the perspective of e-business. The products are discussed in more detail in later chapters.

4.1 Integrated functionality of OS/400 for e-business

The e-business optimal server of choice is the iSeries server. The greatest strength of the iSeries server lays in its superb *availability* and *reliability* features, along with its wide variety of *integrated* OS/400 products offering.

4.1.1 Integrated quality of OS/400: Networking flexibility

The iSeries server provides the flexibility necessary for the network. The following functionality are all integrated into OS/400 and are supplied as no charge components.

4.1.1.1 TCP/IP: The protocol of the Internet

The iSeries server offers robust support for TCP/IP. With the integrated TCP/IP capabilities of OS/400, users can link to networks throughout the world. TCP/IP Connectivity Utilities for iSeries (5769-TC1) supports these features:

- Numerous services, protocols, servers, physical interfaces, and more.
- TCP/IP physical interfaces:
  - Asynchronous support
  - Distributed data interface (DDI)
  - Ethernet version 2.0
  - Ethernet IEEE 802.3
  - Frame relay network
  - IBM token-ring network
  - Integrated xSeries Server for iSeries Firewall for AS/400 (LAN)
  - X.25 (PVC and SVC)
  - X.25 over integrated services digital network (ISDN)
  - Wireless local area network (LAN)
- TCP/IP communications protocols:
  - Domain Name Services (DNS).
  - Address Resolution Protocol (ARP)
  - Internet Control Message Protocol (ICMP)
  - Network File System (NFS)
  - Packet Internet groper (PING)
  - Point-to-Point Protocol (PPP)
  - SOCKS proxy enablement
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)
  - Dynamic Host Configuration Protocol (DHCP)
  - The iSeries acts as a Lightweight Directory Access Protocol (LDAP) server for the X.500 standard
- TCP/IP application protocols include standards for many common applications

  TCP/IP application programming interfaces (APIs) are available to support unique requirements beyond the functions of standard TCP/IP

applications. With this support, you can write applications to the TCP, UDP, IP, and Internetwork Packet Exchange (IPX) protocols directly.

For more details on the OS/400 integrated functions of TCP/IP, refer to:
`http://www.as400.ibm.com/tcpip`

4.1.1.2 Multi Protocol Transport Networking (MPTN)

AnyNet or the Multi protocol Transport Networking (MPTN) architecture allows Common Programming Interface for Communications (CPI-C), intersystem communications function (ICF), and sockets to flow over either TCP/IP or SNA via AnyNet. The TCP/IP applications that can use this support via sockets are FTP, SMTP, POP, LPR, and LPD. When using AnyNet support, APPC over TCP/IP allows CPI-C and ICF applications to run without changes over a TCP/IP network including Client Access/400 applications.

4.1.1.3 Virtual Private Networking (VPN)

Native OS/400 VPN support provides a leading-edge and interoperable IP security function. The OS/400 VPN solution provides client dial-up support for dynamic IP address assignment as well as Layer 2 Tunneling Protocol (L2TP) support for remote access via an Internet Service Provider (ISP).

4.1.2 Integrated quality of OS/400: Web enabling functions

The iSeries server contains numerous tools to enable your applications to the Web including:

- IBM HTTP Server for iSeries, including the popular Apache HTTP server
- WebSphere Application Server Standard Edition
- WebSphere Advanced Edition high-performance EJB server
- CGI support
- Net.Data support
- 5250 Gateway and emulator
- Webserver Search (formerly known as NetQuestion)

Another point to consider is that of the iSeries server as a Java server platform. The potential of Java makes developers reconsider their non-iSeries development environment. The iSeries server perfectly fits the Java platform, not only for development itself, but also for run time (the Java Virtual Machine is under the TIMI). Built-in OS/400 V4R5, the AS/400 Java Toolbox and the Java Development Kit give you the tools to start. The support for Enterprise JavaBeans permits you to grow fast.

4.1.3 Built-in security features of OS/400 for e-business

The iSeries server provides the tools necessary to secure the solution environment with encryption software, the security delivered by the IBM HTTP server, IP packet filtering, SSL support, and so on. Virtual Private Networking (VPN) support (L2TP, IKE, IPSec) is the key technology to protect data in business-to-business relations. In fact, the OS/400 has built-in nearly all features to be a firewall in itself with the help of a router or firewall.

Note

IBM Firewall for AS/400 V4R4 (5769-FW1) will be withdrawn from marketing 29 December 2000. Refer to the Web site at `http://www.iseries.ibm.com/products/firewall/index.htm` for sample configurations using alternate solutions and references to other solutions, such as Check Point Software Technologies, Cisco Systems, or Axent, Inc.

4.1.4 IBM DB2 Universal Database Version 7

IBM DB2 Universal Database Version 7, the world leader in large data warehousing databases, is committed to deliver the best in power, performance, and scalability. This advanced, 64-bit relational database system provides leading-edge performance in e-business and data warehousing environments. The iSeries and DB2 UDB for AS/400 in combination provide the flexibility and adaptability to support any type of workload, small or large.

4.1.4.1 DB2 Universal Database and the integrated file system

The requirements of a high-function, robust database with the capabilities to deliver advanced universal data support include:

- Universal access
- Universal application support
- Universal extensibility
- Universal scalability
- Universal reliability
- Universal management

DB2 UDB for AS/400 has long provided facilities to address most user requirements, many forms of information could be stored on the iSeries server. With Version 4 Release 5, DB2 UDB for AS/400 supports storing and managing all forms of information, including complex objects (such as spreadsheets, word processing documents, and multimedia objects) within the database. This release includes such features as large object (LOB)

support (BLOBs for binary, CLOBs for character, or DBCLOBs for double-byte character), user-defined functions (UDFs), user-defined data types (UDTs), and DataLinks. All of these features, combined with the functions of constraint checking, trigger, and stored procedure support, allow customers to use one database management system to store, retrieve, and manage all of their corporate information.

These functions can be incrementally implemented as business requirements dictate. DB2 UDB for AS/400 provides robust database functionality in the same simple, easy-to-use environment. This allows small businesses to maintain their current application base, while adding new database functionality to their applications in an incremental fashion. No changes are necessary to legacy applications that do not require these new functions.

4.1.5 Universal Database access

DB2 UDB for AS/400 provides a wide variety of universal data access features. It allows customers to access information on the iSeries server using Structured Query Language (SQL) or through OS/400-unique commands and application programming interfaces (APIs). Virtually any program or product that provides data access through one of the standard interfaces for relational data can access data stored and managed by DB2 UDB for AS/400. These industry standards and product-specific interfaces include:

- ODBC Windows Database Access Standard
- OLE DB Windows Information Access Standard
- ADO Windows Information Access Classes
- JDBC Java-based Database Connectivity
- SQLJ Java-based Embedded SQL
- SQL ISO and ANSI Standard SQL
- DRDA X/Open Distributed Database Standard
- CLI X/Open Database Access Standard
- EDA/SQL IBI EDA SQL Standard
- DAL Apple Relational Database Standard APIs
- Net.Data Internet Data Access

In addition to these interfaces, DB2 UDB for AS/400 continues to support integrated database access. All of these interfaces provide direct access to a single common data store with excellent performance. These interfaces allow thousands of products to access the iSeries server without special modifications.

With Lotus Domino being integrated on the iSeries server, virtually all Lotus Domino applications are supported, as well as the existing OLTP, Business

Intelligence (BI), e-business, and client/server applications previously available.

DB2 UDB for AS/400 is completely Web enabled. Not only is the data managed by DB2 UDB for AS/400 accessible through your favorite Web browser, but all host applications are Web enabled. The iSeries 400 server allows customers to make their host-based applications and data available over the Web with no application conversion needed.

DB2 UDB for AS/400 also leverages a common set of IBM products like Visual Warehouse, Intelligent Miner, and DataPropagator. This integration has been enhanced even further with new features like the integrated DB2 OLAP (on-line analytical processing) Server for iSeries, iSeries Agent support for Visual Warehouse, and DB2 DataJoiner support on the Integrated xSeries Server for iSeries.

Although DB2 Universal Database for AS/400 is not covered in this redbook, you can find more information on the Web at:
`http://www-4.ibm.com/software/data/db2/`

The key to complete an e-business solution is to use tools and middleware that build onto the operating system. Refer to the remaining chapters of Part 3 of this redbook and Chapter 12, “B2B: Application solutions” on page 261, for non-IBM tools.

Chapter 5. HTTP Server for iSeries

IBM HTTP Server is the follow-on to the IBM Internet Connection Server (ICS). It is a scalable, high-performance Web server that is available from OS/400 V4R3 onward, as well as other IBM and non-IBM platforms.

IBM HTTP Server is a complete Web server product with advanced security and application development features. With IBM HTTP Server, you have everything you need to quickly and easily establish a Web presence, and start on your road to working the Web for business.

Gartner Group says

The Gartner Group (find them at `http://www.gartner.com`) has provided a good holistic view of the strengths of the AS/400 as a Web server in today's business environment. In fact, the title of the research report is "AS/400 as a Web Server: Perspective". You can subscribe directly to Gartner to read the report.

HTTP is the protocol used to communicate between a browser and a Web server. To implement a static Web site, all that is required is a Web server and an HyperText Markup Language (HTML) authoring tool such as NetObject Fusion.

Note: IBM HTTP Server for AS/400 will become IBM HTTP Server for iSeries with the release of V5R1. Throughout this chapter, we use the respective names to reflect the version that is being discussed.

5.1 HTTP Server for AS/400 (Original): Product description

The primary software backbone of a Web presence is the HTTP server, commonly known as a Web server. It is a TCP/IP application similar in concept to a network file server.

The key uses of the HTTP server are described in this chapter. These uses include:

- File server
  - Serves any kind of file to a browser (for example html, gif, or avi)
  - Caches files for performance

- Application server

  Numerous application models

- Security server
  - Trusted component of the system
  - SSL support
  - Client authentication and access control
- Management server
  - Access logs
  - SNMP MIBs (V4R4)
- Proxy server

  Caching proxy

The purpose of the HTTP server is to respond to browser requests for files, graphics, and other media, such as sound, and to reply in the form of sending HTML pages back to the browser. On the iSeries server, Web pages are stored either in directories in the integrated file system (IFS) or as members in database files in libraries.

HTML is the standard language for Web pages. WWW documents are written in HTML script. The HTML coding describes, to the browser, the appearance of the displayed document. Therefore, an HTML page is a mixture of text and special HTML tags that describe the page. There are tags that describe the structure and presentation of the page, such as heading tags, ordered list tags, and table tags. There are also tags for including an image on the page and tags for creating a link to another page.

The general sequence of operation for a Web server is:

1. A Web browser requests a document.
2. The HTTP server sends the requested document.
3. The Web browser interprets the document and displays it.

HTTP servers can also load and run Java applets from browsers and forward browser-generated requests to Java servlets on the main server.

Application servers (such as the IBM WebSphere Application Server) and transaction servers run on top of HTTP servers to provide the dynamic data and transaction capabilities of the customer e-business Web site respectively.

For the iSeries platform, from an IBM product viewpoint, there are two HTTP servers available:

- The HTTP Server for iSeries
- The Domino HTTP Server

The IBM HTTP Server for iSeries (5769-DG1) is the current Web server packaged with the iSeries server. It is the Web server component of the WebSphere Application Server for AS/400. It can also act as a gateway between the client and the iSeries database, using the CGI interface and Net.Data. Any Web design tool can be used to create content. Figure 27 shows NetObjects Fusion (`http://www.netobjects.com/products/`) as an example.

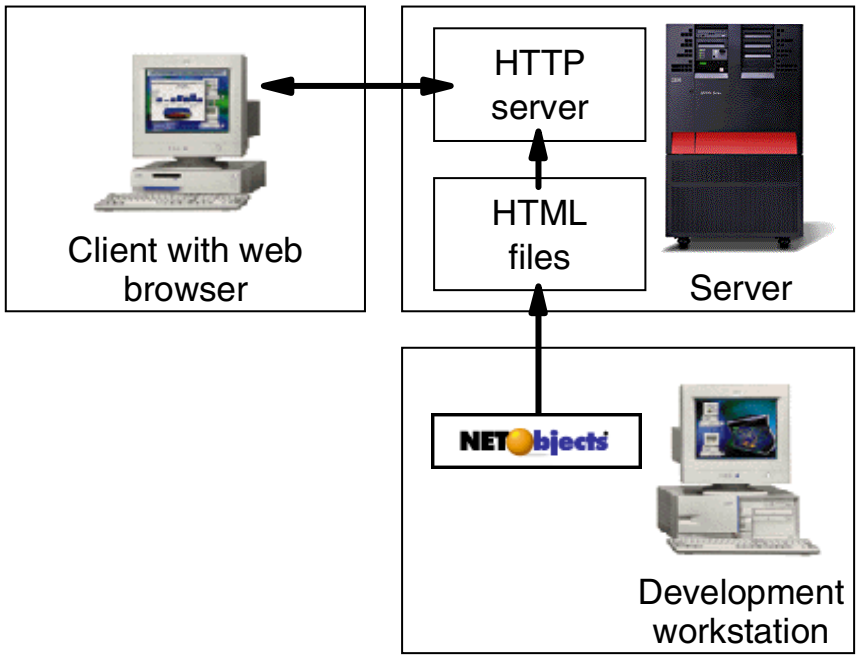


Figure 27. IBM HTTP Server for AS/400

IBM HTTP Server for AS/400 was known as the HTTP Server in OS/400 V3R7 and the Internet Connection Server (ICS) in OS/400 V4R1 through V4R2. At the time, ICS was part of TCP/IP Connectivity Utilities (5769-TC1). With V4R3, substantial enhancements were incorporated into the product, and it reverted back to its former name (along with a newly assigned product number 5769-DG1). It also introduced new SSL support. An Internet Connection Secure Server (ICSS, 5769-NC1/128-bit or 5769-NCE) was introduced at this time to provide SSL support of the V4R1 and V4R2 Internet Connection Servers.

The V4R3 HTTP Server introduced the integration of WebSphere Application Server for AS/400, providing support for Java servlets on the AS/400e server. Given the importance of this product in OS/400 V4R4, the WebSphere Application Server Standard Edition packaged with OS/400 was newly numbered as 5769-AS1.

5.1.1 V4R3 features of HTTP Server for AS/400

V4R3 introduced substantial improvements over the V4R2 ICS product. The major features of the V4R3 product are described in the following sections.

Current HTTP
The IBM HTTP Server for AS/400 implements the HTTP Version 1.1 protocol. This is the current version of the HTTP protocol.

Dedicated subsystem
The previous ICS ran its HTTP server tasks in the QSYSWRK subsystem. The V4R3 HTTP Server tasks run in their own subsystem, QHTTPSVR, for greater efficiency of processing.

Persistent connections
When you enter a URL into your browser's address line or click a link on a Web page, you open a connection between your browser and the HTTP server.

Prior to the availability of persistent connections, each file referenced on the Web page was retrieved using a separate connection. This type of retrieval is tremendously costly for the HTTP server and the network since there is overhead required to establish and terminate each connection.

Persistent connections are the default behavior for an HTTP server that implements the HTTP 1.1 protocol. Persistent connections provide the following advantages:

- Because there is less opening and closing activity, CPU and memory utilization on the HTTP server are reduced.
- Network congestion is minimized because of the fewer number of TCP/IP packets that are required to request the files.
- HTTP requests and responses can be pipe lined on the connection. Using pipe lining, a client can make several requests to the server without waiting for the responses to the requests.

Virtual hosts

In previous versions of the HTTP Server for AS/400, the only way to host multiple Web sites on the AS/400e server and use the default HTTP port (80) is to use different communications adapters in the AS/400e server. If you want to host multiple Web sites through the same communications adapter, only one of the sites can use the default HTTP port. All other Web sites need a separate port assignment. To request those Web sites, the unique port assignment is included as part of the requesting URL.

Starting with the OS/400 V4R3, IBM HTTP Server for AS/400, you can enable virtual hosting. This allows you to host any number of Web sites through one communications adapter. With virtual hosting, you do not need to assign a unique port to each Web site.

Virtual hosting is useful if you need to provide multiple “top-level” URLs for your Web sites or if you provide Internet Service Provider (ISP) services to clients.

Proxy, cache, and local memory cache

The IBM HTTP Server for AS/400 can be configured as a non-caching or caching proxy server. When used as a non-caching proxy, the primary benefit of enabling proxy services is that the IP addresses used on your internal network are not sent out of your network. The proxy service forwards the request from your internal network using the IP address of the proxy server, not the address of the original requester. When the proxy server receives the response, it forwards the response to the original requester.

Proxy caching

With caching enabled, the proxy server can act as a high-speed local store of previously accessed Web pages. When you configure your server as a proxy caching server, you can improve performance. You can also allow users of your internal network to access documents on the Internet. For example, if you frequently access the same set of Web pages from one or more sites, it may be advantageous to activate the caching feature. The retrieved Web page is stored locally on your AS/400e server. Any subsequent accesses to the page occur at LAN speed, rather than at Internet speed.

Web pages can be encoded with a “no-cache” attribute or a specific expiration date. You can also configure the IBM HTTP Server for AS/400 proxy service so that it periodically performs “garbage collection” to remove expired files from the cache. The cache is located in the QOpenSys file system, which provides support for case sensitivity in file names. You configure the maximum size of the cache (which uses AS/400e server disk storage), protocols, and URLs to cache or not to cache.

Proxy logging
Another use of the proxy service (with or without caching) is to log client requests. Some of the data available includes:

- Client IP address
- Date and time
- URL requested
- Byte count
- Success code

With this information, you can construct reports to account for the use of your Web site. The information can be used in a charge-back system, to understand and track marketing trends, for example.

Local memory cache
A proxy cache is traditionally most beneficial to clients on your network since it lets you store files that were retrieved from other Web sites. You can provide a caching service for files on your site using the local memory cache configuration options.

To use a local memory cache, you identify an amount of memory to allocate and a set of files to be cached. When the IBM HTTP Server for AS/400 is started, the files are read into the local memory cache up to the limit of the amount of memory allocated or the limit of the number of files that you allow to be cached.

When a request is received at your IBM HTTP Server for AS/400, the local memory cache is checked first to determine if it has a copy of the requested file. If so, the file is served from the cache, which is significantly faster than if the file is retrieved from disk storage.

Server-side includes
Server-side includes enable the server to process some of the Web pages before the server sends the page to the client. The current date, size of the file, and the last change date of a file are examples of the kind of information that you can include in Web pages that you send to the client.

Automatic browser detection
Use this feature to provide different documents for different clients, allowing the customer Web site to seamlessly exploit the unique capabilities of whatever browser your customers are using.

Server API support

As a follow-on to ICAPI, server API support enables users to write applications to extend or customize how the HTTP server handles client requests.

CGI programming

Corporations and other customers benefit from interacting with browser users by sending and receiving data. In the Web presence arena, this type of transaction is simple, such as collecting the name and address of a browser user who wants to receive a catalog. In general, these interactions start with a form: a Web page that contains input-capable fields and push buttons (like function keys). On the server, for example, the HTTP server needs to hand the input from the form to a program for processing. Typically, on AS/400e servers (and most of the platforms), that program is a CGI program or has Net.Data macros, which place a user-friendly interface around CGI programs. The CGI program receives the form data from the browser, accesses business data and business logic on the AS/400e server, updates or stores information (if required by the transaction), and then builds the Web page that the HTTP server returns to the browser user in response.

There are several techniques available to you to create Common Gateway Interface (CGI) programs for use with IBM HTTP Server for AS/400. The V4R3 release of IBM HTTP Server for AS/400 provides the following enhancements for GCI programming:

- Java and REXX CGI
- Non-parsed headers CGI
- QzhbCgiParse API

Java and REXX CGI

In addition to traditional iSeries languages used for CGI programming (ILE RPG, ILE COBOL and ILE C), you can use Java and REXX to create CGI programs. Java, in this case, means Java applications that use iSeries support for working with the STDIN and STDOUT files or the CGI APIs, not Java servlets. Support for Java servlets is provided with the WebSphere server, which is an add-on to the IBM HTTP Server for AS/400.

Non-parsed headers CGI

Most CGI programs generate response headers and HTML and return the data to the requesting client through IBM HTTP Server for AS/400. However, there may be occasions when you want to generate a response in your CGI program and return it directly to the requesting client.

If the name of your CGI program begins with `nph_` or `nph-`, its output is not converted by IBM HTTP Server for AS/400. You are responsible in your

non-parsed header CGI program for creating a complete HTTP response message, including the HTTP return code and status information.

QzhbCgiParse API

Prior to V4R3 of IBM HTTP Server for AS/400, CGI programs written in ILE RPG or ILE COBOL needed to use several APIs to work with HTML form data:

- QtmhGetEnv to get the value of the QUERY_STRING environment variable and other environment variables
- QtmhRdStin to read data from the STDIN file for a POST request
- QtmhCvtDb to parse data from the QUERY_STRING environment variable or the data retrieved from STDIN

Although the QtmhGetEnv and QtmhRdStin APIs are easy to work with and relatively straightforward, the QtmhCvtDb API is somewhat cumbersome to use.

IBM HTTP Server for AS/400 provides the QzhbCgiParse API, which combines the functionality of the three Qtmh APIs into one. Some of the features of this API include:

- Support for both GET and POST data
- Parses form data into field name and field value pairs
- Uses a list technique similar to other AS/400e server APIs. Rather than determine the format of the data to be retrieved in advance (that is, while coding the CGI program), simply call the API and walk through the list of retrieved values. If your HTML form changes, you do not necessarily need to recode and recompile your CGI program to accommodate the changes.

Persistent CGI

One of the most significant enhancements in V4R3 of IBM HTTP Server for AS/400 for CGI programs is the introduction of the persistent CGI, or persistency. To understand the benefit of persistency, you can compare it with what happens in a non-persistent CGI application:

1. The requester makes entries on an HTML form and clicks a Submit button on the form. The Submit button has an associated URL that invokes a CGI program on the HTTP server. A connection is created between the HTML form and the CGI program.
2. The CGI program is invoked and retrieves the form data (using either the GET or POST method). The CGI program processes the data, prepares a response, and returns the response to the requester using the STDOUT file.

3. After completing the output to STDOUT, the connection between the requester and the CGI program is terminated when the end-of-file indication is sent through STDOUT. At this point, the CGI program ends, since there is no way for any subsequent requests to reconnect to that instance of the CGI program.
4. Subsequent requests start at step 1 again. A new connection must be established between the HTML form and the HTTP server.

Although this scheme is fine for HTML forms that can be processed in a single invocation of a CGI program, most Web transactions involve more than one HTML form or multiple interactions with the same form.

For example, a “shopping cart” application is a form that is continuously added to as the requester selects items from other Web pages. As items are added to the cart, the list of items must be maintained somewhere because in a non-persistent application, there is no program running on the HTTP server to retain the list. Some solutions for maintaining this state information include temporary files on the HTTP server or “cookies” that are written to the requester’s local storage.

How persistent CGI works

In contrast, a Web application that uses persistent CGI does not need to rely on temporary storage schemes to maintain state information. A persistent CGI application can work as described in the following series of events:

1. The requester fills in an HTML form and clicks the Submit button. The requesting URL invokes a CGI program. The connection between the requester and the HTTP server is now active.
2. The CGI program prepares output to send back to the requester. However, a special header record (Accept-HTSession) is returned to the requester along with a “handle” to identify the persistent CGI program that services the requester.
3. Output from the CGI program is sent to the requester in the STDOUT file. As with non-persistent CGI, the connection between the requester and the CGI program is terminated after an end-of-file. However, the CGI program itself remains active, since the HTTP server can identify and use it for additional requests from the same requester.
4. Subsequent requests send the handle along with data from the form. The HTTP server recognizes the incoming request as belonging to the particular instance of the CGI program. The new connection between the requester and the HTTP server is linked to the CGI program that is already running. Because the program never ended, any internal data structures that it had created on previous requests are still available.

Controlling a persistent CGI
Since there is no way to guarantee that a requester completes a transaction, IBM HTTP Server for AS/400 includes directives that let you specify the amount of time a persistent CGI application can be inactive before being terminated. The time-out value can be specified at both the server level and the application level so that you can allow some applications more time to complete.

No indication is sent to the requester when the time out is reached. Also, any database and file processing is your responsibility. For database applications where changes occurred, you typically want to perform a rollback operation when a persistent CGI time out occurs.

NetQuestion
NetQuestion is a powerful, full-text search engine that builds a global Internet or centralized intranet search service. The Information Technology (I/T) department can handle large amounts of information that are typically stored on Web sites. Documents to be indexed by NetQuestion are provided in either plain text or text with HTML markup. CGI scripts and HTML forms are provided for search and administration. Administration can also be done, using command line functions.

Net.Data
The follow on to DB2 WWW, Net.Data is a no-charge AS/400e tool for helping application developers to add live data to their Web pages.

With static HTML pages, data is present that never changes. Therefore, you need to write CGI programs to dynamically build Web pages. You can do this using the IBM HTTP Server for AS/400, which provides a CGI interface. However, it is not easy to write these programs.

IBM Net.Data is a CGI program that allows application developers to easily transform static HTML Web pages into dynamic Web applications using Web macros. Web macros created for Net.Data have the simplicity of HTML with the functionality of cgi-bin applications, making it easy to add live data to static Web pages. Live data includes information stored in DB2 UDB for AS/400 (locally or remotely), databases on other systems, REXX programs, C and C++ programs, RPG programs, COBOL programs, and other sources.

5.1.2 Security features (cryptographic, certificates, and digital ID)

SSL is a security protocol that was developed by Netscape along with RSA Data Security. This protocol ensures that data transferred between a client

and a server remains private. It allows the client to authenticate the identity of the server and the server to authenticate the identity of the client.

IBM HTTP Server provides HTTP secure (HTTPS) transactions with the Secure Sockets Layer (SSL) V3 protocol. Starting with V4R3, IBM includes the Cryptographic Access Provider as a no-charge LPP with OS/400 to use with SSL.

Once your server has a digital certificate, SSL-enabled browsers, such as Netscape Navigator, can communicate securely with your server using SSL. With SSL, you easily establish a security-enabled Web site on the Internet or on your corporate intranet. You can also install digital certificates on the clients in your network so the server can authenticate connections without prompting for a user ID or password.

SSL uses a security handshake to secure the TCP/IP connection between the client and the server. During the exchange signals, the client and server agree on the security keys that they will use for the session, and the client authenticates the server. After that, SSL encrypts and decrypts all of the information in both the HTTPs request and the server response, including:

- The client's URL request
- The contents of any form submitted
- Access authorization information like user names and passwords
- All data sent between the client and the server

The IBM HTTP Server for AS/400 uses public key cryptography from RSA Data Security, for encryption, digital signatures, and authentication. This support is built into OS/400 for optimal performance and is readily available to all TCP/IP servers wanting to use SSL.

There are three versions of the Cryptographic Access Provider:

- 5769-AC1 provides 40-bit encryption
- 5769-AC2 provides 56-bit encryption
- 5769-AC3 provides 128-bit encryption

Only one version is shipped with your OS/400 installation media. The version shipped is based on the country where the AS/400e server is installed. This version complies with United States export laws for computer encryption products and local laws of the country.

Note: By providing the Cryptographic Access Provider along with OS/400, it is possible to use the support provided by this Licensed Program Product (LPP) even if IBM HTTP Server for AS/400 is not installed.

5.1.2.1 Digital Certificate Manager (DCM)

Another change introduced with OS/400 V4R3 is the Digital Certificate Manager (DCM) option. You can install this as option 34 of the OS/400 base installation.

Digital Certificate Manager provides support for generating and maintaining digital certificates. Certificates are used for both server and client authentication.

Although you can generate certificates for your HTTP server to attest to its authenticity, you most likely need to apply for and receive a certificate from a well-known certificating authority (for example, Verisign) if you intend to conduct e-commerce with your AS/400e server. Both the Netscape Navigator and Microsoft Internet Explorer browsers include a list of well-known certificating authorities that are accepted by the browsers to authenticate Web pages from your HTTP server.

Digital ID

Another feature introduced with V4R3 of IBM HTTP Server for AS/400 uses digital IDs to provide client certification. Although we traditionally think of digital IDs to provide verification of the server's authenticity, it is also useful for clients to provide the server with a guarantee of their authenticity.

Client authentication using digital IDs can be used as an alternative to prompt for a user ID and password. Digital ID authentication requires SSL client authentication for HTTP server client certificates. This offers resource protection with:

- Valid client certificates
- Client certificates with certain distinguished names values
- Client certificates associated with AS/400e user profiles
- Client certificates associated with AS/400e validation lists

5.1.2.2 SOCKS support and SSL tunneling

If your environment has a SOCKS-based firewall for access to the Internet, you can use the IBM HTTP Server for AS/400 proxy server to access the destination outside the firewall. Client connections that use SSL are tunneled through the proxy server, eliminating the need to decrypt and encrypt the data at the proxy.

5.1.3 OS/400 V4R4 features of HTTP Server for AS/400

With the advent of V4R4, several improvements are added to the HTTP Server for AS/400. Such enhancements include:

- **Improved management**
  - SNMP subagent support is added. This allows HTTP server statistics to be forwarded to an SNMP network manager upon request. The subagent allows the use of any SNMP-capable network management system, such as TME 10 NetView, TME 10 Distributed Monitoring, or HP OpenView to monitor the server.
  - Log reporting provides the ability to define access reports, generate reports, view reports, and maintain report files using a GUI based on report templates. The ability to remotely access HTTP server logs, statistics, and status is also introduced with V4R4. Parameters such as the number of hits on an iSeries Web page are readily accessible using this interface.
  - Log archiving allows the customer to manage and maintain their log files.
  - The standard Extended Log File Format is supported. This allows a broader use of industry-wide log analysis tools and log file customizing. The extended format allows more data to be saved in the access log files and allows more control over which data is stored in these files. Numerous analysis tools are available that use this new log format.
  - Web server error logs now contain messages presented in the customer's language of choice.
  - Serviceability of the Web server is improved with additional trace points, additional information in service traces, and improved FFDC information.
  - Error logs are NLS enabled.
- **Performance improvements**

  HTML files are dynamically cached in memory when a URL request for that file is processed. Subsequent requests for that file are handled without the need to do a file input or output. This results in better static page serving performance without manual configuration.

  Multi-threaded CGI programs are now supported. Multi-thread programs can often be more efficient than single threaded ones.
- **LDAP support**

  The HTTP server uses Lightweight Directory Access Protocol (LDAP) to store configuration information and user authentication information. LDAP integration allows you to use directory services for server configuration and authentication.

- **Tools enablement**

  New APIs to the HTTP server are provided to allow third-party management tools to query the value of certain configuration directives as well as use the Web server's mapping rules for a URL.

  The following additional HTTP methods are now supported:

  - **PUT**: Allows a resource identified in a URL to be stored on the AS/400e server
  - **DELETE**: Allows a resource identified by a URL to be deleted from the AS/400e server
  - **User Defined methods**: Additional methods with user-defined behavior may be implemented

- **Additional security function**

  Certificate administration is centralized in the Digital Certificate Manager (DCM) product. The HTTP Server for AS/400 is a certificate customer.

  Platform for Internet Content Specifications (PICS) support is added.

- **iSeries Webserver Search Engine**

  With V4R4, the HTTP server now has a no-charge built-in search engine to support both SBCS and DBCS search data. DBCS encompasses documents written in all languages including Chinese, Japanese, and Korean. The engine allows the customer to perform full text searches on HTML and text files sorted in an OS/400 file system from any Web browser. The iSeries Webserver Search Engine allows browser-based administration and provides Net.Data macros to customize the search and search results.

You can find more information on the IBM HTTP Server at:
`http://www.iseries.ibm.com/products/http/httpindex.htm`

5.2 HTTP Server for iSeries (Powered by Apache): Product description

IBM HTTP Server for iSeries has been enhanced to include the popular Apache HTTP server. This initial release is based on Apache Version 2.0

Functional overview

Apache, a freeware HTTP server, is open-source software that implements the industry standard HTTP/1.1 protocol with a focus on being highly configurable and easily extendible. It is built and distributed under the Apache Software License by the Apache Software Foundation. IBM HTTP Server for

iSeries incorporates Apache Version 2.0. Apache is available for other operating systems at: http://www.apache.org/

Once an Apache-based server for the iSeries is configured appropriately, existing CGI programs and WebSphere applications will run on the Apache server with no changes required.

The following enhancements have been added to the Apache server available on the iSeries:

- Full-function, task-oriented, Web-based user interface for server administration and configuration
- Authentication using LDAP, iSeries user profiles, and validation lists
- Full, native SSL support, including client authentication and association between client certificates and validation lists or iSeries user profiles
- Static and dynamic local file caching
- CGI support for C, C++, RPG, COBOL, REXX, and CL
- CGI APIs (except QzhbCgiParse)
- Net.Data support
- Persistent CGI support
- Support for CGIs running in named activation groups
- Configuration, instance, and group file APIs
- Global and instance-specific configuration settings
- Webserver Search Engine support
- Support for most OS/400 file systems
- Support for configuration files stored in LDAP
- Support for TRCTCPAPP and serviceability
- User-defined methods

New functions

The following functions were not in the IBM HTTP Server for AS/400 (Original), but are available with IBM HTTP Server for iSeries (Powered by Apache):

- Authorization enhancements: Support for denying specific users, groups, or hosts
- SSL enhancements: More specific customization of Secure Sockets Layer with the server
- Headers control: Ability to control expires and other headers
- URL rewriting (limited function in the initial group PTF)
- Dynamic virtual hosting
- Control the number of CGI jobs that are started with server (and user profile)

- More customization of directory listings
- Support for TRCTCPAPP
- Support for configuration files in threadsafe IFS file systems

Functions not available

The following functions were supported in IBM HTTP Server for AS/400 (Original), but will not be available in IBM HTTP Server for AS/400 (Powered by Apache):

- Access log reporting
- Web usage mining
- Platform for Internet Content Selection (PICS)
- Server API

For more information

You can find more information on the HTTP Server for iSeries (Powered by Apache) at: `http://www.iseries.ibm.com/products/http/httpindex.htm`

Chapter 6. WebSphere Application Server for AS/400

What is WebSphere Application Server? In a nutshell, it is a set of tools for developing, testing, running, and measuring Java-based Web applications. WebSphere Application Server is the IBM Java-based Web application deployment environment for server-side applications and JavaBeans. It helps customers deploy and manage Web-based applications ranging from simple Web sites to powerful e-business solutions. This chapter discusses WebSphere Application Server for AS/400 in more detail.

6.1 Introduction to IBM WebSphere Family

WebSphere Application Server for AS/400 is a product component in the IBM WebSphere family. This section provides a brief overview of the IBM WebSphere family and how WebSphere Application Server for AS/400 products fit in the picture.

6.1.1 Server concept review

As the name of the product itself says it, WebSphere Application Server is a product to build and maintain an application server. In that case, what is an application server in an e-business context? To clearly understand this, we present this concept in Figure 28. The following sections explain the primary components.

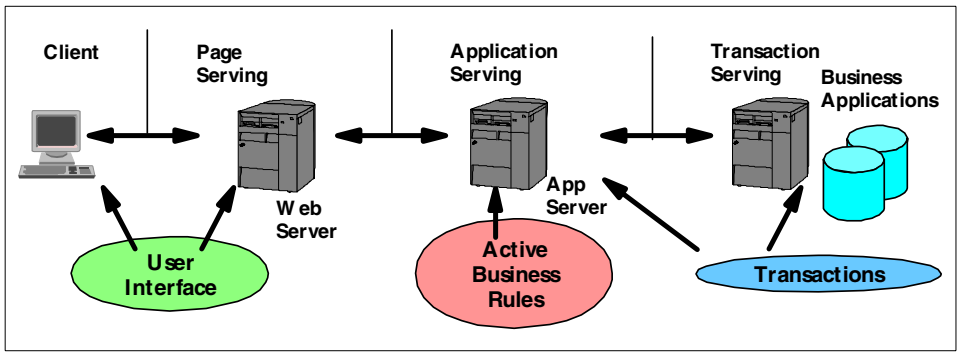


Figure 28. Web server, application server, and transactions server

6.1.1.1 Web (or HTTP) server

The Web server's primary purpose is to serve Web pages to a client browser. A business application for this technology is to develop a static Web catalog

to list product descriptions (including graphics and pictures) and pricing. It also includes instructions for ordering the product using a non-Web means.

6.1.1.2 Application server

An application server mediates between the Web and business applications. This type of server can be described as “Web middleware”, or a middle tier in a three-tier e-business environment. In the tiered environment, the bottom tier is the HTTP server and the highest tier is the business database (that is, DB2) and the business logic (traditional business applications such as order processing). An e-business example of this kind in a dynamic Web site environment involves automatically linking the Web catalog to price changes in the customer’s price database.

An application server is a set of routines or software allowing the user to run server-side applications, such as servlets, JavaServer Pages and Enterprise JavaBeans. These applications can serve as a link to the existing “legacy” applications or database information.

Application servers naturally support Web-based applications. They provide the services and tools to generate Web-user interfaces. They support all of the Web-related protocols and infrastructure. Application servers apply active business rules and run business logic (filters, rules, and so on). Several different Web programming models are typically employed, including HTML, CGI (implemented using the HTTP server in the WebSphere environment), and the above mentioned Java servlets, and JavaServer Pages (JSPs).

Typical features of application servers include:

- Managing browser and Java-based client sessions
- Support to run server-side e-business logic
- Enabling connections to back-end computing resources for data and transaction processing
- Simplified development and deployment of Web-enabled e-business applications
- Support of connections to legacy data and applications
- Application deployment and management tools

Application servers work in concert with a development tool that supports servlets and JSPs. WebSphere Studio provides this support.

6.1.1.3 Transaction server

Transaction servers are often the same as, or a more capable version of, the application server. A transaction server is required when a customer wants to extend e-business across the entire spectrum of business processes to encompass transaction processing and run the actual state of the business.

A catalog company wanting to acquire a transaction server can introduce browser-initiated ordering of their products. This requires *integration* of Web customer ordering with the existing order processing system. The Web application performs some level of processing before accepting and confirming an order.

If a company using a transaction server wants to use the existing ordering program, then *connectors* can be employed. Connectors are Java classes that interact with another program or data source. Connectors work with a program on the same machine or serve as a proxy to a program running on another machine.

The transaction server customer can take this concept further and evolve the Web ordering process to automatically integrate with the inventory system. Potential applications include an automatic ordering of products when their inventory falls below a set threshold level. This involves transactions scoped across more than one data source (also known as *distributed transactions*).

As well, *persistence* requirements can also grow along with the acceptance of the Web application. *Persistenc*e refers to the process of moving data between the customer's relational database and Java objects.

Transaction servers often employ Enterprise JavaBeans (EJB), a key technology for dealing with distributed transaction and complex persistence issues in a Web application. In addition to a Web development tool supporting servlets and JSPs, a transaction server can also employ a Java Integrated Development Environment (IDE) such as VisualAge for Java Professional Edition (included with WebSphere Studio).

6.1.2 Overview of IBM WebSphere family

WebSphere is a brand name for a set of IBM products designed to make it easier and more productive to build, deploy, and manage a dynamic Web site. WebSphere's architecture enables you to build business-critical applications for the Web.

The IBM WebSphere family consists of the following components:

- WebSphere Application Server
- WebSphere Studio
- WebSphere Performance Pack
- WebSphere Site Analysis

For the most current information on IBM WebSphere support, go to the site at: `http://www.software.ibm.com/webservers`

6.1.3 The IBM WebSphere Application Server

As the popularity of Java continues to increase, the emerging program model that people are expected to move to is the application server model. The IBM offering for the iSeries server is the WebSphere Application Server. It is a Java-based servlet engine that is built on top of the native Java virtual machine on the iSeries server. It provides Java servlet API support, which is defined by Sun Microsystems.

If you write to the Java servlet API standard, your application is portable across any operating system and any environment that supports servlets. This is a big reason why Java servlets are a popular interface to write to.

Figure 29 shows how servlet support is provided on the iSeries server. The browser-based application is at the bottom of the figure. A database or file or some other resource that the browser application wants to access through a servlet are shown at the top of the figure.

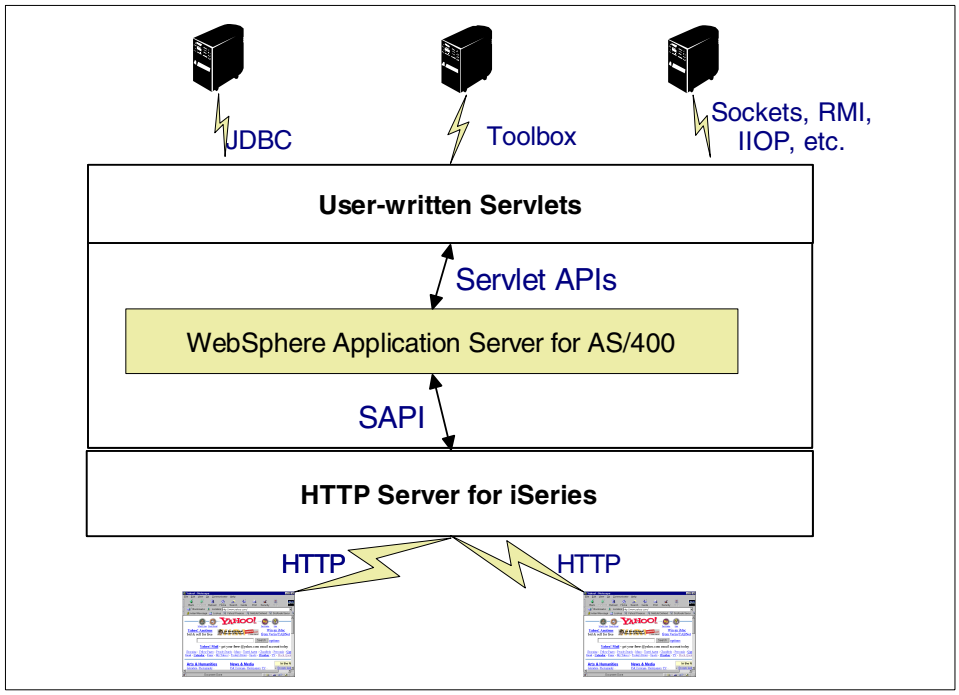


Figure 29. AS/400 servlet support

The browser always interfaces to the Web server, whether it is the Web server built into OS/400, Apache, or Netscape. In the case of the AS/400 system, it is the IBM HTTP Server (Original) (5769-DG1).

The WebSphere Application Server is an add-on product to the HTTP server. It provides servlet API support. The user-written servlet interfaces with these APIs. The WebSphere Application Server interfaces to the Web server using the Web server's API. On the iSeries server, it uses the Server API (SAPI) interface, which is also called IBM Connection API (ICAPI). This path is used on the iSeries server because the HTTP server provides this level of API support. The WebSphere Application Server interfaces to the IBM Web server, which, in turn, interfaces to the browser.

On other platforms, the WebSphere Application Server also supports the Apache API, ISAPI, which is the API that the Microsoft Web server supports, and NSAPI, which is the Netscape API. This means that the WebSphere Application Server provides servlet support for almost any Web server.

The WebSphere Application Server comes with a graphical user interface to manage servlets and control who has access to them. Since most

applications access a database, there are many advanced database access capabilities available. This includes a connection manager to improve performance when accessing remote databases and data access beans that make accessing the database even easier for Java programmers.

The WebSphere Application Server also comes with JavaServer Pages support. The JavaServer Pages technology is a simple but powerful way to dynamically generate HTML on the server side. With JavaServer Pages technology, you can quickly and easily create Web pages with dynamically generated content. When you write JavaServer Pages, you call reusable server components (such as JavaBeans or servlets) to automatically create data objects from an HTML file. This allows you to cleanly separate the generation of dynamic content from its presentation.

There are three editions of the WebSphere Application Server:

- **Standard Edition**: The Standard Edition is intended for use by Web application developers who focus on the issues of presentation logic, data access, and the business logic that resides in the middle tier. The focus is on servlet run times, HTML, and support for JavaServer Pages (JSPs). This combination provides an easy-to-use set of capabilities for presentation and data access-oriented Web applications.
- **Advanced Edition**: The Advanced Edition adds support for Enterprise JavaBeans. EJB components are reusable, portable, server-side business logic components. IBM has adopted the EJB model to deliver productivity and flexibility to customers who are building sophisticated Web applications. This unified component model, supported across the IBM product line, makes it easier for customers to build, run, and manage the new generation of e-business applications, using tools and platforms that best meet their business needs.
  - EJB support includes:
  - Bean-managed and session-managed persistence using entity beans
  - Full support for session beans
  - Relational database connectivity using JDBC (allowing support of the Java Transaction Service, or JTS using EJBs)
  - Support for EJBs to MQSeries and CICS for even more robust transactional support
- **Enterprise Edition**: Integrating enterprise applications is the focus of the Enterprise Edition. This edition adds to the EJB and services support of the Advanced Edition the ability to integrate applications across the enterprise using the robust architecture and services provided as part of

the CORBA standard. Enterprise Edition leverages the connectivity and application integration capabilities of IBM Component Broker technology, which has been incorporated into WebSphere. In addition, users can deploy EJBs either to the same EJB container that is part of the Advanced Edition or to the Component Broker container.

The AS/400e and iSeries servers do not support the Enterprise Edition of WebSphere Application Server.

The WebSphere Application Server is available in three different versions. Table 1 on page 37 identifies the editions and versions of the WebSphere Application Server that are available for OS/400 V4R3, V4R4, and V4R5.

In OS/400 V4R3, the IBM WebSphere Application Server for OS/400 is included with IBM HTTP Server for AS/400 (5769-DG1). The WebSphere Application Server for OS/400 is functionally equivalent to the WebSphere Application Server Version 1.1.

In OS/400 V4R4 (and V4R5), the WebSphere Application Server was taken out of the IBM HTTP Server product and packaged as its own product, WebSphere Application Server for AS/400 (5769-AS1, a feature of OS/400). This facilitates its use with the HTTP Server for iSeries, as well as with other Web server technologies.

As with V4R3, the WebSphere Application Server for AS/400 continues to be available as a no-charge, optional licensed program. The V4R4 and V4R5 product is shipped with the base operating system. It provides WebSphere Version 1.1 support.

It can be upgraded to WebSphere Application Server Standard Edition Version 2.03 through group PTF SF99027. The Standard Edition includes support for Java servlets, JavaServer Pages, and an XML parser.

WebSphere Application Server Standard Edition Version 3.0 is available as program product 5733-AS2 (56-bit encryption support) or 5733-AS3 (128-bit encryption support). Group PTFs are available which upgrades these products to the 3.02.2 level. They require JDK 1.1.7 or 1.1.8 support.

WebSphere Application Server Advanced Edition Version 3.0 has been available since February 2000 as program product 5733-WA2 (56-bit encryption support) or 5733-WA3 (128-bit encryption support). It is a chargeable licensed program option. The Advanced Edition includes support for Enterprise JavaBeans, scalability support, and workload management support. A group PTF is available that upgrades this product to the 3.02.2 level. It requires JDK 1.1.7 or 1.1.8 support.

In October 2000, Version 3.5 for both the Standard Edition and Advanced Edition of WebSphere Application Server became available.

WebSphere Application Server Advanced Edition Version 3.5 is available as program product 5733-WA3 (128-bit encryption support). Version 3.5 does not offer 56-bit encryption support because US government export regulations have been relaxed. WebSphere Application Server Advanced Edition Version 3.5 requires JDK 1.2.2 support.

WebSphere Application Server Standard Edition Version 3.5 is available as program product 5733-AS3 (128-bit encryption support). WebSphere Application Server Standard Edition Version 3.5 requires JDK 1.2.2 support.

Complete details on the PTF and service information for WebSphere are available at:
`http://www.as400.ibm.com/tstudio/websphere/services/service.htm`

6.1.4 WebSphere Studio for AS/400

WebSphere Studio is optimized specifically for WebSphere Application Server applications. However, you can also use many of the Studio tools for the development of applications for other Web servers or Web application servers. You run the WebSphere Studio development tools on a Windows workstation. You can deploy the resulting application to an iSeries server running either WebSphere Application Server Standard Edition or Advanced Edition as shown in Figure 30.

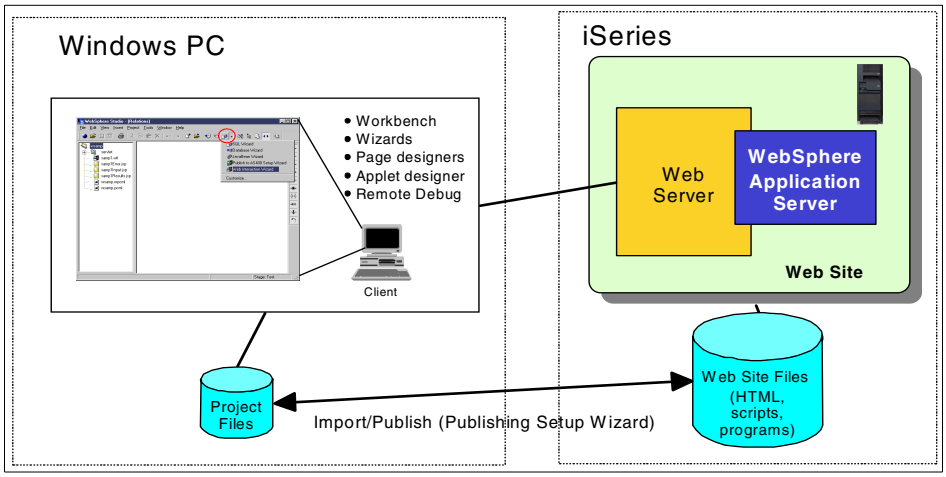


Figure 30. WebSphere Studio overview

WebSphere Studio provides tools you can use to manage your Web application project and for the creation of HTML, Java, and JSPs including graphics and database access. WebSphere Studio maintains project files in the file system and provides support for team development and version control tools. The deployment features of WebSphere Studio enable you to configure the projects to deploy to a number of locations, such as the WebSphere Application Server or the WebSphere Test Environment of VisualAge for Java.

WebSphere Studio for AS/400 now incorporates AS/400 Affinity. AS/400 Affinity is designed for OS/400 application developers who want to develop e-business applications but do not have time to acquire the additional skills required to develop Web-enabled applications. WebSphere Studio for AS/400 now gives application developers, with traditional AS/400 programming skills, the ability to quickly develop e-business applications without having to learn Java and other Web-application skills. You (the developer) can concentrate on the underlying business application logic residing on the OS/400 host, using current ILE RPG (or any other high-level language) skills. You can use the intuitive aspects of WebSphere Studio for AS/400 to design the Web-based front end. And, you can generate the JavaServer Pages (JSP) and Java servlets required to enable the new e-business application.

WebSphere Studio for AS/400 SmartGuides (Wizards) takes you through the steps required to create Web input and output pages. It allows you to define the parameters associated with the design-time controls on the Web pages and link the fields on the Web pages to the parameters in the business logic. You can do all of this without having to deal with JavaServer Pages, JavaScript code, and servlet code.

WebSphere Studio for AS/400 gives you the ability to create new e-business applications using a Web-based front end that communicates with the business logic of an ILE language program residing on the OS/400 host. You can create your Web input and output pages with WebSphere Studio and its Page Designer. Or, you can generate input and output pages from templates listed in the new Web Interaction wizard.

The AS/400 Web Interaction wizard also helps you perform these tasks:

- Define the parameters associated with the design-time controls on your Web pages, without the need to deal directly with JavaServer Pages (JSP), JavaScript code, and servlet code.
- Link the fields on your Web pages to the parameters in the ILE business logic.

Use the Publishing Setup wizard to identify your iSeries servers and to define the publishing information used by WebSphere Studio's publishing function. Once this is done, you can deploy an iteration of your application for testing or deploy the final version for production purposes.

WebSphere Studio for AS/400 includes the following OS/400-specific functions:

- A Web Interaction wizard that creates the underlying code used to communicate between the Web pages and the business logic
- OS/400-specific, design-time controls for your Web input and output pages
- A Publishing Setup wizard designed to help you set up your publishing information before you deploy iSeries e-business applications to either a test or production environment

WebSphere Studio contains a number of wizards that guide you through such tasks as SQL statement generation and creation of Web pages to interact with databases and JavaBeans. You can also use the WebSphere Studio Page Designer to edit these generated pages. The following tools are included in WebSphere Studio Version 3.5:

- Studio Workbench
- Studio Wizards
- Page Designer
- WebArt Designer
- Applet Designer

6.1.5 WebSphere Performance Pack

WebSphere Performance Pack is a set of tools for caching, load balancing, and Web-site replication. This is not an IBM iSeries licensed product, but it is available from the IBM Software Group.

WebSphere Performance Pack comes in two versions. The first is *Cache Manager*. It provides a proxy caching capability. The HTTP Server for iSeries Web server has caching proxy built into it. The Cache Manager is an enhancement over that. It uses more sophisticated algorithms and has some additional features. It is basically a Web server that acts as a caching proxy and does that very well. It also has a built-in Platform for Internet Content Specifications (PICS) filtering platform. This is a self-rating system. You can put information on your Web page that basically rates the contents of the Web page for violence, nudity, and so on. In the caching proxy, you can filter out those types of sites. Anybody using the caching proxy to access the Internet

will never see the filtered content in their browser. This is especially useful in an educational environment where you're trying to filter that type of information away from the students.

Performance Pack Cache Manager is not currently available for the iSeries server.

Availability Services provides the network dispatching capability or load balancing capability. This is needed for scalability or high availability services. For example, you may want multiple iSeries servers to serve requests.

Perhaps you have one Web address, but you need more than one server to serve the requests to deliver adequate response time. You need multiple machines to look like one machine. It is usually a load balancer that does that. All the requests come into the load balancer and they are forwarded to the different Web server machines based on their capability of handling the requests. The load balancing software does not run on the iSeries server. It runs on an external box including an IBM 2216 Router, AIX, SUN, or Windows NT box. It is a front end to multiple systems, including multiple iSeries servers. This provides for scalability and high availability.

6.1.6 WebSphere Site Analyzer

The WebSphere Site Analyzer tools provide site administration and analysis tools to administer and manage the usage of a Web site. The collected statistics reside on the server, for example, an iSeries server. This data is downloaded to a Windows 32-based workstation with the site analysis tools installed. The tools perform the analysis and have graphical display functions to display the results. This is not an IBM iSeries licensed product, but it is available from IBM Software Group.

The Site Analyzer Tools includes functions such as:

- An administration site visualizer
- A report generator and builder
- A content analyzer to scan the Web site and identify such items as duplicates, unavailable resources, and content with excessive size
- A usage analyzer to scan the Web logs for hits, requests, paths, agents, and so on (analysis can be scheduled and results placed in a database)

6.1.7 Putting it all together

WebSphere products are available for many servers, including IBM servers. Figure 31 on page 116 shows the family of WebSphere products and whether

they are used for building, running, or managing e-business. Section 6.2, “WebSphere Application Server for AS/400: Product description” on page 117, discusses the WebSphere products that apply to the iSeries server family.

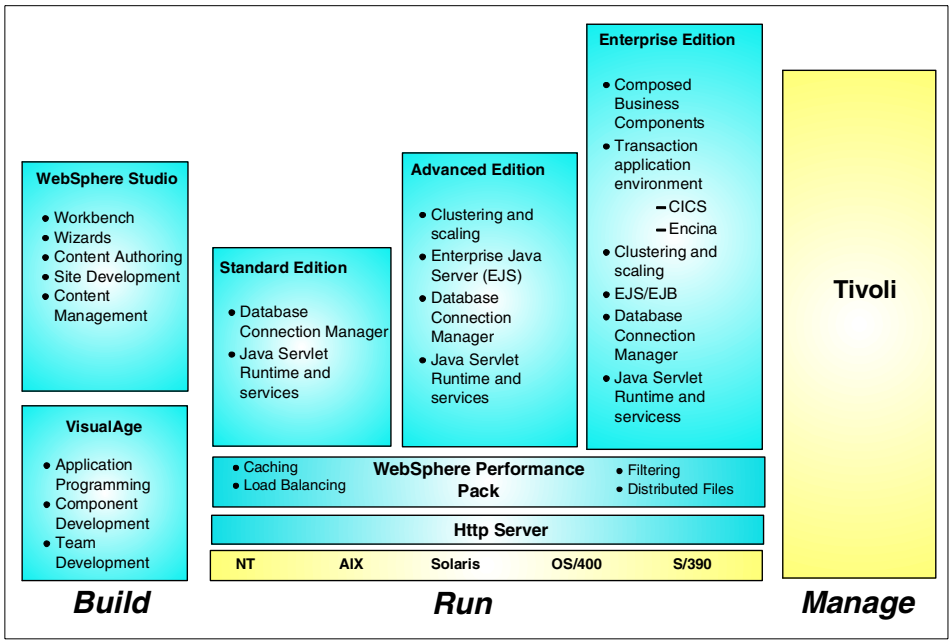


Figure 31. The WebSphere product family

The functional highlights include:

- WebSphere Studio and VisualAge for Java are used to build the applications.
- WebSphere Application Server is used to run the applications. It works in conjunction with the HTTP server.
- WebSphere Performance pack is used for caching, load balancing, and Web-site replication.
- Tivoli is used for change management.

For the latest information on WebSphere on the iSeries server, go to the site at: `http://www.as400.ibm.com/products/websphere/index.htm`

6.2 WebSphere Application Server for AS/400: Product description

This section explains the products of IBM WebSphere Application Server for AS/400. Figure 32 provides an iSeries-specific view of the WebSphere product family. Each member is described in the following sections.

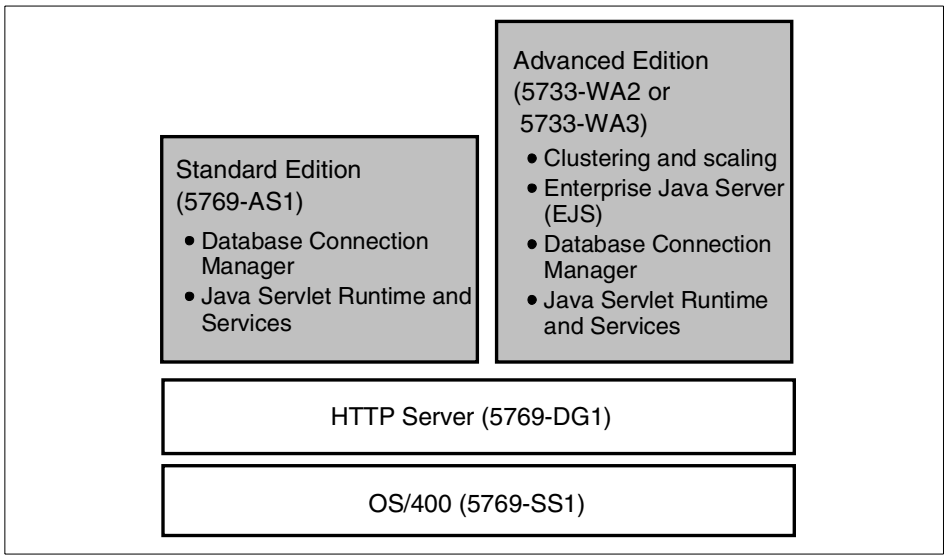


Figure 32. AS/400 WebSphere product family

The iSeries server currently supports the Standard and Advanced Edition of WebSphere both at Version 3.5.

For the latest versions of the products supported for the iSeries, refer to:
`http://www.as400.ibm.com/products/websphere/index.htm`

6.2.1 WebSphere Application Server for AS/400: Standard Edition

WebSphere Application Server for AS/400 Standard Edition is a no charge Licensed Program Product (LPP), 5769-AS1. It comes in OS/400 as part of the base package. It features the following components and services:

- Plug-in for IBM HTTP Server for iSeries
- Support for servlets, including:
  - Java-based servlet engine
  - Implementation of the JavaSoft Java servlet API
  - IBM extensions and additions to the servlet API

- JavaServer Pages (JSP) support
- XML Document Structure Services
- Connection management feature to cache and reuse JDBC connections
- WebSphere Application Server Manager, the graphical administration interface for WebSphere Application Server

6.2.1.1 Plug-in for IBM HTTP Server for iSeries

The IBM HTTP Server for iSeries handles static content, Common Gateway Interface (CGI) program invocations, and proprietary plug-ins. The run-time environment (WebSphere Application Server) plugs into IBM HTTP Server for iSeries using plug-in APIs. Then, it routes servlet requests to the servlet manager, which then takes care of handling the request and passing the data back to the client.

6.2.1.2 Servlets

Within the middle-tier and the Web Application Server environment, the focus is on the Servlet Engine, which is Java-based. The servlet run time provides the Sun/JavaSoft APIs for the Java servlet environment, including the servlet life cycle: init, service, and destroy. Servlets can be preloaded, so that when a client request comes in, a servlet is loaded and waiting to act on it.

Servlets send and receive most of their data through output and input streams. These streams are supplied each time a servlet is invoked using service callback. A popular specialization of servlets provides a function designed to make it easy to read parameters from a URL and send HTTP output in response to that URL request.

The servlet manager (part of WebSphere Application Server Manager) creates instances of the servlets, deploys them, manages their execution, and provides tracing and monitoring facilities for them. The servlets themselves handle HTTP requests, maintain an HTTP session with the client, produce presentation logic using HTML, stream, and non-transactional business logic. Servlets can also call component functions or routines built as JavaBeans. These beans can be called to connect to, or interface with, remote systems of various types and formats using different native APIs.

6.2.1.3 JavaServer Pages (JSP)

The WebSphere Application Server servlet engine also handles requests for JavaServer Pages (JSP) or server-side HTML scripting. These dynamic requests are normally not as “short-lived” as static ones. Therefore, the limit of how many concurrent requests this engine can handle is smaller than for static serving, and the response time is longer. This is to be expected, since

we are now running applications and dynamic content, and not just sending static bytes back to the client.

6.2.1.4 XML

WebSphere Application Server supports XML document structures and can generate, validate, parse, and serve XML content.

6.2.1.5 Connection Manager

The Connection Manager manages a pool of relational database connections. Coupled with some data access JavaBeans, the Connection Manager connects to a remote database, pulls in required data, saves it into a local cache, and disconnects. The JavaBeans then work with the data in the cache to dynamically generate Web content.

6.2.1.6 WebSphere Application Server Manager

WebSphere Application Server Manager is a browser-based graphical interface. It allows you to change WebSphere Application Server settings and to configure and monitor Java resources, WebSphere Application Server security, JDBC connections, active sessions, and logs.

After you install WebSphere Application Server, you can locate the Manager by directing your browser to the URL:

```
http://your.server.name:9090
```

Here, *your.server.name* is the host name of your Web server.

6.2.1.7 WebSphere Standard Edition V3.5

This latest version of the product features:

- Support for JavaServer Pages:
  - Support for specifications .91 and 1.0
  - Extended tagging support for queries and connection management
  - An XML-compliant DTD for JSPs
- Support for the Java Servlet API 2.1 specification including automatic user session and user state management
- High-speed pooled database access using JDBC
- XML server tools, including a parser and data transformation tools
- Additional integration with IBM VisualAge for Java to help reduce development time by allowing developers to remotely test and debug Web-based applications

6.2.2 WebSphere Application Server for AS/400: Advanced Edition

WebSphere Application Server Advanced Edition Version 3.5 is available as program product 5733-WA3 (128-bit encryption support). Version 3.5 does not offer 56-bit encryption support because US government export regulations have been relaxed. WebSphere Application Server Advanced Edition Version 3.5 requires JDK 1.2.2 support. It is a chargeable licensed program option (LPO) and priced per iSeries CPU.

The WebSphere Application Server Advanced Edition 3.5 for AS/400 is part of an extended brand of IBM products for e-business. It goes beyond the capabilities of the WebSphere Standard Edition product, servlets, and JavaServer Pages (JSP) support, to provide support for Enterprise JavaBeans (EJBs). EJBs allow you to develop sophisticated server-side components (or objects) for your business. This model may include:

- Business applications
- Internet- or intranet-based applications that require integration into existing iSeries applications
- New business applications that require complex database integration with a heterogeneous, multi-database vendor environment

When used with development products, such as VisualAge for Java Enterprise Edition 3.0, WebSphere Application Server Advanced Edition 3.x provides a modern development environment for building Java applications. This environment provides the facilities to provide similar levels of integration, security, and transactional support that was previously only available to the traditional AS/400 developers utilizing ILE languages. The component technology that is used, EJBs, is *the* standard within the Java community for building components.

WebSphere Application Server Advanced Edition 3.x implements the EJB Version 1.0 specification, with some 1.1 specification enhancements, particularly in the area of the finder helper methods. For more information on EJBs, refer to the Sun Microsystems Enterprise JavaBeans Technology Web pages at: `http://java.sun.com/products/ejb/index.html`

6.2.2.1 Advanced Edition product features

WebSphere Application Server Advanced Edition 3.x product features include:

- Support for JavaServer Pages:
  - Support for specifications .91 and 1.0
  - Extended tagging support for queries and connection management

  - An XML-compliant DTD for JSPs
- Support for the Java Servlet API 2.1 specification including automatic user session and user state management
- Full support for the Enterprise JavaBeans (EJB) 1.0 specification, including both SessionBeans and EntityBeans (container-managed and bean-managed persistence)
- Deployment support for EJBs, Java servlets, and JSPs with performance and scalability improvements, including:
  - Application-level partitioning
  - Load balancing
- High-speed pooled database access using JDBC
- XML server tools, including a parser and data transformation tools
- A Web site analysis tool for developing traffic measurements to help improve the performance and effectiveness of your Web sites
- Additional integration with IBM VisualAge for Java to help reduce development time by allowing developers to remotely test and debug Web-based applications
- Enhanced support for distributed transactions and transaction processing
- Improved management and security controls, including:
  - User and group level setup
  - Method level policy and control

Performance and scalability

The Advanced Edition includes application-level workload management and clustering, with enhanced container deployment environment services for EJBs, servlets, and JSPs. Improved transaction management intelligently deploys and executes across multiple applications and components, therefore, optimizing object management and performance.

The Advanced Edition focuses on higher performance and scalability across the deployment environment. It addresses load balancing, application partitioning, and workload management for EJB components. This involves enhanced and added “container” deployment environment services for servlets, EJB, and JSP technology. The improved transaction management has intelligence on the type of applications being executed and can deploy and execute across multiple applications and components accordingly. Object management and performance are optimized.

Transaction management
The Advanced Edition includes deployment and management capabilities for Java applications and EJB components that allow powerful interactions with enterprise databases, transaction processing systems, and other applications.

The EJB server provides better monitoring and control of containers used within the server. It offers enhanced Java Transaction Service (JTS) support within the EJB server, better interoperability, and a major rewrite of the deployment environment.

A single, robust server offers better control, more flexibility, and better serviceability for the deployment and management of JavaBeans, Java servlets, JSPs, and applications built to the EJB specification for execution and transaction management.

Container management and persistent storage with DB2 Universal Database help provide a high-performance transactional environment using servlets and EJBs.

Site analysis
A completely new and exciting package of site usage and analysis tools are included with WebSphere Application Server Advanced Edition. Site analysis focuses on developing basic traffic measurement functions while allowing the user to gauge traffic volume (hits, visits), identify traffic sources (domains, subdomains, referrers), and manage site integrity (link verification, site conformance). These are the key features enabling the user to enhance and improve Web site content and performance, making Web sites more enticing, informative, and ultimately beneficial to the “bottom line” of your business.

The new “server” contains the analyzers whose responsibility is to transform the raw data into valuable information and store it into the provided DB2 UDB database. The client provides administrative, visualizing, and report-generating functions. From the client, the user may schedule analysis tasks to run at a specific time or time interval. Progress status is broadcast to interested clients and displayed as appropriate and necessary. Once the analysis is complete, users can generate reports or view the data. The system offers many predefined, ready-to-use reports.

6.2.3 Final look: Architectural view of all components

We covered all major technology components of WebSphere Application Server and their relationships with other components. We also covered how the WebSphere Application Server product is implemented on the iSeries

server. Let's take a final look at the architectural view of all related components as shown in Figure 33. See if you can explain comfortably each component and their relationship.

Note

Customers who need iSeries Support Line for WebSphere need to purchase the Internet Product group of support offerings.

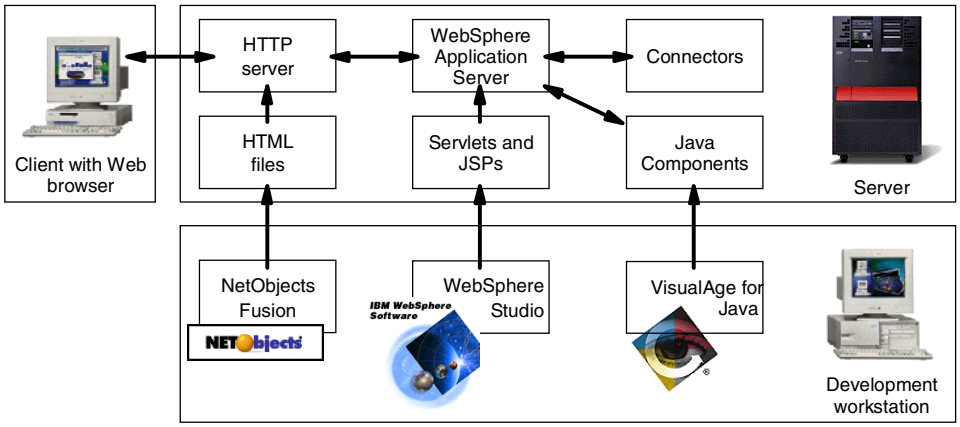


Figure 33. WebSphere Application Server with HTTP server and tools

6.3 Case study: Welch's Foods, Inc.

We conclude this chapter with a case study of a real-life customer testimonial from Welch's Foods, Inc. This case study shows how Welch's Foods used WebSphere Application Server for AS/400 to build their e-business and achieve success with it.

Vintage IT infrastructure with IBM e-business solution

"We wanted to run our new Web-based application on our AS/400 because that's where our company data resides, and we wanted a programming environment that gave us the flexibility to run portions of the application on any other platform as well. Java was the obvious answer. Our investment in Java is paying big dividends," says *Deepak Mohapatra, Manager of Operations and Networks, Welch's Foods, Inc.*

Founded in 1869, Massachusetts-based Welch's Foods, Inc. (`http://www.welchs.com`) is a well-established firm by any standards. Welch's

Foods, Inc. has 1,350 employees with headquarters in Concord, Mass. The company is owned by the 1,500 members of National Grape Cooperative.

Welch's has been an American household name for many years. It is well known for its juices, jams, and jelly products. Welch's also comes to our attention as a company embracing technology to make its business more effective proving that "old" doesn't mean old-fashioned.

Welch's first became an IBM AS/400 customer in 1993, when it moved its core business applications from the mainframe to the AS/400 system. With IBM Business Partners Marcam and J.D. Edwards providing ERP and financial packages, the AS/400 system became the center of the company's information technology (IT) strategy.

By 1995, with the bread and butter applications established on the AS/400 system, Welch's began to explore ways to use information technology (IT) to expand business opportunity. Welch's manager of operations and networks, Deepak Mohapatra explains, "We were looking at ways to deliver decision support systems and to better include our suppliers and brokers in our business process." Welch's built an AS/400 data warehouse that provided the sales force with timely information and aided in decision support. This effort eliminated about forty-thousand reports from their former mainframe days.

Beginning in 1996, the company began to look at ways to use the emerging Internet to improve its business. By 1999, Welch's had rolled out three major new AS/400e applications all Web-based, all using Java and the IBM WebSphere Application Server.

Welch's has kept loyal customers over the years by providing consistently high-quality products. Concord grapes taste just like they did in 1869 when the world was low-tech, but it's AS/400e high-tech solutions that help to bring these quality grape products to your table in 2000.

Solving a promotion challenge

We have all seen special promotions like free tasting samples when we shop for groceries. Welch's, like most other food product producers, conducts these promotions at the retail locations as a way to attract new buyers. In fact, the company budgets $100 million in its marketing development fund each year to support various types of in-store promotional events. Management had always seen this as a necessary investment, but a nightmare to manage and measure. Deepak Mohapatra explains, "We had a complex process based on spreadsheets that tried to keep track of what events had taken place and what moneys were owed. We were never really sure what had happened and our "unresolved" spending account was very

large.” In addition, there was no quick and easy way to measure the results of a promotion event to see if it actually had a positive effect on sales.

As the Welch's IT staff first sought ways to get the Marketing Development Fund process under control, it looked at outside service offerings. Eventually Welch’s rejected them because they required Welch's brokers and retailers to invest in equipment and IT skills. What was really needed was a system that was “universally” available to all participants, even those sales people who worked out of their homes. Welch's decided to develop a system in-house and use the Internet and browser-based PCs to connect.

This led to the development of an AS/400e-based Java solution where Welch's brokers, retailers, sales staff, and customer service representatives all see the same data regarding promotional events and the associated payments. This improved promotion management system is expected to yield between 10 and 20 percent savings, returning their investment in six months.

Java for programming

According to Deepak Mohapatra, “We knew that our application had to run in a browser to satisfy the no-client maintenance requirement. We looked at ActiveX controls, but rejected that approach because it would also involve installation of code on the PC. So Java was our answer.” Once deciding on Java, Welch's approach was to make all Java code server-based.

Coding began in 1997. The programming model consists of servlets for AS/400e code and DHTML and Swing 1.2-based applets for the client side. By early 1999, all retailers and brokers were communicating with Welch's AS/400e server from browser-based PCs and thin clients using the Internet backbone. Java applications provide screens to enter promotional events and track them through the payment cycle. What was once a nightmare has become a well-managed process where all parties see the same data. Results are noted immediately, and decisions can be taken to repeat successful events or cancel unsuccessful ones.

Welch's went through extensive education of its programming staff to create the skill base for the new application set. Today, five programmers are fully skilled at object-oriented programming methodologies and Java coding. This resulted in a bonus of two other Java-based applications rolled out in 1999. Welch's employees can now access human resource files and receive self-service 24 hours-a-day, 7 days-a-week for most personnel and benefit-related matters. A new forecasting and reporting system allows sales people and brokers provide their forecasting data using the Internet and shows them the actual shipments in real time.

WebSphere as the server

When Welch's first wrote its prototype for the Marketing Development Fund application, IBM WebSphere Application Server was not available. Then, the IT group wrote its own AS/400 server engine.

Welch's was one of the first beta customers of the IBM AS/400e Model 720 in 1998. That system was dedicated to the new Web-based application. When WebSphere Application Server was released in 1998, Welch's restructured the application to take advantage of the servlet APIs and monitoring functions.

Working with IBM all along, the addition of WebSphere Application Server to the application went very well. Mohapatra summarizes his company's IBM experience in the project. He says, “We had direct access to the Java and WebSphere development teams. This was very helpful in giving us the skills we needed to write and deploy this very successful AS/400 Java application.”

With the final implementation of WebSphere, the Welch’s project achieved 100% ROI in less than six months, a 10% to 20% reduction in operational costs, and enhanced customer service.

For more case studies of AS/400e, iSeries, and WebSphere interaction, see the site at: `http://www2.software.ibm.com/casestudies/swcsweb.nsf/swgSearch?SearchView&Query=AS/400+AND+WebSphere`

6.4 References

This following sources provide references for WebSphere Application Server for AS/400 for both Standard Edition and Advanced Edition.

Web sites

- WebSphere Application Server for AS/400 home page: `http://www.as400.ibm.com//tstudio/websphere/docs/doc.htm`
- For initial sizing of the iSeries server for WebSphere Application Server: `http://as400service.ibm.com/estimator`

  This site provides the sizing service for multiple workload types. From the pull-down menu, under Workload Type, select **WebSphere**.
- IBM e-business home page: `http://www.software.ibm.com/e-business`
- AS/400 North America home page: `http://www.as400.ibm.com/na`

  Among other things, this site provides the quickest access to AS/400 Web Events including a “Lunch and Learn” series.

- Link to various iSeries products and their PTFs at:
  `http://www.as400.ibm.com/misc/map.htm`

Publications

- *Building AS/400 Applications for IBM WebSphere Standard Edition 2.0,* SG24-5635
- *WebSphere Scalability: WLM and Clustering Using WebSphere Application Server Advanced Edition*, SG24-6153

Chapter 7. WebSphere Commerce Suite for AS/400

What is WebSphere Commerce Suite? And why should it be your choice among so many possible, nice alternatives to build your e-business sites? One quick answer is “because your primary interest is *e-commerce*”.

No other aspect of e-business has garnered more attention than e-commerce. After all, consumers spent $30 billion online in 1999, and that figure is expected to grow more than $300 billion in the next four years. But, that is just part of the story. E-commerce spending between businesses is expected to be at least ten times more than consumer e-commerce.

WebSphere Commerce Suite V4.1 for AS/400 (formerly called Net.Commerce V3.2 for AS/400) is the optimal choice if you agree to the previous point and are willing to implement e-commerce sites quickly and yet in a highly reliable manner as possible. That is, within e-business, if your business metaphor is e-commerce (both B2C and B2B), then WebSphere Commerce Suite V4.1 for AS/400 is the best path to consider.

7.1 Features and uses of the WebSphere Commerce Suite

As mentioned earlier, e-commerce is the area where opportunity lies. Your business needs to take advantage of this trend. E-commerce is about much more than simply opening up a new, online sales channel. It’s about using technology to streamline your business model, creating savings, and increasing efficiency. It’s about lowering costs and establishing closer, more responsive relationships with your customers, suppliers, and partners.

You can link dealers and factories online, reducing both lag time and paperwork. You can move procurement online by setting up an extranet that links directly to vendors, cutting inventory carrying costs and becoming more responsive to your customers. You can streamline your financial relationships with customers and suppliers by Web enabling billing and payment systems.

The end result? While building customer loyalty, companies can reduce costs by improving order processing efficiency. They can maintain fill rates while reducing inventory and warehousing expenses. And, they can lower the actual dollar costs of sales transactions.

WebSphere Commerce Suite V4.1 for AS/400 (WCS) is the IBM answer to the needs of building and benefiting from a successful e-commerce site. It is designed and architected for dynamic, flexible Web sites integrated with enterprise information.

WCS is an extensible framework with an emphasis on scalability, flexibility, security, and performance. Shoppers can easily find your products with flexible shopping metaphors. It works in a wide variety of environments and handles a wide range of applications.

WCS is a cross-platform architecture. IBM offers various versions of products for AIX, OS/390, OS/400, Solaris, and Windows NT. WebSphere Commerce Suite V4.1 for AS/400 is the IBM e-commerce product for the AS/400e and iSeries servers.

7.1.1 WCS: Design overview

As shown in Figure 34, WebSphere Commerce Suite V4.1 for AS/400 is actually a series of servers that run together to allow you to administer your e-commerce site from a collection of PC-based tools. That is, WebSphere Commerce Server runs on top of WebSphere Application Server (WAS), Advanced Edition. WCS enjoys a tight integration with the DB2 UDB for AS/400 database where both your configuration information and store and shopper data is kept. WAS, in turn, runs on top of the IBM HTTP Server.

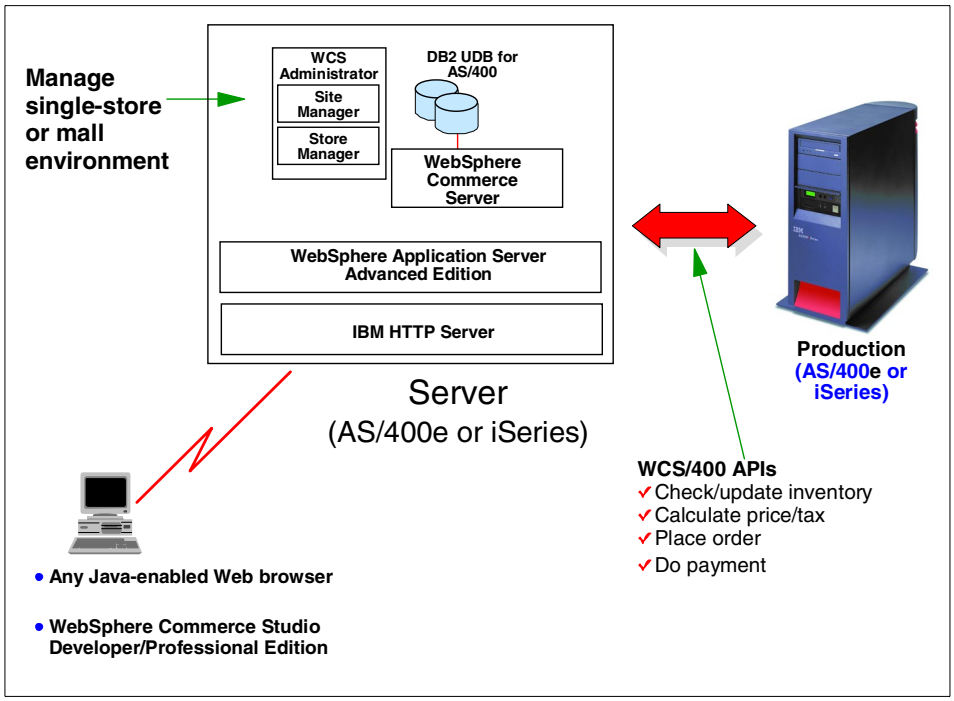


Figure 34. WebSphere Commerce Suite V4.1: How it works

You administer the WebSphere Commerce Server using two separate tools running on a PC (or like device).

One tool is the WCS Administrator, which includes a Site and Store Manager. These are browser-based applications. You can use any Java-enabled Web browser to work with the details of your e-commerce site. For example, you could use this administrative interface to update the price of an item or create a new shopper group (a means to give a group of shoppers a discounted price).

The other tool set is provided by the WebSphere Commerce Studio. This is a collection of very powerful Windows-based applications that allow you to access some of the advanced features of WCS including managing all the file assets such as images, HTML, JSPs, and Net.Data macros. For example, you would use the WebSphere Commerce Studio to create a new store full and complete with images, HTML and the supporting dynamic logic provided by Net.Data (or edit an existing one) and then “publish” the store to the iSeries to be served. The Professional Developers edition also includes a Catalog Architect to visually work with a catalog of maybe thousands of items (versus the WCS Administrators one-at-a-time approach) and Blaze Advisor that allows you to personalize your Web site for each customer.

WCS also comes with a set of APIs that allow you to override the default manner of behavior to allow you to integrate a WCS customer order with your existing back-end business solution. Your production iSeries server could be in the same system as the WebSphere Commerce Server, but is shown in Figure 34 on page 130 separately as an example.

7.1.2 WCS: Advantages

The real questions should be: Which particular product or mechanism should you use to achieve your goal? What benefits would you see from that choice compared to other alternatives? Here’s why WCS can be your best friend in many cases.

7.1.2.1 Packaged offering with robust integrated functions

First of all, WCS is the premier IBM offering that allows many businesses to quickly conduct consumer-to-business transactions through the Internet. It is not an iSeries-specific benefit, but is still important to be reminded (especially when there are multiple choices of doing similar jobs on the iSeries server with another product offering than WCS).

It is a merchant solution that provides a framework to conduct business over the Web in a secure and scalable manner. It supports both business-to-business and business-to-consumer environments.

7.1.2.2 Secured e-commerce solutions

IBM WCS works together with a relational database and a secure Web server to give users and companies a simple and secure environment. It is also combined with IBM Payment Manager and other secured payment methods such as SET Secure Electronic Transaction and CyberCash (for more information about the IBM Payment Manager, see 7.3, "IBM WebSphere Payment Manager" on page 143). Combined with the integrated security features of OS/400, this is an outstanding advantage.

7.1.2.3 Seamless iSeries integration

WCS uses built-in Web serving, Java and, database capabilities. This way, the iSeries server extends such benefits of WCS as:

- Built for business reliability
- Industry's best availability
- Industry leading security to protect information assets
- Extensive scalability to accommodate growth
- Legendary service and support

Most importantly, you can maximize the impact by Web enabling your already robust back-end applications using WCS.

7.1.2.4 Industry standards architecture

The WCS architecture itself is an industry standards implementation. It is a cross-industry and cross-platform architecture. This is not an exception for its iSeries implementation WCS.

7.1.2.5 Scalability

WCS is designed to be scalable to meet the needs of the small to large business. Merchants can take advantage of their existing operating environment and expand to larger systems as their electronic traffic grows. The benefit of this scalability-minded characteristic of WCS architecture can be greatly multiplied when it is combined with, or physically realized on, the ever-increasing hardware scalability of the iSeries server.

7.2 WebSphere Commerce Suite V4.1 for AS/400: Product description

This section provides the product description of WebSphere Commerce Suite V4.1 for AS/400. It includes the mandatory and optional software product

requirements to use WCS and the product components. It then summarizes with an entire architecture view.

7.2.1 Software packaging (5798-WC4)

When you purchase WebSphere Commerce Suite V4.1 for AS/400, you not only receive 5798-WC4 (the licensed product that is the base server engine for WCS/400), but you also receive other server software that runs on your iSeries and client-based application development software that will run on a Windows NT system. This is *the suite* of applications that makes up WebSphere Commerce *Suite*.

7.2.1.1 Server software packaging

The applications in this suite of software that run on the server (AS/400e or iSeries) and are packaged with WebSphere Commerce Suite V4.1 for AS/400 (5798-WC4) are:

- WebSphere Application Server, Advanced Edition
  - 5733-WA2 (56-bit) or
  - 5733-WA3 (128-bit)
- WebSphere Payment Manager (5733-PY2)
  - IBM WebSphere Payment Manager 2.1 Cassettes for SET
  - IBM WebSphere Payment Manager 2.1 Cassettes for CyberCash
- Blaze Advisor Server
  - Blaze Software for personalization
- XML4C parser
- Hot Media
- WebSphere Application Server for Windows NT, AIX, and Solaris

  Interestingly enough, WebSphere Application Server for Windows NT, AIX, and Solaris is provided for you to install the WAS Administrator Console on a Windows NT system to administrate your WAS server environment on your iSeries server.

7.2.1.2 Client (development) software packaging

The applications in this suite of software run on a client or development system. This system is a PC running Windows NT. For this reason, you need to purchase WebSphere Commerce Studio.

Your developers' time is valuable, and so are the tools that help increase their productivity. IBM WebSphere Commerce Studio, Developer and Professional Developer Editions (based on the award-winning IBM WebSphere Studio,

Version 3) deliver a complete design and development environment that speeds creation and maintenance of your e-commerce site.

Your choice is between the Developer and Professional Developer editions.

WebSphere Commerce Studio, Developer edition
If you choose the Developer edition, you receive:

- WebSphere Studio
- Commerce Suite Studio extensions:
  - Store Creator Wizard
  - Store Profile Editor
  - Store publishing enhancements
- WebSphere Commerce Suite for NT

  This is a developer's license that is provided to allow you to develop and test on a Windows NT platform before you publish to your production AS/400e or iSeries server.

WebSphere Commerce Studio, Professional Developer edition
If you choose the Professional Developer edition, you receive all of the features in the Develop edition, plus:

- Blaze Advisor Builder (for personalization)
- WebSphere Catalog Architect

7.2.2 Software requirements in OS/400

In addition to the software that is provided with your purchase of WebSphere Commerce Suite V4.1 for AS/400 (5798-WC4), you need the following required products and features of OS/400. Most, are included as part of OS/400 or are available as no-charge program products.

- OS/400 V4R4 or above (5769-SS1), which includes:
  - DB2 UDB for AS/400
  - Digital Certificate Manager (option 34)
  - QShell Interpreter (option 30)
- IBM HTTP Server for iSeries (5769-DG1), which includes:
  - Net.Data
- Crypto Access Provider
  - 5769-AC1 (40 bit) or
  - 5769-AC2 (56 bit) or
  - 5769-AC3 (128 bit)

- Java Developer Kit 1.1.7 (5769-JV1)
- AS/400 Toolbox for Java (5769-JC1)
- TCP/IP Connectivity Utilities (5769-TC1)

And, optionally, you can choose:

- AS/400 Client Access for Operations Navigator and 5250 access to the iSeries
  - AS/400 Client Access Family (5769-XW1)
  - Client Access/400 Optimized for Windows (5769-XD1)
- Domino for AS/400, V4.6.2 or above
- MQSeries: Optional; for integration with ERP
- LDAP server or servers: An LDAP server is available as part of OS/400

7.2.3 Hardware sizing for the iSeries server

To size a system to handle an estimated WCS workload, please use the IBM Workload Estimator for iSeries found at:

`http://as400service.ibm.com/estimator/index.html`

7.2.4 WebSphere Commerce Suite for AS/400: Product components

Figure 35 on page 136 provides an overview of WebSphere Commerce Suite V4.1 for AS/400 with its product components. Some of the components are part of WebSphere Commerce Suite V4.1 for AS/400. The components are explained in more detail in the sections that follow.

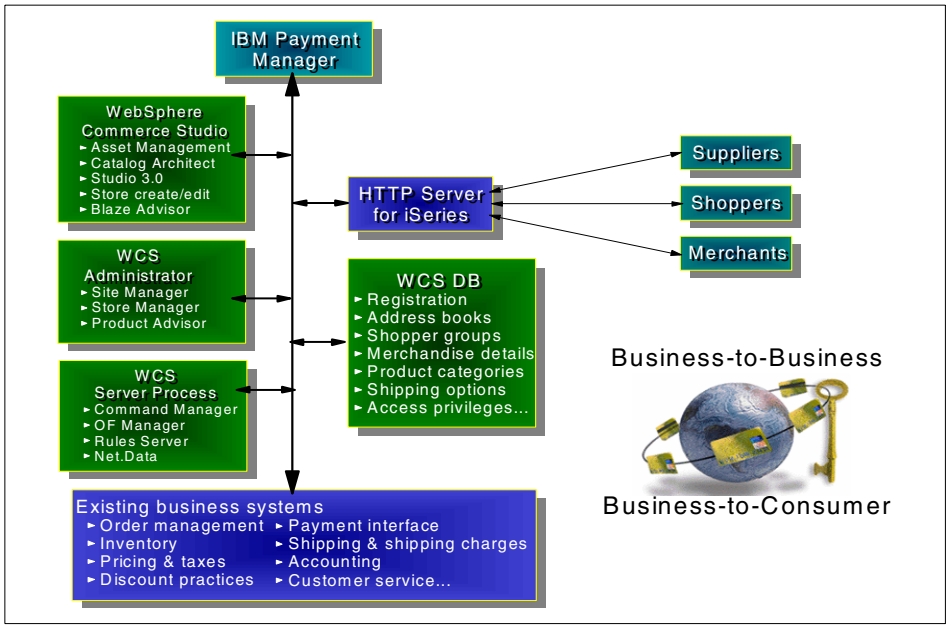


Figure 35. WebSphere Commerce Suite V4.1 for AS/400 product components overview

As shown in Figure 35, the product components of WebSphere Commerce Suite V4.1 for AS/400 include:

- HTTP Server for iSeries
- WCS Administrator
  - Site Manager
  - Store Manager
  - Product Advisor
- WCS Server Process
  - Command Manager
  - OF Manager
  - Rules Server
  - Net.Data
- WCS DB (database)
- IBM Payment Manager (see 7.3, “IBM WebSphere Payment Manager” on page 143 for more information about this separate product that is shipped with WCS).

  IBM WebSphere Payment Manager (5733-PY2) is part of the package when you buy WCS. It also can be purchased separately and be the

Payment Manager for any other e-commerce or e-business solution on your iSeries.

- WebSphere Commerce Studio
  - Asset Management
  - Catalog Architect
  - WebSphere Studio 3.0
    - Page Designer
    - HotMedia
    - IBM PerfectPhoto
  - Store creation wizards and editors
  - Blaze Advisor for personalization

In addition to these components, you need:

- Web browser and a Windows NT system to run the WebSphere Commerce Studio
- DB2 UDB for AS/400
- Existing back-end applications or business systems

7.2.4.1 Web browsers and HTTP server components

Figure 35 highlights the Web browsers and HTTP Server for AS/400. Web browser users can be suppliers, shoppers, or merchants depending on the nature of the service you provide. Many of the interfaces to the WebSphere Commerce Suite (WCS) product are also provided using Web browsers.

All these requests come into the iSeries server through HTTP Server for iSeries. Likewise, all the responses to the Web browser are served by this component.

7.2.4.2 WebSphere Commerce Suite V4.1 for AS/400 Administrator

At the beginning of this chapter, we said that you should build the site as quickly as possible. You also want to dynamically change, update, enhance, or customize the sites. WCS Administrator is a core component that enables most of these functional requirements.

You can use the WCS Administrator to manage an electronic store or mall. You can easily enter store and product information and tailor product information to suit your merchandising requirements. Changes will appear automatically on your Web site.

The WCS Administrator contains two data management applications:

- **Site Manager**: Creates and manages commercial Web sites.
- **Store Manager**: Use to develop an online catalog and to manage such information as shipping options, shopper groups, and customer numbers. Use the simple online forms to enter and update information in the WCS database.

The information you can keep in the database and display in your electronic store includes:

- Store or mall name
- Logo location
- Contact information
- Mission statement
- Policies
- Types of services and products
- Currency used
- Merchandise offered (including descriptions, product number or stock keeping unit, images, prices, availability dates, dimensions, weight, and so on)
- Product categories
- Shipping options and services
- Shopper groups
- Information about the people who have access to the store's database

7.2.4.3 WebSphere Commerce Suite V4.1 for AS/400 server process

This component, which is the actual server engine of WebSphere Commerce Suite V4.1 for AS/400, is explained in 7.2.5, “Architectural view of WebSphere Commerce Suite for AS/400” on page 140.

7.2.4.4 DB2 UDB for AS/400 component

Almost everything you create with WebSphere Commerce Suite V4.1 for AS/400 is stored in DB2 UDB for AS/400. DB2 UDB for AS/400 plays a role in the storage of the resources required to run a WCS implementation.

Two databases

There is one distinction that you are required to make. There are two databases:

- A WCS database
- Existing back-end applications database

Technically, you can run a transactional site purely with the WCS DB without interfacing with an existing back-end database, but it is not practical.

As shown on Figure 35 on page 136, WebSphere Commerce Suite V4.1 for AS/400 DB includes:

- Registration
- Address books
- Shopper groups
- Merchandise details
- Product categories
- Shipping options
- Access privileges

Importing business data into WCS

Many WCS customers already have their own back-end applications for running their business. These back-end applications access and use data in existing databases that contain information relating to customers, products, billing, and inventory control.

When a WCS instance is created, the database created has sufficient information in it to allow the mall to be administrated. However, it has no product, category, or price information unless the option was selected to install a demonstration store or mall.

The WCS database tables contain all the information WCS needs to store information about the mall or store and its operation, such as products, categories, and prices. Initially these product, price, and category tables are empty. This information must be fed into the WCS database.

This information can be added to the database from the WCS Administrator screens. However, if there is a large amount of data to be added to the database, manual addition can be impractical. In this case, customers may want to load data from their current back-end system for the initial dataload phase and for ongoing changes.

Fortunately, there are a number of options available for loading data from the back-end database into the WCS database. These options need to be considered and used for the initial loading as well as for keeping the WCS data and the back-end system data fully synchronized.

7.2.4.5 Back-end business systems

The final component in the diagram in Figure 35 on page 136 is back-end business systems. This component is also referred to as back-end applications, or legacy applications or systems. Whichever name you choose to call it, this is where the richest business functions exist and the source of endless potential of benefits when you successfully “Webulate” or Web enable it.

7.2.5 Architectural view of WebSphere Commerce Suite for AS/400

Figure 36 provides the architectural view of WebSphere Commerce Suite V4.1 for AS/400 implementation. In fact, this is a detailed view of the WCS Server Process component shown in Figure 35 on page 136.

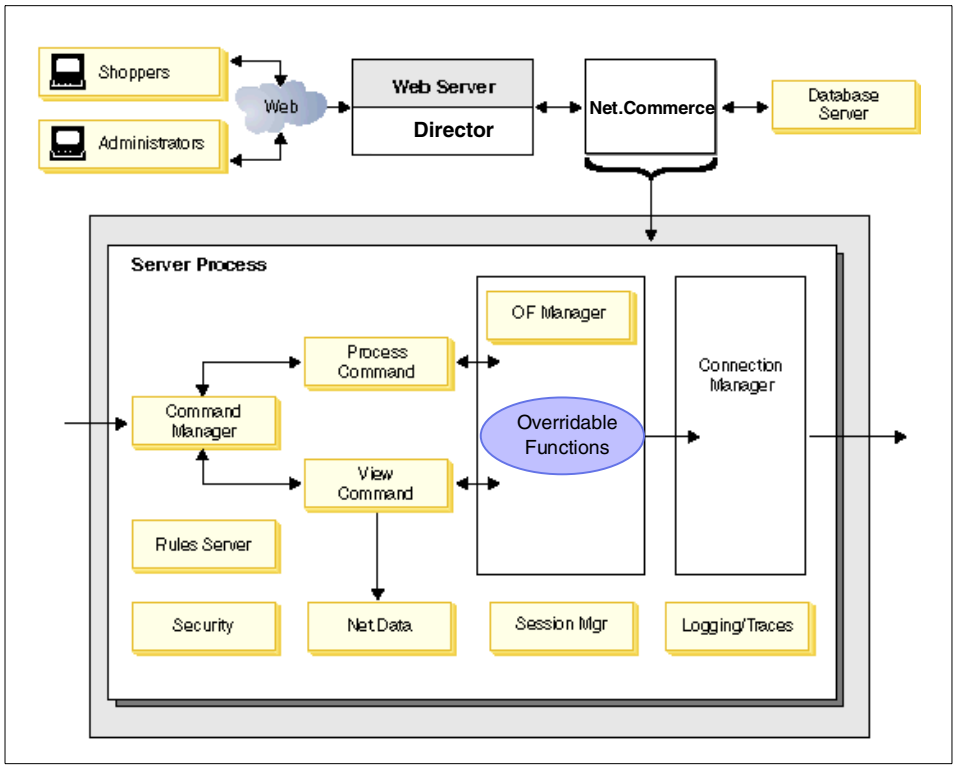


Figure 36. Architectural view of WebSphere Commerce Suite V4.1 for AS/400

Director
The director provides the same function as WebSphere Commerce Plug-In of WebSphere Commerce Suite V4.1 for AS/400. It is a component of the WCS server that parses HTTP requests from shoppers and performs a cache lookup. If the appropriate cached page is not found, it routes the requests to the WCS server engine, based on a workload balancing scheme.

7.2.5.1 WCS server process components

People use WebSphere Commerce Suite V4.1 for AS/400 from their Web browsers. This includes actual clients, such as shoppers or business partners, who do business with you over the Web and with you as the administrator of your e-business site. Web browser users interface with WCS via a graphical user interface. When they click the browser expecting the server to provide the function required, the actual request is delivered to the iSeries server in the form of commands.

The WCS server process, therefore, is a set of functions to process the commands. The WCS server consists of several components that interact to process C++ and Net.Data commands. Again, the details and commands of the implementation of this server engine are transparent to the users (that is not just your clients, but your administrators as well).

Command Manager
The Command Manager locates the required command and executes it. Commands are separated into two distinct groups: view commands and process commands. *View commands* perform viewable tasks (for example, displaying an order using the OrderDisplay command). *Process commands* perform business functions (for example, process an order using the OrderProcess command).

Overridable functions (OF)
Overridable functions are used to extend the business logic in commands. WCS is a front-end application that provides the tools to quickly set up a mall from which shoppers can browse and order. WCS uses API functions to implement the shopping process. The ideal situation is to map the entire site flow directly to an existing WCS API. However, if WCS meets only part of your requirements, you must tailor and extend the system by using the WCS commands and overridable functions.

Another potential usage of OFs is when you link your WCS applications with your back-end applications. You can do this in a number of other ways including using MQSeries. However, OFs can be a favorable option if you have a C++ programming skill set.

Connection Manager
The Connection Manager manages a pool of database connections.

Security
An important aspect to a successful e-commerce site is security. Your customers are concerned with the security of their personal information as it is transmitted across the Internet and as it is used throughout your order processing environment. In addition, you should be concerned with securing your information assets and systems. The security component performs authentication and access control tasks.

WebSphere Commerce Suite V4.1 for AS/400 provides features to help you implement your security strategy including:

- Authentication
  - Basic authentication (user ID and password)
  - X.509 authentication (the use of client certificates)
- User Registry
  - WCS database and commands for user registration
  - Lightweight Directory Access Protocol (LDAP)

Net.Data
Net.Data is an application that facilitates interaction between shoppers and the database. A Net.Data macro is a file that retrieves data from the database and displays it as a formatted Web page. It contains functions that usually execute SQL queries, HTML tags (which can also contain JavaScript code), and Net.Data statements.

WebSphere Commerce Suite V4.1 for AS/400 allows the option to use JavaServer Pages (JSP) to create Product and Category display pages. JSP technology allows you to combine scriptlets of Java code into your HTML pages. It also allows you to implement standards-based technology for your catalog display pages and create templates for catalog and product pages in the Commerce Studio environment. You can also take advantage of precoded Commerce Suite JavaBeans.

At a high level, both Net.Data macros and JSPs do the same thing. They allow you to wrap HTML with logic to create dynamic Web pages. Net.Data uses a macro language, while JSPs use Java.

Session Manager
The Session Manager is responsible for tracking user sessions.

Logging and tracing
The logging and tracing component provides diagnostic functions and error handling.

Rules server
The Blaze Advisor Rule Server processes rules and provides personalized product recommendations based on a user-defined set of rules.

7.3 IBM WebSphere Payment Manager

Payment transactions are an integral part of e-business success. As more and more merchants move online, there is a growing need to seamlessly integrate payment capabilities with existing software for an easy-to-use, end-to-end, more secure e-commerce solution.

IBM WebSphere Payment Manager (5733-PY2) enables service providers to quickly and easily link merchants to payment processors to handle today's complex e-commerce transactions. This latest e-commerce solution helps service providers grow revenue from existing merchants while attracting new ones.

Packaging

IBM WebSphere Payment Manager (5733-PY2) is a separate product that is shipped with WCS). It is part of the package when you buy WCS. You can also purchase it separately and use it as the payment manager for any other e-commerce or e-business solution on your iSeries server.

With IBM WebSphere Payment Manager, service providers can extend their businesses and ability to be profitable with new offerings that can provide a clear advantage over their competition. IBM WebSphere Payment Manager offers payment hosting capabilities for Internet Service Providers (ISPs), Commerce Service Providers (CSPs), Application Service Providers (ASPs), banks, and financial institutions.

Figure 35 on page 136 shows how the Payment Manager fits in the e-commerce architecture on the iSeries server.

The Payment Manager provides payment services on the Internet by taking credit card payments from consumers. The Payment Manager runs for a merchant and is used in conjunction with online shopping software such as WCS. It supports the SET Secure Electronic Transaction protocol developed by Visa, MasterCard, IBM, and others. In addition to the SET protocol, a

cassette (a software plug-in to an architected interface used to expand the number of payment types) for CyberCash is also available.

The Payment Manager can obtain credit card approvals and capture funds by communicating with a payment gateway, which runs at a bank (typically called an *acquirer*). In addition, it can process deposits and credits or perform reversals.

SET Secure Electronic Transaction

SET is an open-network payment-card protocol that provides greater confidentiality, greater transaction integrity, and less opportunity for fraud on each transaction point than any other existing secure payment system. The process involves a series of security checks performed using digital certificates that are issued to participating purchasers, merchants, banks, and payment brands.

There are five main parties involved in a SET transaction:

- Cardholder
- Merchant
- Issuer: The customer's financial institution, which provides the payment card to the customer and the payment to the merchant
- Acquirer: The merchant's financial institution, which enables the merchant to accept a payment card brand and issues the captured payment to the merchant
- Certificate Authority (CA): A trusted third-party that can certify the identities of the customer, the merchant, and the acquiring institution to each other

Four of these parties require their own SET software. The issuer communicates with the acquirer over a secure network or other communications channel. Therefore, it does not need a secure Internet implementation.

SET has four components:

- **Cardholder Wallet**: A component that is run by an online consumer enabling secure payment card transactions over a network. SET Cardholder Wallet components must generate SET protocol messages that can be accepted by SET Merchant, Payment Gateway, and Certificate Authority components.
- **Merchant Server (Payment Server)**: A component that is run by an online merchant to process payment card transactions and authorizations.

It communicates with the Cardholder Wallet, Payment Gateway, and Certificate Authority components.

- **Payment Gateway**: A component that is run by an acquirer or a designated third party that processes merchant authorization and payment messages (including payment instructions from cardholders) and interfaces with private financial networks.
- **Certificate Authority**: A component that is run by a certificate authority that is authorized to issue and verify digital certificates as requested by Cardholder Wallet components, Merchant Server components, or Payment Gateway components over public and private networks.

Some benefits to merchants for implementing SET are:

- Increased sales from existing online shoppers who can now more confidently expand the number of merchant sites where they shop
- Additional sales from consumers who were traditionally constrained from electronic shopping due to their concerns about security on the Internet
- Increased savings through a reduction of exception handling
- Reduced costs associated with fraud

For more information about SET, refer to the Web site at:
`http://www.setco.org`

7.4 References

This section provides various references for WebSphere Commerce Suite V4.1 for AS/400.

Online Manuals
Most of the manuals of WebSphere Commerce Suite V4.1 for AS/400 are delivered as online documentation installed on the iSeries server. If WebSphere Commerce Suite V4.1 for AS/400 is installed on your server, you can find its manuals under the iSeries integrated file system directory /QIBM/ProdData/NetCommerce/html/MRI2924/ncbooks. All the online manuals are PDF files.

Web sites
- IBM WebSphere Commerce SuiteHome Page:
  `http://www.ibm.com/software/webservers/commerce`
- AS/400 and IBM iSeries WebSphere Commerce:
  `http://www.iseries.ibm.com/ebusiness/wscommerce.htm`

- IBM WebSphere Commerce Community:
  http://www.ibm.com/software/websphere/commerce/community
- IBM WebSphere Application Server:
  http://www.iseries.ibm.com/products/websphere
- PartnerWorld for Developers, iSeries:
  http://www.iseries.ibm.com/developer
- IBM Payment Server Home Page:
  http://www.ibm.com/software/webservers/commerce/payment
- IBM Redbooks: http://www.redbooks.ibm.com
- IBM AS/400 and iSeries reference information:
  http://www.ibm.com/servers/eserver/iseries
- IBM Information Center: http://www.ibm.com/iseries/infocenter
- Service offerings at IBM Support: http://www.ibm.com/servers/support

Publications

- *WebSphere Commerce Suite Fundamentals, Version 4.1*, GC09-2994
- *Demo CD-ROM* (based on Net.Commerce V3), GK3T-2318
- *Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice*, SG24-4978
- *Payment Server V1.2 for AS/400: Secure Transactions in e-commerce*, SG24-5199
- *Building AS/400 Client/Server Appls with Java*, SG24-2152
- *Building AS/400 Applications with Java*, SG24-2163
- *Building AS/400 Appls with WebSphere Standard Edition*, SG24-5635
- *Net.Commerce V3.2 for AS/400: A Case Study for Doing Business in the New Millennium*, SG24-5198
- *Building e-commerce Solutions with Net.Commerce: A Project Guidebook*, SG24-5417

Chapter 8. Lotus Domino for AS/400

This chapter provides an overview of Lotus Domino and related e-business and e-commerce products. It will help you to understand the products that are required and to know why the iSeries is a great platform on which to deploy Domino applications. It offers concrete examples of how Domino can be deployed in the enterprise. After reading this chapter, you should have a clear understanding of what Domino can do in terms of e-business application development.

8.1 Lotus Domino overview

Domino for AS/400 was ported from the UNIX version of the Domino product. The native Domino version became available in February 1998, followed by Release 5.0 for AS/400e customers in 1999. The iSeries server includes Dedicated Server for Domino (DSD). These are specially tuned iSeries Models 270 and 820 to give optimal performance to the Domino workloads. They are easily identified by the yellow badging. The DSD is often called the "Bumble Bee". For more information about DSD, see:
`http://www.iseries.ibm.com/domino/dsd.htm`

There has been a great deal of writing and many case studies on developing workflow, collaboration, e-mail, and database applications. This is where Domino's strengths lie, as a collaboration and workflow solution. However, that is not all that Domino can build. It can also build Internet-related applications, which are explained in this section.

As a server, the iSeries server provides reliability and scalability to enhance existing business applications. It can also reach new levels of collaboration and coordination, which was previously not possible with other Domino servers.

Domino base functions
Domino is an entire infrastructure and framework with which you can build new applications or extend existing iSeries applications to Notes or Web-based clients. It includes an application server, Web server, tools, processes, security, and other base functions on which, and with which, you can build applications.

Tools within the Domino environment to enable Web technology include:

- **Dynamic and transactional Web site**: Development tools, business data, applications, middleware, and architecture to design and build dynamic Web sites
- **Workflow**: Managing the movement of documents and information through the organization
- **Collaboration**: Helping people work together by managing information, data, schedules, documents, communication, and business processes
- **Database**: Store and manage non-traditional, data-like graphics, pictures, scanned images, scanned signatures, documents, and relational data
- **E-mail**: Mail serving and management
- **Application development**: Tools, environment, security, features, and functions to develop a wide variety of applications

These base functions can be built into a wide range of applications. The functions are provided and managed by the Domino server and services. Within the broader Domino environment, there are features (such as messaging and replications services) and tools (like Domino Designer and Java) on which, and with which, you can build applications.

Domino server and services

The core of Domino is a document database server used to run various applications. The Document database server is the base of Domino, complemented by the following services:

- **Database replication services**: Allow several replicas (full or partial related copies of a database) over different Domino servers or Notes clients. These services synchronize the replicas in a scheduled or an on-demand basis.
- **Messaging services**: Allow the sending and receiving of mail. Many protocols are supported, including:
  - Notes Mail
  - Post Office Protocol 3 (POP3)
  - Internet Message Access Protocol (IMAP)
  - Lightweight Directory Access Protocol (LDAP)
  - Simple Mail Transfer Protocol (SMTP)
  - Multipurpose Internet Mail Extensions (MIME)
- **Search services**: Provide the capabilities of a full text search engine to search documents in databases, attachments, and files across an entire domain.

- **Security services**: Include:
  - Multiple user and object access levels
  - User groups
  - Digital signatures
  - Public key encryption
  - Object integrity checking
- **HTTP services**: Support Hypertext Transfer Protocol (HTTP), Hypertext Markup Language (HTML), and Network News Transfer Protocol (NNTP). Domino Internet security includes support for the Secure Sockets Layer (SSL) protocol and X.509 certificates. Use your existing Internet infrastructure with Domino and be assured that your system works with other systems based on open protocols.

Domino is available on many other platforms, including IBM OS/390, IBM AIX, Linux and other UNIX operating systems, as well as Microsoft Windows NT.

Lotus Domino for AS/400 is the native implementation of the Lotus Domino server on the iSeries platform and is also one of the best performing servers.

New Lotus Domino for AS/400 record

The iSeries server Model 840 powered by Silicon-on-Insulator (SOI) and copper wiring technologies has set a new world record for Lotus Domino scalability and performance. It successfully handles 75,000 concurrent NotesBench R5 Mail Users with an average response time of 276 milliseconds.

Improvements have continued to support even more throughput.

8.1.1 What a Domino application is

In generic terms, a Domino application is a set of actions within a piece of software that the designer considers a unit of work. The application is made of processes, data, business rules, and so on. Lotus developers and users refer to a set of related Domino databases (.nsf files) as an application. It can be one or more databases.

These databases contain forms, pages, views, framesets, agents, scripts, applets, HTML pages, and so on. Applications can import, update, or access data from host systems, including DB2 UDB for AS/400 and native Domino databases.

Note

Additional information about Lotus Domino is available at:
`http://www.lotus.com/domino`

Information about Lotus Domino for AS/400 is available at:
`http://www.iseries.ibm.com/domino`

Developer information is located under “All About Notes & Domino” link at:
`http://www.notes.net`

8.1.2 Domino for AS/400 functions

Domino for AS/400 combines the iSeries value proposition of integration, ease of use, reliability, and scalability with the world's leading groupware offering. Domino provides the application development infrastructure. Combined with AS/400 reliability, the iSeries is a great server on which to run Domino applications. Features include:

- **Integrated messaging**: The messaging infrastructure in Domino is robust and is the *foundation* for applications, not the end result.
- **Business architecture**: The design of Domino is specific to business process automation and supports many kinds of data and human interactions.
- **Clustering support**: Clustering offers high availability switchover on failure. Cluster servers redirect database open requests to other servers in the cluster to provide users with uninterrupted access to important databases. This means greater availability on the iSeries server because the it manages automatic restarts. The Internet Cluster Manager (ICM), available from Release 5.0, lets you use Domino clusters to provide failover and workload balancing to HTTP clients (Internet browsers) when they access Domino Web servers.
- **Rapid application development environment**: The environment is visual (point and click), rapid (wizards and forms), client/server, Web-enabled, Java-enabled, and event driven.
- **Suite of connectors to enterprise applications**: Lotus has developed many connectors to access data from many systems and databases. This makes Domino a good integration or consolidation point and environment to build new, high-impact, business process-oriented applications.

- **Versatile database**: Structured and unstructured data, such as plain text, rich text, scanned images, video, and audio, can be combined and are the foundation for building high-impact applications.
- **Distributed environment**: Allows documents and applications to be shared through an intuitive interface (Notes Client or Web browser) across LANs, WANs, and the Internet.
- **Security**: Domino supports leading security and has access controls (user authentication), digital signatures (documents and mail), encryption (communications such as SSL and mail). Execution Control Lists (ECLs) prevent unauthorized Domino applications from causing damage.

Lotus Domino provides rapid application development tools, including LotusScript, to help you design and develop collaborative applications for your users that can extend to the Internet. LotusScript allows you to easily integrate your existing DB2 UDB for AS/400 data into these applications. For example, a customer service application could access an existing DB2 UDB for AS/400 customer master file for such information as name, address, and phone number. DB2 UDB for AS/400 integration works both ways. Your AS/400 accounts receivable application might update your Domino customer service database to alert your customer service representatives to any outstanding accounts receivable issues.

With Domino for AS/400, you can tackle many business problems that require organizing unstructured data or managing the flow of information. At the same time, you can link this workflow and unstructured information to the wealth of business data in your DB2 UDB for AS/400 databases.

The application integration options available to you depend on what you want to achieve. For example, a Notes client or a Domino server may need to access DB2 UDB for AS/400 data, or an iSeries application may need to access a Domino database.

Domino makes developing applications easy to automate business processes. Organizations with heterogeneous platforms and operating systems can benefit from the seamless cross-platform design of application databases created with the Domino Designer for Notes client. With the Domino Designer, you create applications from a graphical user interface (GUI) with little knowledge of programming or scripting languages.

Pre-defined templates (for example, xxx.ntf files) make setting up and configuring applications fast and easy. Experienced application developers can create more complex workflow, GUI-based client/server, or network-centric applications.

8.1.3 Why Domino on the iSeries server

As a server, the iSeries server provides reliability and scalability, which are essential in the dynamic and often unpredictable world of e-commerce. Domino for AS/400 makes it easy to integrate the business data in DB2 UDB for AS/400 databases with the Web-enabled applications you build with Domino. This is based on these factors:

- **Reliability:** The iSeries server boasts the most robust reliability in the market, with an incredible record of 99.97% uptime. If a specific application should fail, the iSeries server's unique architecture, which logically insulates applications from one another, assures uninterrupted performance for other applications.

  Each Domino Partition runs in its own subsystem, which enables Domino to benefit from this feature.

- **Availability:** Self-diagnostics, remote administration, and the debugging capability of the iSeries server ensure maximum availability. The iSeries integrated file system architecture allows for centralized backup and recovery for iSeries applications and data, including Domino. With the Domino Enterprise Server, availability is further enhanced by the ability to run multiple partitioned servers on one physical system and clustering to provide server failover support and dynamic load balancing.

  An exclusive feature of Domino for AS/400 is the automatic restart of the Domino server. IBM has built a watchdog job that starts automatically when you start a Domino server and it monitors the Domino server. In the event of a failure, it automatically brings down the failing Domino server in a controlled manner and restarts it without affecting any other Domino partitions or other applications running on the system.

- **Scalability:** Unsurpassed single footprint scalability is available with Domino for AS/400. With one administrator and one backup to perform, administration costs are minimized as user capacity is maximized.

- **Server consolidation:** Scalable Domino platforms reduce the number of servers required to support an entire enterprise. The fewer the number of servers you have, the lower the complexity and cost of administration and management for a groupware solution are.

  With Domino on the iSeries platform, you can consolidate multiple Domino servers into one box. Servers are all "under one roof", so administration costs are a fraction of what is required for server farms.

  In addition, with all the Domino servers on the same iSeries server, you don't need to replicate your relational databases and applications to other servers to make them accessible. All the Domino servers have access to

the same relational data and traditional applications. In addition, when you run multiple partitions on one physical box, the partitioned servers communicate over an internal network, which does not add any network traffic to your LAN.

- **Integration with the IBM DB2 UDB for AS/400 database:** Domino for AS/400 applications take advantage of direct real-time access to the DB2 UDB for AS/400 database. Users embed the results of an iSeries query into a Notes document without the need for programming. Domino for AS/400 allows access to DB2 UDB for AS/400 from Domino application development tools and APIs. This access is direct (that is, it is in microcode). This enables fast, reliable, secure, and scalable access, without the need for an ODBC driver or a middleware communications layer, when the application code runs on the server rather than the client.

  The Lotus Enterprise Integrator product (a separately purchased product, formerly known as Lotus NotesPump) provides for data movement between DB2 UDB for AS/400 and Domino, and direct real-time access to DB2 UDB for AS/400 data from Domino applications with no programming required. The iSeries implementation adds unique capabilities to Lotus Enterprise Integrator. Using a feature of LEI, iSeries access rights can be mapped to Domino access rights for unprecedented security implementation. You can enable replication of access rights between a DB2 table and a Domino database.

  An exclusive feature of the iSeries implementation of Lotus Enterprise Integrator is the propagation of deleted records, so that any operations you perform (update, insert, or delete) can be synchronized between your Domino databases and DB2 UDB for AS/400 tables.

  For more details on positioning the various DB2 UDB for AS/400 access alternatives, refer to the redbook *Lotus Domino for AS/400: Integration with Enterprise Applications*, SG24-5345.

- **Total cost of ownership:** The primary benefits of installing Domino for AS/400 include data integration and centralized management. iSeries with Domino offers customers a single hardware platform to manage both line of business applications and groupware applications to coexist and extend to Notes and browser clients through Domino without adding multiple servers or middleware. Customers with a Domino for AS/400 implementation do not need to invest in an additional hardware platform, operating system software, and the skills needed to maintain this environment.

8.2 Technology and architecture: Domino for AS/400 solutions

Domino is heavily used in places that have a direct impact on such areas as customer support, lead and customer management, workflow, publishing and distribution of information, competitive analysis and positioning, product improvement processes and development, communications, electronic commerce, and customer service.

The focus of this redbook is specific to e-business applications that can be built with, and on top of, Domino. As reviewed in Chapter 2, “Building e-business sites: Phased approach” on page 49, customers typically go through three distinct phases when building e-business solutions. Domino can provide functionality to build and support all three types: Web presence, dynamic site, and transactional site.

The following sections define and show representations of the implementation of these phases with Domino.

8.2.1 Web presence definition

As outlined earlier, Web sites serve static Web pages. The content of what the visitor sees is not changed by user interaction. In simple terms, the role of the HTTP server is to receive requests from browsers for Web pages, locate the pages, and send them to the requesters. The browser communicates with the HTTP server using URLs that contain the location of the pages that the user wants. The Domino HTTP server provides the static content.

8.2.2 Dynamic site definition and architecture

A dynamic site, by definition, has read-only access to data on back-office business systems through a Web browser. The back-office systems we refer to and access are iSeries-based applications (ERP, home-grown, legacy, and others) and Domino applications (.nsf database). An example is a customer accessing the order status of an item purchased or a supplier checking the inventory level at a customer warehouse.

Figure 37 represents an overview of the components of a Domino dynamic site implementation. It shows the flow of information through the databases, server, and network to the user. As defined earlier, dynamic sites allow access to Domino applications through a Web browser through the Internet, intranet, or extranet. The client user has read-only access through the browser to back-office and Domino databases and the Domino application server. Typically these applications move documents, automate processes, provide information, and respond to queries.

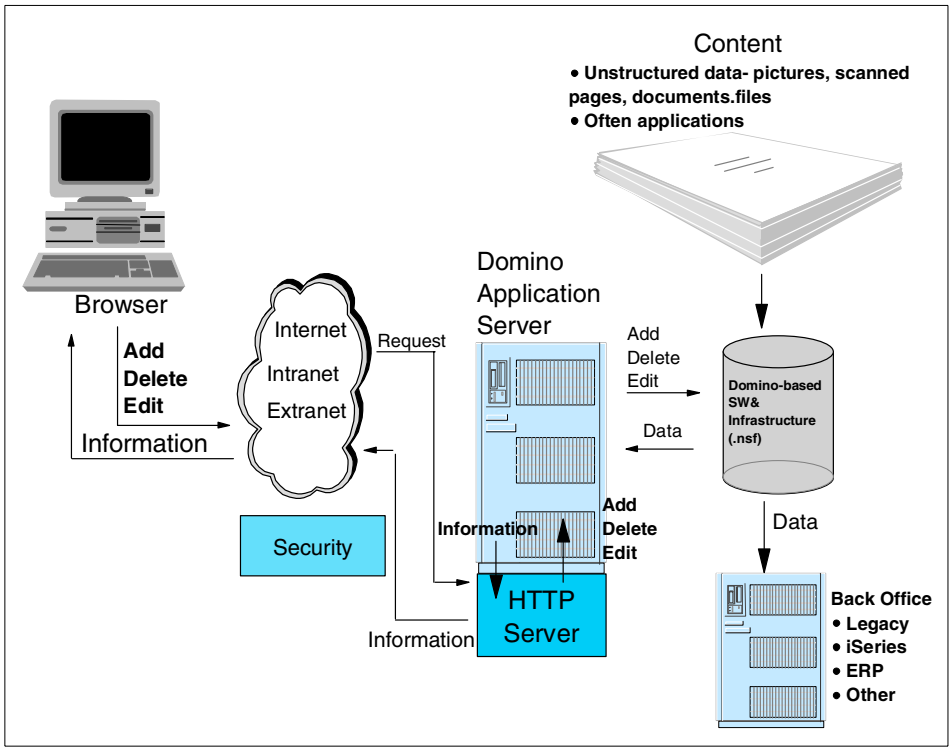


Figure 37. Domino dynamic site architecture

Domino can be used to build the following types of dynamic site solutions:

- Enhance and extend existing legacy applications and provide workflow and collaboration functions not easily added with legacy development methods and tools, without having to replace those core systems. At the same time, you can link this work flow and unstructured information to the wealth of business data in your DB2 UDB for AS/400 databases to enhance the value of what you've already built.
- Consolidate existing system interfaces that are not currently connected. And if they are connected, provide significant benefits.
- Automate business processes (for example, communications and e-mail) and discuss databases, movement of documents (workflow, scheduling, publication of information), distribution of documents, and access to information more accurately and with better timing.
- Rapid application development tools, templates, wizards, and data access tools allow quick development.

- Integrate existing DB2 UDB for AS/400 data into new Domino applications. Your Domino application can access your existing DB2 UDB for AS/400 customer master file for such information as a customer's name, address, and phone number.
- Security for which Domino supports leading security and has access controls (user authentication), digital signatures (documents and mail), encryption (communications: SSL and mail), and protection from active content (ActiveX, JavaScript, executables, self-extracting .zip).

Domino provides this functionality with its base server functions, host integration, development tools, and messaging infrastructure. This list begins to outline the broad scope and impact these applications can have for your business.

8.2.3 Transactional site definition and architecture

A transactional site has add, delete, and edit access to data on back-office business systems through a Web browser. The back-office systems we refer to and access are iSeries-based applications (ERP, home-grown, legacy, and so on) and Domino applications (.nsf database). Some examples are a customer accessing and updating their personal information, a distributor placing an order for an out-of-stock item with their supplier, or a customer placing an online order. The transactional site architecture is similar to the diagram shown in Figure 37 on page 155.

Transactional sites can be divided into three main categories for analysis purposes:

- **Business-to-business**: This type of transactional site provides external supply chain (partners, vendors) add, delete, and edit access to company databases. Some examples include ordering products, filing sales forecasts, and accessing and updating account data. Applications on the host system process the transaction, update data, and trigger other events depending on the application design. This type of solution typically has some form of user authentication (login and ID), a forms-based information request or query function that is user defined, and a forms-based data entry function for the transaction detail.
- **Business-to-consumer**: This type of transactional site provides consumers with add, delete, and edit access to company databases. This is different than a dynamic site where users can view only. Transactional sites have add, delete, and edit functions. Customers can update their personal information, be added to databases, complete information forms, application forms, and so on. This information is then processed. The corresponding applications on the host manage the transaction. As with

business-to-business solutions, this typically has some form of user authentication (login and ID), a forms-based information request or query function which is user-defined, and a forms-based data entry function for transaction detail.

- **E-commerce (e-store):** The e-commerce implementation involves actually transacting commerce and purchasing products or services. business-to-business (B2B) and business-to-customer (B2C) sites provide data and information and allow add, delete, and edit functions, but there is no shopping. E-stores, as they are commonly referred to, can have shopping carts, shipping functions, product catalogues, order tracking, some form of back-office integration, order processing, and accept credit payments, among other possible functions.

  Figure 38 graphically presents the steps that customers follow in the shopping process.

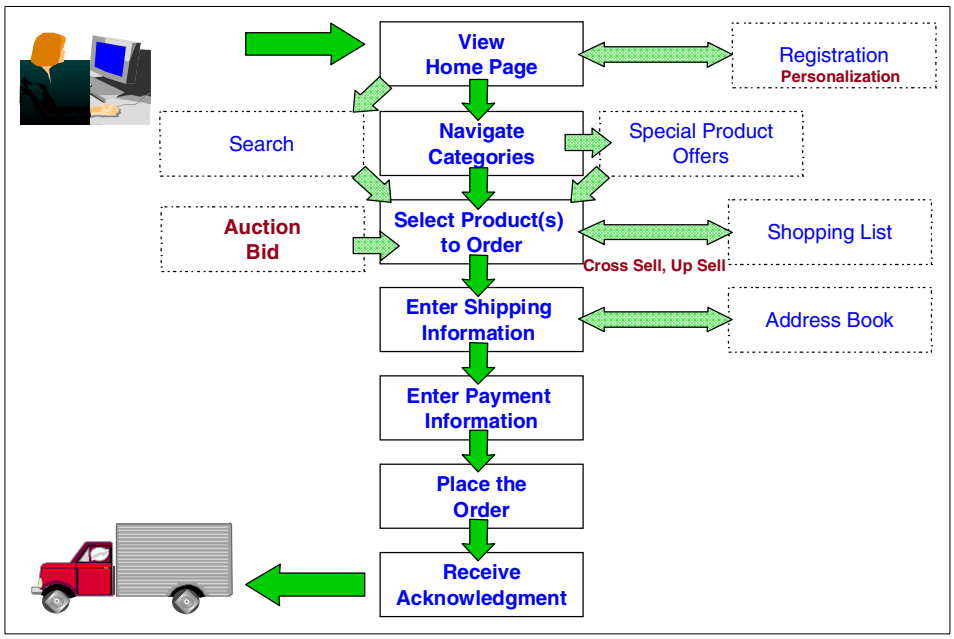


Figure 38. Electronic store shopping model (generic)

To enable this process, such functions as shopping cart, shipping, product catalog, order tracking, user registration, taxation, back-office integration, order processing, credit cards, and payment must be integrated into the application. These are not standard features of Domino but can be built using the tools provided with Domino.

Refer to Chapter 12, “B2B: Application solutions” on page 261, for information about how to find e-store solutions offered by IBM and IBM Business Partners. Also, refer to Chapter 7, “WebSphere Commerce Suite for AS/400” on page 129, for a product that already has the shopping metaphore built as is shown in Figure 38 on page 157.

8.3 AS/400 Domino products: Web presence, dynamic, transactional

This section describes the AS/400 Domino products available to provide Web presence, dynamic data, and transactional solutions.

8.3.1 Domino HTTP Server

Domino for AS/400 is a combination of several server types. Among these are a Notes, POP3, and IMAP4 mail server, a database server, database replication server, and others, such as an LDAP server.

It also incorporates its own HTTP server, which may be used together or as an alternative for the HTTP Server for iSeries. The main strength of Domino HTTP Server is its unique ability to dynamically translate Notes documents into HTML, allowing Domino databases to be seen as HTML documents on the Web.

The Domino HTTP Server supports the following features:

- **Notes to HTML conversion**: HTML Web pages served by Domino automatically convert valid URLs into links. Views can be sorted from a Web browser by clicking the view’s column title. Static HTML files can be published by using a Domino server.

  In HTML code, hotspot links are translated into anchor (<A>) tags.
- **Passthru HTML**: Passthru HTML allows you to include HTML markups without Domino HTTP translating the codes. HTML code is included in a form, document, or About and Using documents that Domino interprets during the page translation. Passthru HTML lets you use Web-only text formatting, links, images, commands, and programs. Using passthru HTML, you can combine Domino features with HTML code.
- **Current Notes Client support**: Domino HTTP Server supports Domino 4.6 and R5 clients.
- **Security**: Database ACLs can control access by Web browsers, as well as traditional Notes clients. Domino supports name and password authentication, and Domino HTTP Server supports SSL Version 3.0
- **HTTP**: Domino HTTP Server supports HTTP 1.1.

- **NNTP**: Domino HTTP Server supports Native Internet News Protocol (NNTP).
- **Java**: Domino HTTP Server provides support for Java servlets and applets.
- **Virtual servers**: Support in Domino HTTP Server is included for virtual servers, which allows more than one Web site with a separate DNS name to exist on a single server partition. With R5, you can run multiple virtual servers in a single partition.
- **WebSphere support**: The WebSphere Plug-in for Domino is available for WebSphere Application Server Standard Edition V3 and WebSphere Application Server Advanced Edition V3. With the WebSphere Plug-in for Domino, it is possible to define the WebSphere Application Server as the servlet and JSP engine in the Domino HTTP server configuration. Additional integration functions are considered.

Note: Domino R5.0 contained the Domino plug-in for HTTP server for AS/400, to allow Domino databases (*.NSF) to be served through the IBM HTTP Server for AS/400 instead of the Domino HTTP Server. That is, it allows you to select built-ins for HTTP or HTTP Server for iSeries so both won't need to be configured.

8.3.1.1 Coexistence of the Domino and iSeries HTTP servers

The Domino HTTP Server and HTTP Server for iSeries each use port 80 by default. Depending on whether each server is attempting to bind to the same TCP/IP Interface, there might be a port conflict. This can be resolved two ways:

- Only run one of the HTTP servers. This is typically done by disabling autostart on either the HTTP Server for iSeries or configuring Domino *not* to run the HTTP server.
- Configure the HTTP servers to use different ports or different TCP/IP addresses.

Note: If multiple Domino partitions are installed, each partition running HTTP must use unique TCP/IP ports or addresses.

The IBM HTTP Server runs in the QHTTPSVR subsystem. The Domino Web servers run in their designated Domino subsystem.

8.3.1.2 Using Domino HTTP Server versus HTTP Server for iSeries

Domino HTTP Server is *functionally equivalent to* V4R3 of HTTP Server for AS/400. Both products are based on the same base code. Subsequent versions were developed independently.

Generally, HTTP Server for iSeries is better tuned for static HTML and offers better functionality in its V4R5 release. It should be used as the preferred Web server. However, if the customer's Web site is mostly based or coded using Domino, or if NNTP is used, choose the Domino HTTP Server.

8.3.2 Application server: Domino for AS/400

Lotus Domino for AS/400 requires an AS/400e server based on PowerPC (RISC) technology. Although it also runs on all RISC-based AS/400e models with available capacity to support it, there is a price and performance advantage in running Domino on the newest technology server models (iSeries) that are optimized for client/server applications.

In addition, keep in mind that the workload and function mix must be considered relative to the hardware specifications. For example, applications with highly CPU-intensive functions (such as database lookups and full text indexing) should not be expected to perform well on systems with low processor speed, low CPW rating, and no cache. The same is true for an Intel-based system. That same system may fit nicely for a simple mail workload, supporting up to the number of users rated for it in the sizing guidelines.

AS/400e server recommendations

The AS/400e server should meet the following minimum requirements:

- 256 MB base memory for application serving, 128 MB minimum for mail server.
- 0.5 to 1.0 MB memory per active mail user.
- 490 MB DASD for Domino executables and data (such as data directories and help files).
- 50 MB DASD per enrolled and active mail user. This estimate is a planning guideline. The amount of space needed varies, depending on how much information each user stores and the Domino applications that are installed.
- At a minimum, two DASD arms are recommended for best performance.

For more detailed planning you can start at the IBM Workload Estimator for iSeries found at: `http://www.as400service.ibm.com/servlet/EstimatorServlet`

Administration workstation

The workstation you use to administer a Domino server on the AS/400e server requires the following hardware:

- A PC with an Intel Pentium processor
- 48 MB of RAM minimum
- 70 MB minimum disk space; 112 MB recommended (above operating system requirements)
- A color display supported by the operating system
- A mouse
- A printer (optional)

Software requirements

The general AS/400e server software requirements are:

- Operating System/400 (OS/400) V4R2 or later, 5769-SS1 (V4R3 is the minimum requirement for supporting servlets)
- Pre-requisite PTFs listed under Support, QMUs, QMRs, and PTFs on the Web at: `http://www.iseries.ibm.com/domino`
- TCP/IP Connectivity Utilities for AS/400, 5769-TC1 (no additional charge; comes with OS/400, but needs to be explicitly installed)
- If you plan to use Java with Domino, you need to install these options:
  - OS/400 Qshell Interpreter option, 5769-SS1 (option 30)
  - AS/400 Developer Kit for Java, 5769-JV1

If you plan to use Operations Navigator for Domino, you need:

- OS/400 Host Servers option, 5769-SS1 (option 12)
- Client Access for Windows 95/NT, 5763-XD1 (no additional charge; comes with OS/400, but needs to be explicitly installed)

  Client Access for Windows 95/NT must be installed on the AS/400e server to receive fixes. The Windows 95/NT client may be installed directly on the PC from the PC CD-ROM. Using Operations Navigator does not require a Client Access license, and you do not need to install 5769-XW1 on your AS/400e server.

If you plan to develop applications with Notes C or HiTest APIs (C, C++, or HiTest), either of the following options is required:

- To compile on the AS/400e server, use the AS/400 ILE C Compiler (5769-CX2).
- To compile on a client workstation, use VisualAge C++ for OS/400, 5716-CX4, with built-in OS/2 client support, or 5716-CX5, and VisualAge C++ for Windows Version 3.5 for Windows 95 or NT support (for GUI development tool).

The latter is a PC product 5622-880, Part# 33H4980. The prerequisites are the same as for Operations Navigator that were previously outlined.

In the case of Notes C++ APIs, only the client compiler option is available.

The following software is required for a workstation on which you use Domino administrator functions to manage Domino for AS/400:

- An operating system supported by the Lotus Notes Release 5.0 client (5.0 runs under Windows 95, Windows 98, Windows NT 4.0, Mac PowerPC 7.6 and 8.5) or by the Lotus Notes Release 4.6 client (4.6 runs under AIX 4.1.5 or 4.2.1, HP-UX 10.20, Sun Solaris Intel Edition 2.5.1, Sun SPARC Solaris 2.5.1, Windows 95, Windows NT 4.0; 4.1 and 4.5 also run under Windows 3.1, OS/2, and Macintosh)
- TCP/IP installed and configured
- Lotus Notes R5 client

The following software is required for a workstation on which you use Operations Navigator to manage Domino for AS/400:

- A Windows 32-bit operating system (such as Windows 95 or NT)
- IBM Client Access support for Windows 95 or NT
- TCP/IP configured and used to connect to the AS/400e server
- Lotus Notes R5 client
- For Notes 4.6, include Notes Designer

Refer to Appendix A, “Sizing and performance” on page 273, for further Domino sizing information.

Networking requirements

Domino for AS/400 runs with the TCP/IP network protocol. TCP/IP is delivered with OS/400 but must be configured. TCP/IP knowledge on the iSeries platform is a necessary prerequisite for successful implementation of Domino for AS/400.

Other miscellaneous considerations

An add-in called *DB2 for AS/400 Notes Import Program* (also called the *DB2 Client Import Library*), which enables the iSeries query embed using the Notes client, is available at no charge for optional use with Domino for AS/400 from Notes client machines running Windows 3.1, Windows 95, Windows NT, or OS/2. It requires a client ODBC connection to the AS/400e server using TCP/IP, such as the ODBC driver which ships with Client Access/400. Keep in mind the following points:

- For Windows 95 clients, the pre-requisites are the same as for Operations Navigator, mentioned earlier, and no fee-based Client Access license is required.
- For Windows 3.1 clients, a 5769-XW1 license is required for each concurrently active user of this type.
- For OS/2 clients, a 5769-XY1 license is required for each concurrently active user of this type.

More information about the DB2 Client Import Library and a link to download it is available at `http://www.iseries.ibm.com/domino` under Related Products. For more information about Client Access/400, go to the site at:
`http://www.iseries.ibm.com/clientaccess`

If the optional product Lotus Enterprise Integrator is used, it needs to be installed on the same AS/400e server as Domino for AS/400. The journaling setting in LEI (or DECS) must match whether the file is journaled.

Directory synchronization requires the installation of Domino software AS/400 Integration (option 1) and HiTest C API (option 6).

The ability to edit the Notes.ini file from a 5250 screen requires PTF SF45296 of OS/400 V4R2 or PTF SF49052 for OS/400 V4R3. V4R4 and V4R5 include the EDTF function.

8.3.2.1 Domino for AS/400 licensing

Domino can be ordered in two ways:

- Through Lotus as written
- Through IBM as a Licensed Program Product (LPP)

As a Lotus product, it is ordered through Lotus authorized software resellers. Domino for AS/400 is packaged on a yellow Lotus CD-ROM. A Domino for AS/400 license includes the core Domino server, the SMTP MTA, and many facilities to support integration with iSeries services and data.

Domino Advanced Services, which provides clustering, partitioning, and charge-back billing capabilities, is provided with the base code and appears on the installation menu. Optional companion products, such as Lotus Enterprise Integrator, are ordered and packaged separately.

Lotus software maintenance provides upgrades for the purchased licenses at no additional cost within a one- or two-year period. Otherwise, upgrades are purchased as though they are new product acquisitions.

It is a great savings to purchase maintenance at the same time as initial licenses. The maintenance (upgrade strategy) for Lotus software is handled differently from the IBM strategy for most iSeries products. By understanding this, you will avoid confusion and maintain customer satisfaction. We recommend that customers obtain detailed information from a Lotus authorized software reseller.

For volume licensing (as compared to an individual retail sale), Lotus uses a contract called a *Passport agreement*. The Passport agreement authorizes customers to use Lotus software products purchased from resellers, typically at a volume discount. It also enables customers to purchase maintenance so they can have no-charge software upgrades for one or two years upon installation. This includes maintenance releases of code, which occur approximately once per quarter, to provide fixes to reported defects and provide functional enhancements.

A typical example of a Passport contract for a small- or medium-sized business is a *Volume Purchase Agreement (VPA)*. Additional contract types are available for larger volume software acquisitions. A Volume Purchase Agreement uses a point structure, in which each product is assigned certain point values. Additional points are accrued for purchase of maintenance and support.

Note

Additional information on Domino for AS/400 software licensing, hardware, and software requirements is found at: `http://www.iseries.ibm.com/domino/`

8.3.3 Legacy system integration

Domino on all platforms offers a set of application programming interfaces (APIs) to allow access to Domino databases from external applications. The iSeries server supports C and C++. Domino for AS/400 also supports Java application development. iSeries applications interface with Domino applications by calling programs written using these APIs.

Data Integration for Domino applications is achieved through connectors and Domino, specifically ODBC, Lotus Enterprise Integration (Lotus Enterprise Integrator, formerly known as Lotus NotesPump), and Domino Enterprise Connection Services (DECS). Each of these services uses Lotus Domino Connectors to access a variety of data sources. Depending on the functionality, performance, application and so forth, the data integration option selected varies.

8.3.3.1 Lotus Domino Connectors

Lotus Domino Connectors allow Domino server applications to connect, authenticate, and translate data between Domino and external RDMS, ERP, transaction systems, and text/file source data. This allows applications to integrate and have interactive access to enterprise source data. The Lotus Domino Connectors work in tandem with Lotus Enterprise Integrator (formerly NotesPump) and Domino Enterprise Connection Services. The architecture is represented in Figure 39.

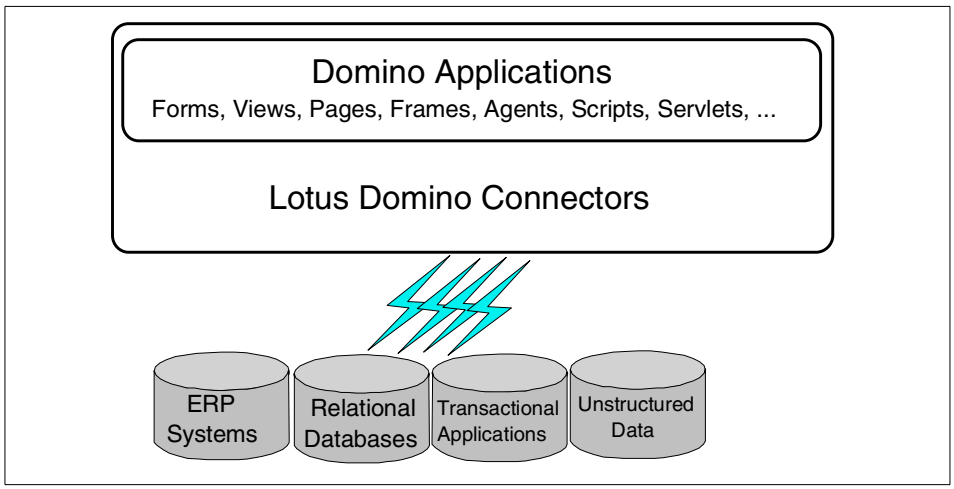


Figure 39. Lotus Domino Connector architecture

Connectors manage connection, login authentication, and data translation across enterprises data sources. They are developed using vendor-supplied APIs (DB2, Oracle, Sybase, MS SQL Server, ODBC, EDA/SQL, MQSeries, ERP applications, and others) and are integrated with the Lotus Connector API.

8.3.3.2 Domino Enterprise Connection Services (DECS)

DECS is the name for a Lotus developed technology. It is a feature of Domino, beginning with Release 4.6.3, to supply an easy-to-use, forms-based interface to achieve live, integrated connectivity to external data from Domino applications in real time. With DECS, you can view, create, update, or delete external, back-end data, directly and transparently through a Notes client or Web client. Using DECS, developers work with a template-based application that enables them to integrate external source data with any Domino application. This represents a significant advance in Domino application development capabilities.

DECS provides seamless real time integration between Domino applications and data in enterprise systems. This allows developers to store application data in either the Domino object store or another system such as a relational database.

Domino application developers enhance existing Domino applications by identifying named forms and individual fields to interact with data supplied from external source servers. Developers simply identify a Domino form field as the key field within a DECS real time connection definition form. They also identify other form fields to hold query result information, supplied through real time access to an external server source, such as a DB2 UDB for AS/400 table.

Users with either a Notes client or Web browser gain immediate benefits by accessing the Domino form and entering specific key values for query to the external data source. This action causes the Domino Extension Manager to recognize this user initiated event and transfer the key values to the external source server to perform a query, (or update) on behalf of the end user, using the key value supplied. The query result is sent back to the user in the Domino application, in real time.

8.3.3.3 Lotus Enterprise Integrator (LEI)

LEI provides a server-based, multi-tasking environment to define and administer high-volume data exchange, synchronization, event-driven, and real-time transfer of data across Domino Connector sources. This service provides a declarative interface to control data exchange between Domino and the various data sources in an organization, for example scheduling, remote management e-mail notification and so on.

Note: For those of you who are familiar with earlier version of LEI, the LEI Connector is the enhanced equivalent to the NotesPump Link.

Lotus Enterprise Integrator features

On the iSeries server, the LEI server is a 64-bit application, which executes as an ADDIN server task to a native 64-bit Domino server on the iSeries server. Access to DB2 databases, Domino databases, and file system objects is allowed using the LEI server running on the iSeries server. Direct (native API) connections to databases such as Oracle and Sybase are not supported on the iSeries server, nor is the ODBC Connection allowed. Access to non-DB2 data sources is achieved by using the IBM DataJoiner product on a Windows NT Integrated xSeries Server (IXS), outboard Windows NT, AIX, or HP_UX platform to reroute DRDA requests from the iSeries server to the target data source. Databases that support DRDA or an AS/400 Requester/Driver can be accessed without using DataJoiner.

Note

Additional information about Lotus Enterprise Integrator for AS/400 is found in the *Lotus Integrator 3.0 User Guide.* You can find this document on the Web by searching for LEI on the site at: `http://www.lotus.com/home.nsf/`

Domino applications written with LotusScript have the ability to call external applications using SQL statements. SQL can call a stored procedure that is an iSeries RPG program, for example. Another technique is for the programs to communicate using a data queue. For example, you can write information to a DB2 UDB for AS/400 table and have a never-ending RPG program read or update it (as a way of communicating) to process requests.

8.3.3.4 LEI/DECS purpose

LEI/DECS is a tool typically used to synchronize between Domino databases and relational data sources, such as DB2, Oracle, and so on. LEI/DECS can be used with or without clustering Domino servers. However, since the iSeries server only supports LEI/DECS connections to DB2 (locally or via DRDA), an alternate platform can be used to synchronize between Domino for AS/400 databases and, for example, Oracle. In that scenario, you might have a Domino for Windows NT server synchronizing between Oracle and a Domino database using LEI. Then, you might replicate those synchronizations onto a Domino for AS/400 server through Domino clustering replication.

Clustering and LEI/DECS are administered using separate tools (clustering via Domino Administrator and LEI/DECS via their respective admin databases. Both of these are “centrally managed” in that one set of configuration data can be used to control many servers or server instances. The configuration can be accessed and modified from any appropriately connected Notes client.

8.3.3.5 ODBC and AS/400 data access

You can write Java agents and servlets that use JDBC to access DB2 UDB for AS/400 (and non-IBM databases, such as Oracle). This section examines data access.

Lotus Notes Domino application development tools, regardless of platform, allow for access to relational data by using ODBC. Many existing applications use LotusScript:Data Object (LS:DO) or include @db formula statements with an ODBC parameter specified. With Domino for AS/400, the application code looks exactly the same from the programmer's perspective.

The iSeries differentiation is what OS/400 does with the database request after it leaves Domino. When the code runs on the Domino for AS/400 server, as in the case of server agents or scheduled agents, OS/400 intercepts the ODBC request and processes it by making direct calls to DB2 UDB for AS/400. No ODBC driver or middleware communications layer is necessary. This provides a shorter path length for optimum performance, as well as eliminating the cost and work associated with installing, configuring, and managing an ODBC driver. This benefit assumes you do not have client-based Domino functions or others that use ODBC directly.

Database terminology comparison

There are some differences in terminology between the iSeries server, SQL, and Notes. Table 2 shows the terms that have similar meaning between the three environments.

Table 2. Database terminology

DB2 for AS/400 relational database	SQL database	Domino database
Library	Collection	Database (.nsf file)
File	Table	Form/View
Record	Row	Document/Row
Field	Column	Field/Column
Query	Query	Selection Formula

On the iSeries server, there are libraries that contain groups of data. This is similar to the concept of a collection in SQL and a Domino NSF database. Within a library on the iSeries server, files are synonymous with a table in an SQL collection and a view in Domino.

One example of a data item is known as a *record* in an iSeries file. This record is the same as a *row* in an SQL table and a document in a Domino database view. Within the record, there are individual data elements called fields on an iSeries server. A *column* is a data element within an SQL row. A data item in a document is called a *field*.

Lastly, on the iSeries server and in SQL, you create queries to obtain data from the data repositories. In Domino, you use selection formula to perform this function.

Note

The redbook *Lotus Domino for AS/400: Integration with Enterprise Applications*, SG24-5345, provides a broad understanding of the different ways to integrate Notes and Domino applications with DB2 UDB for AS/400 data and iSeries applications in a Domino for AS/400 environment. In selecting the most appropriate tool, it is important to understand the trade-offs of each, such as where it can be used, the prerequisites, the required skills, scope of data or application access, and performance considerations.

8.3.4 Security

Domino supports leading security and has access controls (user authentication), digital signatures (documents and mail), encryption (communications, SSL, and mail), and protection from active content (ActiveX, JavaScript, executables, self-extracting .zip). The iSeries server is known for its security. The combination of the application security of Domino and AS/400 security makes for very secure systems.

8.3.5 Development tools

Several development tools are described in this section, including Domino Designer, VisualAge for Java, and WebSphere Studio.

8.3.5.1 Domino Designer

Lotus Domino Designer R5 is an integrated Web application development environment used to rapidly build and deploy secure e-business applications to connect enterprise data with strategic processes.

The key features of Domino Designer R5 include:

- **Comprehensive IDE**: A task-oriented, integrated environment to create secure, end-to-end Web solutions.
- **Rapid development**: Create applications based on Web-ready templates, add workflow, security, and other services using Domino Objects.
- **Native access to enterprise data**: Quickly and seamlessly connect Web applications to live data in relational databases, ERP applications, and transaction systems.

- **Open standards support**: Use any HTML authoring tools, Java IDEs, and scripting tools. Supports such Web standards as Java, JavaScript, HTML 4, CORBA/IIOP, and OLE.
- **Build locally, deploy globally**: Supports Web users worldwide in their native languages with Domino Global WorkBench.

Domino Designer comes with the following tools:

- **Page Designer**: WYSIWYG HTML authoring with complete control over page design and layout. Add styled text, image maps, tables, Java, and ActiveX components to applications.
- **Frameset Designer**: Used to create multi-paned interfaces for Web applications, without HTML coding. Automatically maintains target links.
- **Outline Designer**: Design an entire site, link content to the site design, manage the links, and automatically generate a personalized site map.
- **Domino Objects**: Rapidly build applications that access system services like security, messaging and workflow in your applications, using your choice of Java, JavaScript, CORBA/C++, or LotusScript.
- **Domino UI Java Applets**: Enhance the functionality of browser-based applications with Notes design elements like a view with resizable columns, multiple document selection, and rich text with no programming.
- **Forms Designer**: Create professional-looking forms (surveys, visitor registration, and so forth) quickly.
- **Programmer's Pane**: Provides universal, consistent access to all programming languages and scripts.
- **Instant Feedback**: Preview as you go in your choice of client, including Lotus Notes, Microsoft Internet Explorer, and Netscape Navigator.

Domino Designer gives you live access to enterprise data and applications, via support for Domino Enterprise Connection Services (DECS). This feature offers:

- **Comprehensive connectivity**: DECS supports a wide range of enterprise systems, including DB2, Oracle, Sybase, ODBC, EDA/SQL, SAP, PeopleSoft, J.D. Edwards, Oracle Applications, MQSeries, CICS, and more.
- **High performance, real-time connectivity**: DECS manages persistent, parallel, pooled connections from Domino to external data sources, enabling efficient, simultaneous data access.

- **Your choice of development options**: Connect to enterprise data non-programmatically via the DECS interface or programmatically from LotusScript.

Domino Designer allows your choice of popular Web development tools, including:

- **Extensive Java support**: Develop Domino Web applications in your favorite Java IDE, such as Symantec Cafe, Borland JBuilder, or IBM VisualAge for Java. Embed and manipulate Java applets, integrate Java servlets, and even create Domino server agents in Java.
- **Integration with NetObjects Fusion and Microsoft FrontPage**: Create HTML pages and links in your favorite site authoring tool and save them directly to the Domino object store.
- **Use with Lotus eSuite DevPack**: Domino Designer templates enable easy integration with a powerful set of pre-built, pre-tested, Java applets including spreadsheet, word processor, calendar, address book, and CGI gateway.

Domino Designer lets you create and maintain multilingual Web applications, with Domino Global WorkBench, so you can:

- **Make your applications multilingual**: Work the Web in your site visitors' native languages. Maintain multilingual Web sites with minimal additional overhead, while slashing translation costs.
- **Synchronize translated content**: Manage the release of documents across different language versions of your site.

To find out more about Domino Designer and other Domino development tools and services, visit the Lotus Web site at: `http://www.lotus.com`

You can also refer to the Developer Central site at:
`http://www.lotus.com/developers`

8.3.5.2 WebSphere Studio

Domino is optimized for application and collaborative services (messaging, content management security, and so forth) and convergent applications (automating unstructured business processes, information flow). WebSphere is optimized for such infrastructure services as basic HTTP, server-side Java Web services, distributed transaction management, Java standards, Java programming model, EJB, and performance management.

Domino is becoming more of a development environment (with WebSphere for JavaServer Page (JSP) and Enterprise JavaBean (EJB) support).

WebSphere and application development sites are incorporating workflow and messaging more into the solutions.

The overall integration strategy by Lotus and IBM is building towards an integrated and consistent Web application platform. Figure 40 illustrates this organization.

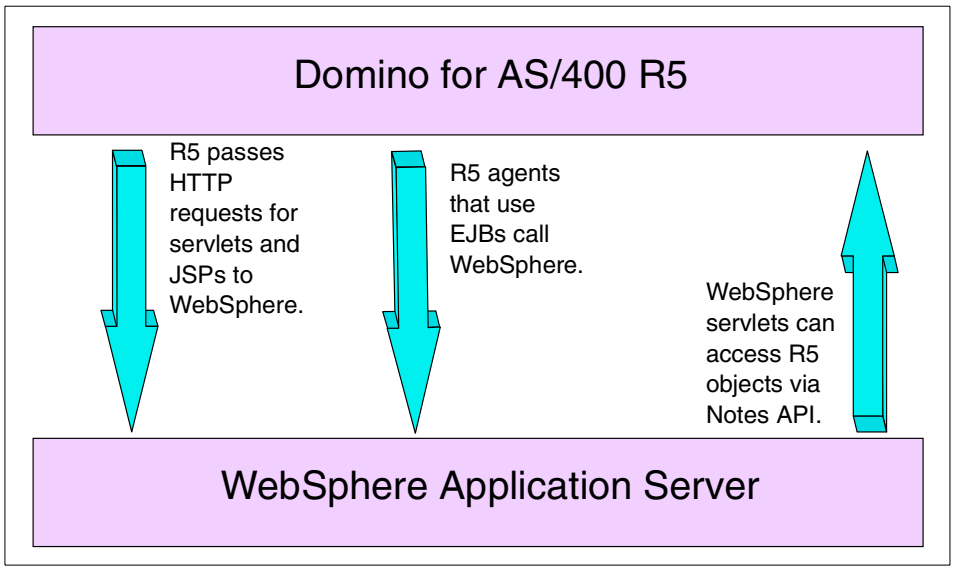


Figure 40. Applications span Domino and WebSphere

This includes common application services (directory security), common programming models (JavaBeans, servlets, JSPs, EJBs, Domino objects), a common set of developer tools (Java and eBasic as a common scripting language for JSPs), common enterprise data connectivity structure, and combined solution packages.

Refer to Chapter 6, “WebSphere Application Server for AS/400” on page 105, for additional information.

8.3.5.3 WebSphere Commerce Suite integration with Domino

WebSphere Commerce Suite (WCS) programs are based on DB2, which Domino applications can connect to using DECS and LEI. Typically these two products do not integrate automatically by themselves, but work in a complementary way. One scenario is a WCS transactional site with Domino workflow, messaging handling communications, and business process management in the organization.

Refer to Chapter 7, “WebSphere Commerce Suite for AS/400” on page 129, for further information.

8.3.5.4 Domino Toolkit for Java

Toolkit for Java enables developers to write, test, and debug Domino agents using their favorite Java Integrated Development Environment. Visit `http://www-1.ibm.com/servers/eserver/iseries/domino/domdevtools.htm` for further information.

8.3.5.5 VisualAge for Java

VisualAge for Java is for programmers. It is a full-blown Java IDE that has all the tools you need to write, debug, and test Java applets, servlets, and applications. It integrates with Domino through the free Domino Toolkit for Java. Please visit `http://www.iseries.ibm.com/developer/` for additional information.

Note

The Domino Application Studio is a comprehensive set of leading Web development tools and servers for designing and developing Web applications. Products not covered include NetObjects Fusion, NetObjects Bean, and Script Builder. Additional information is available at:
`http://www.lotus.com/home.nsf/welcome/appstudio`

8.3.5.6 Payment systems

Domino allows connection to industry standard authentication authorities (Visa, MasterCard, CyberCash, Amex, and so on). It supports SET Secure Electronic Transactions and Electronic Commerce Modeling Language (ECML).

When processing a transaction, the Domino application uses standard APIs to communicate with payment authorities. Domino developers code with these APIs and instruct the application to verify payment at the point in which it is needed in the purchase process. Once confirmation is received or rejected, the transaction continues or an error message is shown.

When do you need payment transactions? The answer depends on the type of payment, such as:

- Real time: When a user submits a form
- Real time: When an order is being fulfilled
- Asynchronous: As a step in a workflow process
- Asynchronous: On a mixed schedule – Authorize now, capture later

Where do you need payment transactions? The options include:

- Web agents (QueryOpen, QuerySave, OpenAgent)
- Scheduled agents
- Notes client events

8.3.5.7 Electronic Store: ezMerchant

Developing and deploying an electronic store requires specialized skills to develop the various underlying technologies (cookies, encryption, data translation, API integration, and so on). It is not a task customers typically undertake. The e-store market is dominated by packaged products that (after being installed and setup) create a functional online store. Currently, ezMerchant by BinaryTree.com is the only offering available that is totally developed using, and is on, the Domino Application Server environment.

ezMerchant is the only electronic store currently offered for sale as a package. It offers a full and complete shopping basket, catalogue, order tracking, and other functions, all implemented on Lotus Domino. This solution is developed by BinaryTree.com, an IBM and Lotus Business Partner. It is a feature-rich commerce solution to create fully functional corporate Web sites, electronic store fronts, and business-to-business solutions.

Some major features include site creation, a product catalog, commerce options, order processing, customer registration, and so on. Additional information is available at: `http://www.binarytree.com`

Refer to 12.2.2, “BinaryTree.com” on page 264, for further information on BinaryTree.com.

8.4 Scenarios: Positioning, sizing, performance, and implementation

This section provides customer testimonials, quotes, sizing information, and references for iSeries customers who use Domino products to build their e-business transactional site solution.

8.4.1 Domino positioning and best use scenarios

Domino offers a rich set of capabilities, enriched by the numerous solutions available from Lotus, independent software vendors, and business partners. Everyone has a different view of what the product does, and every view is usually correct, since Domino means different things to many people. This makes positioning and defining best-use scenarios difficult. You can perform different tasks with the product so the best-use scenario is usually “it depends”.

Traditionally Domino (or Notes as it was called) was thought of only as e-mail, calendaring, and basic workflow. This is still valid, and many companies use the product simply for regular groupware and communication functions.

Domino can be used for building and running more robust and complex applications, as has been described throughout the Domino sections in this handbook, particularly when it comes to e-business.

If you now think of Domino as just a groupware product or just collaboration, think of it as an integrated messaging and Internet application software platform on which you can build and deploy various e-business applications. These applications are enterprise strength and can be deployed to the Internet, intranet, and extranet. They provide a variety of high-impact business process-oriented solutions.

In general, the applications built with Domino are not as transaction oriented or scaled as large as WebSphere or Net.Commerce solutions. However, that is not the value proposition of Lotus Domino. This product focuses on business-process automation, building new information-based applications, managing unstructured data and information, flexibility, integration with back-office computers and databases, and business-oriented solutions. These are all elements of e-business. Consider Domino as a platform to deploy your business on the Internet.

8.4.2 Sizing information

Probably the number one question about Domino for AS/400 is “How big an iSeries server do you need to run Domino for *x* number of Notes users?” You can find the answer on the IBM Domino sizing home page at:

`http://www.iseries.ibm.com/domino/`

Once you reach this site, select **Sizing for Domino** from the **Shortcuts** pull-down list. Among other resources, this will lead you to the IBM Workload Estimator for iSeries found at:

`http://as400service.ibm.com/servlet/EstimatorServlet`

As with every performance estimate (whether a rule of thumb or a sophisticated model), treat it is as an estimate only. Particularly with a robust product like Domino that offers so many different capabilities with a chameleon-like ability to conform to the habits of the users, each installation has unique performance characteristics and demands. The typical disclaimers that go with any performance estimate (“your experience might vary...”) are especially true with Domino. These sizing estimates are general guidelines, but their accuracy in all circumstances cannot be guaranteed.

For more detailed sizing information, particularly if you plan to add Domino for AS/400 to your existing iSeries workload, refer to your IBM consultant and look to the BEST/1 modeling tool. The BEST/1 capacity planning tool has been updated (with PTF SF47087 of 5769-PT1 on V4R2) to include three Domino workloads. This helps size the system for adding Domino to an existing V4R2 workload. Descriptions of those workloads are available through the Support Center and are also posted at the ILIC Web site:
http://www.software.ibm.com/is/ibm-lotus

8.4.3 Domino for AS/400 performance

When the AS/400e division introduced its fourth generation of 64-bit technology, it achieved a NotesBench “Mail only” mark of 27,030 users with a 0.08 second response time. This benchmark used 12-way AS/400e servers, which is the second highest NotesBench number ever. The RS/6000 has achieved about 1000 more users, as has the AS/400e (with lower CPU utilization and faster response times than the RS/6000 model). This means the AS/400e and iSeries servers have the raw processing power to manage the batch server type workloads demanded of a server running Domino.

Since then, the AS/400e and iSeries servers have topped 75,000 mail users.

The iSeries server’s well documented reliability is increased further with the implementation of Domino clustering technology. The iSeries server, on its own, has only hours of downtime per year. Multiple Domino instances can run concurrently (up to 99 on large machines) and be managed by the iSeries server. If one of those instances fails, the iSeries server manages the failure, re-distributes the processing across the cluster, and re-starts the failed server.

Note

A NotesBench audit is an independently verified measurement of the capacity and price-performance of a platform (hardware and operating system combination) running a standard Domino mail workload. See
http://www.notesbench.org

Other additional links related to performance include:

- Domino for AS/400 Performance home page:
  http://www.iseries.ibm.com/developer/domino/perform
- Maximizing application and server performance in Domino:
  http://www.iseries.ibm.com/developer/domino/perform/maxperform.pdf

Document references to server performance and Domino include:

- *Lotus Domino for AS/400: Performance, Tuning, and Capacity Planning,* SG24-5162. This redbook shows a detailed approach for:
  - Measuring Domino application performance
  - Analyzing throughput and response times
  - Extrapolating measured data and doing capacity planning for growth
  - Tuning Domino servers and OS/400 resources for optimal use
  - Considering partitioning, clustering, and sizes of databases
  - Using text search, indexing, and views
  - Measuring, tuning, and extrapolating Domino HTTP server workloads
- *Performance Considerations for Domino Application*s, SG24-5602. This redbook investigates how to consider performance and capacity when designing and developing a Domino application. It also discusses how to analyze a Domino application that does not meet its performance and capacity goals.

Note

The 1999 IDC white paper *A TCO Analysis of IBM AS/400 Dedicated Server for Domino versus PC Servers* is available on the Web at:
`http://www.iseries.ibm.com/conslt/dsd_tco.htm`

This white paper discusses the cost differences between two popular Lotus Domino servers: the AS/400 and the PC. To quantify these differences, an analysis of 500 and 1,000 user configurations for each platform was done, as well as a calculation of three-year total cost of ownerships (TCOs) for both models. Refer to the above Web site for more information.

8.4.4 Scenarios: Implementation examples using Domino

This section includes write-ups of selected case studies, that is, actual customers using Domino for various functions of e-business. Included are dynamic site and transactional site case studies. The problem to solve and the solution chosen is described for each case study selected.

8.4.4.1 Dynamic site scenario case study: Saab Cars USA

To empower dealers with up-to-date product and service information, Saab Cars USA, Inc. built a corporate intranet. It was intended to:

- Improve communications between Saab and its network of more than 225 independent retailers nationwide
- Integrate multiple back-end business systems with the Web

- Minimize the software and hardware required at retail sites

Technology solution
Saab developed an Intranet Retail Information System (IRIS), incorporating the IBM AS/400e server and Lotus Domino running natively on the AS/400e server.

The new solution offers the following benefits to Saab:

- Improved information flow to retailers, with two-second response time for any transaction
- Seamless integration with existing IBM S/390 and AS/400e servers, both housed in different physical locations
- A “serverless” environment at dealership locations, with Web browsers as the only software required on the client
- Simplified systems management through the use of the AS/400e server and the server in multiple roles

A solid anchor for a complex e-business initiative
In the past, information about service, ownership, warranties, and parts was scattered among three systems:

- An AS/400e server at Saab's U.S. headquarters in Norcross, Georgia, which houses the company's core business applications
- An IBM S/390 server at Saab's parts distributor
- A variety of management systems at Saab retail locations

IRIS seamlessly pulls information together from all these sources into DB2 UDB for AS/400, the built-in relational database on the company's AS/400e server. An AS/400e server running Lotus Domino for AS/400 was added as the main Web and messaging server. When a dealer requests data, IRIS pumps it from DB2 UDB for AS/400 into Domino databases residing natively on the AS/400e server.

“The AS/400e server give us the solid reliability and performance we need to make our intranet fly,” says Jerry Rode, director of information technology.

Integrated AS/400 functionality simplifies the intranet
The next piece to the IRIS puzzle moves the data from the AS/400e server to the retailer clients using a satellite network. Saab chose IBM Business Partner Jacada in Atlanta, Georgia, to develop Java-based graphical clients, using Jacada's Web-to-host product, Jacada for Java.

Jacada for Java generates interfaces for the AS/400e server and mainframe applications using Java technology. The advantage of Jacada for Java is its transparency. It supports online transaction processing with Saab's intranet without requiring changes to existing back-end applications. “Now we can tap the rich functionality we've built up in our business systems, and move forward from there,” says Rode.

From the retailers' point of view, all they see is a Java applet. And, all they need to view the data is a Web browser. “With this approach, our retailers do not require a server on site,” says Rode. “... this is a great benefit since it greatly simplifies systems management for them.”

At the time of the initial roll out of IRIS, the Jacada for Java Web-to-host solution could reside only on a Windows NT server. Now that Jacada has announced a new version of Jacada for Java on the AS/400e platform, the AS/400e server fills this role.

“We'll be moving to the AS/400e server as soon as the Jacada for Java code becomes available,” says Rode. “It's one less system we'll have to manage, and we know the AS/400e server will accommodate more concurrent users than our NT server.”

He adds, “With the built-in capabilities of the AS/400e server, we can standardize our headquarters hardware platform. Everything runs native on the AS/400e server. We don't need an NT server for one application, UNIX for another, and Novell for yet another. That means we don't need a lot of disparate skills to support our systems.”

A solid foundation for growth

Saab is working to develop a system for forwarding warranty, sales, and repair information directly from retailer locations to Saab's AS/400e server. This will eliminate costly paperwork and rekeying.

“IRIS will retrieve data automatically from our retailer systems,” says Rode. “We won’t have to wait for batch transfers.”

For example, Saab will be able to track service reports daily and determine whether a particular part is causing recurring warranty claims. The problem can then be corrected at its source, in the manufacturing process.

For the future, Saab plans to roll out IRIS worldwide to all its retailers. The company will tap into the power of the AS/400e servers located at 14 data centers around the world.

“The AS/400 strategy is right for our business,” says Rode. “Intranet and extranet applications are where the field is headed. The recent changes in the AS/400 architecture positions us very well to move in that direction. And, its capabilities keep growing as our environment is growing. It gives us a solid foundation for the future.”

For further information on this case study, refer to this Web site:
`http://www-1.ibm.com/servers/eserver/iseries/casest/saab.htm`

8.4.4.2 Dynamic site scenario case study: U.S. Olympic Committee

Established in 1978, the United States Olympic Committee (USOC) is the coordinating body for all U.S. Olympic-related athletic activity. Almost 500 employees work from its headquarters and training center in Colorado Springs, Colorado; Lake Placid, New York; and Chula Vista, California.

Like the aspiring athletes it supports, the USOC is committed to excellence in serving American sports. The USOC is continually ranked among the top 100 nonprofit organizations in the United States. Its primary goal is to assist in finding opportunities for every American to participate in sports, regardless of gender, race, age, geography or physical ability. It does that by:

- Supporting sports that are, or want to be, included in the Olympic or Pan-American Games
- Supporting the bids of U.S. cities to host the Olympic Games
- Hosting more than 25,000 athletes each year at three Olympic Training Centers

In fact, the USOC allocates grants that are worth more than $30 million to athletic programs and $115 million to member organizations every year.

In coordinating all U.S. Olympic-related athletic activity, the USOC must provide timely, accessible information. It lead the way in the early 1990s by creating a centralized data repository on the AS/400 platform with information on athletes, training centers and ongoing competitions. Data is accessed via PCs in a high-speed wide area network, through Client Access for AS/400.

Technology challenge

The United States Olympic Committee (USOC) wanted to provide information to a wider community of users, including athletes and the media. Initial applications included:

- Drug education
- Insurance policy benefits and administration
- Press Box (including results, event schedules, athletes’ biographies)

Technology solution

The USOC implemented the new applications using the Domino server. The benefits included:

- Easy access through an intuitive graphical user interface
- Simplified systems administration
- High level of security
- Stable environment
- Excellence in delivering timely, accurate information

Integrated AS/400e functionality simplifies the intranet

In early 1997, the USOC raised the bar, setting new goals for information delivery. It saw a need to make information more timely and easy to access for the USOC community of users, including developing athletes and the media. The challenge was to make it easy and intuitive for these people to access information from any PC anywhere in the network. The USOC cleared this hurdle by implementing Lotus Domino on AS/400 on the Integrated Netfinity Server.

“This enables us to provide graphical applications while taking advantage of all the strengths of our AS/400e server,” says Becky Autry, MIS director, USOC.

“Our goal is to achieve an absolutely intuitive graphical user interface integrated with our DB2 database on the AS/400e. The AS/400e offers numerous advantages,” adds Cal Hedgeman, senior manager for application development at the USOC. “The ease of administration and high level of security and reliability are very important. We had done a lot of PC-based application development in the past, but felt the AS/400e offered a stronger platform to enable us to integrate complex applications. It's a very stable environment.”

Setting new standards for information delivery

The USOC decided to start with three small applications:

- Drug education
- Insurance policies and administration
- Press Box

The first two applications provide information directly to athletes on drug use policies for the Games and insurance. For example, an athlete can check information on various drug categories, for example anabolic steroids. In the past, the USOC delivered this information (primarily) on paper or by telephone. Now, it can offer it electronically to everyone at the training centers.

According to Mark Vogtner, associate director, MIS, “With these Notes applications, we'll be able to provide information faster and keep it current. Plus with Notes easy-to-use, intuitive tools, we can dig in and pull out exactly the information that our clients need.”

The third application, called Press Box, is an Internet-based Notes application. Its users are media people writing stories about the U.S. athletes. Press Box debuted at the 1998 Winter Games, providing up-to-date information about participating U.S. athletes to the media at Nagano, Japan. It replaced a PC-based bulletin board system that was set up each time at past Games. Amid a complex tangle of telephone lines, modems, and interfaces, Vogtner recalls, “It was always a challenge to get the system up and running, especially overseas. Press Box, on the other hand, can be accessed from any PC with a Web browser. It requires no special set up at the Games.”

With Press Box, users can send text files instantly, using a Web browser. This process greatly simplifies the flow of media information from the Games, making it more immediate.

For the future, the USOC has plans for a number of additional Notes initiatives. For example, building an Executive Information System (EIS), using the Domino server running natively on the AS/400e server.

“Our goal is to exploit the rich graphical functionality of Notes and the equally rich database capabilities of our AS/400 with its DB2 database,” says Vogtner.

“Notes is the foundation of our information delivery system, replacing some printed reports and faxed documents. It's a leap forward for us in terms of the quality of service we provide to our clients.”

8.4.4.3 Transactional site scenario case study: Longmont Police Department

The Longmont Police Department in Longmont, Colorado, needed to replace its out-dated, unreliable police report-writing system running on a Novell platform. It needed a more efficient accurate system that would integrate with its records management process.

Technology

Through the force of Lotus Domino for AS/400, the department was able to establish a stable, reliable platform on which it deployed a highly functional crime report writing application that has drastically cut the time it takes to

write and file reports, allowing officers to spend more time on the streets protecting their community.

Terry Lahoun, Police Commander, explains, “After evaluating a number of systems, we decided to go with Lotus Domino for AS/400 because of platform stability with DB2 UDB for AS/400 integration that didn't force us to create a lot of infrastructure. At that time, I didn't realize how much we were trying to do depended on a solid e-mail system. With reliability, integration, and e-mail, Lotus Domino for AS/400 tied it all together for us.”

Report writing has long been the bane of every police officer. Dramas on television characterize police singing the blues of this time-consuming process. Report writing keeps them tied up for hours behind their desks, instead of out on the street performing their more important duties for the public.

In reality, writing crime scene and arrest reports that are both accurate and complete is crucial to the legal process and pivotal to the criminal justice system.

To reduce the time and hassle associated with the report writing process, the Longmont Police Department turned to Lotus Domino for AS/400. With the goal of allowing officers to spend more crucial time on the street, the department deployed a new application to make report writing faster and easier, while increasing accuracy.

This was not the department’s first attempt to enhance its reporting system. In the late 1980s, the department deployed an initial application for an automatic police reporting system written in d-base residing on a Novell server. At that time, only about 12 officers used the program. Then the department decided to increase the number of officers on the system and, at the same time, put laptops in many of the patrol cars.

“We actually had two versions of the program: the network version and the stand-alone version for the laptops in the field,” says Lahoun. “But we had a lot of ergonomic problems using the laptops within the cars, so the officers just quit using them and preferred the network version within the department. The entire force of nearly 60 officers eventually moved over to the network version that resided on a Novell server.”

As more users moved onto the networked d-base program, the department began to experience system crashes and conflicts due to downtime. When the system needed to upload the crime reports to its record management program that resided on its AS/400 Model 500, every officer needed to get out

of the system. “Due to the fact d-base rebuilds indexes, every officer needed to be out of the system when we uploaded the reports; we were forced to run the transfer only twice a day,” says Jan Kilgore, Director of MIS for the department. “There was a 12-hour delay in getting the crime reports into our records management system.”

The department realized it needed to install a more stable, reliable platform than was offered by its present Novell server and deploy a more functional report writing application. “Our driving force was to move to another platform that could handle more users and offer more functionality,” says Commander Lahoun. “We wanted to stay in an automated format and not step backwards to paper reporting. I didn't want to use a dictation system. Many forces use dictation systems, but you can show a higher degree of accuracy and efficiency with an automated police reporting system than with a dictation system.”

After evaluating a number of systems, the department implemented Lotus Domino for AS/400 “... because of platform stability and its DB2/400 integration that didn't force us to create a lot of infrastructure,” Lahoun says.

With the installation of the AS/400e Model 170, combined with the power of Lotus Domino, the department now deploys a crime reporting application that offers connectivity and functionality, while working seamlessly with its records management system. “We have been very pleased with the reliability and performance of our AS/400,” says Elizabeth Copley, Program Analyst for the department. “Another important reason we went with Lotus Domino was the flexibility and functionality of working within a Domino environment. Our present and future applications call for the Workflow capabilities of Domino. The system is always reliable; these trusty AS/400s are fabulous.”

Working with its IBM and Lotus Business Partner, Midrange Performance Group (MPG), Boulder, Colorado, the department confidently deployed its newly developed application, Lotus Report Writing System (LRWS). “Now the officers click on the LRWS icon that opens a front-end screen. Everyone is greeted by one main screen as opposed to the previous d-base system that offered officers certain menus and supervisors different menus,” says Randy Watson, President of MPG. “It's very intuitive in nature. The dialogue boxes, the drop-down boxes, the radio button are all things that enforce consistency. Even those officers that know very little about the application can easily get into the system and quickly write a report.”

The new reporting system takes advantage of Notes' ability to organize and associate information from different sources. When an officer gathers notes from a crime scene and begins writing a report, the new system provides the

option to automatically attach the original dispatch request. The report writing system is linked to the Computer Aided Dispatch (CAD) system to eliminate duplicate entry of information and increase accuracy.

“When the officers are adding persons to their reports, such as suspects, victims, complaints, or witnesses, there is a function within the report called a Person Lookup that uses Open Database Connectivity (ODBC)”, says Elizabeth Copley. “The officer can put in part of a name or a full name and all of a sudden we bring up a subfile of all the possible name matches based upon what is keyed into his selection criterion box.”

This function has saved on keying time by bringing in all standard information on the person, such as date of birth and address. There is no need to constantly rekey the same information, increasing both productive and accuracy. “The CAD look up is a real timesaving activity bringing in the information that resides within the AS/400 Model 500,” Copley says.

To make things simpler and quicker for the officer, the report writing system offers a variety of forms appropriate for different incidents. “You can write an offense form, a vehicle form, a property form, a hate crime form, a press release, and then you can write an officer narrative. When the officer finishes, the report is escalated to the next level for a supervisor's approval,” says Commander Lahoun. “The nice feature is that if the supervisor disapproves, a reason for disapproval is put into that document, and an e-mail notification is sent back to the officer stating why the report was disapproved. The cycle now repeats itself.”

Domino's embedded Workflow capabilities keep the report moving until it is approved and finally becomes an official police report. It is then uploaded to the AS/400e server, flowing over into another view for records management department to do a final quality check.

“Now it takes only a matter of a few minutes to transfer data to the records management system,” says Jan Kilgore. “We run it several times a day to keep the records current, saving us nearly 75% of the time it used to take. But more importantly, nobody knows what is happening behind the scenes because we never need to be down.”

Response to the new system has been very positive. The previous process in the records department required nearly ten steps to file and transfer, now accomplished in just a couple of seamless steps.

Lotus Domino for AS/400 also helps the department in its national crime reporting to the Federal Bureau of Investigation (FBI). “We have a program

called National Incident Bureau Reporting System (NIBRS), which provides data to the FBI. There is certain data they need, not the narrative, but the statistics, such as victims, arrestees, suspects, and the crime itself," says Commander Lahoun. "It's very complex to pull out that information; there are about 52 data elements that actually get submitted. They are embedded in our report writing. We used to ask the officers to fill out a NIBRS report, but this was time consuming for them. With Lotus Domino, we have been able to eliminate the NIBRS form we had within the old report writing application, because now that data is pulled out automatically, without the officers worrying about it."

Lotus Domino for AS/400 positions the department to begin plans to continue to build upon the connectivity of its present system. "The whole force has seen advantages to the system with regard to our case management," says Lahoun. "For me, the bigger possibility is to network the actual report writing system to all the components of the criminal justice system. Networking to others with the whole criminal justice system such as the District Attorney's office, probation officers or judges would be a great benefit. If we had the application on their PCs, they could pull the report themselves. With Lotus Domino, it's just so much simpler."

Other future plans may even include the creation of a subset of the report writing program that the department can put out on the Web. For minor crimes, citizens could fill out a smaller version of the application instead of coming into the department. "These reports would be filtered to our administrative report takers who would then follow up with the citizen," says Jan Kilgore. "This also allows the police to spend their time dealing with issues that best use their resources."

Lotus Domino for AS/400 gives the Longmont Police Department the ability to allocate its resources to better serve the community. Instead of sitting behind a desk, Longmont's officers make the best use of their training and skills by being able to spend more time protecting the community.

For further information on this case study, refer to this Web site:
`http://www-1.ibm.com/servers/eserver/iseries/casest/longmont.htm`

8.4.4.4 Business-to-consumer case study: Copart, Inc.

Copart Incorporated (`http://www.copart.com/`) is an automobile salvage dealer and reseller service provider. They offer salvage services primarily to insurance companies that buy and sell salvaged vehicles through auctions. Founded in 1982, the company began with a single location using telephone and fax to contact customers. Today, using technology as a key enabler, it

operates more than 60 facilities in 30 states, with 1,100 employees and annual revenues exceeding $100 million (U.S.).

Copart's technology base is an e-business solution running on IBM AS/400e server. Copart uses the AS/400e server in a number of different roles:

- Web server
- Production
- Solution development
- Offsite backup

And, the company plans to launch Lotus Domino to deliver Web-based electronic mail and workflow capabilities.

Technology challenge

To grow its business, Copart Inc. looked to information technology (IT) as a key enabler. It wanted to implement a new Web-based system to:

- Help it reach more customers in a shorter time frame
- Enable customers to conduct transactions electronically with Copart
- Help Copart realize increased operating efficiencies
- Provide a scalable foundation on which to grow its business

Technology solution

Copart Inc. chose the AS/400e server to deliver online bidding on the Web. The benefits include:

- More timely customer response to bids
- Increased business
- Time and cost savings
- Greater reliability, scalability, and flexibility

Double-edged challenge: Build a business and contain costs

The business challenge Copart faced was how to increase efficiencies within its business processes to enhance customer service, without increasing costs.

“We decided an electronic interface would help reduce costs and improve the delivery of information to buyers,” says David Bauer, senior vice president and chief information officer at Copart Inc. “Prior to our new e-business solution, we used a paper-based system creating a heavy administrative workload. Today, with our AS/400e Web-based solution, we have two-way communications that enable customers to bid electronically on cars and receive critical information on their transaction.”

"In addition, Web-based bidding has greatly reduced the time spent by Copart employees on the telephone and enabled them to accept more business," adds Bauer. "They save an average of seven hours per day on the telephone doing administrative chores and are able to divert this time to enhance revenue-producing activities with customers. We estimate that this will translate into substantial cost savings annually."

A proactive strategy for increased business

Using the AS/400e e-business solution, customers can:

- Place bids electronically
- Receive confirmation that bids were received
- Receive bid status and notification if they are among the "Top Five" bidders
- Rebid if they are displaced from the "Top Five"
- Receive confirmation that they have won (or lost) as well as essential information such as closing price, location of the automobile, and claiming procedures

"Our business has virtually exploded since we went online using the AS/400e," says Bauer. "In addition to opening new facilities this year, we've experienced extraordinary growth."

Also, by being able to notify bidders if they drop out of the Top Five (and their chances of obtaining their automobile are slim), Copart believes that it has generated additional and generally higher bids.

According to Bauer, Copart is now planning a Domino application to send e-mail automatically via the Web to customers.

"When we initially introduced online bidding, our customers took to it like fish to water," he says. "Today, they can see their bidding results online by accessing our Web site. With Domino, we can now take it a step further and actively send them e-mail. We'll move from a passive approach to an active system that electronically prompts customers to actions such as increasing the amount of their bid or even bidding on another automobile."

With the help of Tailored Systems, an IBM Business Partner providing leading-edge e-business solutions, Copart is also deploying a high-availability Real Vision imaging system on the Web using the AS/400e server. This system will enable Copart to take digital pictures of automobiles up for auction and post them on the AS/400e server for Web access by customers.

"Our online bidding system is an excellent example of the Web-serving capabilities of the AS/400e," says Bauer. "Also, the scalability and flexibility of

the AS/400e platform means we are strategically positioned to incorporate new initiatives such as Domino that will help us grow our business even more."

For more informtation about this case study, see
http://www-1.ibm.com/servers/eserver/iseries/casest/copart.htm

8.4.4.5 Domino-based electronic store ezMerchant case study: Blake Equipment

Blake Equipment traditionally had strong sales and a sterling reputation for quality and customer service, but they realized that they were missing an opportunity with the growth of the Internet. Looking to capitalize on the new Digital Economy, Blake Equipment wanted to create a new e-commerce enabled Web site for their successful Connecticut-based water products business. The challenge was that what Blake wanted to build was an entirely new brand identity, H20products.com, on the Web.

Needs assessment

With the water product industry being a niche market, a highly focused approach was necessary to capture their target audience's attention on the Web. What Blake Equipment needed was an e-commerce enabled Web site to provide a secure environment for transactions as well as the ability to communicate with their back-end systems. Blake also wanted a phased, integrated approach, so that different components could be added as time and budget allowed.

BinaryTree.com's solution

BinaryTree.com's ezMerchant solution (Version 2.0) allowed Blake Equipment to expand quickly into this new medium using only a fraction of the costs and resources normally required for an e-commerce Web site. In looking for a site builder to not only develop an appealing Web site, but also meet their criteria for secure e-commerce transactions and seamless back-end connectivity, Blake Enterprises settled on BinaryTree.com's ezMerchant.

Some competitive differentiation of ezMerchant includes shopping cart capabilities, secure credit card transactions, and above all, Web site ease of use. Internet surveys confirm that if a customer finds the information they are looking for within three mouse clicks, the chances of them returning to that site are reduced considerably. Thanks to ezMerchant's intuitive site builder, H20products.com customers can easily access the information they need.

“ezMerchant makes interacting with our customers easier and more efficient. With Binary Tree’s tools and services we are able to serve our customers better,” according to Fred Cuda, President of H20products.com.

BinaryTree.com recommended that Blake Equipment use Domino for their messaging needs as well. On the basis of this recommendation, Blake Equipment upgraded their server from an AS/400e server Model 600 series to the Model 720. In doing so, they also implemented a Lawson system to handle their back-end needs. The decision to go with Lawson was made after Blake Equipment initially settled on an alternate back-end system, but found that BinaryTree.com's expertise with Lawson, coupled with ezMerchant's high degree of compatibility, made Lawson the most attractive choice.

Real-time information exchanged between the backoffice and the H2Oproducts.com Web site provided improves customer service and facilitates faster customer access to their accounts. H20products.com was up and running in two months, with a site that met all of their objectives.

By using H20products.com as a primary business tool, customer service can be far more effective, with updated orders and information available immediately on the Web. With real-time data available, customers know the inventory status of products they are interested in, without waiting for a sales person to get back to them. Their new ezMerchant site, H20products.com, Blake Equipment signs up new distributors over the Web, which reduces their overhead costs. This site also gives Blake the opportunity to expand nationally, as opposed to region by region.

The software they use includes:

- BinaryTree.com’s ezMerchant 2.0
- Domino Server R5
- Domino R5 Clients (Administrator, Client, and developer)

The hardware they use includes the IBM AS/400e 720 server.

Additional product information is available at: `http://www.binarytree.com`

8.4.5 Additional reference material

Additional Domino for AS/400 case studies available on the Web at:
`http://www.iseries.ibm.com/casest/casemenu.htm`

The redbook *Developing an e-business Application Using Lotus Domino for AS/400*, SG24-6052, shows you how to create an e-business application with Domino for AS/400. It describes the tools and techniques used to develop a

sample e-business application with Internet, intranet, and extranet functions, as well as the infrastructure and security implemented for the application.

New Enterprise Integration Functions for Lotus Domino for AS/400, SG24-6203, was recently published and covers how to call OS/400 stored procedures from Domino for AS/400 and how to use Java in Domino to develop integrated Domino and AS/400 applications. There is also a section on integrating SAP R/3 with Domino for AS/400.

Up and Running with Domino for AS/400, SC41-5334, helps new Domino users install and configure Domino for AS/400 Release 5.0.1 or later on a Dedicated Server for Domino.

Lotus Domino for AS/400 R5: Implementation, SG24-5592, Domino R5 is Lotus' new generation of Internet products. This redbook helps you implement and administer this exciting new release on the iSeries server with Lotus Domino for AS/400. This redbook is written for technical specialists and programmers, who are IBM customers and Business Partners.

Chapter 9. Other IBM e-business products for the iSeries server

This chapter discusses some of the many IBM products that run on iSeries servers to enable an e-business environment, including Host-On-Demand, Host Publisher, and MQSeries. Many other solutions are available, some of which serve the same functions as these featured products and others to further complement them.

IBM PartnerWorld for Developers sponsors an iSeries Application Development Tools Network. This iSeries program is a resource used for IBM and non-IBM tools and currently sponsors over 100 business partners. The Tools Network program is described on the Web at:
`http://www.as400.ibm.com/developer/tools/index.html`

This site also offers a reference for non-IBM e-business products. You can also find this information in Chapter 12, “B2B: Application solutions” on page 261.

9.1 IBM WebSphere Host On-Demand

IBM WebSphere Host On-Demand gives you secure access to your host applications and data, using a simple, Java-enabled Web browser. With secure access to your host data and with pSeries, iSeries, and DEC/UNIX emulation, Host On-Demand requires no client installation or middle-tier server. Centralized deployment means that you always have the latest version of code, and software maintenance costs are minimized, with only the Web server needing code installation.

9.1.1 Quick, easy access to critical host data

The browser-based access of IBM WebSphere Host On-Demand gives users a simple way to reach critical host data, without requiring any software to be installed on the client. IBM WebSphere Host On-Demand uses the power of Java technology to open the doors to your host data whenever you need it, wherever you need it, directly from your browser. Simply click a hyperlink to launch IBM WebSphere Host On-Demand Java applet. This Web-to-host connectivity solution provides secure Web-browser access to host applications, so you can take existing host applications to the Web without programming.

With support for TN3270E, TN5250, VT52, VT100, VT220, and IBM CICS Java Gateway access, users have a single interface to their key host data. Because IBM WebSphere Host On-Demand is Java technology-based, its

interface has the same look and feel across various types of operating environments. IBM WebSphere Host On-Demand also provides a default graphical user interface (GUI) to help simplify the experience for users who are unfamiliar with traditional green screens.

9.1.2 A cost-effective approach

You can save money in product deployment and maintenance by installing IBM WebSphere Host On-Demand on a Web server, eliminating the need to manage individual user desktops. Users can connect directly to a host system, such as an IBM iSeries or pSeries server, eliminating the need for extra hardware and software required by three-tier solutions. IBM WebSphere Host On-Demand can be installed on nearly any server platform, accommodating various size organizations and branch offices.

The latest version of IBM WebSphere Host On-Demand is downloaded each time a user accesses the software, eliminating the need to install code on client workstations. With the Cache Client feature and the new SmartCache in IBM WebSphere Host On-Demand, Version 5.0, users download once. Then subsequent uses originate from their hard drives, while code updates are handled in the background through the Smart Cache. Users are always productive, and always have the latest version of code, with no distribution costs.

IBM WebSphere Host On-Demand can be installed on many platforms, including Microsoft Windows NT, IBM AIX and Linux, as well as iSeries and pSeries enterprise servers.

9.1.3 Connect directly to any Telnet server

With IBM WebSphere Host On-Demand, the emulation functionality is contained in the client applet. This eliminates the need for a middle-tier server, which can be a performance and security issue. Once the applet is served to the client, it's easy to connect directly to any standard Telnet server that provides the best access to the required data. The Telnet connection can be changed as often as user requirements for new data change. You can access an unlimited number of host sessions concurrently. Because no middle-tier server is required, you are not restricted by its capacity.

9.1.4 Simplify Web connectivity

For organizations with iSeries servers, IBM WebSphere Host On-Demand is the first choice for providing users with connectivity back to the iSeries server, with features that include:

- 5250 host printing
- 5250 device IDs
- 5250 file transfer
- Native system installation
- ENPTUI emulation support
- Database On-Demand

9.1.5 Create new e-business applications

A rich Java tool set, including Host Access Beans for Java, IBM Host Access ActiveX controls, and IBM Host Access Class Library API, help enable customers to rapidly create custom e-business applications to achieve competitive advantage. Because IBM WebSphere Host On-Demand is part of the WebSphere product family, applications developed using the tool set can be incorporated as part of other WebSphere software projects. This preserves your IBM WebSphere Host On-Demand investment and helps provide a quick start to the Web and e-business.

The Host Access Class Library API provides access to 3270, 5250, and VT data streams. These class libraries allow you to use mission-critical information in new ways, such as integrating data from one application with another.

Host Access Beans for Java provide host connectivity and emulator functions through simple, component-based development tools, like IBM VisualAge for Java. You can use these beans to rapidly create custom applications that allow you to deliver the specific functions you want to include in your host access applications. These object-oriented beans help minimize your development efforts through software reuse. Application developers who are familiar with ActiveX tools can use IBM Host Access Controls – a set of ActiveX controls used to provide the functionality found in Host Access Beans for Java.

9.1.6 Security enhanced access across the global Internet

Using Secure Sockets Layer (SSL), Version 3.0, IBM WebSphere Host On-Demand extends host data access across intranets, extranets, and the Internet with added security. Mobile workers can establish security enhanced communication with an enterprise host. With client and server certificate support, IBM WebSphere Host On-Demand can present a digital certificate (X.509, Version 3) to the Telnet server for authentication. Extending this support to certificate authorities, including Tivoli SecureWay Trust Authority, allows enterprises to benefit from industry standard public key infrastructure (PKI) methods.

9.1.7 Manage large numbers of users

With IBM WebSphere Host On-Demand, administrators can permit users to create their own IDs, so large numbers of users can manage themselves. By employing a combination of features – configuration migration from IBM Personal Communications and import and export from IBM WebSphere Host On-Demand – configurations can be shared, managed and distributed easily among groups defined by administrators. The administrator panel provides a familiar tree view, providing easy management for IBM WebSphere Host On-Demand users and groups, allowing all user and group information to be displayed on one screen.

For even greater enterprise management, IBM WebSphere Host On-Demand can use an LDAP server to store IBM WebSphere Host On-Demand configuration information. This includes LDAP storage of all user, group, and session-configuration information, such as keyboard mappings, macro definitions, session parameters, and a migration facility for existing IBM WebSphere Host On-Demand profiles.

9.1.8 IBM Screen Customizer

IBM Screen Customizer automatically converts host screens into a graphical presentation that is easily customizable without any programming. Designed with patented screen recognition, IBM Screen Customizer can be used with both IBM WebSphere Host On-Demand and IBM Personal Communications. There is no risk to the host application, and no impact on the host workload. Users can simultaneously access multiple hosts with multiple sessions running, without any degradation in host response time. After users install IBM Screen Customizer, they never have to see a *green screen* again.

9.1.9 International language support

IBM WebSphere Host On-Demand is multilingual and is available in 21 languages, including double-byte character set (DBCS) languages. Support for the European currency symbol 1, as well as keyboard and code-page support for many more languages, is also provided. All language versions are available on the same media, and multiple language versions can be accessed concurrently, making it a truly international product.

9.2 IBM WebSphere Host Publisher

Making host information available through the Internet is an integral part of successful e-business. Web-to-host integration is one of the fastest ways to

extend existing business-critical applications to your employees, trading partners and customers.

IBM WebSphere Host Publisher is a Web-to-host solution built to address the unique characteristics of the Internet. A key component of the IBM WebSphere Host Integration solution, WebSphere Host Publisher allows you to integrate multiple sources of data, including host and database applications, as a single Web page with no change to back-end systems. It provides the high level of security you need for your Web-based environment, offering support for Secure Sockets Layer (SSL) encryption and authentication, as well as DES-encrypted passwords. And IBM Network Dispatcher, included with WebSphere Host Publisher, provides load balancing and failover so that large enterprises can maximize WebSphere Host Publisher performance, throughput, and reliability.

With WebSphere Host Publisher, you can provide important information directly to users, reduce the expense of call centers, and improve customer service. Simply build and deploy Web self-service applications to provide host access to virtually any user with a standard Web browser, a user-friendly vehicle for users who are unfamiliar with traditional green screens.

WebSphere Host Publisher consists of two major components:

- **Host Publisher studio**: Provides an easy-to-use customization environment to create Integration Objects that can be used to dynamically access back-end data sources.
- **Host Publisher server**: Includes WebSphere Application Server and provides the run-time environment to execute Integration Objects, reusable beans for Java applications, created with Host Publisher studio.

You can create Integration Objects and Web pages using Host Publisher studio, publish them to Host Publisher server, and provide transparent host access to users. The inclusion of IBM WebSphere Studio gives you the ability to scale and reuse your original Integration Objects as part of new advanced Web applications and to add new business logic.

9.2.1 Host Publisher studio

Host Publisher studio automatically generates Integration Objects, which encapsulate interactions and data retrieval. Integration Objects are used in fully customizable HTML pages and can be reused by other Java application programs created outside WebSphere Host Publisher.

Host Publisher studio generates fully customizable HTML output with embedded JavaServer Pages (JSP) tags. Point-and-click features allow Web

administrators to map specific fields on a Web page using the HTML tags to point to specific fields on the host emulation screen or database table. This bi-directional implementation allows information to be updated from the Web interface, making it simple to connect users to back-end systems across the enterprise. You can also enhance the generated HTML with your favorite Web authoring tool, such as WebSphere Studio, to meet your particular style and image requirements.

WebSphere Studio Professional Edition is included as part of WebSphere Host Publisher. WebSphere Studio provides all the tools you need for complete WebSphere Host Publisher e-business implementation, such as HTML or JSP editing, new business logic creation, and site deployment enhancements. Host Publisher studio runs on the Microsoft Windows NT, Windows 2000, Windows 95, and Windows 98 operating systems.

9.2.2 Host Publisher server

Host Publisher server includes the IBM WebSphere Application Server, Standard Edition and other run-time components, such as connection management, license monitoring, run-time administration, and log and trace management.

Host Publisher server offers enterprise-class performance, scalability, and availability through several key features, such as object chaining, connection pooling, load balancing, and failover support. Object chaining improves performance and flexibility by breaking complex tasks into manageable subtasks.

These reusable objects can be chained to provide the most efficient flow through a complex application or used by other applications requiring the same subtask. Connection pools, which are defined in the Host Publisher studio, eliminate the overhead of establishing, connecting, and disconnecting separate connections for each host request.

Host Publisher server runs on WebSphere Application Server Standard Edition. WebSphere Host Publisher uses WebSphere Application Server to support the run-time environment for applications that use Integration Objects created by Host Publisher studio. Integration with WebSphere Application Server provides WebSphere Host Publisher applications with access to IBM connectors, including IBM MQSeries. You can also reuse Integration Objects within new, WebSphere-based applications or use WebSphere software and your favorite Java interactive development environment (IDE), for example IBM VisualAge for Java, to add new business logic to WebSphere Host Publisher implementations.

WebSphere Host Publisher provides a run-time version of WebSphere Application Server. If you need or already use the advanced features of IBM WebSphere Advanced Edition or IBM WebSphere Enterprise Edition, you can use your products to support the WebSphere Host Publisher run-time environment.

Host Publisher server runs on the IBM OS/390, IBM OS/400, IBM AIX, Sun Solaris, and Microsoft Windows NT and Windows 2000 operating environments. This allows applications created with the common Host Publisher studio to run unchanged in all environments.

9.2.3 Start simple

WebSphere Host Publisher can help your business reduce costs and minimize complexity associated with exploiting Internet technologies. You can create new composite applications that mine information from various data sources without modifying your back-end systems. Integration of back-end systems is performed on the Host Publisher server, and then, HTML is delivered to the end-user Web browser. Easy-to-use GUIs, such as wizards, allow Web designers to create and manage new composite applications. These applications combine multiple back-end data sources to form a Web page. Task-oriented prompts guide the designer through the creation process recording host and database interactions, identifying desired data, and labeling selected data for retrieval. When the Web page is completed, it is published to the Host Publisher server for production access by users.

9.2.4 Provide flexibility

This open, industry-standard software supports applications running on traditional host systems, such as 3270, 5250, and virtual terminal (VT). WebSphere Host Publisher also connects to Java applications and databases with Java Database Connectivity (JDBC) interfaces, such as the IBM DB2 Universal Database and databases from Oracle and Sybase.

Host Publisher Remote Integration Object (RIO) support enables remote Java applications or applets running on a remote client or server to execute Host Publisher Integration Objects as though they were physically running on the WebSphere Host Publisher server.

Through its XML gateway, WebSphere Host Publisher provides access to existing 3270 and 5250 applications in an XML format for use within new e-business applications. Additionally, the HTML mapper capability provides a load-and-go HTML entry-level emulator for 3270 or 5250 application access. Without customization, existing 3270 and 5250 applications can be extended

as HTML to Web users. This ability is targeted at users who need occasional access to the host application and do not yet have desktops enabled for Java applications.

9.2.5 Scale reliably

Scalability, high availability, and reliable security that are combined with support for a wide range of back-end applications make WebSphere Host Publisher an easy, cost-effective way to extend your business-critical applications to the Internet.

Through multi-platform run-time environment support of the OS/390, OS/400, AIX, Solaris, and Windows NT operating environments, your WebSphere Host Publisher applications can be deployed in your varying and growing computing environment. The inclusion of IBM Network Dispatcher enables you to spread the load of user network traffic across a pool of Host Publisher servers.

9.2.6 Grow fast

Beyond the movement of key host applications to the Web, the need to expand your business applications with new capabilities is important in the emerging e-business marketplace. With WebSphere Host Publisher, you have the ability to reuse, expand, and compound your initial investment in Host Publisher Integration Objects in new business applications that can grow with the needs of your company.

More and more companies find the need to deliver host applications and data to handheld devices. WebSphere Host Publisher, coupled with IBM WebSphere Transcoding Publisher, can dynamically extend the reach of host data and applications beyond the Web to pervasive technologies, such as SmartPhone devices and personal digital assistants.

9.2.7 WebSphere software platform: Building on a firm foundation

IBM WebSphere Host Publisher is part of the IBM WebSphere software platform – a comprehensive set of integrated, award-winning e-business solutions. No matter where you are in the e-business cycle, the WebSphere software platform allows you to grow at the speed the market demands. Building on this robust platform, you can connect diverse IT environments to maximize your current investments and leverage existing skills. Deliver your core business applications to the Web using industry standards like Java technology and XML. And create next-generation applications that differentiate you from the competition.

9.3 IBM MQSeries for AS/400, V5.1, 5733-A38

The IBM MQSeries family provides an open, scalable, industrial-strength messaging and information infrastructure. This enables tighter integration of business processes and dynamic workload distribution to form an efficient enterprise- or community-wide system. The MQSeries family consists of three products:

- **MQSeries messaging**: Provides robust middleware for secure and reliable communications across 35-plus platforms (5733-A38)
- **MQSeries Integrator**: Centralizes and applies the rules for transformation and routing of application messaging contents as transmitted by MQSeries (5801-AAR, Feature #6002)
- **MQSeries Workflow**: Enables the capture, visualization, and automation business processes using predefined workflow modules (5801-AAR, #3604)

9.3.1 Business integration with the MQSeries Family

Business integration means that different computer systems, employees, business partners and suppliers work together to provide the best and most innovative service to the enterprise and its customers. The barriers of diverse computer systems, geographic boundaries, time differences, language and format differences, and different methods of working can all be overcome with the MQSeries family.

Business integration means that it is possible to:

- Connect customers, suppliers, partners, and service providers, while maintaining security and control, to enable newly built and re-engineered applications for more effective business processes (for example, supply-chain management).
- Make mergers and acquisitions a success by integrating dissimilar IT infrastructures from two or more companies so they can work as a single entity.
- React more quickly to market trends and opportunities because IT systems are flexible and dependable, and no longer constraining.
- Different hardware and software platforms can behave as if they were designed to work together.

9.3.2 MQSeries for AS/400 features at a glance

- Heterogeneous any-to-any connectivity from desktop to mainframe (over 35 platforms supported).
- A single consistent API, shielding developers from networking complexity.
- Allows a business to integrate disparate islands of automation.
- Time-independent communication.
- Guaranteed one-time delivery.
- Retains close integration with OS/400.
- Makes full use of built-in iSeries features (for example, journaling and CL command interface).
- Enhances work management and security for greater ease-of-use.
- An interactive interface to MQ commands to make administration easier for users familiar with other MQSeries platforms.
- Support for remote administration through the MQ Explorer, an element of MQSeries for Windows NT, V5.1.
- Improved scalability of MQSeries applications on the iSeries server by increased maximum message and message queue sizes (100 MB and 2 GB respectively).
- MQSeries Bindings for Java and support for multi-threaded programs to assist in integrating MQSeries into advanced e-business applications.
- The most widely used message-queuing software on the market, with more than 66% market share.
- A copy of Candle Command Center Admin Pac for MQSeries is included with MQSeries at no additional charge. Candle Command Center Admin Pac for MQSeries is a selection of Candle Corporation solutions for testing MQSeries applications, configuring MQSeries networks, and managing MQ-based computing enterprises.
- Used by more than 5,000 customers to solve their business integration issues, operating in two-thirds of the top 100 North American and European banks, and installed in 350 out of the top 500 IBM customer sites.
- EuroReady.
- Prerequisites include OS/400 V4R5 or V4R4.

9.3.3 Application programs and messaging

The IBM MQSeries range of products provides application programming services that enable application programs to communicate with each other using messages and queues. This form of communication is referred to as *commercial messaging*. It provides assured, one-time-only delivery of messages. Using MQSeries means application programs can be so that the program sending a message, continues processing without having to wait for a reply from the receiver. If the receiver, or the communication channel to it, is temporarily unavailable, the message can be forwarded at a later time. MQSeries also provides mechanisms for providing acknowledgements of messages received.

The programs that comprise an MQSeries application can be running on different computers, on different operating systems, and at different locations. The applications are written using a common programming interface known as the message queue interface (MQI), so that applications developed on one platform can be transferred to another. When two applications communicate using messages and queues, one application puts a message on a queue, and the other application retrieves that message from the queue.

Supported programming languages and compilers for application development include:

- ILE C/400
- ILE COBOL/400
- ILE RPG/400
- ILE C ++ for AS/400
- AS/400 Developers Kit for Java
- VisualAge for C++ for AS/400

9.3.4 Queue managers

In MQSeries, queues are managed by a component called a *queue manager*. The queue manager provides messaging services for the applications and processes that the MQI calls and the programs issue. The queue manager ensures that messages are put on the correct queue or that they are routed to another queue manager.

Before applications can send any messages, you create a queue manager and queues. MQSeries for Windows provides the utilities to assist with this and to create other MQSeries objects that are needed by applications.

9.3.5 Supported platforms

MQSeries/400 supports a message exchange with other users of the MQSeries on over 35 IBM and non-IBM platforms. A complete list of supported platforms can be found on the Web at:
`http://www-4.ibm.com/software/ts/mqseries/platforms/`

Client software provides a remote interface to a MQSeries server. An MQSeries client for the iSeries is currently not available. If CICS for OS/400 is installed, MQSeries can interoperate with CICS transactions and can link them with non-CICS environments.

For more information on MQSeries, go to the site on the Web at:
`http://www-4.ibm.com/software/ts/mqseries/`

9.3.6 Version 5.1 enhancements

The enhancements for MQSeries Version 5.1 include:

- Support for multiple queue managers
- Cluster (or groups) of queue managers
  - Dynamically share workload
  - Balance workload
  - Reroute workload if a system component fails or network path becomes unavailable
- Administration of clusters of queue managers is easier and quicker
- Queue managers in the same cluster can be on different platforms or physically remote from one another
- MQSeries jobs now run in their own subsystem and job attributes can be set by the user (for example, priority)
- Multi-threaded application support assists in integrating MQSeries into advanced e-business applications

9.3.7 Migration and upgrade

Technical guidance on migrating from a down-level version of MQSeries for AS/400 to V5.1 is provided by an Instruction Document SupportPac, obtainable prior to new product availability. Download it free from the MQSeries SupportPac library at:
`http://www-4.ibm.com/software/ts/mqseries/txppacs/`

A valid Software Subscription entitles the user to new level MQSeries products at no charge.

MQSeries for AS/400 V5.1 is also available in shrink-wrapped format as product 5798-AAR, feature #5610.

9.4 IBM MQSeries Integrator for AS/400 and DB2 V1.1, 5801-AAR

MQSeries Integrator is powerful message-brokering software to provide real-time, intelligent rules-based message routing and content transformation and formatting. It seamlessly integrates applications, databases, and networks. MQSeries Integrator is optimized for high-volume, in-storage transformation of messages.

The iSeries platform joins the other key platforms on which MQSeries Integrator is available. The refresh level of the product at V1.1 is consistent across the MQSeries Integrator offerings on all distributed platforms and is available concurrently for all platforms. Usability, serviceability, and GUI improvements help the user:

- Be more selective when viewing the components that make up a format
- Use a find option
- Browse relationships between components at different levels
- Use import or export functions to extract from a database into a file to edit or import into other systems
- Try formats and definitions using a visual tester

MQSeries Integrator offers these advantages:

- Makes it easier to integrate applications and data enterprise-wide
- Provides faster access to information
- Shortens time to market
- Improves customer service and reduces overall costs
- Opens up the information in IT systems to suppliers and customers
- Helps leverage the value chain to improve quality and accelerate responsiveness to change
- Relieves the burden of modifying applications every time they are integrated, that is, connected in new ways
- Transforms and routes data outside the application, without the need for valuable programming and communications skills

Messages pass through a central connection point, the hub, which acts as the core for holding enterprise intelligence. It is designed to handle any volume of

traffic, no matter how heavy or complex. The hub maintains two types of knowledge:

- Knowledge of the applications to enable transformation of message formats. If packaged applications are involved, application templates enable a quick start to integrate those applications with the rest of the enterprise.
- Knowledge of business rules and information requirements to enable intelligent routing of information to where it is needed. Rules, whether complex or simple, can be defined to tailor information flow.

9.4.1 Transformation

The knowledge of each application is stored once in the hub and while intelligent routing (see below) decides where each message is to go. It is translated into the appropriate format. Supplied with the information definition of each application, the transformation engine supplies data in the right format to any receiving application, without the sending application being modified.

9.4.2 Intelligent routing

Intelligent routing encapsulates business knowledge of how information should be distributed between message-sending and receiving applications throughout the enterprise. This knowledge is stored in the hub as a set of rules that are applied to each message that passes through the hub. Messages are distributed according to criteria applied to the values of fields within the message.

9.4.3 Supported platforms

MQSeries Integrator supports message exchange with other users of MQSeries on over 35 IBM and non-IBM platforms. A complete list of supported platforms can be found on the Web at:

`http://www-4.ibm.com/software/ts/mqseries/platforms/`

Part 4. B2B: Business transform through applications transform

This part discusses B2B in general and how you can use the iSeries to transform your business way through transforming its existing core applications. iSeries products are available to build an e-business site.

This part is for both business decision makers and technical people. It will help you understand the transformations you must make to your business processes for e-business. To that end, this part discusses B2B connectors, those key technologies and actual products that help you to establish the connections between your *core applications* and your customer-facing solutions.

Many companies have already “solved” such integration problems. This part surveys some of the pre-packaged Applications Suites suites that are meant to save time and effort over building your own application suites using the connectors discussed above.

And, finally, this part discusses the service offerings that are available for you with a focus on this B2B transformation.

Summit strategies

Summit Strategies has released a very well written report that is entitled “IBM’s iSeries 400: AS/400's B2B Transformation” from January 2001. It introduces the concept of B2B and the role of the iSeries. For a copy of this report, go to `http://www.summitstrat.com/`

Chapter 10. B2B: Transforming business processes for e-business

What is B2B? The title of this chapter says it all. Business-to-business. It transforms your business processes, the way you do business to achieve your business goal better, to thrive in your business area, to lead the pack of the industry, and in many cases, to simply survive. You achieve this transformation by transforming your existing *core* applications into *Web-enabled* applications. This chapter sets the foundation for the rest of the chapters in this part.

10.1 Why should we care?

The theme of this section is “transform business”, that is, transforming the way you do business and the flow in which the processes interact with each other. This is not something you do alone. You need to transform business processes and workflow of your business partners, your suppliers and customers alike, along with you. You want and need this transformation to take place together and toward the same direction. The changes have to be almost revolutionary rather than evolutionary.

At the same time you are moving forward, you also want to ensure that you are not losing your “feet on the ground” perspective and understanding of the value your company delivers to its industry and the relationships you have built with your trading partners and customers. Often, B2B can be defined as “back to the basics”. That means preserving your core-business value and logic while extending your reach is the ultimate business goal.

You question, “Changing the way we do business in a revolutionary fashion? And we do this along with our business partners? Not just that but the whole industry and community and the world move together in a same direction? Are we dreaming?”

No, you are not. It is absolutely possible. In fact, it is happening even as we speak. Many perceive this transform as a business financial improvement strategy. Some perceive it as survival strategy. In our opinion, both are right.

The following sections look at what B2B is and explain how you can participate in this trend in a reliable fashion ahead of your competitors and with your partners.

10.1.1 What is B2B?

First, what is B2B? What is e-business? Are they related or even the same thing? What do they have to do with business transformation? Let's start the argument by attempting to define e-business appropriately.

Some have defined e-business as "every sort of business you conduct over the Web." Maybe others focus on the "e" portion of e-business.

We focus on the "business" portion of e-business. Most likely this is due to the solid ground placement of the iSeries in the world of business servers.

For example, let's say an auto maker has published their Web site to advertise new models, their company, and various events around the world that they sponsor. The site is always refreshed with the current information on a daily basis. Can you call this e-business? According to the first definition, yes, you can. But does it impact the "business" in a critical fashion? Can the company afford for the site to go down for a couple of hours? How about a couple of minutes? Can it ever afford to lose the online service at all? If it's going to be an e-business site, the answers to these questions should be "yes". If the answer is "no", it is not an e-business.

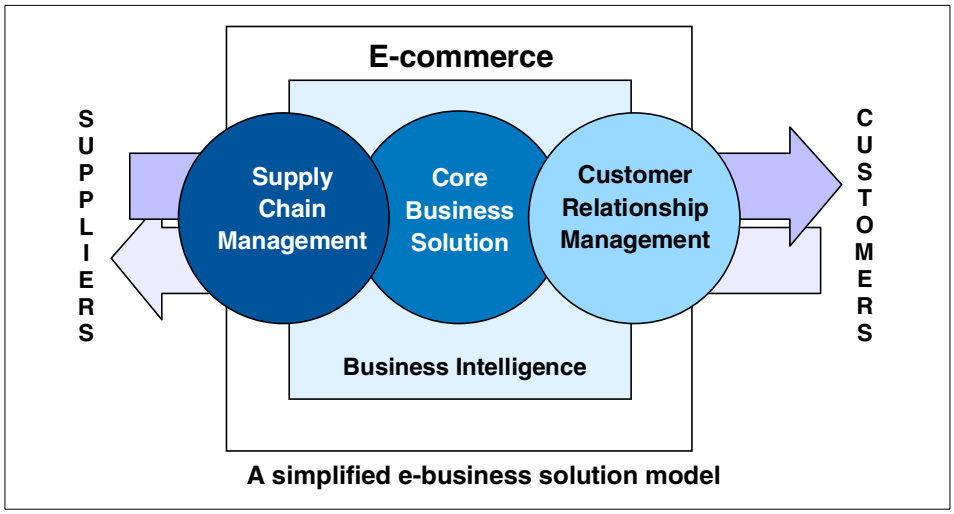


Figure 41. e-business definition

Look at Figure 41. In the middle of the picture is the "Core Business Solution". You may call it differently, for example: an ERP system, LOB, back-end application, or even a legacy system. To a large majority of iSeries

customers, it is an RPG application. No matter how you call it, that is your existing core business engine and that's where the real power is.

What is business transform? It is the process of extending the power of the existing applications set to your Internet, intranet, and extranet audience so that they can access the core applications set, basically at their finger tips through a Web browser.

This introduces new layers of application processes such as:

- **Supply Chain Management (SCM)**: For your suppliers' audience to access your core
- **Customer Relationship Management (CRM)**: For your customers' audience
- **Business Intelligence (BI)**: For your internal staff such as sales, planning, marketing, and business executives, and e-commerce, which allows your customers and business partners to purchase your goods and track orders over Internet

Our definition of e-business is to "transform key business processes by leveraging traditional IT and Internet technologies". This is exactly what we mean by B2B. It is why and how you should transform the business processes.

Before we continue, let's summarize some of the terminology:

- Synonyms referring to existing business applications are:
  - Core Business Solution
  - LOB (Line-Of-Business)
  - Back-end application
  - ERP system
  - Transactional system
- Extending the existing business application to an Internet audience includes the terms:
  - e-business
  - B2B
  - Business transform

Of course, each of these names carries its own meaning. Maybe it is not fair to generalize these names and group them together as if they are really all identical. But remember we are talking at a conceptual level, and at that level, they make sense.

10.1.1.1 iSeries e-business strategy

From the IBM @server iSeries division's perspective, the iSeries e-business strategy in 2000 is an installation-base strategy. Over time, as IBM delivers e-business references on iSeries, we will be able to focus on opportunities outside our installation base. iSeries customers will be able to evolve from "brick and mortar" business models to "brick and click", B2B and B2C models. This means that you will be able to continue to leverage your existing IT infrastructure. And at the same time, you can take advantage of the exploding opportunities that are available by extending your business-critical applications to the Web.

It's important for iSeries customers to understand that their predicament is not unique. Today, only 2% of companies are "green field". That means that 98% of customers have to extend existing applications. This includes customers with Windows NT, UNIX, S/390, and their environments that face the same Web challenges (and opportunities!) as iSeries customers.

iSeries customers will be able to take advantage of the iSeries server's optimized infrastructure (Java, Domino, Web serving, and so on) to migrate applications to the Web. This includes existing LOB applications, such as Billing, A/R, Inventory, for example, as well as front-office applications, such as SCM, CRM, and e-commerce applications.

10.1.1.2 Comparing B2B with B2C

We all understand what B2C is. That is the business of buying and selling over the Web. That is online shopping. That is Amazon.com, eToys.com, and Nordstrom.com. This is already part of our society and is rapidly becoming one of the key staples of it. In the U.S. alone, the volume of online Christmas shopping in the year 2000 was expected to reach $12 billion. We all know B2C. It's big and keeps growing rapidly. So how is B2B different from B2C?

Look at Figure 42. In the upper left corner of the picture, a little boy is drinking milk. When he's out of milk, his family buys more milk via a transaction that could take place at a local food store *or* via the Web. This is marked as 1 in the picture and is where the B2C action takes place. This simple B2C transaction will spawn a numerous number of following transactions between so many different business organizations. As you can easily see in the picture, all the transactions from 2 through 10 are B2B transactions.

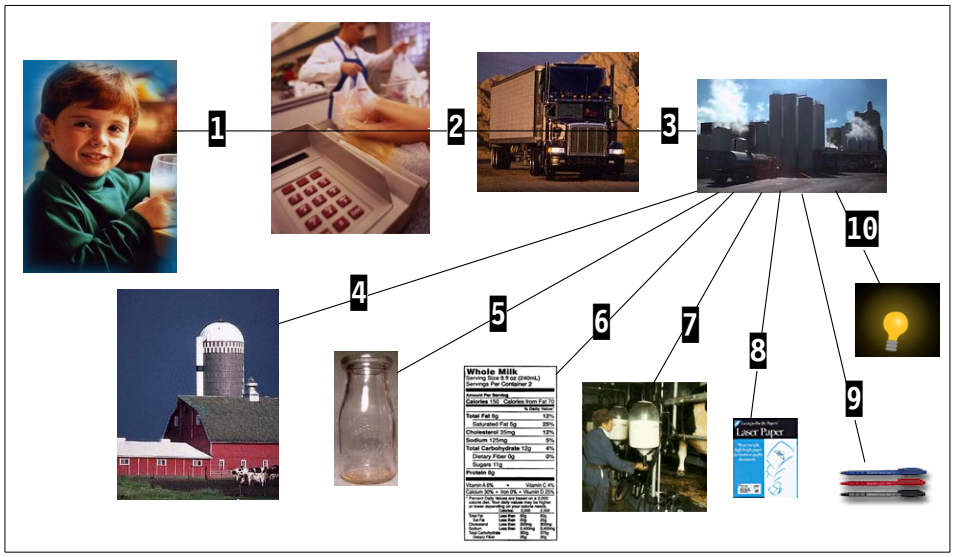


Figure 42. B2C to B2B ratio

Year 2003 projections put this into a business figure perspective. It is expected that some $300 billion work transactions will take place in B2C, while this figure in B2B would be some $3.2 trillion.

10.1.1.3 B2B differences from B2C

At this point, it is beneficial for us to summarize the differences between B2C and B2B.

Different connection topology

You may want to put your B2B server deep in your trusted intranet loop and give tight access control. You will want to open the access authorities only to those trusted business partners or internal staff.

In the case of B2C, you might not want to enforce the same level of access control rules simply because it will not make much business sense. Therefore, you might consider putting your B2C server within the demilitarized zone (DMZ; describes a place in your network between the public Internet and your internal intranet). Let each business process go through the security control mechanism whenever they need to come to the core part of your business engine, which resides beyond your bastion firewall.

Different system capacity requirements

The B2C server should be optimized from the performance perspective to handle a large number of concurrent requests at the expected peak time.

Literally you have no control over this type of workload, and you need to be prepared for the worst (but the best for your business). At any rate, the most important requirement here is to provide the acceptable response time to each online shopper.

In case of B2B server, however, what's more important here is reliability and the availability of the services. It's common in this setting that a single unit of work will involve multiple transactions with multiple data updates in multiple DB files so secured commitment control is a must feature to be implemented.

Different types of purchases and authorization
Compared to B2C, B2B has these characteristics from a purchase and authorization perspective:

- Scheduled, repetitive
- Larger quantities and dollar value
- Open Requisitions (POs)
- Different authorizations within business customers
- MRO purchases

  Maintenance, Repair, and Operations
- Direct purchases (Supply Chain)
- Often by SKU
- Unique contracts and terms and conditions for different business customers

Different personalization
In the case of B2C, the most common personalization technique is analyzing the favorite goods that the online shopper has and trying to provide the best candidates for their additional purchases. For example, let's say you run a book store on the Web. If you have a customer who has a pattern of buying thriller novels, you can personalize their site to introduce new thriller novels in the market on their next visit.

In the case of B2B, the customer may be a procurement officer for a company. For this customer from their business perspective, a volume purchase discount rate table, for example, may be more important than the preference of the color of fabric on a sofa.

10.1.2 What's driving the adoption of B2B

There are a number of factors that drive B2B, but the three most important driving forces of this trend are:

- Business exerting their buying power
- Businesses expanding their market reach
- Businesses reducing the cost of transacting business

10.1.2.1 Business exerting their buying power

B2C buying power and its consequences – the risk of lost business – can be quite significant. In the case of B2B, this can be enormous. In many cases, an unfortunate incident of consecutive mistakes can drive a company out of business. For example, suppose you are an auto-parts supplier. Let's assume that your customers, who are major auto makers, standardized their parts procurement processes and demanded all suppliers adopt this new B2B way. What if your order-taking application has not been B2B enabled? Even if it has been B2B enabled, what if there have been a couple of serious incidents of not being able to meet your customers delivery schedule due to the errors in your workflow?

10.1.2.2 Businesses expanding their market reach

You can expand your market reach through proper and efficient management of an electronic catalog or electronic configurator that provides:

- Integrated enabling infrastructure
- Can integrate with existing back-end systems
- Improved service to buyers or distributors

10.1.2.3 Businesses reducing the cost of transacting business

When you think of every business organizations' primary goal of increasing profit, there are always two ways of achieving that goal. The first way is to increase the revenue. The second way is to reduce the costs. Many view B2B as a way to increase revenue, but B2B has a larger much larger potential of increasing profit by decreasing the operations costs of a business organization. This can be achieved by reducing the following costs:

- Search costs or information costs
- Bargaining costs or decision costs
- Policing costs or enforcement costs

Table 3 shows the case of IBM procurement cost savings by implementing B2B.

Table 3. Benefits of B2B adoption: IBM results in e-procurement

Procurement step	Before	Now
Purchase Order Processing Time	30 days	1 day
Contract Cycle Time	6 to 12 months	30 days

Procurement step	Before	Now
Average Length of Contract	40+ pages	6 pages
Rate of “Maverick Buying”	30%	Less than 2%
Internal Satisfaction With Procurement	40%	More than 85%

The original objective of the IBM e-procurement system was to leverage IBM’s global purchasing power by installing common processes and applications across the corporation and Web-enabling critical functions. IBM now has a leadership position in exploiting IT for the procurement of non-production goods and services. Since 1993, IBM has saved $9 billion.

10.2 B2B business models

In B2B computing, there are three emerging models. They are represented in:

- Buy-side
- Sell-side
- e-marketplace

In Figure 43, the bottom box is the Trading Partner Agreement. This is not a model, but such as agreement plays an important role in this area. You should understand how these models map to the needs and capabilities of your customers’ companies.

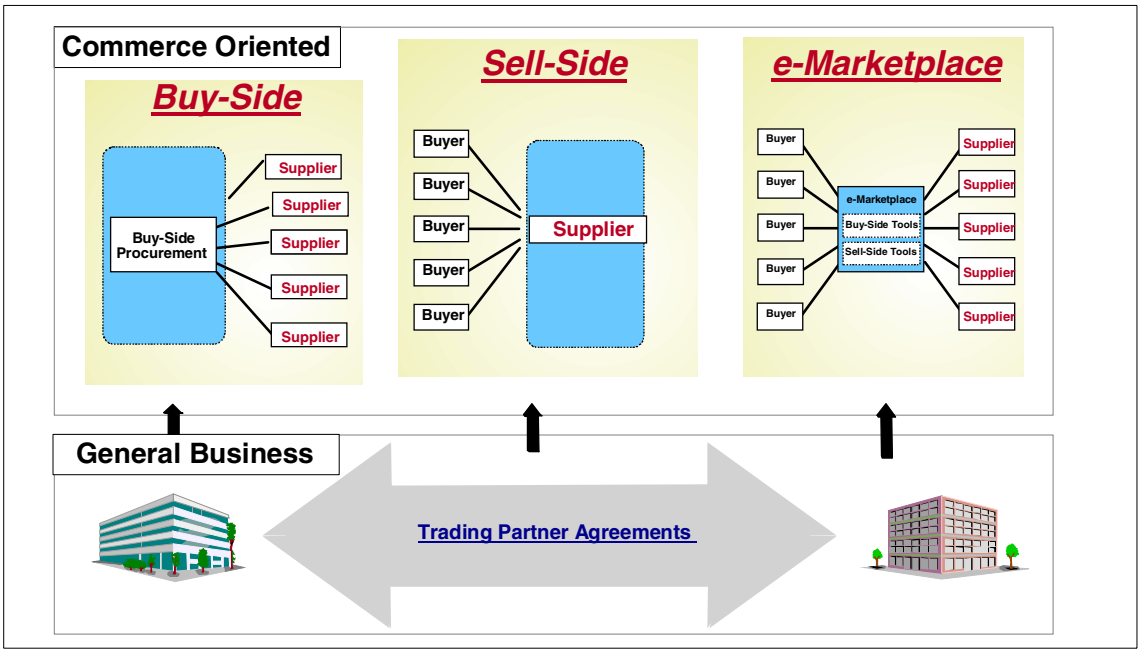


Figure 43. B2B business models

The following sections examine all three of these B2B business models.

10.2.1 Buy-side model

A buy-side solution, also known as a *one-to-many model*, restructures the procurement process within a company. Buy-side solutions have typically been adopted by Fortune 500 companies as a way to have consistent, entity-wide procurement systems.

In this model, multiple catalogs (and specific items within those catalogs) are consolidated with negotiated prices and any necessary pre-authorizations to provide a consistent interface to the company's buyers. For example, a buy-side solution might offer a computer-based interface that presents the company's furniture buyer with only those pieces of furniture that have been pre-authorized and approved for purchase.

This sort of buy-side model eliminates rogue buying and guarantees that the company has the best pricing and volume discounts.

10.2.2 Sell-side model

While the buy-side model represents a one-to-many relationship, the sell-side model has a many-to-one relationship. The most common implementation of a sell-side B2B application is posting a catalog on the Internet that is available to other businesses. The analogous model in the B2C space is Amazon.com. However, in the B2B model, the buyers are not the end consumers that are buying books. Instead, the distributors that are buying the product from the manufacturer are the consumers.

This model allows the procurement staff to view all products, enter their own orders, and check the status of an order. Sell-side solutions also enable companies to apply the information they have about their customers to tailor the message that the Web site presents to each customer. Customization can maximize revenues from cross-selling and up-selling. This model provides marketing departments with a level of granularity and detail that they could only have dreamed of in the past.

10.2.3 e-marketplace model

The third emerging B2B model is the wave of the future. This model is also known as many-to-many, metamarkets, EcoNets, online marketplaces, and others. We refer to these markets as “hubs and spokes”.

Types of e-marketplaces include online auctions, exchanges, and vertical portals. The iSeries server is a natural for filling the role of the e-marketplace hub because of its reliability, scalability, and security strengths. However, you should also recognize that the iSeries (or AS/400e to be more correct in the historical sense) already lives and prospers in the traditional spokes of business. Businesses require robust, efficient software that enables the spokes to connect to multiple hubs.

10.2.4 Trading Partner Agreement (TPA)

Companies engaging in B2B e-commerce – regardless of the model they are in or whether they are suppliers, buyers, or an e-marketplace provider – all need a certain set of business rules. They also need a common data structure so that they can exchange required data without any confusion or problems. Trading Partner Agreement (TPA) provides that method.

We attempt to illustrate this with an example. Consider an auto parts manufacturer who sells to a major auto maker. These partners could implement a TPA to streamline their communication, as follows:

- The auto maker creates the TPA (using an editing tool) and sends the parts manufacturer a tpaML (which is the new IBM XML dialect that is

defined in 11.2.5, “XML (eXtensible Markup Language)” on page 232) template containing its vital information.

- The parts manufacturer adds vital information about itself to the template and electronically returns the completed TPA.
- The TPA is processed (at both locations) by a code generator. The customized B2B integration software is created, which enforces the rules of interaction established in the tpaML.
- The auto maker can issue a purchase order electronically to the parts manufacturer. Acknowledgement or modifications to the order can be transmitted back to the auto maker.

10.2.5 Why is it important to understand these B2B models?

Understanding these B2B models is important for two main reasons:

- Reason 1: It is expected the whole industry will evolve based on these models. At least, these are the models upon which upcoming IBM @server (which includes iSeries) strategies, technologies, and products will be based.
- Reason 2: These are the models we adopted to explain almost everything for the rest of the B2B part, which includes connectors, solutions, and services.

10.3 B2B: Not writing new applications but transforming existing ones

B2B (or e-business) is not about writing a whole new set of applications to replace existing ones. Although this is could be an option, it is a questionable alternative in terms of feasibility and justifiability.

Instead, you should extend your existing applications to both the Internet and your intranet. To do this, you need *connectors*, which are defined in the next chapter.

Chapter 11. B2B: Connectors

What are B2B connectors? B2B connectors are the technology used to integrate your business to your suppliers' and buyers' business to avoid manual processing of business-to-business transactions. This, in turn, reduces processing time and transaction costs.

11.1 Connectors: What and why?

Because e-business is an evolving process, integration with customers, suppliers, e-marketplaces, and your back-end (legacy) systems is a key issue. This section looks at various options available for integrating your Web application with your customers, suppliers, e-marketplaces, and your back-end applications.

Connectors do not give you a ready-to-run application (for this, you need to consider Chapter 12, "B2B: Application solutions" on page 261). Rather, they provide universal access from your Web application to your existing data, applications, and transactions. They help you transform your business to an e-business without making major changes to your existing applications.

11.1.1 Enabling new business processes

A common view of connectors for the new B2B business processes is shown in Figure 44 on page 222. There is a requirement for the customer, supplier, and e-marketplace in the new front-end process, as well as for the back-end process that you must integrate with your legacy applications.

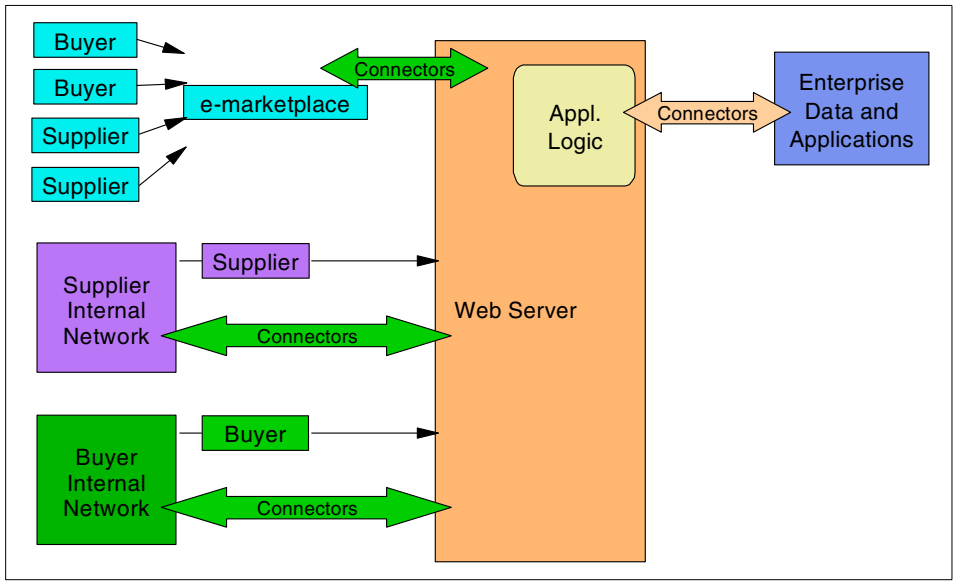


Figure 44. Common view of the connectors for the new B2B processes

Each e-marketplace, buyer, or supplier uses a different business protocol to communicate with the outside world. As a versatile application, your application should be able to talk to these different business protocols.

There are three different types of B2B scenarios when you deal with outside applications:

- **e-marketplaces**

  Your Web application can act as a supplier or buyer in different e-marketplaces. You could buy or sell products with different e-marketplaces.

- **Supplier network integration**

  Your Web application can interact with a supplier's internal network directly from your procurement system.

- **Buyer network integration**

  Your Web application should be able to provide both a remote and local catalog for the buyer network.

Depending on your business transaction, you need to interact with your back-end legacy or ERP application components. Based on the type of business transaction and the back end involved, there could be various ways

in which connectors can be implemented. Some examples include MQSeries, API calls, Remote Function Calls, Remote Procedure Calls, JDBC, DB2 Extenders, and so on.

Figure 45 shows how you can use various technologies to connect and integrate your Web application with back-end applications and the outside world.

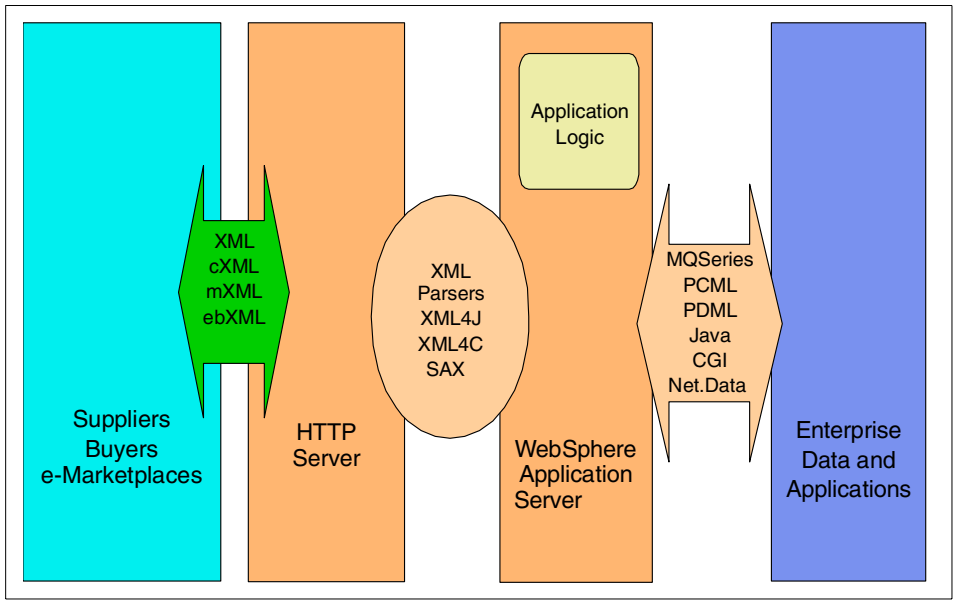


Figure 45. Various technologies used to integrate and connect your Web application

The entire integration of your Web application with the outside world and internal systems can be divided into three stages:

1. Communication with the outside world

   Communication with the outside world is done through XML documents or a different type of XML, such as cXML by Ariba SupplierLive or mXML by Metiom and so on.

2. Front-end processing

   Front-end processing involves converting and validating an XML document to a form that is recognized by your Web application. For this reason, various tools and technologies are used that are described in detail later in this chapter. Such tools include XML parsers like XML4J (XML Parser for Java) or XML4C (XML parser for C++). Document Type

Definitions (DTD) are used for validation and describing the relationship within data.

3. Back-end processing

   Back-end processing involves communication and integration with your back-end or legacy applications. For this reason, you can use such technology as MQSeries, a Java application, Program Call Markup Language (PCML), or Panel Definition Markup Language (PDML).

11.2 Connector technologies and products

B2B, then, is the art and science of connecting your business logic and processes to other business logic and processes, whether they are on your same iSeries server or another system.

Connectors, such as WebSphere Application Server, AS/400 Toolbox for Java, MQSeries, Domino for AS/400, eXtensible Markup Language (XML), Net.Data, Common Gateway Interface (CGI), and CORBA, all provide you with the mechanism to make these connections.

If you are looking for a program product to help you integrate many of these connectors from a single place, go to 11.3, “Connect for iSeries: Pulling them all together” on page 249.

11.2.1 WebSphere Application Server

You can use WebSphere Application Server (WAS), along with the HTTP server, to run the front-end process of the third-party request. WAS can provide the services for the application servlet engine to run your servlets, with the HTTP server providing the HTTP protocol handling.

WAS supports XML document structures and can generate, validate, parse, and serve XML documents.

11.2.1.1 Using CCF connectors with IBM VisualAge for Java

The task of connecting an application to a back-end data store is relatively standard. It follows the same basic pattern whether you are considering the interactions between applications, servlets, EJBs, message queueing systems, relational databases, transactional systems, or some other pieces of enterprise infrastructure.

The IBM Common Connector Framework (CCF) recognizes that most interactions follow a standard pattern. It provides a standard Java-based infrastructure for integrating various system components together.

The CCF solves the problems by providing:

- A common client programming model for connectors that greatly reduces the learning curve for an application developer
- A common infrastructure programming model for connectors
- A plug-in interface for higher-level tools, making them independent of a particular connector

The CCF is easily applied within a feature-rich component environment such as the environment provided by WebSphere Application Server to support servlets or Enterprise JavaBeans.

11.2.1.2 The MQSeries connector

VisualAge for Java's MQSeries Common Connector Framework (CCF) Connector classes provide a higher-level Java interface, which conforms to the IBM CCF. This interface simplifies some of the programming tasks associated with the MQSeries Client Classes for Java native programming interface. It is consistent with the CCF interfaces implemented by other IBM connectors.

Programs written using the MQSeries CCF Connector classes can communicate with programs that use the standard MQSeries programming interface (the MQI) or with programs that use the MQSeries Client Classes for Java interface. The other applications can be running on any of the systems to which MQSeries has been ported.

11.2.1.3 IBM DB2 JDBC drivers

The JDBC driver consists of the JDBC client and the JDBC server. The JDBC client driver is loaded along with applet or servlet.

Figure 46 on page 226 shows the most common scenario with a Web server and servlet. The servlet uses an application driver and can connect to the DB2 server located on the same machine or on a different machine. The best way to access the DB2 database is to install the DB2 client on the same machine as the Web server.

Calls to JDBC are translated to DB2 CLI calls through Java native methods. This requires the DB2 Client Application Enabler (CAE) component to be installed at the client. A JDBC request flows through the DB2 CLI to the DB2 server through the normal CAE communication flow.

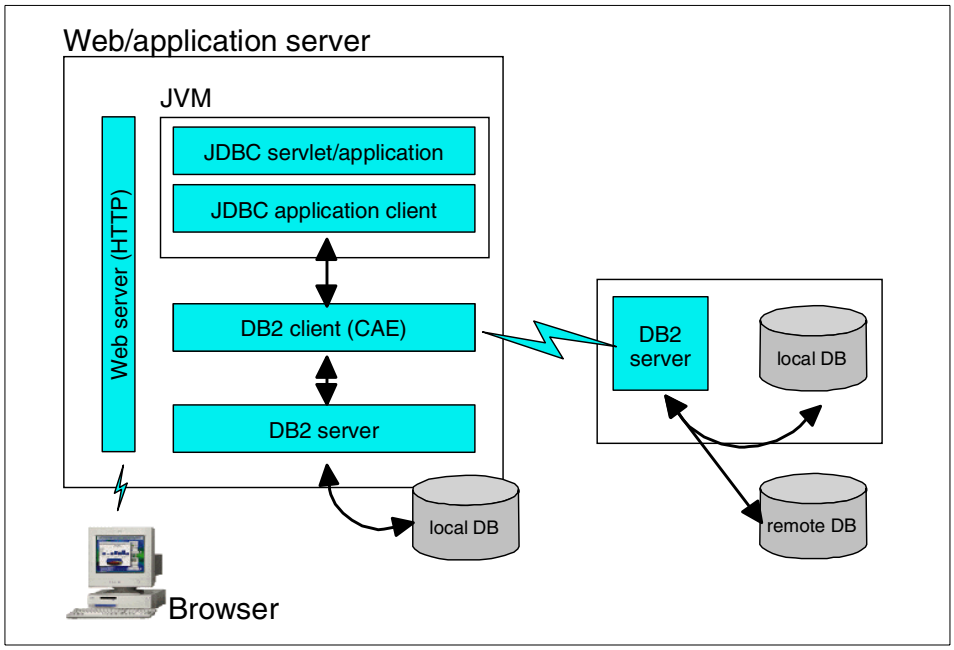


Figure 46. DB2 JDBC application driver architecture in a servlet

11.2.2 AS/400 Toolbox for Java

AS/400 Toolbox for Java is a library of Java classes that give Java programs easy access to iSeries data and resources. These classes allow you to access legacy or ERP application data through Java programming. That is, AS/400 Toolbox for Java delivers a ProgramCall class, which provides a means of calling any iSeries program from a Java program. More specifically, the ProgramCall class uses the iSeries Host server ProgramCall driver to call any iSeries program object.

AS/400 Toolbox for Java includes the XML4J parser (IBM XML Parser), Program Call Markup Language (PCML), an XML dialect, to support Java programs calling iSeries applications programs and Panel Definition Markup Language (PDML), an XML dialect, to support GUI panel definition.

AS/400 ToolBox for Java access classes

AS/400 Toolbox for Java access classes represent iSeries data and resources. The classes work with iSeries servers to provide an Internet-enabled interface to access and update iSeries data and resources. That is, the `com.ibm.as400.access` package has interfaces, classes, and exceptions to access various iSeries resources.

The servlet classes that are provided with AS/400 Toolbox for Java with the access classes, which are located on the Web server, give you access to information located on your iSeries.

Figure 47 shows how the servlet classes work between the browser, Web server, your Web application, and iSeries data.

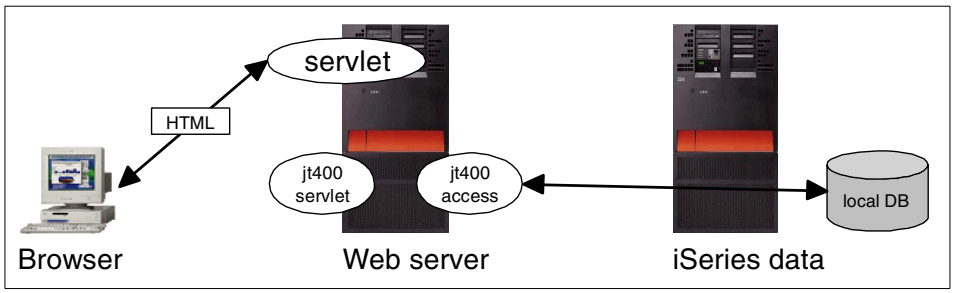


Figure 47. How servlet classes work between the browser, Web server, and Web application

A browser connects to the Web server that is running your Web application servlet. The jt400Servlet.jar and jt400Access.jar files reside on the Web server because the servlet classes use some of the access classes to retrieve data and use the HTML classes to present the data.

11.2.3 MQSeries

MQSeries enables Web applications to use message queuing to communicate with your back-end legacy or ERP applications. Programs within a Web application communicate with a back-end legacy or ERP application by writing and retrieving application-specific data (messages) to and from queues without having a private, dedicated, or logical connection to link them. Therefore, the Web application is shielded from the mechanics of the underlying communications.

MQSeries is used in a client/server or distributed environment. Web and legacy programs can run on one workstation or on different machines on different platforms.

MQSeries Queue Manager ensures that messages reach the target queues. It can also provide a confirmation of delivery option that is unlike the OS/400 built-in data queue function.

Figure 48 shows how MQSeries can be used in the Web application environment to communicate with back-end applications.

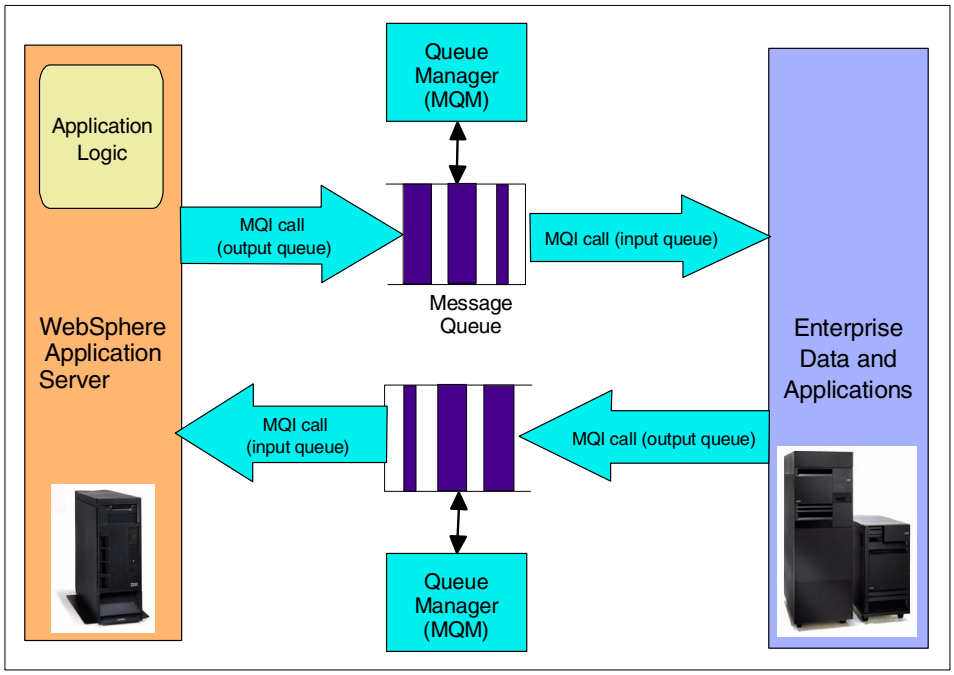


Figure 48. MQSeries with back-end applications

Application programs invoke functions of the queue manager by issuing API calls. For example, the MQPUT API places a message on a queue to be read by another program using the MQGET API.

A program may send messages to another program that runs in the same machine as the queue manager. Or it may send messages to a program that runs in a remote system, such as a server or a host. The remote system has its own queue manager with its own queues. Application programmers do not need to know where the program to which they are sending messages runs. They place their message in a queue and let Message Queue Manager (MQM) worry about the destination machine and how to get it there.

The functions of Message Queue Manager (MQM) include:

- Manages queues of messages for application programs.
- Provides an application programming interface – the Message Queue Interface (MQI).
- Uses networking facilities to transfer messages to another queue manager when necessary.

- Provides additional functions that allow administrators to create and delete queues, alter the properties of existing queues, and control the operation of the queue manager. These functions are invoked through the Run MQSeries Commands (RUNMQSC) utility.

11.2.3.1 MQSeries Client for Java

MQSeries Client for Java provides a set of Java class libraries that permit Java applets on a Web browser, or stand-alone Java applets, to access MQSeries applications over the Internet without needing any other MQSeries code on the client machine.

11.2.3.2 MQSeries Integrator Version 1.1

MQSeries Integrator is a powerful message-brokering software that automatically distributes information to those who need it. When you use MQSeries to transport messages across different computing platforms, it routes information according to enterprise-defined rules, transforming and reformatting it to suit the receiving application. Dynamic reformatting performs conversions, transformations, and message enrichment to meet the needs of the enterprise. MQSeries Integrator seamlessly integrates applications, databases, and networks.

Usability features for those defining broker actions include graphical user interfaces, editable definition files, and a visual tester.

Predefined templates mean MQSeries Integrator can exchange information with major packaged applications such as those from PeopleSoft GL and SAP. They enable new systems to be productive immediately.

The highlights of MQSeries Integrator include:

- Forms the message brokering layer of the IBM business integration framework
- Makes adding, extending, or replacing applications in an MQSeries network simple and easy
- Applies intelligent routing to seamlessly integrate applications and networks
- Enables application-to-application message transformation
- Supports custom built and predefined application libraries
- Supports PeopleSoft GL, SAP R/3, and S.W.I.F.T. templates from New Era of Networks Inc. (NEON)
- Includes new usability features and improved graphical user interfaces

For more information about MQSeries Integrator Version 1.1 (Version 2.0 is not yet available for the iSeries server), see:
`http://www-4.ibm.com/software/ts/mqseries/integrator/`

11.2.4 Domino for AS/400

There are several ways to integrate Lotus Domino and iSeries applications. Lotus Domino applications can interact directly with enterprise data and business logic, and they can use a messaging/queuing layer to post work to an iSeries transaction monitor. iSeries applications can use APIs to update or retrieve Lotus Domino data.

Domino.Connect is an Enterprise Integration solution that integrates Notes and Domino applications with a broad range of RDBMSs, transactions, and enterprise application systems.

Domino.Connect supports these features:

- Is a complete environment for developing and deploying Internet and intranet business applications.
- Enables Notes clients and Web browsers to access enterprise data and applications.
- Allows developers to incorporate database capabilities into Domino-based Web sites and applications.
- Allows secure distributed access and database integration, along with complete messaging, calendaring and scheduling, and replication.
- Allows Domino applications to interact with users to update or create complex transactions with back-end systems.

Domino.Connect consists of a number of components that are the plug-ins, drivers, applications, and tools that integrate the enterprise. They can be classified into the following categories:

- Relational Database Connectivity: Power through integration – NotesPump, NotesSQL, LotusScript Data Object, DB2 Plug-In, and ODBC
- Transaction Connectivity: MQSeries Plug-In
- Application system integration: SAP R/3 Plug-In, Lawson, SSA, and J.D. Edwards
- Developer's interface: LotusScript Plug-In toolkit

Connectors provide the strategic "plumbing" for enterprise integration. They provide native connectivity, via a consistent object model, to external data sources. They allow Lotus Domino applications to connect, authenticate, and

translate data with relational database (RDBMS), ERP, and transaction processing systems.

Connectors can be accessed through visual data mapping tools, such as Lotus Domino Enterprise Connection Services (DECS) or Language Enterprise Integrator (LEI). Alternatively connectors can be programmatically accessed via the Lotus Software Extension (LSX) for Lotus Domino Connectors or Lotus Connectors Java classes. Connectors are “open” so the design specifications are published, and anyone can write a connector using the Lotus Connector Toolkit. Connectors can interact directly with the enterprise data, or they can interact with the business logic.

11.2.4.1 Java, AS/400 Toolbox for Java, and Lotus Domino

The iSeries Java environment interacts with the Lotus Domino environment in the same manner as on the other platforms. Agents can be written in Java to run on the iSeries server. Lotus Domino agents using the Lotus Domino Object classes can run on either the server or the Lotus Notes client. Server Java agents use the built in Java Virtual Machine (JVM) that is shipped with OS/400. Client agents use the JVM of the client operating system.

Agents on either the client or the server may then use Java constructs, such as Remote Method Invocation (RMI), to interact with other enterprise Java applications. Or they may use the AS/400 Toolbox for Java to invoke an iSeries program object or post a transaction to an iSeries application’s data queue.

- **Java classes**: Internally, Java Lotus Domino classes execute the same C++ code as the LotusScript Lotus Domino back-end objects. The only difference is the language syntax.
- **Java agents**: Java classes can also be created as agents in a Lotus Domino database that are executed like any other Lotus Domino agent. They have full agent scheduling and triggering abilities from the client and the server. Java agents can be run in the foreground or the background and can be replicated.

  Java agents complement the LotusScript agents and, to a large degree, can be used interchangeably when dealing with back-end operations.
- **Java applets**: Java applets allow a Lotus Domino developer to create a richer GUI environment for the end user. Applets are dynamically downloaded from the server, executed on the client’s machine, and work with either Web browsers or Lotus Notes clients.

Java applications differ from applets in that they are not dynamically loaded from the server. They are similar to traditional executables in this respect.

- **Java servlets**: Java servlets only run on the server. A servlet is invoked by a client request and responds directly to the client. Typically, a servlet is used to provide a high performance link to a back-end system and display the results on the client in HTML format. However, servlets are not restricted to serving only HTTP requests and may, in fact, converse directly with any suitable client application (usually an applet). A servlet can access both local and remote Lotus Domino databases.
- **Lotus Connector Java Classes**: The Lotus Connector (LC) classes are available to use through Java classes. Developers must import these classes.
- **Java Database Connectivity (JDBC)**: Developers can access enterprise data using Sun Microsystem's Java Database Connectivity (JDBC) API. JDBC provides access to DB2 UDB for AS/400 data whenever Java is used with a TCP/IP connection to the iSeries server. It permits developers to send SQL statements to DB2 databases and retrieve the results.
- **Application programming interfaces (APIs)**: The Lotus Domino for AS/400 product includes low-level C and C++ APIs. These APIs allow the programmer to operate on Lotus Domino objects programmatically, without having to use the Lotus Domino interface. Programs using these APIs can open, create, or delete Lotus Domino databases and manipulate Lotus Domino database elements.

11.2.5 XML (eXtensible Markup Language)

XML is often used to exchange business data between enterprises or within an enterprise's own business unit. XML adds value to these relationships because it provides a common and consistent means of data definition.

XML is a standard and specification for describing data with markup tags. A user defines the set of tags and tag rules to meet their needs. Tags can describe a data or a structure. XML provides additional flexibility because documents are self describing.

Tags allow the creation of such XML dialects as:

- Wireless Markup Language (WML)
- Commerce XML (cXML)
- Electronic Business XML (ebXML)

Each XML document consists of an element that is specific to that document. Figure 49 shows the structure of an XML document. An element with content has a start tag and an end tag, with the content in between the two tags. Elements can be organized into a structure, much like files are organized. The nesting is reflected by the position of the start and end tags. Any element can have an attribute that is used to further define the element.

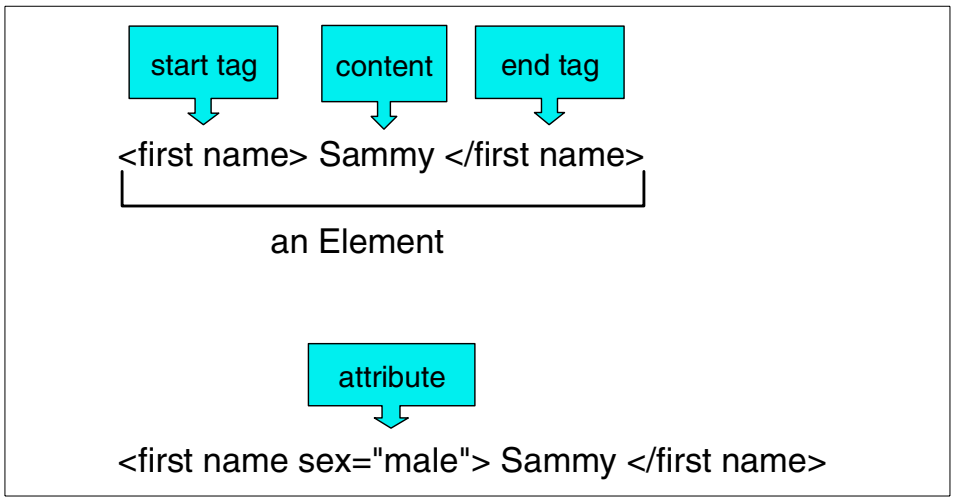


Figure 49. Structure of an XML tag (element)

The following example shows an XML document used for a catalog. The elements have been indented to easily show the hierarchy of the document:

```
<?xml version="1.0"?>
<catalog season="fall">
  <name>Wally's Fall Outdoors Apparel</name>
  <item>
       <itemname>Wool Jacket></itemname>
       <type>
          <typename>Male</typename>
          <cost>$50.00</cost>
          <description>Lite weight Wool Jacket</description>
          <number>4901195M</number>
          <weight unit="pound">1.5</weight>
          <shippingcost/>
       </type>
       <type>
          <typename>Female</typename>
          <cost>$57.50</cost>
          <description>United Lite weight Wool Jacket</description>
          <number>490394W</number>
          <weight unit="pound">1.2</weight>
          <shippingcost/>
       </type>
   </item>
</catalog>
```

The XML document contains information about two wool jackets: a male version and a female version. Within each <type> </type> tag set is the information for each item (jacket). The <shippingcost/> tag does not have an end tag, but rather a back slash (/) after the term shippingcost within the start tag. This is known as an *empty tag*, which has no content and can be used as a place holder for later use.

11.2.5.1 Document Type Definition

The XML document has an associated *Document Type Definition (DTD)* that defined the valid content of the XML document and the relationship between the elements within the document. The DTD defines the structure of the XML document, any rules about the relationship between elements, and any rules that are particular to a specific element. The DTD expresses the hierarchy and the nesting of elements within the document structure.

The following example shows the DTD for the XML document example shown above:

```
<!ELEMENT catalog (name, item*)><ELEMENT name (#PCDATA)>
   <!ATTLIST catalog season (winter|soring|summer|fall) #REQUIRED>
<!ELEMENT item (itemname, type*)>
   <!ELEMENT itemname (#PCDATA)>
   <!ELEMENT type (typename,cost,description,number,weight,shippingcost)>
      <!ELEMENT typename (#PCDATA)>
      <!ELEMENT cost (#PCDATA)>
      <!ELEMENT description (#PCDATA)>
      <!ELEMENT number (#PCDATA)>
      <!ELEMENT weight (#PCDATA)>
         <!ATTLIST weight unit (pound|kilogram|gram|ton) #REQUIRED>
      <!ELEMENT shippingcost (#PCDATA)>
```

An element is defined in this form as <!ELEMENT element name>. An element may contain other sub-elements, which are enclosed in parentheses () and listed after the element name. For example, an element type in the above example has the sub-elements typename, cost, description, number, weight, and shippingcost.

Some elements contain attributes defined in an attribute list <!ATTLIST> statement. For example, element weight has the attributes pound, kilogram, gram, and ton. An attribute is used to add meaning to a particular tag and must be defined within the DTD.

Each element of the document is defined separately for its data type. The #PCDATA (meaning Parsed Character Data) indicating the character (text) data. Currently XML documents only consist of character data.

11.2.5.2 eXtensible Stylesheet Language (XSL)

XML documents can be viewed in XML-enabled browsers by using eXtensible Stylesheet Language (XSL) stylesheets to format the document. Using XML in business-to-business transaction, and as the common communication vehicle between disparate applications, requires a mechanism that will read and interpret the XML document into computer friendly form.

XSL consists of two parts:

- **XSL Transformations (XSLT)**: A language to transform XML documents
- **XML vocabulary**: To specify formatting semantics (XSL Formatting Objects).

Figure 50 shows how the XML document example on page 233 (catalog.xml) looks in the Internet Explorer browser.

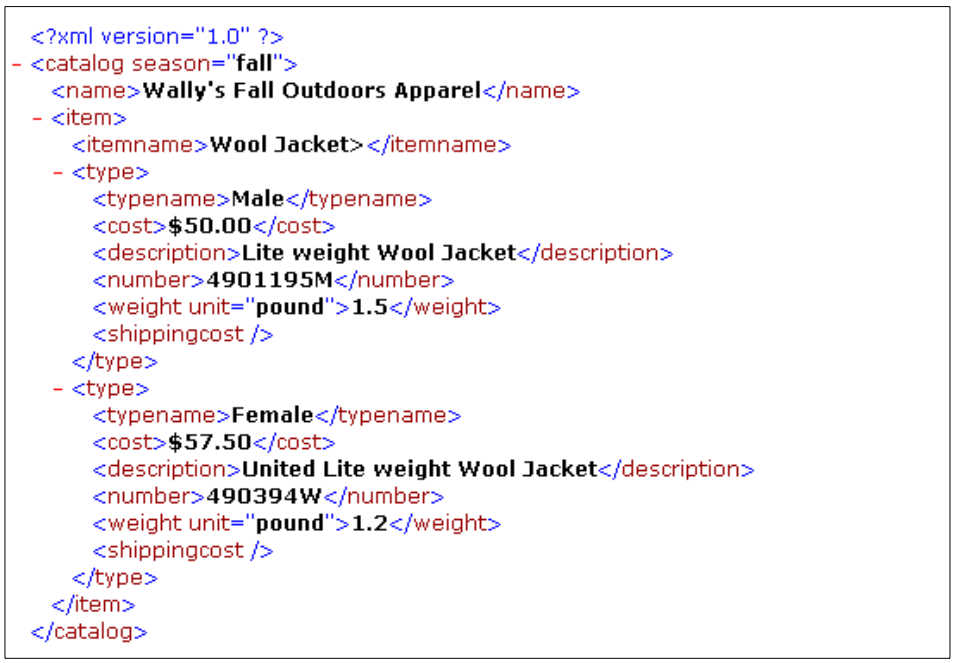


Figure 50. Example: catalog.xml in Internet Explorer

Figure 51 on page 236 shows an example of cXML order request message from the Ariba SupplierLive procurement application.

cXML OrderRequest Message

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM
"http://xml.cXML.org/schemas/cXML/1.1.009/cXML.dtd">
<cXML payloadID="3223232@ariba.acme.com"
    timestamp="1999-03-12T18:39:09-08:00" xml:lang="en-US">
  <Header>
    <From>
      <Credential domain="AribaNetworkUserId">
        <Identity>admin@acme.com</Identity>
      </Credential>
      <Credential domain="AribaNetworkUserId" type="marketplace">
        <Identity>bigadmin@marketplace.org</Identity>
      </Credential>
      <Credential domain="BT">
        <Identity>2323</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="DUNS">
        <Identity>942888711</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="AribaNetworkUserId">
        <Identity>admin@acme.com</Identity>
        <SharedSecret>abracadabra</SharedSecret>
      </Credential>
      <UserAgent>Ariba.com Network V1.0</UserAgent>
    </Sender>
  </Header>
    <Request deploymentMode="test">
      <OrderRequest>
        <OrderRequestHeader orderID="DO1234" orderDate="1999-03-12"
type="new">
          <Total>
            <Money currency="USD">12.34</Money>
          </Total>
          <ShipTo>
            <Address>
              <Name xml:lang="en">Acme</Name>
              <PostalAddress name="foo">
                <DeliverTo>Joe Smith</DeliverTo>
                <DeliverTo>Mailstop M-543</DeliverTo>
                <Street>123 Anystreet</Street>
                <City>Sunnyvale</City>
                <State>CA</State>
                <PostalCode>90489</PostalCode>
                <Country isoCountryCode="US">United States</Country>
              </PostalAddress>
            </Address>
          </ShipTo>
          <BillTo>
            <Address>
              <Name xml:lang="en">Acme</Name>
              <PostalAddress name="foo">
                <Street>123 Anystreet</Street>
                <City>Sunnyvale</City>
                <State>CA</State>
                <PostalCode>90489</PostalCode>
                <Country isoCountryCode="US">United States</Country>
              </PostalAddress>
            </Address>
          </BillTo>
          <Shipping trackingDomain="FedEx" trackingId="1234567890">
            <Money currency="USD">12.34</Money>
            <Description xml:lang="en-us">FedEx 2-day</Description>
          </Shipping>
          <Tax>
            <Money currency="USD">12.34</Money>
            <Description xml:lang="en">foo</Description>
          </Tax>
          <Payment>
            <PCard number="1234567890123456" expiration="1999-03-12"/>
          </Payment>
          <Comments xml:lang="en-US">Anything well formed in XML can go
here.</Comments>
        </OrderRequestHeader>
```

Figure 51. Example: cXML document from Ariba SupplierLive

11.2.5.3 XML parser

Application programs require a means to access the individual pieces of information (elements) contained within each XML document. This is accomplished by using an XML parser to render the document in a structured form (hierarchical tree structure), which allows each element of the document to be accessed and manipulated.

The XML parser uses the DTD to validate the document, which involves ensuring the XML document follows all the rules specified in the DTD. For example, the DTD rules can specify the valid set of tags, the valid element nesting rules, and the attributes that are associated with a particular element.

11.2.5.4 Document Object Model (DOM) tree

An XML parser uses both the DTD and the XML document to create a Document Object Model (DOM) tree, which presents the document hierarchically. The DOM is a representation of the document used by applications at run time to query and update information within an XML

document. The DOM provides a group of APIs that allows access to the elements within the tree. When you use DOM APIs, you can access, change, delete, or add any element within the XML document.

Figure 52 shows an example of the DOM tree for the document from XML document example on page 233 (catalog.xml). It does not represent the content within the tree structure. The tree must be visualized as a grove or forest of trees representing that content rather than this single structure. Each rectangle in following diagram represents a node in the tree, and each oval represents an attribute. To keep Figure 52 simple, only the element and attribute names are included.

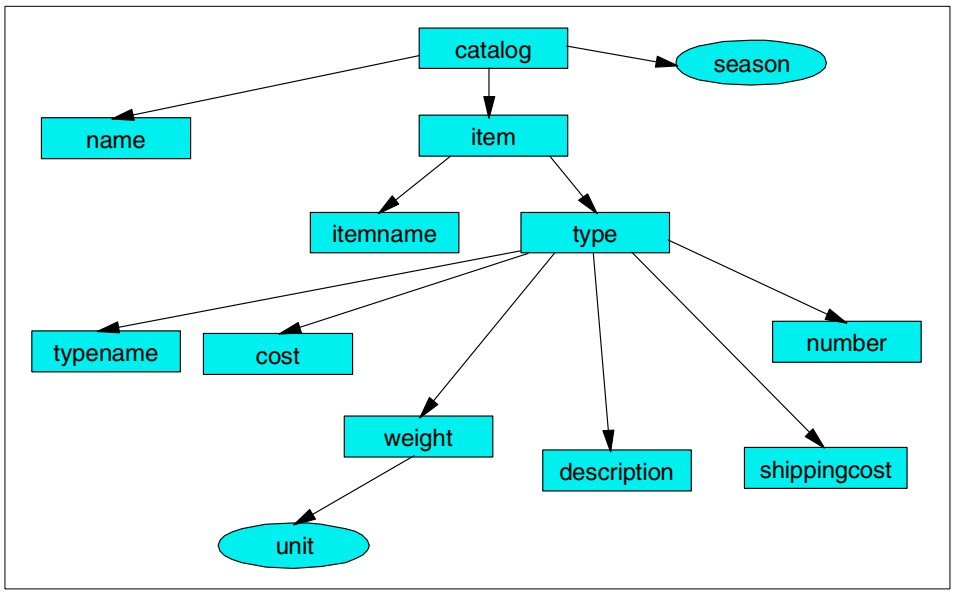


Figure 52. Example: Document Object Model tree for catalog.xml

By following the tree (hierarchical) structure, the methods allow traversing the tree using method calls for the parents and children. The DOM provides methods to traverse the tree created by the parser. The method accesses the elements within the tree using the parent-child relationship.

11.2.5.5 Java XML parser (XML4J)

IBM provides the XML4J parser in various products including the AS/400 ToolBox for Java and on the IBM alphaWorks site (`http://www.alphaworks.ibm.com/`). Because this parser is written in Java, it is portable to operating systems with a Java Virtual Machine (JVM).

The XML4J parser allows you to access the information contained within an XML document.

11.2.5.6 IBM XML parser for C++ (XML4C)

XML4C is based on Apache's Xerces-C XML parser, which is a validating XML parser written is a portable subset of C++. XML4C integrates the Xerces-C parser with IBM International Components for Unicode (ICU) and extends the number of encoding supported to over 150.

It consists of three shared libraries (two code libraries and one data library) that provide classes for parsing, generating, manipulating, and validating XML documents.

11.2.5.7 SAX parser

Often iSeries applications are not equipped to process XML documents. There are two solutions to this problem:

- Change the application. This method is cumbersome, time consuming and, in many cases, simply not a viable alternative.
- Transform the XML data into an acceptable format that can be read by the application.

Data queues are often used by conventional iSeries applications for business transactional data. Therefore, converting XML data into a data queue format is a logical method of causing a traditional iSeries program to interface with and use an XML document.

The SAX specification was put together by numbers of xml-dev mailing lists. SAX is a defacto standard for interfacing with the XML parser, even though it is not a World Wide Web Consortium (W3C) recommendation. The SAX parser implements event-driven APIs for parsing XML documents, which means SAX provides access to the XML document structure and contents via events.

The following example contains a portion of the AddressBook.xml document for the entry "James Edward Washerton". The example includes the XML tags that show both the structure of an address entry and the address content.

```
_________Document_____________________________     _____________Events_________________
<?xml version="1.0" encoding="UTF-8"?>           startDocument
<AddressBook>                                    startElement
<AddressEntry>                                   startElement
<Name title="Mr.">                               startElement
<FirstName>James</FirstName>                     startElement characters
                                                 endElement
<MiddleName>Edward</MiddleName>                   startElement characters
                                                 endElement
<LastName>Washerton</LastName>                   startElement characters
```

```
                                                    endElement
</Name>                                             endElement
<Address>                                           startElement
<PostalAddress type="home">                          startElement
<Street>123 Main Street</Street>                     startElement characters
                                                    endElement
<City>Our Town</City>                                startElement characters
                                                    endElement
<State>MN</State>                                    startElement characters
                                                    endElement
<PostalCode>55489</PostalCode>                       startElement characters
                                                    endElement
<Country>US</Country>                                startElement characters
                                                    endElement
</PostalAddress>                                    endElement
<eMailAddress jewash@aol.com</eMailAddress>          startElement characters
                                                    endElement
<Phone type="home"> 507-334-6565</Phone>             startElement characters
                                                    endElement
<Phone type="work"> 507-334-6688</Phone>             startElement characters
                                                    endElement
</Address>                                          endElement
</AddressEntry>                                     endElement
</AddressBook>                                      endElement endDocument
```

While parsing this document, the SAX parser generates an event for each node that is encountered (shown in the above example in the right-hand column). Each event can be captured by the application program invoking the parser. The application program implements event handlers to process these parsing events. Conveniently, these event handler methods can be extended to perform actions appropriate to a particular parsing event.

The application implements the DocumentHandler and ErrorHandler. The DocumentHandler, the primary event handler interface, “listens” for the various events created by the parser. The ErrorHandler handles the errors generated during the parsing process.

The important difference between SAX and DOM parsers is that each of these parsers supports the two major types of XMI APIs:

- SAX for event-based APIs
- DOM for hierarchical or tree-based APIs

The SAX parser does not build an internal tree. While processing the XML document, it reports parsing events to the application through callbacks. When using the SAX parser, the application must be written (or modified) to implement handlers that deal with the different parsing events.

The DOM parsers represent the XML document in an internal tree (hierarchical) structure. The DOM APIs are used by the application to navigate within the tree. These APIs allow all forms of manipulation of the tree content, including additions, deletions, and updates.

Figure 53 shows input to and output from the XMLtoDQSAX class. It shows the transformation of an XML document into messages on a data queue that can be processed by an iSeries program. It does not show the iSeries program that would process the data queue messages.

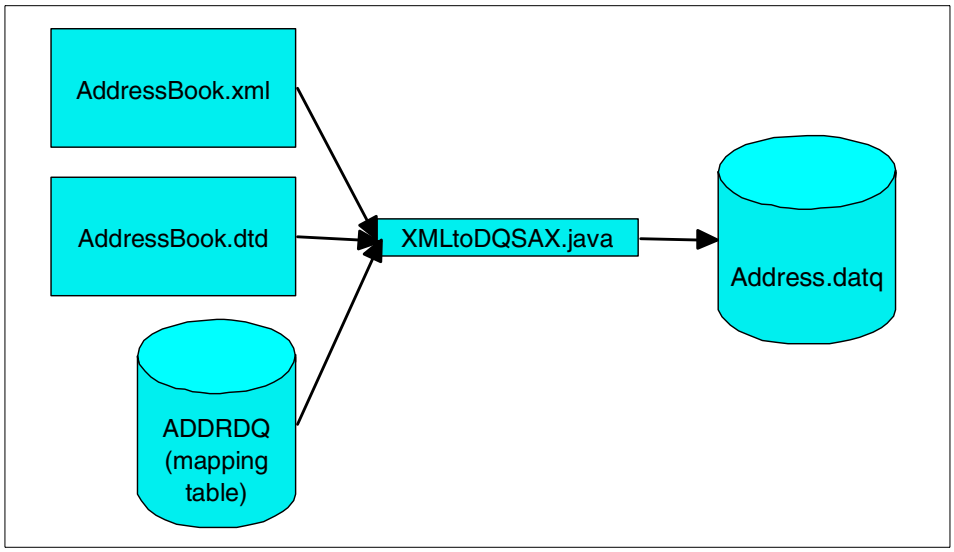


Figure 53. Input and outputs to the XMLtoDQSAX class

The XMLtoDQSAX class processes AddressBook.xml using these steps:

1. It validates the input XML document using the AddressBook.dtd DTD.
2. It maps the XML document contents by element name to the data queue field names using the ADDRDQ DB2 table. The ADDRDQ table defines the cross reference (mapping) between the elements and attributes in the XML document and the fields within the data queue message format.
3. It creates complete messages on Address.datq.

11.2.5.8 DB2 XML Extenders

In the future with DB2 XML Extenders, the iSeries developer will be able to map XML document details directly into DB2 UDB for AS/400 tables. Meanwhile, the currently provided Java class (insertXMLintoDB2.java class) can be used. This is accomplished by using the XML4J parser and key Document Object Model (DOM) APIs that are incorporated into the insertXMLintoDB2.java class.

Figure 54 shows the input to and results from insertXMLintoDB2.java. It shows both the catalog.xml document and the catalog.dtd document as input.

It also shows access to a DB2 UDB for AS/400 table named MAPCATALOG. Each catalog is processed to create an insert statement. A complete insert statement is used to update the table CATALOG.

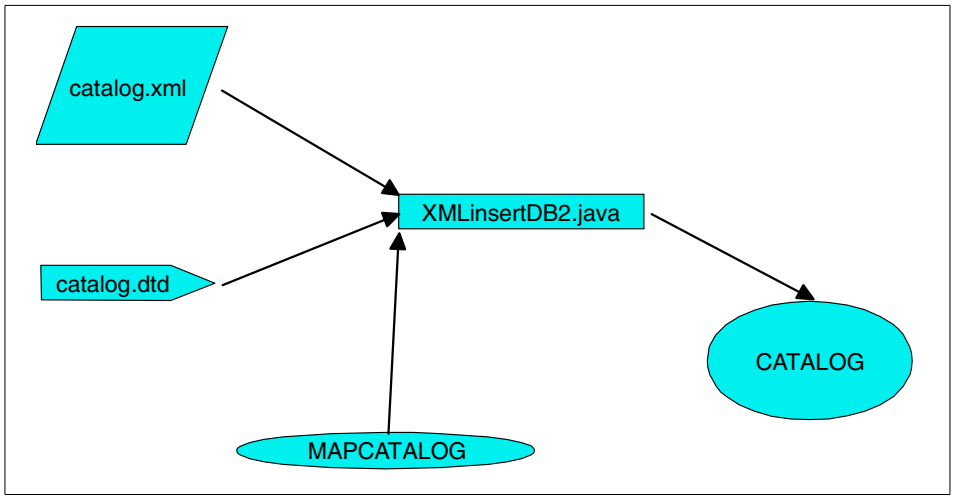


Figure 54. Input and results used in insertXMLintoDB2.java

The MAPCATALOG table provides the cross reference between the insert statement and the XML document. It determines whether an element from the XML document is included in the insert statement and which field (by position) that content is to replace.

11.2.5.9 Lotus XSL Transformer

Lotus provides an XSL transformer, which can be used to translate an XML document into HTML. There are several ways XML can be included in a designer application and serve the data to an XML parser.

You can enter XML tags that describe data on a form or a page. By treating the contents of the form or a page as HTML, you can serve XML to an XML parser that can interpret the tags. XML describes the data being presented. To format and style the data on the form or page, a stylesheet is used that is created with XSL to transform the data into HTML.

11.2.5.10 tpaML

This specification defines the language for creating electronic Trading Partner Agreements (TPAs) between two business partners (parties to the TPA). Like the Trading Partner Agreements used in electronic data interchange (EDI), these TPAs define the “information technology terms and conditions” that enable business documents to be electronically interchanged between

partners. However, these TPAs are not paper documents. Rather, they are electronic documents, written in XML, which can be processed by computers at the partners' sites to set up and then execute the desired business information exchanges.

A TPA describes all the valid visible, and therefore enforceable, interactions between the parties. It is independent of the internal business processes of each party. Each party builds its own internal business process to satisfy these external TPAs and interface them to the rest of its business processes. However, the internal business processes are, in general, not visible to other parties (unless desired by the service providers themselves). The intent is to provide a high-level specification that can be easily comprehended by humans and yet is precise enough for enforcement by computers.

Information in the TPA includes:

- **Identification**: Identifies uniquely the TPA document and the parties involved.
- **Communication**: Specifies the transport protocol or protocols and electronic addresses of the parties.
- **Security**: Defines the certificates used for authentication, non-repudiation, digital envelope, and other security parameters.
- **Invocation-independent properties**: Specify overall properties of the TPA, for example, the valid duration of the TPA.
- **Data definition**: Describes the formats of the data being passed around.
- **Role definition**: Describes each of the roles specified in the TPA that can be filled by specific parties.
- **Action list**: Describes the requests each party can issue to the other. These actions are the independent units of work. The action definitions define the associated message flows between the invoker and the service provider, responsiveness, failure handling, and other attributes.
- **Sequencing rules**: Describe the valid action invocation sequences in each party.
- **Global properties**: Describe the default properties of various fields in the TPA, for example, responsiveness.
- **Comments**: Describe the handling of disputes, termination of the TPA as a whole, and other exceptional conditions.

11.2.5.11 PCML

Program Call Markup Language (PCML) is a tag language that helps you call iSeries programs, with less Java code. PCML is based on XML, a tag syntax you use to describe the input and output parameters for iSeries programs. PCML uses a program-call-document framework designed to make it easier for the programmer to call an iSeries program from a Java program. PCML enables you to defined tags that fully describe iSeries programs called by a Java application.

PCML allows you to write less code. Ordinarily, extra code is needed to connect, retrieve, and translate data between an iSeries server and Java Toolbox objects. However, by using PCML, your calls to the iSeries with the AS/400 Toolbox for Java classes are automatically handled. PCML class objects are generated from PCML tags and help minimize the amount of code you need to write to call iSeries programs from your Web application.

When used in conjunction with the ProgramCallDocument class delivered in the AS/400 ToolBox for Java, PCML fully describes all parameters, structures, and field relationships for a given iSeries program call. The AS/400 Toolbox for Java support interprets the PCML, calls the iSeries program, and allows the programmer to use any data returned from the called iSeries program.

Although, PCML was designed to support distributed program calls to iSeries program objects from a Java platform, you can also use it to make calls to an iSeries program from within an iSeries environment. Figure 55 on page 244 shows how to make program calls to the iSeries using PCML.

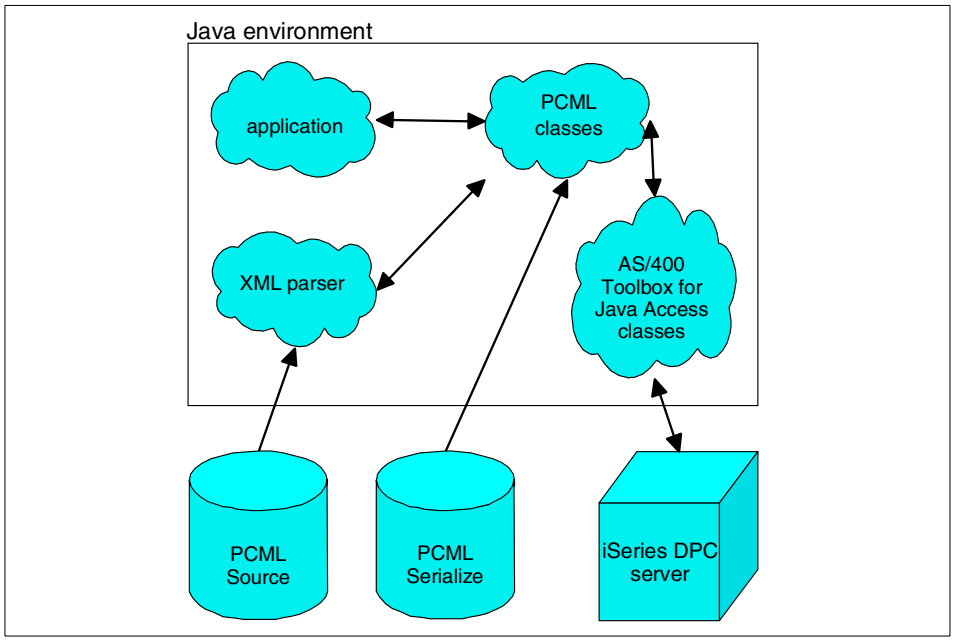


Figure 55. Making program calls to an iSeries program using PCML

When your Web application constructs the ProgramCallDocument object, the IBM XML parser reads and parses the PCML source file. After the ProgramCallDocument class is created, the application program uses the ProgramCallDocument class methods to retrieve the necessary information from the iSeries through the iSeries distributed program call (DPC) server.

To increase run-time performance, the ProgramCallDocument class can be serialized during your product build time. The ProgramCallDocument is then constructed using the serialized file. In this case, the IBM XML parser is not used at run time.

11.2.5.12 PDML

Panel Definition Markup Language (PDML) is a new user interface (UI) data representation language that is based on XML to define a platform independent language for describing the layout of GUI elements. Panels can be defined by using the Graphical Toolbox, which is included in the AS/400 Toolbox for Java Modification 2. The Graphical Toolbox also provides a run-time API to display the panels. The API displays panels by interpreting the PDML tags and rendering the user interface as Java Foundation Classes.

Here are some examples of PDML tags:

<panel>	Defines a panel
<title>	Specifies the title of the panel or field
<size>	Specifies the size of the panel or field
<label>	Defines a label on the panel (static text field)
<location>	Specifies the location of the field on the panel
<button>	Defines a button on the panel
<textfield>	Defines a text field on the panel

Some of the tags used in PDML are:

- **<pdml></pdml>**
  Used to identify the start and end of a PDML definition.
- **<panel name=mypanel1> and </panel>**
  Used to identify the start and end of a panel called mypanel1.
- **<title>xxx</title>**
  Sets the title used when displaying a panel. If the panel is the only content of a window, then this is the window title.
- **<label name=label_1>Name</label>**
  Defines a text label (output only) to show on the display.

Graphical Toolbox

The Graphical Toolbox, a set of UI tools, makes it easy to create custom user interface panels in Java. You can incorporate the panels into your Java applications, applets, or Operations Navigator plug-ins. The panels may contain data obtained from the iSeries server or data obtained from another source, such as a file in the local file system or a program on the network.

The GUI Builder is a “what you see is what you get” (WYSIWYG) visual editor for creating Java dialogs, property sheets, and wizards. With the GUI Builder, you can add, arrange, or edit user interface controls on a panel, and then preview the panel to verify that the layout behaves the way you expected. The panel definitions you create can be used in dialogs, inserted within property sheets and wizards, or arranged into splitter, deck, and tabbed panes. The GUI Builder also allows you to build menu bars, toolbars, and context menu definitions.

The Resource Script Converter converts Windows resource scripts into an XML representation that is usable by Java programs. With the Resource Script Converter, you can process Windows resource scripts (RC files) from your existing Windows dialogs and menus. These converted files can then be edited with the GUI Builder. Property sheets and wizards can be made from RC files using the resource script converter along with the GUI Builder.

Both the Resource Script Converter and the GUI Builder rely on PDML as the means to represent the user interface layout. Once your panels are defined in PDML, you can use the run-time API provided by the Graphical Toolbox to display them. The API displays your panels by interpreting the PDML and rendering your user interface using the Java Foundation Classes.

11.2.5.13 Pervasive computing

Pervasive computing devices (PvC) are one of the latest types of devices to emerge on the scene. These devices, ranging from cell phones to WebTV, can be used to conduct business by connecting via the Internet or a private network to business applications running on an iSeries server.

XML plays an important role in this new mobile environment. XML, which is used to define portable data, provides both the basis for the various XML dialects used by these devices and the tagged business data displayed on these devices. The transformation of the business data (represented by a dialect) into a form tuned to the device is accomplished by using XSL and a stylesheet transformer, like LotusXSL from IBM.

The viewing space dictates the amount of information that can sensibly be presented on these devices. For example, cell phones have under two square inches of viewing space, and a palm device has under four square inches of viewing space. The different input mechanisms affect the dialogue structure and limit the way responses are made. Many of these devices support a subset of HTML with their owns tags mixed in. Cell phone manufacturers use Wireless Markup Language (WML) as the standard markup language. WML is a dialect that contains some HTML and some unique tags. A WML document is composed of one or more cards to carry out the dialog with the cell phone user. A card would equate to an application display screen.

Some Wireless Application Protocol (WAP) gateway vendor products can dynamically format existing Web page content (HTML and graphics) into WML, which is then compiled into a compressed byte stream that is sent from the WAP gateway to the phone device's micro browser. Similarly, these products enable user input from a WML browser by translating from the compiled WML sent to the gateway to WML and sending it to the Web server resources. Additionally, products such as IBM WebSphere Transcoding Publisher cannot only do this for WML devices. However, they can customize HTML and graphics content for devices such as Palm Pilots.

11.2.6 Net.Data

Net.Data simplifies the writing of interactive Web applications through macros. Using Net.Data macros, you can execute programming logic, access

and manipulate variables, call functions, and use report-generating tools. Net.Data processes the macro to produce output that is displayed by a Web browser. Macros combine the simplicity of HTML with the dynamic functionality of Web server programs, making it easy to add live data to static Web pages.

The Web server invokes Net.Data as a CGI application when it receives a URL that requests Net.Data services. The URL includes Net.Data-specific information, including the macro that is to be processed. When Net.Data finishes processing the request, it sends the resulting Web page to the Web server. The server passes it on to the Web client, where it is displayed by the browser.

Net.Data supplies language environments that can be used to access data sources and to execute your legacy or ERP application containing business logic. As an example, you can use the SQL language environment to both read from and update information in your DB2 database.

With Net.Data, you can add a user-written language environment in a pluggable fashion. Each user-written language environment must support a standard set of interfaces that are defined by Net.Data and must be implemented as a service program.

11.2.7 Common Gateway Interface (CGI)

The Common Gateway Interface (CGI) specification defines how the server is expected to call external programs and how those programs should return the document that they produce back to the HTTP server. CGI is a standard for interfacing external applications with information servers, such as HTTP servers or Web servers. The current version of CGI is 1.1.

CGI support allows you to build interactive forms for use on the Web. You can write a program in C, RPG, COBOL, or Java that can accept and interpret input from the form, perform an action, and return the resultant output to the client in HTML.

A programmer creates a gateway program that is specifically coded to handle the parameters (usually sent as name-value pairs). The fields used to query and send data back to the iSeries CGI application are hidden to the user as part of the HTML of a Web page link's FORM or ISINDEX query box. In response to the user request, the gateway program executes and returns the results to the client in the form of more HTML.

11.2.8 CORBA

Common Object Request Broker Architecture (CORBA) encompasses a series of standards and protocols for interprocess communication in a heterogeneous environment. Using CORBA, developers can easily write applications for many different operating systems at once, in any number of languages. It has become a standard inclusion along the Apache Web server in software packages such as IBM WebSphere.

CORBA is an architectural standard proposed by the Object Management Group (OMG). This is an industry standard organization. This standard allows you to create object descriptions that are portable among programming languages and execution platforms.

CORBA encompasses a series of standards and protocols for interprocess communication in a heterogeneous environment. With CORBA, developers can easily connect processes running on different machines, with different operating systems, and with code written in different languages. The CORBA specification has quickly caught on as a standard method for interprocess communication.

The CORBA specification only defines a set of conventions and protocols that must be followed by CORBA implementations. It is left to vendors and developers to translate this specification into a working implementation. CORBA does not make any restrictions on language or on the underlying operating system. Because of this, implementations of the CORBA specification have been created for a wide variety of operating systems including UNIX, Windows, and OS/400, and for many languages, including C, C++, Java, Ada, LISP, Python, and even COBOL. Any CORBA implementation that matches the defined interfaces and adheres to the defined protocols is allowed to communicate with other CORBA implementations.

Objects, ORBs, POAs, BOAs, and IORs

Objects are at the heart of every CORBA application. Objects reside on various machines throughout the distributed environment and are tasked with performing duties defined by their implementation. In the standard two-tier architecture, the objects are often thought of as the servers in the system. However, unlike standard servers, objects have the ability to move around if needed. A client communicates to an object through an object reference. This is a pointer to the object that allows requests for operations and data access to be sent from the client to the server via an Object Request Broker (ORB). An ORB is best thought of as the traffic cop in the system. It knows whether

requests should be routed to implementations contained within itself or to another ORB running on another machine.

Every object on the ORB must have an implementation. This implementation is code written to perform tasks on the server machine. In other words, the implementation is what does the actual work of the object. An implementation can be in any language. It is allowed to perform tasks supported by the language, operating system, and underlying hardware. Examples of such implementations are database adapters and code designed to interface with a legacy library.

When a request reaches the ORB for which it is intended, the request is passed to an object adapter. The portable object adapter (POA) and its predecessor, the basic object adapter (BOA), form a link between an object's implementation and its presence on the ORB. During the creation of an object on the ORB, the developer must specifically link a newly created object reference and its implementation. The object adapter then informs the ORB that it wants any requests for the new object reference to be routed to it. With object adapters, the ORB is no longer burdened with the task of keeping track of object implementations.

Every object on the ORB has an Interoperable Object Reference (IOR). An IOR is a global identifier string that identifies the machine on which its associated object is located and the interface that the object supports. If given the IOR for an object, a client can use standard function calls on the ORB to turn it into an object reference. With the information contained in the IOR, the ORB knows what type of object is being referenced and the machine to which all requests should be routed. The simplest way for a client to access the IOR of a server object on the ORB would be for the server object to write its IOR to a file. The client could then read the IOR from this file and have the ORB resolve it into an object reference.

11.3 Connect for iSeries: Pulling them all together

Connect for iSeries is targeted for sellers of goods and services that suddenly realize that their buyers are now in control of what trading protocols will be used for the exchange of goods. Buyers, in an attempt to streamline their purchasing process and reduce their expenses, purchase new procurement software from companies like Ariba or Metiom. This software has the ability to communicate to multiple seller organizations using XML-based protocols such as Commerce XML (cXML). The buyer is motivated to do business with any sellers that support this new protocol because it helps the buyer to further reduce their expenses.

The seller must support these new protocols or be at a competitive disadvantage to those sellers who support the protocols. These protocols represent new ways of receiving and responding to orders from their buyers. For maximum efficiency, the seller must be able to tie these new orders into existing order processing software.

11.3.1 The Connect for iSeries solution

One of the goals of the Connect for iSeries solution is to provide a highly configurable or pluggable architecture that is extremely easy to use, yet easy to extend. This extensible framework is supported by a series of graphical tools that allow the product to be installed and configured, and then customized solutions to be developed, deployed, and managed. Figure 56 shows a very high level overview of the Connect for iSeries product.

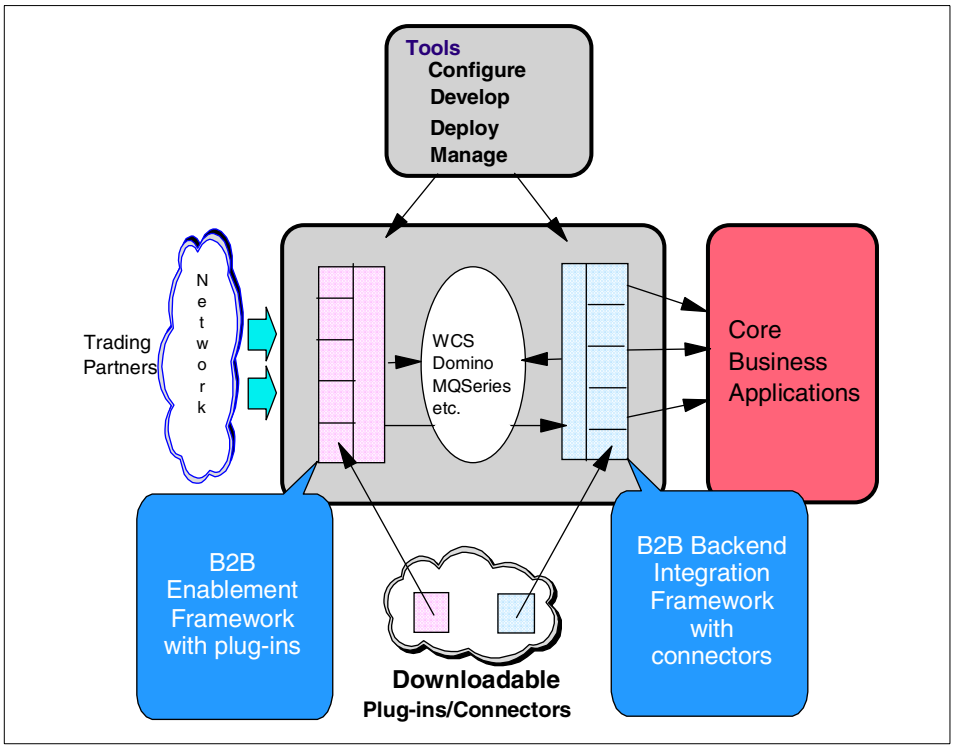


Figure 56. High-level overview of the Connect for iSeries product

A front-end B2B enablement framework handles the interfacing with the various trading partners over a variety of connectivity mechanisms and protocols. A collection of plug-ins is provided or can be downloaded from the Web that supports a multitude of mechanisms used by buyers and

e-marketplaces to submit B2B requests such as order placement, order status checks, and catalog maintenance.

A back-end integration framework deals primarily with the processing of the B2B requests by tying them into existing ERP, supply chain, and other core business applications. A collection of connector types is provided or can be downloaded from the Web that supports three popular interfacing mechanisms: program calls, queues, and Java methods. For each connector type, one or more connector instances can be created. A connector instance is described by an Application Connector Document (ACD), which describes the interface and the data to be passed to the application. This ACD is unique for each application being accessed.

Building an ACD does not involve writing code. However, adequate knowledge of the interface to the application being accessed is required. ACDs may be provided for popular core business applications either by IBM or its business partners. Custom ACDs can be built for other applications using tools provided with the product.

Connect for iSeries makes use of other middleware including WebSphere Application Server, WebSphere Commerce Suite, MQSeries, and Domino to process some of the B2B transactions. Usage of these middleware tools depends on the types of transactions being processed and their availability in the customer's environment. The configuration tool is used to install the appropriate plug-ins and connectors that enable these products.

Connect for iSeries is built using industry standard Java and XML. It is written in 100% Java for ease of development and portability. A messaging interface is used to transfer XML documents between components of Connect for iSeries. Enterprise JavaBeans are not currently used in this product.

11.3.2 Run-time architecture

The concept behind Connect for iSeries is relatively simple as shown in Figure 57 on page 252.

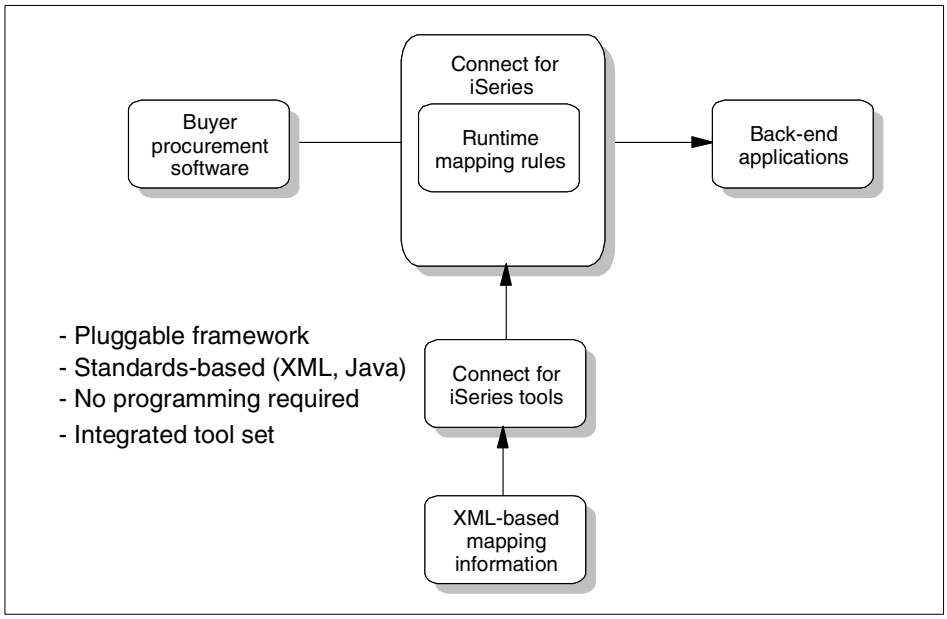


Figure 57. Connect for iSeries concept

Connect for iSeries receives messages from your buyers' procurement software and maps those messages into your existing back-end applications. The rules for mapping these messages is provided as a series of XML documents that can be created by the Connect for iSeries tools. No special programming is required. Connect for iSeries is implemented as a pluggable framework that makes it easy to add additional trading partner protocols in the future.

Version 1.0 of Connect for iSeries is written to handle two trading partner protocols: cXML from Ariba and mXML from Metiom.

11.3.3 Architecture of Connect for iSeries Version 1.0

The primary run-time components of Connect for iSeries are the Front-end Delivery Gateway and the Back-end Flow Manager. These two components accept incoming requests from trading partners (represented by the Buyer block in Figure 58), map those requests into back-end applications, and return an appropriate response to the trading partner.

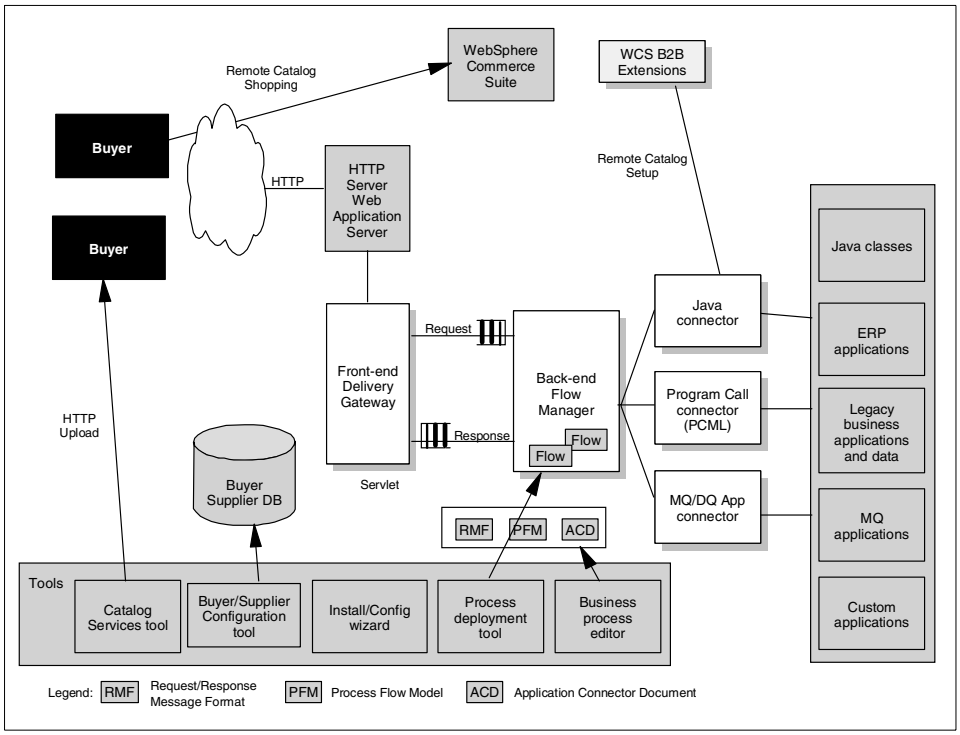


Figure 58. Architecture of Connect for iSeries Version 1.0

In Version 1.0, the front-end and back-end components must reside on the same system. However, in future versions, they will be able to run on separate systems. This is possible because the front end and back end use MQSeries request and response queues to communicate to each other.

The front-end delivery gateway is implemented as a Java servlet. As such, it requires the services of an application servlet engine in order to operate. Two choices for servlet engines are available:

- WebSphere Application Server Standard or Advanced Edition can be used in combination with the IBM HTTP Server to provide hosting for Java servlets and HTTP protocol handling respectively.
- Domino has a built-in servlet engine and an HTTP server that can also be used to host Java servlets.

Requests from Ariba or Metiom marketplaces are sent via HTTP to the front-end delivery gateway servlet. The front-end delivery gateway is responsible for examining and validating these requests. Any authentication

and authorization information associated with the particular request is validated. Some requests will need to be transformed to a normalized format consumable by the back-end flow manager. Some requests can be passed exactly as they are received. All requests will be preceded by a common header agreed to by the front-end and back-end components.

Once the request is validated, authenticated, and transformed, it is placed on an MQSeries queue for processing by the back-end flow manager. The front end waits for an associated response from the back end on an MQSeries response queue.

The back-end flow manager receives the request from the front end and is responsible for transforming these requests into something that an existing back-end application can process. This transformation is done based on a set of rules supplied by the customer. These rules are specific to the customer's installation.

The transformation rules are in the form of a Process Flow Model (PFM), which can be built using the supplied Process Flow Creation Tool.

In Connect for iSeries Version 1.0, only a single step is allowed to be specified in a PFM flow. Subsequent versions will allow multiple steps to be specified and, therefore, allow multiple business processes or applications to be processed from a single incoming request.

There are various ways that back-end applications receive requests to process transactions. In Connect for iSeries Version 1.0, three ways are supported:

- Direct program calls of ILE applications
- Java method calls
- Requests sent via a queuing mechanism (MQSeries queues or iSeries data queues)

Additional connector types will be provided in subsequent versions of Connect for iSeries.

In order for the back-end flow manager to communicate to an application using one of these connector types, it must be provided with the format of the messages and parameters that the application expects. This description is provided by the customer or application provider in the form of an Application Connector Document (ACD). An ACD is an XML document that adheres to the DTD defined for the appropriate connector type. There are ACD DTDs for each of the three supported connector types. An ACD for the Java connector and the program call connector indicates the name of the class or program to

be called and a description of the parameters it expects to be passed. An ACD for the queue connector indicates the name of the queue to use and the format of the message to be placed on that queue.

11.3.4 Application Connector Type (ACT)

An Application Connector Type (ACT) is an XML DTD, which serves as a template for capturing information that defines a particular instance of a B2B connector. Each type of B2B connector supported by Connect for iSeries has a corresponding ACT, which defines the properties and input/output data formats supported by the particular connector type. In Version 1.0, ACTs will be shipped with the Connect for iSeries product; user-defined ACTs will not be supported in this release. ACTs are used by the Business Process Editor tool to determine the set of properties and input/output message formats to prompt for when creating and editing the definition of a connector instance.

Users of the tools provided with Connect for iSeries will not have to modify or even be aware of the ACTs shipped with the product. The information captured within an ACT will be exposed during editing and creation of an ACD by the Business Process Editor tool.

11.3.5 Application Connector Document (ACD)

An Application Connector Document (ACD) is an XML document that defines the properties and input and output parameters supported by a given connector instance. An ACD is associated with an ACT that defines the named properties and input/output message formats supported by the given type of connector. While there will be one ACT per supported connector type, there can be multiple ACDs – one for each connector instance. ACDs are created and updated using the Business Process Editor tool and are referenced by one or more Process Flow Models, which define a business process that will handle one or more types of B2B requests. ACDs also serve as a means to describe the interface to customer or ISV-written business solutions that want to integrate with the Connect for iSeries product.

11.3.6 Process Flow Model (PFM)

A Process Flow Model (PFM) is an XML document that defines the business process that will be used to handle a particular B2B request. In Version 1.0, business processes will be restricted to a single connector instance invocation. In future releases of Connect for iSeries, PFMs will be extended to allow multiple connectors to be invoked in response to a B2B request and will provide support for more advanced control flow constructs (for example: if-then-else, case, iteration, and so forth). In addition to defining the particular

connector instance used to handle a B2B request, PFMs also define field mappings between:

- The request message provided with the B2B request and the input parameters supported by the connector instance
- The output parameters returned by the connector instance and the response message defined for the particular B2B request

PFMs are created and updated using the Business Process Editor tool. PFMs can be used to describe how a given customer or ISV solution can be used to handle a particular type of B2B request and can serve as the vehicle for integrating these solutions with the Connect for iSeries product.

11.3.7 Protocol Message Format (PMF)

A Protocol Message Format (PMF) is an XML document that identifies the input and output fields associated with a given B2B request. Each type of B2B request supported by Connect for iSeries has a corresponding PMF that defines the request and response messages associated with the B2B request that can be mapped to or from a connector instance by a PFM. For XML-based message formats, the PMF identifies the XML DTD or schema associated with the request type and the subset of fields defined by the DTD or schema that are applicable for B2B requests of this type. The PMF also allows alias names to be associated with elements in the request or response message for the B2B request. This enables more meaningful and intuitive names to be used when mapping the B2B request fields to a connector instance.

A fixed set of PMFs will be shipped with the Connect for iSeries product in Version 1.0 – one PMF for each B2B protocol or transaction supported. No support will be provided for user-defined PMFs in Version 1.0. However, this is considered a potential requirement for a future release. Users of the tools provided with Connect for iSeries will not have to modify or even be aware of the PMFs shipped with the product. The information captured within a PMF will be exposed during editing or creation of a PFM by the Business Process Editor tool.

11.3.8 Business Process Editor tool

The Business Process Editor tool is a Java GUI tool used to create and update information associated with connector instances (for example ACDs) and to create and update business process flow information designed to handle particular B2B requests (for example PFMs). The Business Process Editor runs on either a Windows NT or Windows 2000 client and can be used

while connected via a TCP/IP network to the iSeries host that supports a B2B instance or in standalone mode where no connection exists to the host containing the Connect for iSeries run-time environment. A publish operation is supported by the Business Process Editor to transfer complete ACD or PFM definitions to an iSeries host where it can be deployed to a B2B instance running on that host.

The Business Process Editor is presented as a single GUI panel (Figure 59) with separate tabs used to navigate from Application Connector edit mode to Process Flow edit mode.

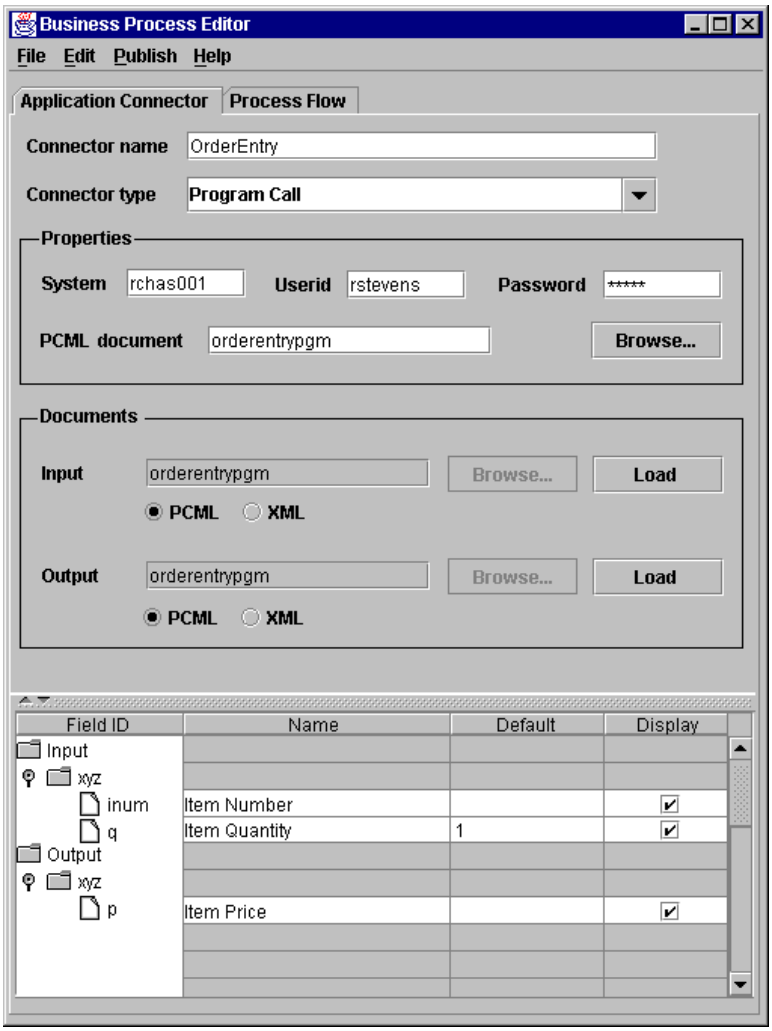


Figure 59. Business Process Editor - Application Connector

The Application Connector tab is used to capture information required to define a connector instance, including the properties and input/output fields supported by the connector instance.

The Process Flow edit tab (Figure 60) describes the process flow model. It includes the type of B2B request to be serviced and the connector instance to be invoked to handle the request. It also includes information about how data provided with the B2B request is mapped to input fields supported by the connector instance and how data returned by the connector instance is mapped to the response data format supported by a particular B2B request.

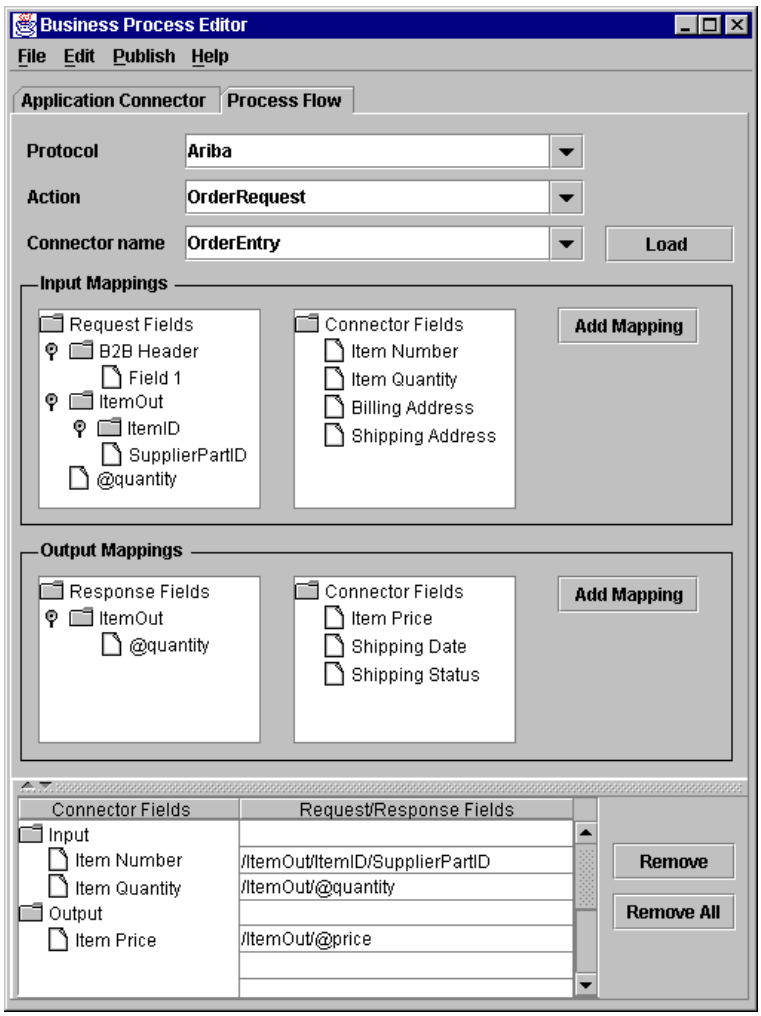


Figure 60. Business Process Editor - Process Flow

11.3.9 Process Deployment Tool

The Process Deployment Tool is a Web-based interface used to deploy a complete business process flow (PFM) and associated connector instance definition (ACD) into the run-time environment supporting a particular Connect for iSeries instance. The Process Deployment Tool runs within the context of a given Connect for iSeries instance. It allows the business process flows defined for that instance to be updated or extended. In essence, the Process Deployment Tool converts one or more PFM/ACD pairs into the run-time meta data format required by the Flow Manager run time responsible for handling incoming B2B requests within a given Connect for iSeries instance.

The process of deployment includes associating a particular business process flow to the type of B2B request the flow is to handle. B2B requests can be qualified by B2B marketplace, B2B protocol, transaction type and subtype, buying organization, and supplier. Therefore, the act of deployment involves:

- User selection of values for all of B2B request qualifiers
- User identification of the business process flow that will handle these requests
- Generation of run-time meta data representing these selections
- Storage of this information in the run-time environment associated with the B2B instance

These four steps features are supported by the Process Deployment Tool provided with Connect for iSeries.

The Process Deployment Tool is a Web-based interface (Figure 61 on page 260) used to deploy a defined business process flow and associated connector instance definition into the run-time environment associated with a particular B2B instance where it will be used by the Connect for iSeries flow manager run time to service B2B requests. A key task in the deployment process involves the association of a particular business process flow with the B2B request that the flow is designed to service. Since B2B requests are identified by B2B protocol, marketplace, transaction type/subtype, buying organization and supplier, all of these qualifiers can be specified when deploying a business process flow.

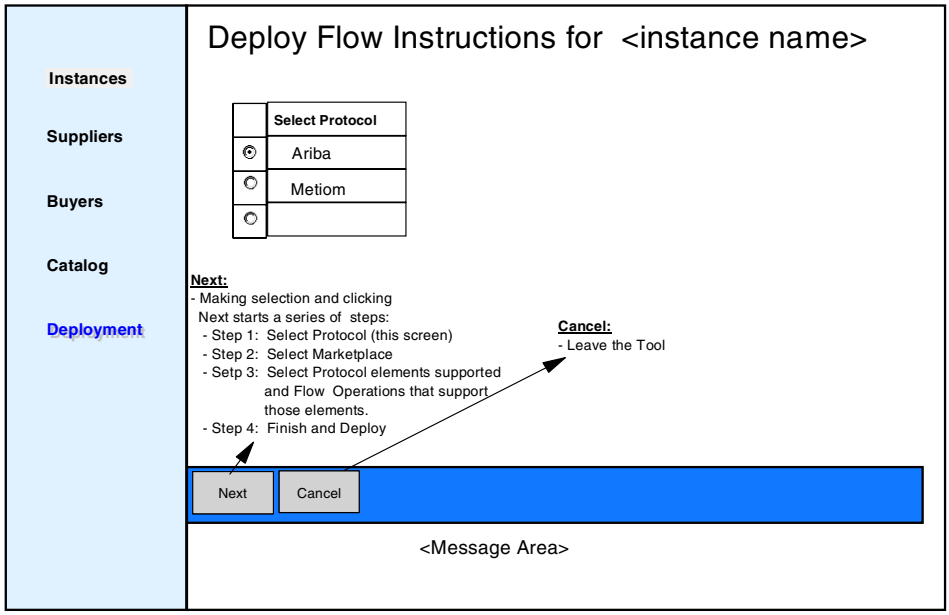


Figure 61. Process Deployment Tool

The Process Deployment Tool provides a wizard-like interface to walk the user through the set of selections required to deploy a new business process flow or to view the set of current business process flows that are currently defined to a B2B instance.

Chapter 12. B2B: Application solutions

B2B computing has become a major force in the industry. From core business to e-business, it doesn't mean changing everything. It means linking the core transaction processing applications that are the heart of your business today directly with your distributors and manufacturers to better support and serve your customers and strategic business partners through new interfaces.

IBM understands the demands of integrating a computing relationship. For that, with the iSeries technology and tools, we offer a set of ready applications. This IBM and Business Partner worldwide solutions Web site is a key place to start your search and understanding:
`http://www.ibm.com/solutions/`

Once you reach this Web site, you see an index to solutions by industry (Figure 62).

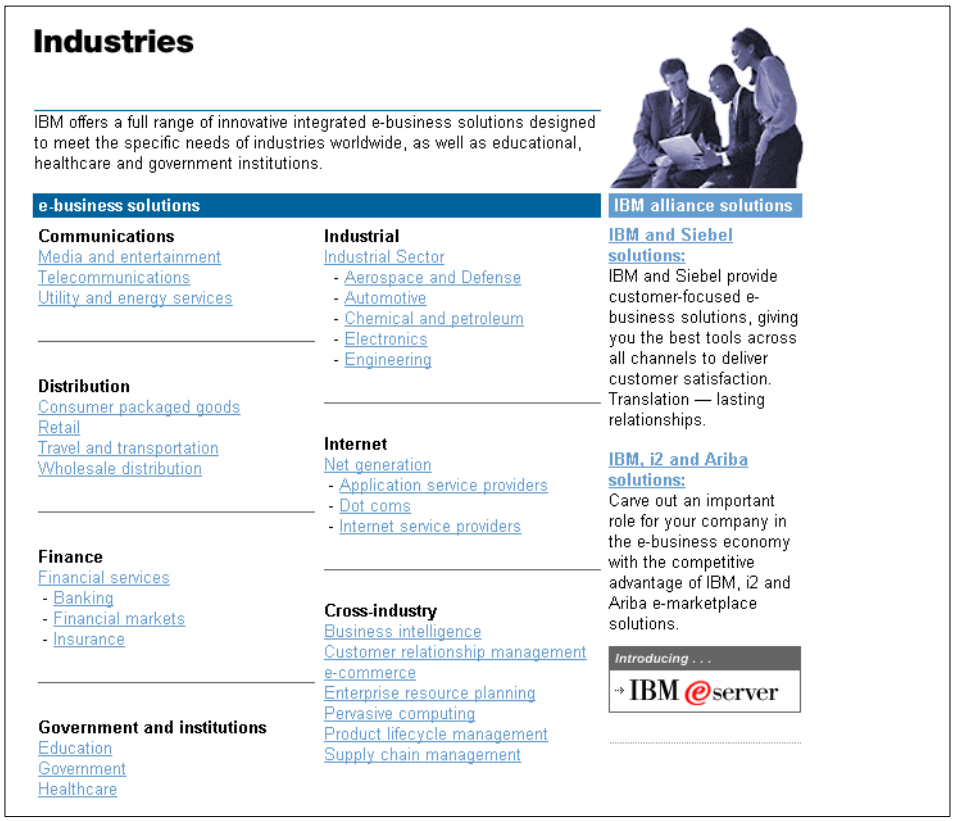


Figure 62. IBM and non-IBM solutions by industry

Another way to search and find a specific solution is through the PartnerWorld for Developers Web site: `http://www.developer.ibm.com/`

Click the **Global Solutions Directory** link under Features. This takes you to `http://www8.software.ibm.com/solutions/isv/igssg.nsf/LanguageSelector/?OpenForm` where you can select your language. Figure 63 shows the page that appears, which offers a powerful solution search tool. You can search by keyword against the solution provider name, the solution name and description, and even a business problem description. Notice on the right that you can refine your search by operating system.

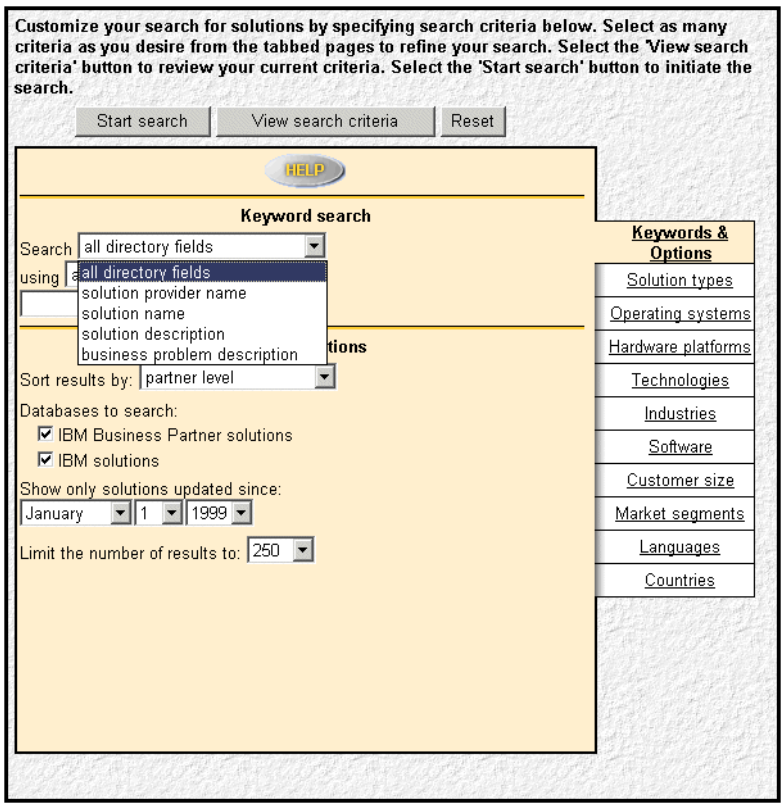


Figure 63. Global Solutions Directory

12.1 IBM solutions

For an e-commerce solution that is right for both B2C and B2B, see Chapter 7, “WebSphere Commerce Suite for AS/400” on page 129.

For a payment engine to handle the integral part of e-business success (money!), see 7.3, “IBM WebSphere Payment Manager” on page 143.

And, while products like Domino itself are not “solutions”, many companies provide Domino-based portals and solutions that might fit your needs. For this and other solutions, one of the best places to start is IBM PartnerWorld for Developers. PartnerWorld is a world wide program that supports developers who build solutions using IBM technologies.

12.2 Non-IBM solutions

What follows is a brief overview of some of the solutions available to you from IBM Business Partners.

12.2.1 Ariba

Ariba, Inc. is the leading business-to-business (B2B) e-commerce platform and network services provider. Through the Ariba B2B Commerce platform, an open and end-to-end infrastructure of interoperable software solutions and hosted Web-based commerce services, the company enables efficient on-line trade, integration and collaboration between B2B marketplaces, buyers, suppliers, and commerce service providers. The global reach and best-of-breed functionality of the Ariba B2B Commerce platform creates Internet-driven economies of scale and process efficiencies for leading companies around the world.

Ariba can be found at: `http://www.ariba.com`

12.2.1.1 Ariba SupplierLive

Developed by Ariba and IBM, SupplierLive dramatically streamlines the process through which suppliers participate in e-commerce by providing all the components for a complete solution, including hardware, application and middleware software, services and support, financing, B2B connectivity with the Ariba platform, and networking solutions from Cisco Systems. IBM Start Now e-commerce Solutions for Ariba SupplierLive' offers easy access to multiple procurement systems and e-marketplaces from a single platform. It is based on the IBM WebSphere Commerce Suite and the IBM Netfinity and iSeries server platforms, and incorporating Cisco networking solutions.

The key features of Ariba SupplierLive include:

- **Multiple selling models**: Enables suppliers to build a differentiated e-commerce site on the Internet and to connect to buyers and marketplaces across the Ariba Commerce Services Network.

- **Seamless cXML PunchOut**: Allows buyers live access to a supplier's e-commerce site to maintain differentiation and streamline content management
- **Robust implementation for buyer specific catalogs and pricing**: IBM WebSphere Commerce Suite enables customers to create bundles, packages, cross-sells, and up-sells
- **Pre-negotiated prices**: Buyers and suppliers can securely contract with each other for discounted prices
- **Back-end integration**: Orders can now flow directly from Ariba Buyer to IBM WebSphere Commerce Suite without human intervention, and IBM WebSphere Commerce Suite can be integrated directly into the supplier's ERP system
- **B2B interoperability**: Allows suppliers to connect and do business with various B2B ecosystems
- **Continuous connectivity**: Networking solutions provided by Cisco provide suppliers the capacity to handle the increased demand for connectivity

IBM Start Now e-commerce Solutions for Ariba SupplierLive is available via Ariba's Supplier Advisor on the Ariba Commerce Services Network.

12.2.2 BinaryTree.com

BinaryTree.com is a business solution provider that specializes in the development of proprietary software products to help you gain a competitive advantage in today's global economy. They create products that generate interactive e-commerce Web sites and collaborative intranet and extranet portals. BinaryTree.com also delivers software tools that enable companies to move seamlessly between proprietary e-mail systems. BinaryTree.com can be found at: `http://www.binarytree.com/`

12.2.2.1 ezMerchant Collaborative eBusiness Suite

ezMerchant is a tightly integrated complete B2B solution. Robust in all areas of site creation, site presentation, customer management, product catalog, commerce, fulfillment, and site administration. Additionally, the Points Of Exodus (POE) APIs and batch data transfer features found in the Enterprise Manager provide extensive back-end integration capabilities. This all-encompassing approach significantly reduces your Web site development cycle when compared to implementing similar solutions using competitive products. With ezMerchant, a completely enabled B2B site is up and running in one to two months. It also decreases administrative and support costs

moving forward. The entire framework is based on the following three modules:

- **Points Of Exodus (POE) Enterprise Integration framework**: Integration with Existing Corporate Infrastructure ezMerchant can interact in real time with virtually any back-end system, external application, or existing business rule logic using the Enterprise Manager's Points Of Exodus (POE) APIs.
- **Plug-in cartridges**: These cartridges are off-the-shelf integration modules that extend the reach and capabilities of the products in the BinaryTree Collaborative eBusiness Suite. They provide comprehensive and easy to deploy integration to ERP systems and other industry standard connectors.
- **Collaborative Application Portal**: ezMerchant comes bundled with a Collaborative Application Portal. A cost-effective, out-of-the-box solution for developing corporate intranets and extranets.

12.2.3 LANSA

LANSA Inc. is a leading provider of application development tools and services for iSeries, client/server, and Internet applications. Since the company was founded in 1987, it has served the information management and application development needs of organizations throughout North America. LANSA Inc. is a licensed IBM iSeries Application Development Partner, a Partner-in-Development All-Star, and a member of the IBM client/server series. LANSA can be found on the Web at:
`http://www.lansa.com/`

12.2.3.1 LANSA for the Web Commerce Edition

LANSA for the Web Commerce Edition enables iSeries customers to easily and rapidly deploy solutions that extend their core iSeries applications to both customers and business partners.

It allows you to extend any existing iSeries RPG or COBOL application so you can have your Web application tightly integrated with your back-end systems.

You can rapidly build e-business applications for the iSeries server and Windows NT. You can easily build Internet, intranet, and Wireless applications that securely access and update iSeries and Windows NT or Windows 2000 data.

LANSA for the Web is renowned for its short learning curve and rapid development capability. This allows customers to easily configure their e-business solution to meet the needs of their business.

LANSA's 4GL allows you to rapidly generate industry standard graphical HTML, Java, XML, or WML.

The components of the Commerce Edition include, for example, a shopping cart, order entry, intelligent e-mailing, customer registration and authentication, product catalog, inventory inquiry, and so forth.

12.2.3.2 LANSA Object Repository

The LANSA Object Repository allows developers to define business rules without programming. LANSA's high level language allows you to more productively define procedural logic.

Architected for the future, you can rapidly modify LANSA applications because many changes only need to be made in one place – the LANSA Object Repository. They are inherited by all applications, across multiple platforms. You enforce development practices that ensure a rapid response to business and technical change.

The LANSA Object Repository is the key to rapid response to user change requests. By centralizing common logic in the Object Repository, all without programming, you change definitions in one place and all programs behave according to your new rules. You have much less code to maintain because LANSA automatically creates screens and reports and centrally enforces your business rules.

The LANSA Object Repository supports these features:

- You actively encourage reuse by storing common applications definitions such as data formats, virtual fields, domain and referential integrity validations, error messages, and help text in Object Repository.
- These definitions are inherited by programs and ensure data consistency, while promoting simplified programming and good design practices.
- Maintenance is much easier with centralized definitions of common logic in the LANSA Object Repository.
- Multiple language definitions for screen field descriptions, help text, and error messages are defined in the LANSA Object Repository.
- Multiple users can access the language of their choice from a single program version. You design and maintain one version.

The LANSA product family includes:

- LANSA for the AS/400, the premier AS/400 development environment.
- LANSA for Windows, portable development using Windows 95 or Windows NT.
- LANSA for the Web, mission-critical transactions for your intranet or the Internet. You can generate industry standard graphical HTML, JavaScript, CGI, and Java from LANSA.

12.2.4 Magic Software Enterprises

Magic Software is a leading provider of application development technology, application servers, and business solutions to corporations worldwide. Its customers include thousands of solutions providers, as well as enterprises in more than 50 countries. In addition to its development technology and business solutions, Magic offers related consulting and professional services, education, and technical support. Magic solutions allow companies to use their existing business processes and legacy investments, as well as rapidly customize solutions to meet specific enterprise needs. Magic can be found on the Web at: `http://www.magicsoftware.com/`

12.2.4.1 Magic eBusiness Platform

The Magic eBusiness Platform is a revolutionary framework for designing, developing and deploying e-business applications. It provides an industry-leading, productive development environment, Magic eDeveloper, with comprehensive features for deploying, customizing and maintaining large-scale distributed B2B applications.

Magic leverages its technology and applications to provide solutions that are uniquely adaptable, scalable, and meet business needs in a uniquely reliable and timely fashion.

All Magic applications are compatible with a wide range of platforms, databases and middleware. This compatibility allows applications to integrate easily and rapidly to back-end databases and applications.

12.2.4.2 Magic eMerchant

Magic eMerchant is a powerful e-commerce enterprise platform for deployment of a sophisticated B2B e-commerce site. It allows companies to efficiently leverage their existing enterprise resources and integrate emerging technologies to their best advantage. Magic eMerchant allows you to meet the challenge of B2B e-business, with a sophisticated architecture supporting an e-commerce site that is rapidly customized to your business processes and business rules:

- Enable B2B electronic commerce
- Solution enabled agents to browse complex catalog system
- Insert large orders via the Web
- Deliver an e-procurement solution to better manage and follow the procurement and streamline the flow of business between national buyers, providers, and local unit managers to increase productivity
- Equalize pricing and raise overall order quality
- Improve efficiencies in its business

  The eMerchant-based solution is the first Web-based solution of its kind to be implemented in Italy. By improving supply chain efficiencies, food suppliers and distributors can cut time and costs by approximately two-thirds. These are just three examples of many B2B Magic and Magic eMerchant implementations all over the world.

12.2.4.3 Magic eService

Magic eService is an entirely Web-based enterprise-level customer service management system that streamlines service workflow. It provides you with a comprehensive picture of your service department in real time. Keeping both customers and management abreast of service performance in real time, Magic eService's advanced features enable better management and control of the service department, and improved customer service and retention.

Magic eService enables the complete tracking and centralized management of each customer interaction from the first contact to problem resolution. Customers receive immediate and personalized attention through round-the-clock, interactive service, while management tools measure service department profitability, cost and resource allocation for better decision-making. Completely Web-based, Magic eService significantly reduces overhead and enables effective, long-term customer service:

- Provides an ideal long-term service solution for problems that require extended back-and-forth correspondence between service agents and customers.
- Using simple drop-down menus, agents and customers can easily define the most complex technical details of a product or problem.
- All Web interactions, e-mail, and file uploads are centrally tracked and documented as part of a single service workflow.
- Using Internet-based virtual support centers, Magic eService gives you the flexibility to employ agents anywhere. It eliminates the need to maintain physical service centers around the world.

Chapter 13. B2B: Services

This chapter provides a couple of places where you can start your research for good and reliable services resources.

13.1 IBM iSeries Services Network

You can visit this valuable Web site at:
`http://www.ibm.com/eserver/iseries/services`

The main page is shown in Figure 64.

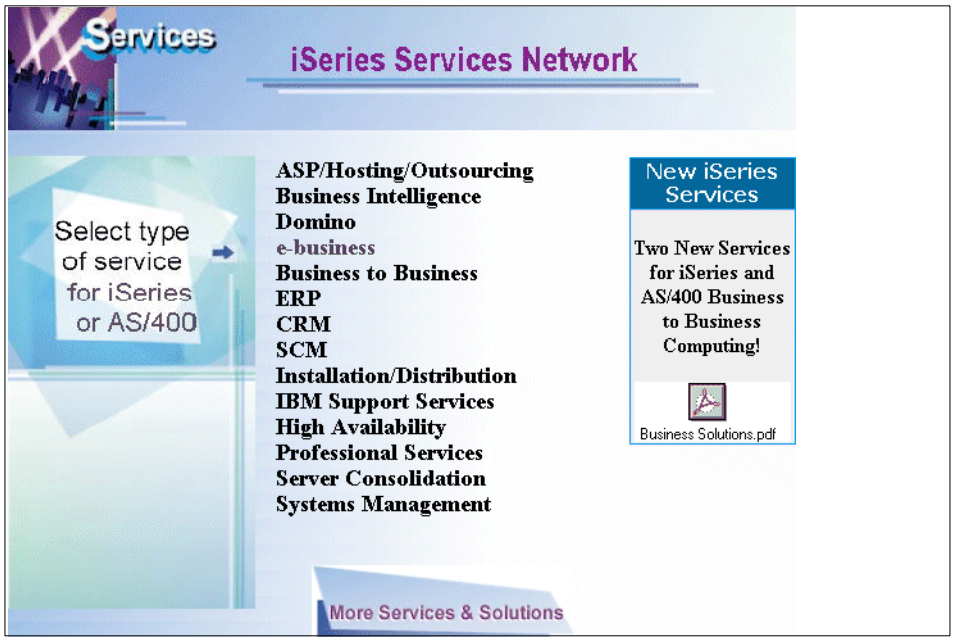


Figure 64. iSeries Services Network

From the main page, you can select either the “e-business” category or the “Business to Business” category to find the list of the services available in your geographical region. Notice that the sidebar on the right in Figure 64 mentions two new services offerings for the iSeries B2B computing. They are:

- iSeries B2B Opportunity Assessment
- Defining an iSeries B2B Solution

These services are offered to iSeries customers to assist in integrating B2B technologies and solutions into your business.

13.1.1 iSeries B2B Opportunity Assessment

Start your B2B journey here. Move from B2B awareness to the solutions and technologies to unharness the potential of B2B in your business.

13.1.1.1 Audience

This service targets businesses that need assistance in taking the first step to leverage B2B technologies in their business. This session helps to assess B2B opportunities and associated requirements and identify potential projects. You will learn about available solution offerings and resources to create a business advantage.

13.1.1.2 Description

Three services are available to iSeries customers and are performed by IBM iSeries e-business consultants or a qualified service provider. They provide up to 16 hours of an e-business consultant's time and knowledge transfer in one or more B2B areas:

- Reviewing and analysizing your e-business and IT strategy as it relates to B2B
- Identifying potential B2B projects to create a business advantage
- Investigating and knowledge transfer in one or more B2B areas:
  - Sell-side commerce
  - Catalogs
  - Marketplace
  - Connection to market place
  - Procurement systems
  - Connectors to ERP systems
  - Business process integration
  - Supply chain
- Analyzing and positioning iSeries solutions and enablers within the context of your e-business and IT strategy and requirements
- Identifying services providers with the skills capable of further defining, implementing, and deploying the customized solution you select
- A summary report provided by e-mail of any reviews, investigations, and analysis performed

13.1.1.3 Contact point

For information and scheduling, contact the Rochester Opportunity Center:

- By telephone in the US and Canada, call 1-888-426-9851. People from Asia Pacific (AP), Latin America (LA), and Europe, Middle East, and Asia (EMEA) should call 1-507-253-7056.
- By e-mail, send a note to RCHAS400@us.ibm.com

13.1.2 Defining an iSeries B2B solution

If the existing portfolio of iSeries B2B solutions does not match your current requirements, you have found the right place. Skilled technical resources are available to build the customized iSeries solution you need to compete.

13.1.2.1 Audience

This service targets businesses that clearly understand B2B opportunities, but need assistance in taking the next steps to further develop project details and architect a solution. Participants receive help in selecting a technology base for implementation that is consistent with your IT strategy and skill and in developing a specific implementation proposal and cost estimate.

13.1.2.2 Description

These customized solutions are developed by IBM iSeries e-business technical consultants within the Customer Technology Center (CTC), part of the IBM Rochester iSeries Laboratory, or qualified services provider.

This offering provides for:

- Getting started with a B2B project requiring a customized solution
- Developing a high level architecture and a design with considerations for:
  - Existing IT infrastructure
  - Integration with existing business applications
  - Technologies available
- iSeries and IBM software technical experts in:
  - Internet and intranet applications including WebSphere
  - Native Java programming
  - Native Domino solutions
  - TCP/IP and sockets application development
  - XML programming
- Mapping project requirements into a high level technical specification
- Developing an initial project management outline with cost projections for subsequent customization and deployment stages

13.1.2.3 Contact point

For more information, please contact:

Mark Even
AS/400 Custom Technology Center
3605 Hwy 52 North
Rochester, MN 55901
Phone: 1-507-253-1313
e-mail: even@us.ibm.com

13.2 eLance for iSeries

If you are not familiar with eLance yet, it is a service site where you can buy and sell services. In a way, it is analogous to eBay. At eBay, you exchange goods, while at eLance, you exchange (or buy and sell) services.

IBM has worked with eLance to provide access to the eLance global marketplace, which connects iSeries buyers and sellers. Buyers can use eLance to access global pricing, talent, and experience by posting project descriptions and opening them up for competitive bidding. For standard services, buyers browse the descriptions of fixed price services and place orders for service. Service providers can use eLance to access buyers from around the globe without expensive marketing efforts.

You can access eLance for iSeries at:
http://www.ibm.com/eserver/iseries/services

Under **Related Links** on the left bottom panel, you find the link to **eLance.com Global Marketplace** at:
http://www.as400.ibm.com/service/elance.htm

As shown in Figure 65, you will find links to both the buyers and sellers of services on the iSeries server.

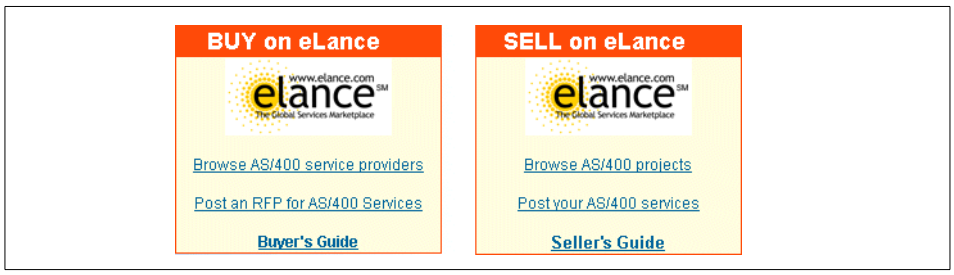


Figure 65. eLance for iSeries buyers and sellers sites

Appendix A. Sizing and performance

This appendix discusses the sizing- and performance-related issues of e-business on the iSeries server.

A.1 iSeries Performance and Capabilities Reference

iSeries V4R5 performance information is available online at:
`http://www-1.ibm.com/servers/eserver/iseries/`

At the sites left of the navigation bar, select **Library** to access the online iSeries library found at:
`http://publib.boulder.ibm.com/html/as400/onlinelib.htm`

Search there for *AS/400 Performance Capabilities Reference Version 4, Release 5*, or go directly to:
`http://publib.boulder.ibm.com/pubs/pdfs/as400/V4R5PDF/AS4PPCP3.PDF`

This is an excellent source of performance information for the iSeries. For performance topics related to e-business, go to Chapter 6 on Web serving performance, which includes details on the HTTP Server for AS/400, WebSphere Commerce Suite, and WebSphere Application Server.

A.2 Understanding the performance components of your system

When you evaluate performance in a Web environment, response time is the most visible measurement, particularly when the end user is the primary objective and recipient of the resulting information. As such, response time is a key factor to consider when sizing a system for performance.

There are three major components that affect performance in Web environments. The components include the:

- **Client**: The client component typically contributes up to 25% of the response time if you use Web browser pages through a modem connection. The memory size of a client is an important factor since many Web-related tasks use large amounts of memory.
- **Network**: Usually the network component has a significant impact on overall performance. It typically contributes up to 60% of the response time.
- **Server**: Server components are described in the following section.

Work with each component when sizing and managing performance in the Web environment. Each component is discussed in the following section.

A.2.1 iSeries server component

There are three major components within the iSeries server component itself. To understand how your iSeries server acts as a Web server, you need to understand the major components of the server. Each component has some sub-components.

The server components and subcomponents are:

- Hardware resources:
  - Processor
  - Memory
  - I/Os
- Application code affecting:
  - Static pages
  - Dynamic pages
  - Servlet, CGI, Net.Data
- Database functions:
  - Get key
  - Get method
  - R/O and R/W

Figure 66 illustrates the components and their sub-components from a performance perspective. The pie chart at the center represents the server-side subcomponents.

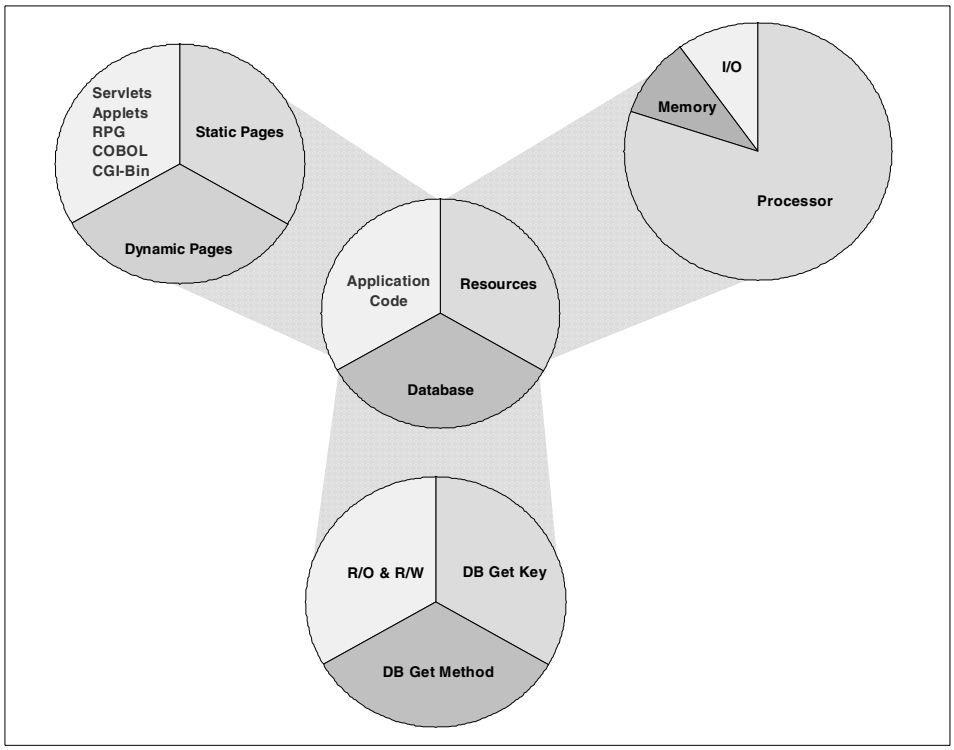


Figure 66. Performance subcomponents

The subcomponents shown in Figure 66 are further divided into the resource subcomponents (upper right corner), the application code subcomponents (upper left corner), and the database subcomponents (lower center).

Note

The pie chart areas of each subcomponent do not necessarily represent the portions or contribution of the subcomponents for performance.

A.3 IBM Workload Estimator for iSeries

To obtain an estimate of the size iSeries server that is recommended to run one or more workloads associated with e-business and or collaboration, such as Domino, Java, or WebSphere, use the IBM Workload Estimator for iSeries 400 (Estimator). Use the Estimator to size a brand new iSeries with all new workloads, size new workloads, or to size the upgrade of an existing server

with the original workload set. IBMers, Business Partners, and customers can use the Estimator to size a single new workload (such as Domino or WebSphere) or to size several workloads running on the same server. The Estimator recommends the model, processor, interactive feature, memory, and DASD necessary for the mixed set of workloads. The recommended processor model is based on the processor utilization, amount of disk, and memory. The interactive features are based on the defined workload. The predicted CPU utilization of the recommended system is presented in graphical format.

The results can be customized. For example, if you want to project for growth, adjust the target processor utilization accordingly. After customization, the Estimator recalculates to determine the best iSeries server options to fit your needs.

The recommended access method for obtaining the IBM Workload Estimator for iSeries is to log on to the Web site at:
`http://www.as400service.ibm.com/estimator`

Note: There is also a download form available for IBM Sales and IBM Business Partners.

We highly recommend you involve IBM Sales or IBM Business Partners before you make any purchasing decisions based on the results obtained from the Estimator. All Estimator results need to be refined by IBM Marketing or a Business Partner.

Appendix B. NLS considerations

This section includes considerations for using e-business in businesses and countries that require national language support.

B.1 General considerations

The most remarkable effect of using the Internet for business is its broader coverage of geographies. You can extend the reach to every corner of the globe thanks to Internet technology. Otherwise, it is impossible. Wherever the server is located, the information is just a click away from a browser. This draws an interesting consideration of the languages that have to be delivered and whether you need multiple systems to serve more than one language. The iSeries server is ideal, because it has well-architected multilingual support functions for the system administration and user application programming.

To serve users better, who are the customers and potential customers, be careful to examine which language is acceptable in the Web pages. If you have to serve in more than one language, think about how you set up and manage the server to achieve the multilingual services.

The iSeries server uses EBCDIC encoding (CCSID) to store data in its DB2 UDB for AS/400 databases and ASCII encoding for the integrated file system. On the other hand, all browsers use ASCII encoding (CCSID). When data is transferred to the browsers, data needs to be converted between the two encoding (CCSID). Figure 67 on page 278 shows an overview of how the code conversion takes place in the HTTP server.

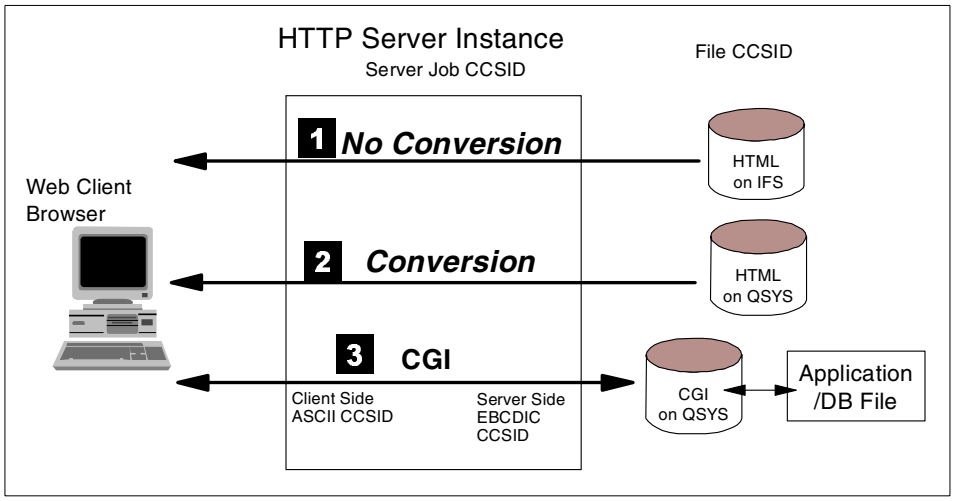


Figure 67. Overview of the code conversion mechanism on the HTTP Server for iSeries

Consider the following points:

- If the page (HTML in the figure) is from an IFS file system, it is not converted.
- If the page is from qsys.lib, it is converted to the CCSID associated with the HTTP server job.
- If the CGI is used to serve the page, the data is converted to the CCSID configured for the HTTP server instance.

If the CCSID, which is set up in the HTTP server, does not match an expected CCSID, which is used in the browser, the characters that are displayed may be meaningless.

There are two types of information regarding Web publishing:

- **Static page**: The contents of pages do not change.
- **Dynamic data**: Some contents of pages change, depending on time, user requested data, and so on. The contents may come from database files.

B.2 Static page

As discussed in Chapter 2, “Building e-business sites: Phased approach” on page 49, the static page is the first phase for doing e-business. This is also the easiest scenario in terms of multilingual services. Focus on the Web

pages to be published and translate them or create new pages for other languages. The following list shows the technique and functions to leverage:

- Different directory for each language:

  Divide the Web pages depending on the language that is displayed on a browser. The users, through a browser, click the button designated to the language. This is the simplest solution. You do not have to set up the server separately for the language, but you have to choose one language for the very first page that may not be understandable for some users (such as Hindu displayed in the US). Perhaps English is the least common denominator for the first page with the buttons captioned by their languages. In this scenario, it is assumed that all pages are stored in the integrated file system in ASCII encoding, so that no text conversion takes place.

- Multiple HTTP server instances to server multilingual sites in one system:

  Set up one HTTP server instance for each language. Then let users know the appropriate URL for the language. The URL may be a different host name or a different port number. To have the multiple server instances, you need IBM HTTP Server for iSeries. Domino HTTP Server does not support multiple instances. In this scenario, the pages can be stored in both the qsys.lib file system and integrated file system. You can configure each server instance to convert data residing in qsys.lib to an appropriate ASCII encoding.

- Synchronizing data between multiple home pages or instances:

  There is no tool to synchronize the multiple pages and instances. You have to maintain each one manually every time you change the contents.

B.3 Dynamic data

When data is retrieved from other resources and stored in Web pages, the application programs need to account for the multilingual environment. For example, when a customer name is retrieved from a database file, the application needs to know from where the data is retrieved. It may be in a different record in the same file, in a different field in the same record, or in a different file. It depends on how you develop the multilingual programs.

Refer to *AS/400 International Application Development V4R2*, SC41-5603, for more information on how to develop multilingual applications. The following list outlines the considerations for a multilingual environment:

- **Different directory for each language**: If data is retrieved from a database and needs to be converted to different languages (that is, to

different ASCII CCSIDs), your application program needs to convert the data based on the language that the other end (browser) uses.

- **Multiple HTTP server instances with CGI**: This is similar to the discussion in B.2, “Static page” on page 278, but there are programming considerations, such as the places where the data is stored. As mentioned before, the CGI program may have to retrieve the data in a different database file or in a different record depending on the database design, for the matching language that the browser user is using. After retrieving the data, the HTTP server converts the data to ASCII CCSID as shown in Figure 67 on page 278.
- **Domino server**: The Domino server is the NLS capable product. It uses UNICODE for internal character representation, which can represent all possible languages. Domino documents are tagged with the language that is used by the Notes user and stored as it is. It is up to the Notes users to display and type in the correct language. That is, when a French Notes user enters French text, other French users can see the text in the correct character images, but German users may not. If data comes from database files, the story is different. It has to be converted from one CCSID to another, that is EBCDIC CCSID to ASCII CCSID even within the same language. Since Domino does not tolerate the locale values associated with the QNOTES user profile for this conversion process, it is not possible to convert to multiple CCSIDs. If you want to serve more than one language to end users, you need more than one system that runs a Domino server.
- **WebSphere**: Since WebSphere runs on top of the HTTP server, the same can be applied as in the HTTP server considerations. Java has a different programming structure from the traditional AS/400 and iSeries programming languages, such as RPG and COBOL. It may be easier to code the multilingual application.

B.4 Browser (client) considerations

When you browse Web pages, use the appropriate language version of the browser or operating system. Otherwise, you may not be able to see the text as it is supposed to be displayed. For example, when you read the Japanese text on your browser, use the Japanese version of Windows operating systems. Japanese text displayed on the English-language version of Windows 98, for example, does not make sense at all.

Note

Microsoft Internet Explorer (IE) offers the ability to display many languages, including DBCS languages, on the English-language version of Windows operating systems. Simply download the fonts that you want to see from the Microsoft IE sites.

Appendix C. iSeries e-business product reference

This appendix is a customer and sales person's reference to each of the products listed in this redbook. It includes the product name, number, the date or version and release it became available, the date it will be (or was) withdrawn from marketing (if announced), and the date service is to be *discontinued*.

Discontinuation of service means that work to resolve defects discovered within the product itself is not done. Typically products go out-of-service some time after the product is enhanced for feature and function.

When a product is withdrawn from marketing, typically improved and equivalent function is available in a product produced at a later time. In this chart, the last column "*Product providing replacement or comparable function*" represents the later product.

Table 4. e-business product reference

Product name	Product number	Date or OS/400 Version Release available	Date withdrawn from marketing, if announced	Date service is discontinued, if announced	Product providing replacement or comparable function
OS/400 V4R3	5769-SS1	09/11/1998		01/31/2001	V4R5 or V4R4
OS/400 V4R4	5769-SS1	05/21/1999			V4R5
Cryptographic Access Provider 40-bit	5769-AC1	V4R4		05/31/2001	
Cryptographic Access Provider 56-bit	5769-AC2	V4R4		05/31/2001	
Cryptographic Access Provider 128-bit	5769-AC3	V4R4		05/31/2001	
WebSphere Application Server for AS/400	5769-AS1	V4R5		05/31/2001	

Product name	Product number	Date or OS/400 Version Release available	Date withdrawn from marketing, if announced	Date service is discontinued, if announced	Product providing replacement or comparable function
SecureWay Host on Demand Version 4.0	5648-C54	V4 9/99			
Cryptographic Support for AS/400	5769-CR1	V4R4			
Screen Customizer for Host Integration Version 1.2	5648-D01	V1 9/99			
SecureWay Host Publisher Version 2	5648-D31	V2 2/2000			
CICS for AS/400	5769-DFH	V4R5		05/31/2001	
HTTP Server for iSeries	5769-DG1	V4R5			
Firewall for AS/400	5769-FW1	V4R4	12/29/2000	05/31/2001	See http://www.as400.ibm.com/firewall
AS/400 Toolbox for Java	5769-JC1	V4R5		05/31/2001	
Advanced Job Scheduler	5769-JS1	V4R5		05/31/2001	
AS/400 Developer Kit for Java	5769-JV1	V4R5			
Lotus Domino Doc	5769-LDD	V3.0			
Net. Commerce for AS/400, Version 2	5798-NC2				5798-WC4 or 5798-NC3

Product name	Product number	Date or OS/400 Version Release available	Date withdrawn from marketing, if announced	Date service is discontinued, if announced	Product providing replacement or comparable function
Net.Commerce for AS/400, Version 3.1	5798-NC3	04/30/1999 V4R3			5798-WC4
Net Question for AS/400	5769-NC5	V4R3		05/31/2001	
Payment Server for AS/400, Version 1.2	5733-PY1	V4R3		01/31/2001	5733-PY2
WebSphere Payment Manager for AS/400, Version 2.1	5733-PY2	V4R4			
OfficeVision JustMail for AS/400	5798-TBT	V4R2			
WebSphere Application Server V3.0.2, Advanced Edition for AS/400 (56-bit)	5733-WA2	V3R0M2		07/31/2001	
WebSphere Application Server V3.0.2, Advanced Edition for AS/400 (128-bit)	5733-WA3	V3R0M2		07/31/2001	
WebSphere Commerce Edition for AS/400, Version 4.1	5798-WC4	V4R4 05/26/00			

Appendix D. Step-by-step guide for building a Web presence site

This appendix provides a detailed description for each step to establish a presence on the Web. We selected the case of building a Web presence site because every one has to pass this stage. Plus, the methodology remains valid for transactional and dynamic data types of Web sites.

D.1 Establishing a Web presence

This section provides information that is necessary to establish a Web presence. With this information, you will be able to publish a Web site to convey information about your company, products, and services. Millions of Web viewers can access information about your company and products through their Web browser.

There are three different kinds of Web sites: Internet, intranet, and extranet. All three use the same technology. The difference is who can see the information. On the Internet, everyone with Internet access can view your Web pages. On an intranet, only people within your company can view the information. On an extranet, only people your company works with, for example suppliers and distributors, can access the Web site. The type of information to publish on the Web site determines the type of Web site.

To establish a Web presence, you need to accomplish these tasks in order:

1. Read about the technology you need to achieve a Web presence. Understand what your environment should look like and what technology is required at each stage of the Web presence development cycle.
2. Learn about what needs to happen before one Web page is produced. In this pre-Web site stage, you set the business objectives you want the Web site to reflect.
3. Learn about the process of developing a Web site. This appendix includes steps to follow when developing a Web site.
4. Look at ways to publish the site to ensure that people visit it.
5. Understand which iSeries products allow you to build a Web presence and be prepared for the next phase of e-business, the dynamic site.

Note: To achieve the benefits of this appendix, you should understand what the World Wide Web is prior to reading it. You should also have beginner's knowledge of Hypertext Markup Language (HTML), which is the markup language that allows Web browsers to translate your information for a Web

user. To understand HTML, refer to Chapter 5, “HTTP Server for iSeries” on page 89.

D.1.1 Technology and architecture definitions

This section describes the terminology of the technology required (or available) for a Web presence in e-business. The components are displayed in Figure 68 and discussed in the following sections.

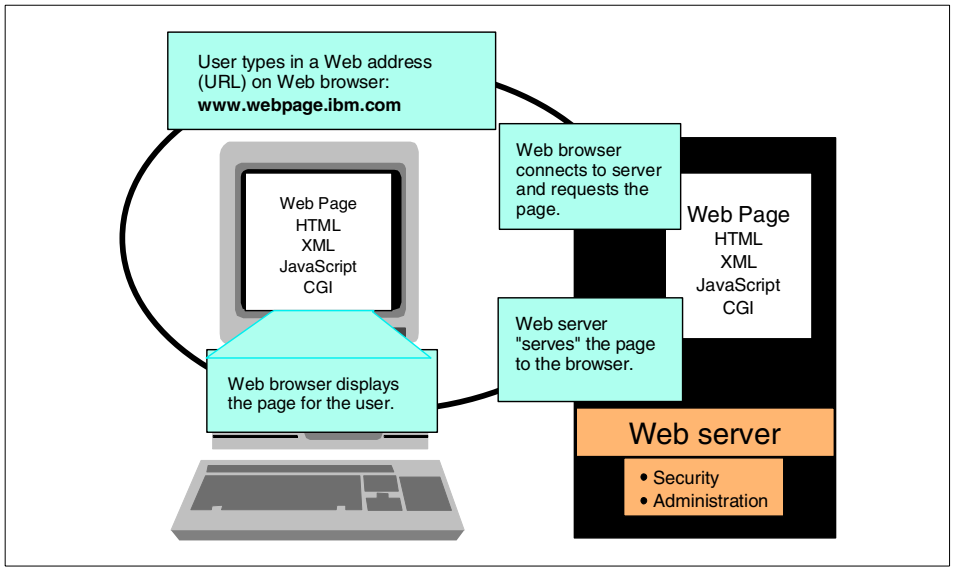


Figure 68. How a Web server works

D.1.1.1 Web site

A Web site is a linked collection of Web pages in an organized structure that resides on a Web server. The Web site is identified by a Universal Resource Locator (URL), which is the Web address, through a Web browser.

D.1.1.2 Web pages

Web pages display text and graphics through Web browsers. Create these pages in HTML, or a combination of HTML and XML, because each are markup languages. Then, add CGIs and JavaScript to add interactive functionality to the Web pages.

These two languages are discussed in the following sections.

D.1.1.3 HTML

Hypertext Markup Language (HTML) is a simple authoring language to display content over Web browsers. It allows for the “mark up” of content with tags and tag elements so a Web browser can read it. You can create HTML code with any word processing software. However, special Web-authoring software allows you to add more style to your Web pages without having to remember all the HTML tags.

D.1.1.4 XML

eXtensible Markup Language (XML) is a markup language that assigns meaning and structure to the content of Web pages. XML is a subset of the Standard Generalized Markup Language (SGML) for defining markup languages to represent structured data.

Many say that XML is to data as HTML is to display. The main point of XML is that, by defining your own markup language with the Document Type Definition (DTD), you can encode the information of your documents more precisely than is possible with HTML.

For more information on XML and its uses, visit the IBM Developer Works Web pages at: `http://www.ibm.com/developer`

D.1.1.5 Cascading Style Sheets

Cascading Style Sheets (CSS) describe how content appears on the Web. Unlike HTML, where a designer gives tag elements the same description repeatedly, a designer describes, with one style sheet, how a page or a group of pages looks.

D.1.1.6 CGI

Common Gateway Interface (CGI) is a programming language that allows Web pages to call applications for use within the Web site. CGIs are commonly used to display dynamic data on Web pages. However, you can use them to allow searches of your Web site and offer dynamic links to static information.

D.1.1.7 JavaScript

JavaScript is a scripting language that allows users to interact with Web page information. All the code is defined in one HTML page and doesn’t have to call any applications to work. For example, when you roll your mouse over an image on a Web page, the interaction changes the image.

D.1.1.8 Web browser

A Web browser is a client program to initiate requests to a Web server and display information that the server returns. Netscape Navigator and Internet Explorer are the two most popular Web browsers. They are described in Chapter 5, “HTTP Server for iSeries” on page 89.

D.1.1.9 Domain name

A domain name is a name that your Web site is known as on the Internet. This is your Web address. For example, ibm.com is the IBM domain name.

The domain name represents your Web server’s IP address on which your Web site resides. Also, the domain name is a part of the Web site’s URL, for example: `http://www.ibm.com`

You can sign up for a domain name through a number of registrars.

D.1.1.10 Web server

A Web server is an application that transmits a file to a Web browser. The Web server serves the pages to the user through a Web browser.

The general sequence of operation for a Web server is:

1. A Web browser requests a document.
2. The HTTP server sends the requested document.
3. The Web browser interprets the document and displays it.

D.1.1.11 Security policy

A security policy is a plan to secure the files on the Web server and the access to Web pages viewable through a browser. On the server side, you secure files at the file or directory level on the Web server by limiting access to who can manipulate them. At the browser level, you can limit access to who views the Web pages by setting up an authentication application. See 4.1, “Integrated functionality of OS/400 for e-business” on page 83, for more information on Web security.

D.1.1.12 Administration controls

This is an application to measure the number of times a user accesses a Web page and delivers tracking reports to show the amount of activity on your Web server. This is important for expansion purposes. When you know which pages are the most popular, you can expand your Web site in that direction and scale your server resources accordingly.

D.1.2 Pre-site considerations: Planning the site

This section takes you through the steps that are necessary before you write any Web pages. Prior to developing the site, set the business objectives you want the Web site to achieve. Determine the site's purpose and ensure that the content is what your potential audience needs.

D.1.2.1 Determining the site's purpose

Before you code one page of HTML and place it onto the World Wide Web, you need to realize what the Web means to your company. It means a global presence. Your Web site and information about your company and its products are viewable by millions of people all over the world. A Web site that is created well is important to your company's image.

Answer the question: Why do you want a Web site? Consider these criteria:

- Do you want to inform people about your products and persuade them to place orders?
- Do you want to persuade other businesses to do business with your company?
- Do you want to inform people about your company?
- Do you simply want to be on the Web because everyone else is?

Thoroughly understand every reason why your company wants a Web site.

Here's an exercise that describes how this activity is done:

1. List all of the reasons your company wants a Web site.

   For example, I want a Web site because:

   - My competitor has one.
   - I want new customers.

2. Translate those reasons to state a purpose.

   For example, the Web site will:

   - Give me an equal presence on the Web as my competitor.
   - Show people why these products best fill their needs.

A typical site about a company includes this information:

- About the company
- Products
- Services
- What's new
- Contact us

Regardless of the content you place on Web pages, it must follow the site's purpose, which is the site's reason to exist in the first place.

D.1.2.2 Understanding the site's audience

In this stage, you need to determine:

- The demographics of the audience
- What the audience needs from your Web site
- What the audience can do with this Web site
- The kind of technology your audience uses

Each of these considerations helps you determine the information to place on the Web pages. For example, if you know that your existing customers are small to medium businesses, ask how you can target information about your company to this market segment.

The audience's needs and what they can do with the information drives the type of content you should place on the site. For example, if a large number of your customers often need to know local sellers of your product, include a Web page that lists the local sellers.

Knowing what technology your audience has can prevent your site from scaring away visitors with inaccessible pages, long loading times, and annoying animations. Creating Web pages that have large graphic files inconveniences the audience because they take forever to load.

Consider how people see the information you present before investing time and money into developing the latest and greatest Web site, only to find out your audience can't view it. You should know what modems, browsers, and plug-ins (do they even know what plug-ins are?) they have, as well as the screen size they use.

D.1.3 Developing a Web site

When you turn the Web site's purpose into a statement, you create project objectives. This section describes the development cycle for Web sites. Once you complete these steps, your Web site is ready to be published.

The Web development cycle stages consist of:

- **Plan:** You've completed most of this phase by determining the site's purpose and audience. Other planning tasks, such as establishing and understanding the budget, schedule, equipment, and staff, can take place with regular project management procedures.
- **Create**: Create the design of the site, the content, and the code.

- **Test**: Have multiple people go through your site to ensure all the links work and the content is correct. View the pages through different browsers and different types of computer displays to ensure that all the colors and pages appear and work as you intended.

D.1.3.1 Planning your Web site

You've established the project objectives by determining the site's purpose. The rest of the planning phase involves typical project planning such as staff, budget, schedule, and assessment.

D.1.3.2 Creating the Web site

In this stage, you need to design and produce the Web site and pages. This section covers the elements involved in this stage.

The Web site design tasks include:

- Organizing site structure
- Determining content that fills the structure
- Labeling the categories

Organize the structure and organization of the information you want to present on the site. A hierarchical structure is common for a simple Web presence. Ensure that as people navigate throughout your site, they always understand where they are and where they are going.

As you plan your site's structure, refer to the book *Information Architecture on the World Wide Web* by Louis Rosenfeld and Peter Morville to achieve a complete overview of all the elements that go into planning a site structure.

For example, a company that wants to publish a site to inform people about their company and products could have a site structure like the example in Figure 69 on page 294.

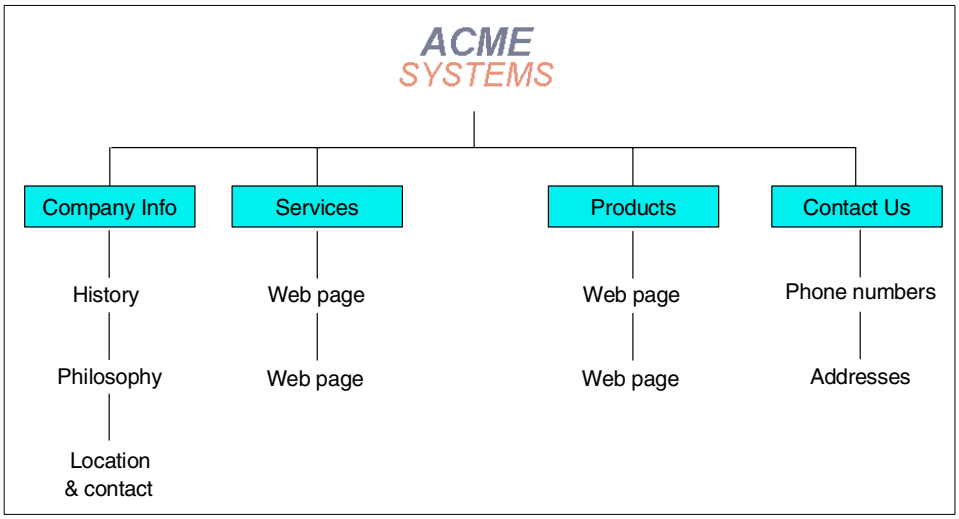


Figure 69. A simple site hierarchy

All information that goes on this site should fall under one of the listed categories. If it doesn't, consider whether that information is appropriate for the site's purpose.

The content is categorized according to this structure, and each category needs a label. Labelling is more complex and goes beyond the scope of this section (see the *Information Architecture on the World Wide Web* book). It is important to consider labelling because it's how your audience can navigate throughout your site.

Label each category with terminology appropriate for the audience. If people cannot find the information they need within your site, they leave it with the click of a mouse. Do your homework and make sure your site speaks the audience's language. Refer to Appendix B, "NLS considerations" on page 277, to understand other language support.

D.1.3.3 Web page

Web page design includes the following elements:

- Laying out the pages
- Determining the text and graphics for the pages
- Designing the navigation

As you design the Web pages, ensure they follow the site's purpose and match the expectations of the site's audience.

For more information on Web page layout, text and graphics, and navigation, read the *Yale Web Style Guide* by Peter Lynch and Sarah Horton. It's available online at: `http://info.med.yale.edu/caim/manual/`

Note

If you do not have the design skills necessary to complete this part of the process, give it to someone in your company who has experience with Web design and graphic design. This part of the development cycle is about design.

The production stage of creating Web pages involves the actual coding and file management. Your pages can be simple HTML or involve CGI code and JavaScript to add a dynamic and interactive element to the Web page's appearance. An example is shown in Figure 70.

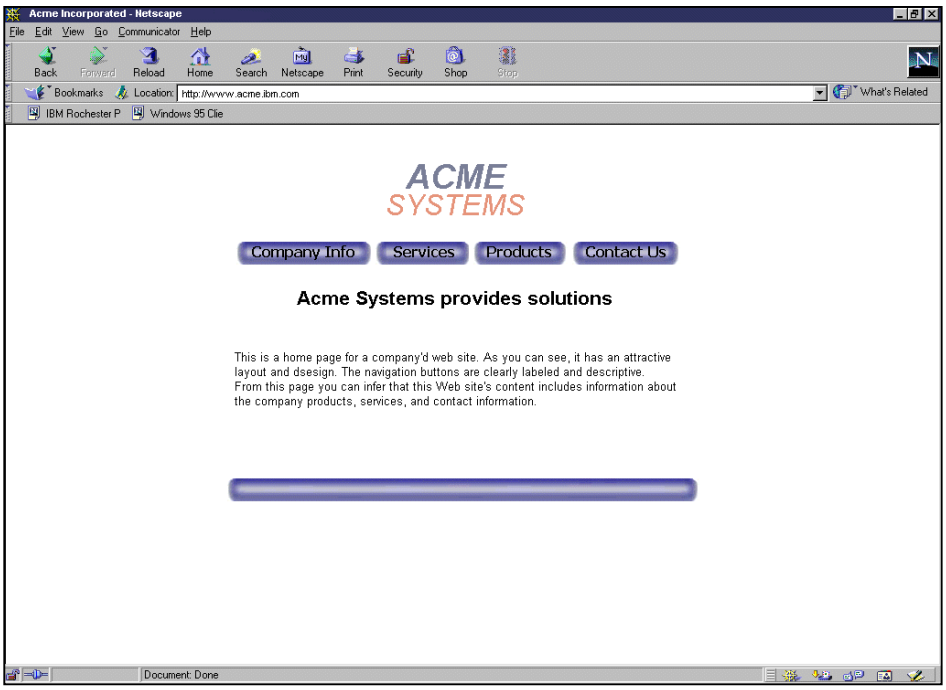


Figure 70. Sample company home page

Consider using a template for all the Web pages in each category to ensure consistency for your users and easier implementation for the developers.

It's important to test the initial templates for how they look on different browsers and browser versions because of functional differences.

Also, view the templates over different speed modems. A 47 KB Web page takes a lot longer to load over a 28K modem than a LAN connection.

Because many Web site's require the work of several developers and designers, construct file management procedures for the staff. This ensures that all work can be accounted for and there are no file overwrites.

D.1.3.4 Testing your Web site

To ensure that anything with your company's name attached to it works perfectly, test the information. The steps to test your Web site are:

1. Stage the site.
2. Conduct quality assurance test.
3. Conduct performance testing.
4. Observe users accomplishing a set of tasks.
5. Test in the client environment.

Staging the site

As you develop Web pages, move them onto a shared Web server that is protected from public view. This gives you more accurate test results because you are viewing the site as if it was on the World Wide Web.

Conducting quality assurance tests

There should be a final check on the content to make sure everything is correct. Ensure the following points:

- Standard spelling and correct grammar
- "Alt" attributes are set for all graphics
- Links go to the correct locations
- Pages, graphics, and scripts display and behave correctly

Conducting performance testing

Ensure that your pages load quickly over slower and faster modems. Long page loading times are an inconvenience for the user.

Observing users accomplishing a set of tasks

Ask users to accomplish a set of tasks with all the pictures and text in place. Verify that:

- Users can find the information they need to find
- Users know where they are in the site
- Users know what is a link and what is not a link
- Users can see important information immediately

Testing in the client environment
Web pages can look and behave very differently depending on the browser, operating system, screen resolutions, and Internet connections. The appearance of colors vary based on what monitors the users are using. The only sure way to obtain an accurate picture of how your site looks is to view the Web pages on their systems.

Test your pages on all targeted browsers. The following items, in particular, behave differently across different browsers and browser versions:

- HTML and HTML extensions (particularly HTML 3.2 extensions)
- JavaScript and other scripting code
- Page layout

Making a backup
Make a backup so that, if necessary, you can restore your files to the server.

D.1.4 Publishing the Web site

Your Web site is ready to go and you've tested it exhaustively. Now it's time for the world to know your company is out on the Web. This section discusses the final steps to accomplish to publish your Web site and advertise its existence.

D.1.4.1 Moving the site to the destination server

Move the site to the destination server where it will be a part of the World Wide Web. If availability is important for your site, make this move during a low usage time such as during the middle of the night. If the files take some time to copy, you may want to take the servers off the network temporarily so that the site is not corrupted with a combination of old and new files.

D.1.4.2 Making search engines find your site

Search engines look for meta information about your site to see if it matches terms for which people are searching. This meta information, or information about the information, is in the form of meta HTML tags. Place descriptive words, key words, subjects, and search terms, for example, in these meta tags to ensure that search engines find your site and return it to a user.

D.1.4.3 Announcing your site on relevant news groups and bulletin boards

Internet news groups are a great way to reach a targeted audience. Many Web users read news groups, and there is no fee for sharing information. Your news group notice should read as a pointer to information of interest rather than as an advertisement. Follow the etiquette and rules of the group.

D.1.4.4 Advertising on major sites

Some sites for Web searches, such as Yahoo and Netscape, provide a way for you to advertise your site. Large numbers of Web users visit these search engines, and you can reach some of these people with an advertisement. These sites charge fees to place ads, so this suggestion applies mainly to commercial sites.

D.1.4.5 Advertising through other media

Attract users to your site by advertising in other media. Radio, television, billboards, and store signs all provide ways to advertise your site.

Advertise your company's Web site address on all communications from your company. Place the address on e-mail signatures, business cards, company stationery, and company brochures.

Remember, your Web site now holds information about your company. The site is the brochure that describes your services, the sales person that sells your product, and the packaging in which you wrap your company. Most of all, the Web site is an important part of your company's image. Place the Web site address anywhere you want your company's name to appear.

D.1.5 Site maintenance

Once your site is published, keeping it current and working 24 hours-a-day and 7 days-a-week is crucial. If your Web site is down or a link is dead when a customer visits it, they walk away with the impression that your site and your company are unreliable and in ill repair.

Keep in mind the important steps explained in the following sections when you manage your Web site.

D.1.5.1 Maintaining links

Broken links frustrate users. Check all of your links periodically, and correct or remove the ones that no longer work. Check an external link more than once before you delete it. It may be only temporarily inaccessible if its server is down.

D.1.5.2 Tracking site activity

The more you know about user activity at your site, the better prepared you are to plan future updates. Tracking reports generated by server software provides you with the following information:

- The number of visits your site receives, which indicates how well it is advertised and how popular it is with users

- The number of disconnects, which indicates technical problems that need correcting
- The pages users link from when they connect to your site, which can help you refine your advertising strategy and even make you aware of categories of users you had not previously considered
- The order in which users view your pages, which help you assess your navigation and information design

D.1.5.3 Keeping users up-to-date on content changes

When appropriate and practical, tell users what content you changed and when you changed it. This information helps them find new material easily. Provide a “What's New” section that displays from the home page, or link directly to the new material from your home page.

D.1.5.4 Maintaining version control

Keep all versions of your site organized so you can update files or refer to previous versions efficiently. Maintain the naming and file directory system you established during the original development of the site. Continue to back up your files and store them safely. Document your procedures for naming and structuring files so others can learn your system easily.

Appendix E. Special notices

This publication is intended to help the iSeries marketing force and AS/400, AS/400e, and iSeries customers who are looking to evolve their iSeries and AS/400e servers to an e-business environment. The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/400 nor Domino for AS/400. See the PUBLICATIONS section of the IBM Programming Announcement for OS/400, 5769-SS1, for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee

that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

e (logo)® 
IBM ®
Advanced Peer-to-Peer Networking
APPN
AS/400e
AIX
CICS
CICS/MVS
CICS/VSE
CICS/6000
COBOL/400
DataJoiner
DB2
DRDA
HotMedia
IBM Payment Gateway
IBM Payment Server
Manage. Anything. Anywhere.
Net.Data
Network Station
OfficeVision/400
OS/2
OS/400
RPG/400
SAA
S/390
SP
System/38
VisualAge
WebSphere
400
cc:Mail
Domino
Lotus eSuite
Passport Advantage
NetView
Tivoli Ready

Redbooks
Redbooks Logo 
AIX
AS/400
AT
AnyNet
CICS/ESA
CICS OS/2
CICS/400
Current
CT
DataPropagator
DB2 Universal Database
Hummingbird
IBM Consumer Wallet
IBM Payment Registry
Intelligent Miner
MQSeries
Netfinity
OfficeVision
Operating System/400
OS/390
PartnerWorld
RS/6000
SecureWay
SupportPac
System/390
Visual Warehouse
Wizard
Lotus
Lotus Notes
eSuite
Notes
Tivoli
TME
Cross-Site
Tivoli Certified

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

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

SET and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix F. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

F.1 IBM Redbooks publications

For information on ordering these publications see “How to get IBM Redbooks” on page 313.

- *iSeries Handbook,* GA19-5486
- *AS/400 e-commerce: Net.Commerce,* SG24-2129
- *IBM SecureWay Host On-Demand: Enterprise Communications in the Era of Network Computing*, SG24-2149
- *AS/400 e-commerce: Internet Connection Servers,* SG24-2150
- *Building AS/400 Client/Server Applications with Java,* SG24-2152
- *AS/400 Internet Security: IBM Firewall for AS/400,* SG24-2162
- *Building AS/400 Applications with Java V2,* SG24-2163
- *IP Network Design Guide,* SG24-2580
- *Cool Title About the AS/400 and Internet,* SG24-4815
- *AS/400 Internet Security: Protecting Your AS/400 from HARM in the Internet,* SG24-4929
- *Unleashing AS/400 Applications on the Internet*, SG24-4935
- *Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice*, SG24-4978
- *AS/400 TCP/IP Autoconfiguration: DNS and DHCP Support,* SG24-5147
- *Lotus Domino for AS/400: Performance, Tuning, and Capacity Planning,* SG24-5162
- *V4 TCP/IP for AS/400: More Cool Things Than Ever,* SG24-5190
- *Net.Commerce V3.2 for AS/400: A Case Study for Doing Business in the New Millennium*, SG24-5198
- *Payment Server V1.2 for AS/400: Secure Transactions in e-commerce*, SG24-5199
- *Using MQSeries on the AS/400,* SG24-5236
- *Web-to-Host Integration Solutions*, SG24-5237

- *Lotus Notes and Domino R5.0 Security Infrastructure Revealed,* SG24-5341
- *Lotus Domino for AS/400: Integration with Enterprise Applications*, SG24-5345
- *IBM Storage Solutions for e-business,* SG24-5356
- *Who Knew You Could Do That with RPG IV? A Sorcerer's Guide to System Access and More,* SG24-5402
- *Building e-commerce Solutions with Net.Commerce: A Project Guidebook,* SG24-5417
- *Lotus Domino for AS/400 R5: Implementation*, SG24-5592
- *Performance Considerations for Domino Application*s, SG24-5602
- *Web Enabling AS/400 Applications with IBM WebSphere Studio*, SG24-5634
- *Building AS/400 Applications for IBM WebSphere Standard Edition 2.0,* SG24-5635
- *AS/400 Mail: Multiple SMTP Domain Names Behind a Firewall,* SG24-5643
- *Business-to-Business Integration Using MQSeries and MQSI, Patterns for e-business Series,* SG24-6010
- *Developing an e-business Application Using Lotus Domino for AS/400*, SG24-6052
- *All You Need to Know When Migrating from IBM Firewall for AS/400*, SG24-6152
- *WebSphere Scalability: WLM and Clustering Using WebSphere Application Server Advanced Edition*, SG24-6153
- *e-Commerce Patterns Using WebSphere Commerce Suite, Patterns for e-business Series,* SG24-6156
- *New Enterprise Integration Functions for Lotus Domino for AS/400*, SG24-6203

F.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at http://www.redbooks.ibm.com/ for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

F.3 Other resources

These publications are also relevant as further information sources:

- *Net.Commerce for AS/400 Installing and Getting Started Guide, V3.2,* GC09-2864
- *WebSphere Commerce Suite Fundamentals, Version 4.1*, GC09-2994
- *Demo CD-ROM* (based on Net.Commerce V3), GK3T-2318
- *Network Products Reference,* GX28-8002
- *Net.Commerce Technologies,* G310-0705
- *Net.Commerce AS/400 Spec Sheet,* G325-6345
- *Work Management Guide*, SC21-8078
- *AS/400 International Application Development V4R2*, SC41-5603
- *e-business with Net.Commerce,* SR23-8849
- *Global Commerce in Info Age,* SV31-3974 (audio tape)
- Morville, Peter and Rosenfeld, Louis. *Information Architecture on the World Wide Web*. O'Reilly & Associates, March 1998 (ISBN: 15-659228-24).

- *Yale Web Style Guide* by Peter Lynch and Sarah Horton. This document is located on the Web at: http://info.med.yale.edu/caim/manual/

The following publications are available in softcopy only from:
http://publib.boulder.ibm.com/pubs/html/as400/v4r5/ic2924/info/index.htm

- *TCP/IP Tutorial and Technical Overview,* GG24-3376
- *Up and Running with Domino for AS/400*, SC41-5334

Consultant papers and white papers

- *Lotus Notes Agent of Change: The Financial Impact of Lotus Notes on Business.* IDC, 1999.
- *Server Selection: Reversing the trend of rising IT costs*. IDC, December 1998.
- *A TCO Analysis of IBM AS/400 Dedicated Server for Domino versus PC Servers*. IDC, 1999.
- *Platform Availability Data: Can you spare a minute?* Gartner Group, October 1998.
- *AS/400 equals cost-effective ERP*. Meta Group, April 1999.

F.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- View the Jupiter Communications Online Holiday Season Study at: http://www.jup.com
- Visit the IDC Research home page at: http://www.idc.com
- IBM @server iSeries server product information: http://www.as400.ibm.com/products
- Gartner home page: http://www.gartner.com
- IBM Redbooks home page: http://www.redbooks.ibm.com
- IBM Firewall for AS/400 home page: http://www.iseries.ibm.com/products/firewall/index.htm
- J. Leleux & Cie S.A. (Belgium) – IBM customer presented in case study: http://www.leleux.be
- Payment processing information: http://www-4.ibm.com/software/webservers/commerce/payment
- IBM AS/400 Business Object Benchmark for Java: http://www.iseries.ibm.com/whpapr/jbob400.htm

- DB2 product family information: http://www-4.ibm.com/software/data/db2
- TCP/IP for OS/400: http://www.as400.ibm.com/tcpip
- HTTP server information:
  http://www.iseries.ibm.com/products/http/httpindex.htm
- Apache Software Foundation home page: http://www.apache.org
- NetObjects site: http://www.netobjects.com/products
- IBM Workload Estimator for iSeries:
  http://as400service.ibm.com/estimator
- IBM Framework for e-business: http://www.software.ibm.com/e-business
- IBM @server iSeries server information for North America:
  http://www.as400.ibm.com/na
- iSeries site map: http://www.as400.ibm.com/misc/map.htm
- IBM WebSphere product information:
  http://www.as400.ibm.com/products/websphere/index.htm
- WebSphere Application Server for AS/400 PTFs:
  http://www.as400.ibm.com/tstudio/websphere/services/service.htm
- WebSphere Application Server for AS/400 documentation:
  http://www.as400.ibm.com//tstudio/websphere/docs/doc.htm
- Web Application Servers: http://www.software.ibm.com/webservers
- Sun Microsystems Enterprise JavaBeans Technology Web pages:
  http://java.sun.com/products/ejb/index.html
- IBM WebSphere views:
  http://www2.software.ibm.com/casestudies/swcsweb.nsf
- WebSphere Commerce Suite:
  http://www.ibm.com/software/webservers/commerce
- WebSphere Commerce Suite V4.1 for AS/400:
  http://www.iseries.ibm.com/ebusiness/wscommerce.htm
- IBM WebSphere Commerce Community:
  http://www.ibm.com/software/websphere/commerce/community
- PartnerWorld for Developers: http://www.iseries.ibm.com/developer
- IBM @server iSeries Integrated Application Servers:
  http://www.ibm.com/servers/eserver/iseries
- iSeries Information Center: http://www.ibm.com/iseries/infocenter
- Quick access to IBM server support: http://www.ibm.com/servers/support

- SET Secure Electronic Transaction: http://www.setco.org
- BinaryTree.com e-business product solutions home page: http://www.binarytree.com
- IBM @server iSeries Dedicated Server for Domino: http://www.iseries.ibm.com/domino/dsd.htm
- Information about Lotus Domino is available online at: http://www.lotus.com/domino
- Lotus Develop Network: http://www.lotus.com/developers
- Note.net home page: http://www.notes.net
- iSeries Client Access: http://www.iseries.ibm.com/clientaccess
- AS/400 consultant reports: http://www.iseries.ibm.com/conslt/dsd_tco.htm
- Lotus home page: http://www.lotus.com/home.nsf
- Lotus Domino Developer's Tools: http://www-1.ibm.com/servers/eserver/iseries/domino/domdevtools.htm
- Domino Application Studio: http://www.lotus.com/home.nsf/welcome/appstudio
- AS/400 case studies: http://www.iseries.ibm.com/casest/casemenu.htm
- Blake Equipment Web site for the ezMerchant case study: http://www.H2Oproducts.com
- IBM Technical Support Technical Information site: http://www.software.ibm.com/is/ibm-lotus
- NoteBench Consortium: http://www.notesbench.org
- Domino performance: http://www.iseries.ibm.com/developer/domino/perform
- Maximizing Application and Server Performance in Domino (white paper): http://www.iseries.ibm.com/developer/domino/perform/maxperform.pdf
- Case study on the Longmount Police Department: http://www-1.ibm.com/servers/eserver/iseries/casest/longmont.htm
- Case study on Saab Cars, USA: http://www-1.ibm.com/servers/eserver/iseries/casest/saab.htm
- Application Development Tools and Middleware for iSeries: http://www.as400.ibm.com/developer/tools/index.html
- MQSeries Family Platforms: http://www-4.ibm.com/software/ts/mqseries/platforms

- IBM MQSeries Family SupportPacs:
  http://www-4.ibm.com/software/ts/mqseries/txppacs
- Summit Strategies home page: http://www.summitstrat.com
- MQSeries Integrator:
  http://www-4.ibm.com/software/ts/mqseries/integrator
- IBM alphaWorks site: http://www.alphaworks.ibm.com
- Global Solutions Directory:
  http://www8.software.ibm.com/solutions/isv/igssg.nsf/LanguageSelector
- PartnerWorld for Developers home page: http://www.developer.ibm.com
- e-business solutions for industries: http://www.ibm.com/solutions
- Magic Software's home page: http://www.magicsoftware.com
- Ariba home page: http://www.ariba.com
- LANSA home page: http://www.lansa.com/
- iSeries Services Network: http://www.ibm.com/eserver/iseries/services
- eLance for iSeries: http://www.as400.ibm.com/service/elance.htm
- AS/400 Performance Capabilities Reference:
  http://publib.boulder.ibm.com/pubs/pdfs/as400/V4R5PDF/AS4PPCP3.PDF
- IBM developerWorks site: http://www.ibm.com/developer

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** http://www.redbooks.ibm.com/

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access **MyNews** at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number _____

☐ Credit card number _____

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Glossary

24x7 An availability requirement meaning 24 hours per day, 7 days per week, with few or no exceptions.

Active Server Pages A dynamic Web page, with the extension .ASP, created with Visual Basic Script or JavaScript. It displays information as a part of the HTML.

Application Programming Interface (API) A predefined set of program subroutine calls that allows one program to communicate with another without either knowing how the other does its job.

Application Server Mediates between the Web and business applications (for example, WebSphere Application Server Standard Edition). It is a set of routines or software that allow the user to run server side applications, such as servlets, JavaServer Pages and Enterprise JavaBeans. These applications can serve as a link to the existing “legacy” applications or database information.

Cascading Style Sheet (CSS) A method that describes the format, style, or look of a marked-up document (for example, a document that uses a markup language to describe its contents).

cookies Bits of information that a Web site gives to a client’s browser. Anytime the client browser requests that site, the site can retrieve the cookie and customize the pages for the client.

Customer Relationship Management Involves supporting, developing, and retaining profitable customers by using e-business technology.

Central Processing Unit (CPU) The part of a computer that includes the circuits that control the interpreting and running of instructions.

cryptography The transformation of data to conceal its information content, prevent its undetected modification, or prevent its unauthorized use.

Dynamic Host Congifuration Protocol (DHCP) The protocol that allows a networked computer to get its networking configuration from a server on the network.

Digital Certificates In computer security, a digital document that binds a public key to the identity of the certificate owner, and therefore, enables the certificate owner to be authenticated. A certificate authority issues a certificate.

Domain Name Service (DNS) A network service protocol that provides a domain name (such as `www.as400.ibm.com`) to IP address (such as `192.27.22.20`) mapping.

Document Type Definition (DTD) A method of defining the tags used in a markup language such as XML or SGML.

e-business A business process transformed to leverage the World Wide Web (Internet, intranet, and extranet) technology for business benefit.

e-business cycle Provides companies a blueprint for how to move through each phase when c eating and deploying applications and providing support for e-business initiatives. Its steps are: build, leverarge, run, and transform.

e-commerce Selling products and services on the Internet.

Enterprise JavaBeans (EJB) A specification of Sun Microsystems, Incorporated, that is part of the WebSphere Application Server Advanced Edition. EJB support allows your application to include sophisticated business components that run on your server. These components may include business logic with automatic distributed transactions and persistence to a relational database.

extranet An intranet whose access for travelling employees or privileged customers and suppliers is partially allowed through a firewall to the Internet.

firewall A special type of network router that connects two networks but only allows

controlled traffic between the two. For example, it allows employees on their intranet to gain access to the Internet but block all access from the Internet in.

FTP In TCP/IP, an application protocol used for transferring files to and from host computers. FTP requires a user ID and possibly a password to allow access to files on a remote host system. FTP assumes that the Transmission Control Protocol is the underlying protocol.

HyperText Markup Language (HTML) A document formatting language in which a specific set of *tags* describe the contents of a Web page, including the headings, bold and italic text, included images, and hypertext links.

HyperText Transport Protocol (HTTP) The protocol that defines how a Web client interacts with a Web server, requesting and receiving Web pages.

HTTP server A server that "serves" Web pages to client browsers over HTTP.

IBM International Business Machines.

Internet A collection of interconnected LANs and WANs that use the Internet Protocol (IP) to route traffic within it.

Internet The "official" global IP network.

intranet An "Internet" and the computers and services available on that network whose access is limited to a particular company or organization. If connected to the Internet, it is usually through a firewall.

Internet Protocol (IP) The protocol that provides for the passing of packets of data between LANs, forming an inter-network of LANs known as an Internet.

ITSO International Technical Support Organization.

Java An object-oriented programming language for portable interpretive code that supports interaction among remote objects. Java was developed and specified by Sun Microsystems, Incorporated.

JavaScript An interpreted programming or scripting language.

JavaServer Pages Pertaining to the Sun Microsystems, Inc., technology that allows dynamic content to be inserted into an HTML page before the HTTP server sends the page back to the browser for display. These pages have a .jsp extension that signals the HTTP process to call the WebSphere Application Server to handle the embedded Java code (known as a scriptlet) or to refer to a JavaBean.

Java Virtual Machine (JVM) The part of the Java environment that is responsible for interpreting the Java codes.

Local Area Network (LAN) A network that allows a number of computers to transmit data between each other directly, usually between any pair of computers but occasionally from one computer to all others on the local network.

Lightweight Directory Access Protocol (LDAP) A network service protocol that allows simple directory-type information (such as names and addresses) to be looked up in an easy and efficient manner.

protocol A set of rules to be followed for some sort of communication. At low levels, it may be definitions of voltage levels on wire and physical cabling, while at high levels it may be the sending of text greetings and responses that establish a conversation.

public key encryption In secure communication, an algorithmic pattern used to decrypt messages that were encrypted by the corresponding private key. A public key is also used to encrypt messages that can be decrypted only by the corresponding private key. Users broadcast their public keys to everyone with whom they must exchange encrypted messages.

router A network device that connects two or more LANs or WANs together to form an Internet.

servlets A Java-based alternative to CGI, running on the server in response to some action by the user such as clicking on a link.

Standard Generalized Markup Language (SGML) A document formatting language in

which *tags* that are extensible and defined within a DTD, describe the content of a document. It is older and more generalized than XML.

Simple Mail Transport Protocol (SMTP) The protocol that allows text messages (e-mail) to be stored and forwarded from the sender to the receiver.

Secure Sockets Layer (SSL) A protocol with which you can establish secure connections between server applications and their clients. SSL has become an industry standard for securing communication between clients and servers.

SET Secure Electronic Transaction A specification for securing payment card transactions over open networks such as the Internet. SET was developed by Visa, MasterCard, IBM, and other technology companies.

tags Sequences of text in a document that are not normally meant to be seen by the reader but rather provide instructions to the application displaying the document on how to display it.

Transmission Control Protocol (TCP) The protocol that provides for the sending of a stream of data from one process on one computer to another process on another computer over packet-based network protocols such as IP. Takes care of breaking the stream into packets, reassembling them in order, and retransmitting packets that are lost or corrupted.

Transaction Server An extension of the application server. A transaction server is required when a customer wants to extend e-business across the entire spectrum of business processes to encompass transaction processing and running the actual state of the business.

Universal Resource Locator (URL) The address of information on the World Wide Web. It is the fully-qualified domain name of a Web site.

World Wide Web (WWW) A network of servers that contain programs and files. It displays specially formatted files that can be hyperlinked to other files throughout the Internet.

Web page Any document that can be accessed by a URL on the World Wide Web.

Web presence A Web site on the World Wide Web. Also the first phase in the e-business phases of adoption.

Web server An HTTP server that “serves” files to a Web browser on a client machine.

Web site A collection of Web pages that resides on a Web server that is managed by a single entity (an organization or an individual). Each Web site has a home page and a URL.

eXtensible Markup Language (XML) A document formatting language in which *tags* that are extensible and defined within a DTD, describe the content of a document while the look is defined within CSSs. It is a newer subset of SGML that is easier to work with.

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-5694-01 iSeries e-business Handbook: A Technology and Product Reference
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. **ibm.com**/privacy/yourprivacy/

(0.5” spine)
0.5”<->0.875”
250 <-> 459 pages

iSeries e-business Handbook

A Technology and Product Reference

Build and deploy B2B and B2C Web-based applications for a competitive advantage

Extend your business critical AS/400 applications to the Web

Discover iSeries for extreme e-business solutions

This technology and product reference guide targets IBM marketing personnel, Business Partners, and AS/400e and iSeries customers who are looking to extend and expand their information server into e-business. It positions e-business on the IBM @server iSeries server against other solutions in the marketplace. Plus, it describes how the iSeries competes in the business-to-business (B2B) and business-to-consumer (B2C) marketplace.

This book shows the solid potential of the iSeries in each facet of e-business solutions. By reading this handbook, you'll gain a broad understanding of how the iSeries serves in an e-business environment. In addition, you'll learn how to sell the iSeries for e-business solutions, including Web presence, dynamic data, and as a transactional site.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-5694-01 ISBN 0738422185