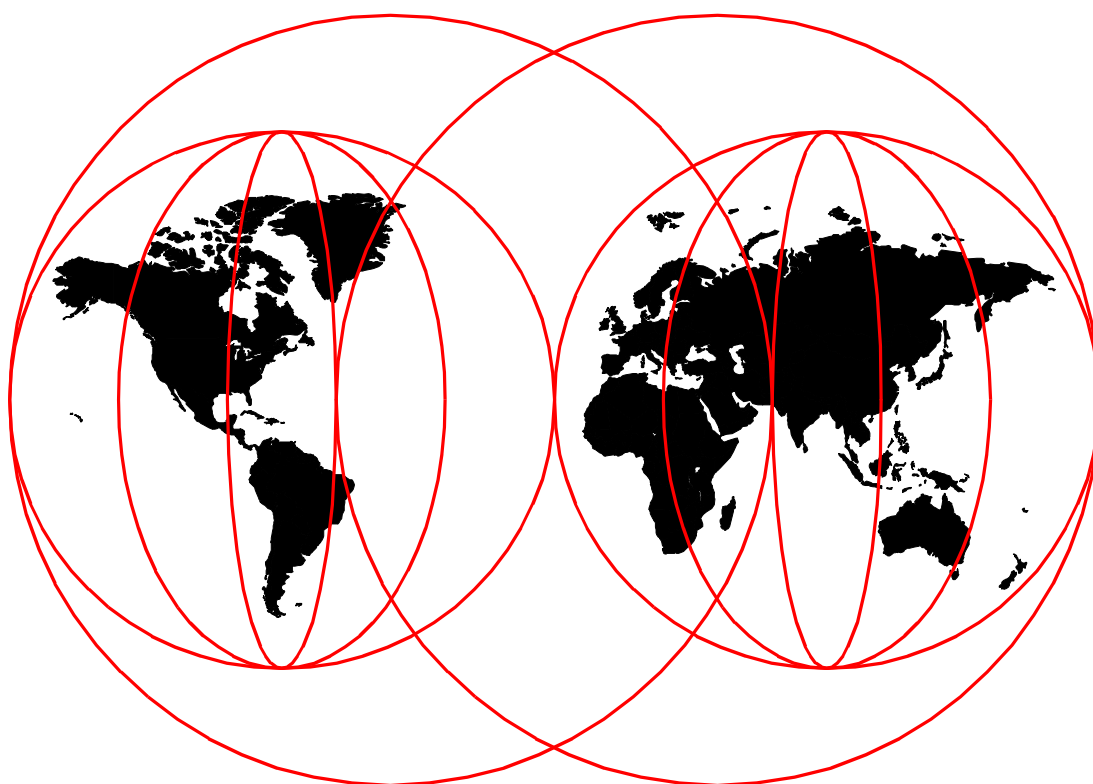


IBM Web-to-Host Integration Solutions

Carla Sadtler, Manda Lackey, Daniel Raisch, Satoshi Takahashi



International Technical Support Organization

<http://www.redbooks.ibm.com>



International Technical Support Organization

SG24-5237-01

IBM Web-to-Host Integration Solutions

December 1999

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, "Special notices" on page 211.

Second Edition (December 1999)

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1998 1999. All rights reserved.

Note to U.S Government Users - Documentation related to restricted rights - Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	vii
The team that wrote this redbook	vii
Comments welcome	viii
 Chapter 1. Introduction	1
1.1 Why the Internet?	1
1.2 Designing your business for the Internet	1
1.3 SecureWay software products	2
1.3.1 SecureWay Host Integration Solution	3
1.4 The WebSphere product family	4
1.5 Web-to-host integration positioning	4
1.6 Solutions for multiple data sources	6
1.7 Specific solutions	8
 Chapter 2. IBM SecureWay Host On-Demand	9
2.1 Host On-Demand server	10
2.1.1 Users, groups, and sessions	11
2.1.2 Redirector	12
2.2 Host On-Demand clients	13
2.2.1 Download client	13
2.2.2 Cached client	13
2.2.3 Locally-installed clients	14
2.2.4 Function On-Demand	14
2.2.5 Host On-Demand Express	14
2.3 How Host On-Demand works	14
2.4 Security	16
2.5 Load balancing and failover support	17
2.5.1 SLP overview	17
2.5.2 Load balancing	18
2.5.3 Failover	18
2.6 3270 host print	19
2.7 5250 host print	20
2.8 Host Access Beans for Java	20
2.9 Host Access Class Library (HACL)	20
2.10 LDAP and Host On-Demand	21
2.10.1 Host On-Demand directory operations	22
2.11 Host On-Demand administration	24
2.11.1 Database On-Demand configuration	28
2.12 Using the client	31
2.12.1 Using Database On-Demand	34
2.13 Where to find more information	40
 Chapter 3. IBM SecureWay Host Publisher	41
3.1 Packaging and software requirements	42
3.2 Components of Host Publisher	43
3.2.1 Host Publisher Studio	44
3.2.2 Host Publisher Server	45
3.3 Load balancing with IBM SecureWay Network Dispatcher	45
3.4 Host Publisher administration	48
3.5 Creating Host Publisher applications	48
3.5.1 Composite applications	50

3.5.2 Integration Object chaining	50
3.6 Creating host Integration Objects	50
3.6.1 Using Host Access.	50
3.6.2 Creating the application.	57
3.6.3 Application results	57
3.7 Creating Database Access Integration Objects	59
3.7.1 Creating the Database Access Integration Object	60
3.7.2 Creating the application.	63
3.7.3 Application results	63
3.8 Using the Host Publisher Studio to build the application	65
3.8.1 Transferring the application to the server.	65
3.8.2 Deploying the application.	66
3.9 Where to find more information	67
Chapter 4. IBM SecureWay Screen Customizer	69
4.1 Screen Customizer features	70
4.2 Requirements for Screen Customizer	70
4.3 Migrating from ResQ!Net.	71
4.4 Supported operating systems (minimum levels).	71
4.5 Supported browsers	72
4.6 Screen Customizer RunTime.	72
4.7 Sample Screen Customizer screens	72
4.8 Screen Customizer Administrator and Customization Studio	74
Chapter 5. Host access APIs and reusable components	83
5.1 Host Access Beans for Java	83
5.1.1 What is a JavaBean?.	83
5.1.2 What are the Host Access Beans for Java?	84
5.1.3 Application development	86
5.2 Host Access Class Library (HACL)	86
5.2.1 HACL programming options	87
5.2.2 HACL for Java	87
5.3 Host Integration Objects	89
5.4 Host On-Demand Connector	90
5.5 Host Access Controls for ActiveX	90
5.6 OHIO.	90
5.7 System requirements.	91
5.7.1 Requirements for the run time environment	91
5.7.2 Requirements for development environment	92
5.8 Where to find more information	93
Chapter 6. Lotus Domino	95
6.1 Lotus Domino and IBM WebSphere.	95
6.2 Lotus Domino and Java	96
6.3 Lotus Domino integration to enterprise	96
6.4 Lotus Domino access to enterprise data and ERP systems	97
6.4.1 Options that require programming efforts.	97
6.4.2 Options that to not require programming	102
6.4.3 Data access roadmap	104
6.5 Access to enterprise transaction systems	105
6.6 LDAP and Lotus Domino R5.0	109
6.7 Where to find more information	110

Chapter 7. WebSphere	111
7.1 Java terms	111
7.2 IBM WebSphere components	113
7.2.1 WebSphere Application Server	113
7.2.2 WebSphere Studio	119
7.2.3 WebSphere Performance Pack	119
7.2.4 WebSphere Site Analysis	120
7.3 Connecting WebSphere Application Server to other subsystems	121
7.3.1 Host On-Demand Connector	121
7.3.2 Connecting to DB2	121
7.3.3 Connecting to CICS	123
7.3.4 Connecting to Lotus Domino	123
7.3.5 Connecting to MQSeries	123
7.3.6 Component Broker	124
7.4 LDAP and WebSphere	125
7.5 Where to find more information	126
Chapter 8. IBM SecureWay FirstSecure	127
8.1 Policy Director	127
8.2 Boundary Server	129
8.3 Intrusion detection	131
8.4 Public Key Infrastructure	132
8.5 SecureWay Toolbox	132
8.6 References	133
Chapter 9. Solutions for accessing CICS from the Web	135
9.1 Terminology	135
9.2 CICS Web solution overview	136
9.2.1 Solutions for Web-enabling CICS and the rest of the enterprise	136
9.2.2 Solutions designed specifically for CICS	138
9.2.3 Complementary products	139
9.3 Making the decision	140
9.4 Connectivity	144
9.5 Connecting to CICS with Host On-Demand	145
9.5.1 Accessing CICS using HOD 3270 emulation	145
9.5.2 Adding Screen Customizer functions	149
9.5.3 CICS Gateway applet	150
9.6 Connecting to CICS with Host Publisher	150
9.7 CICS Web support (CWS)	151
9.7.1 CICS Web support connection methods	152
9.7.2 Calling a Web-aware program using a CWS direct connection	153
9.7.3 Calling a Web-aware program using an indirect connection	154
9.7.4 CICS Web support and the 3270 Web bridge	156
9.8 CICS CORBA client support	158
9.9 NetCICS	159
9.10 CICS Transaction Gateway	161
9.10.1 CICS Transaction Gateway applet architecture	163
9.10.2 CICS Transaction Gateway servlet architecture	164
9.11 Where to find more information	164
Chapter 10. Solutions for Accessing IMS from the Web	167
10.1 Solution Overview	167
10.2 Connectivity	169
10.2.1 Connecting to IMS with TCP/IP	169

10.2.2	APPC Interface to IMS	173
10.2.3	MQSeries	173
10.3	Host Publisher	174
10.4	Host On-Demand	175
10.4.1	Accessing IMS using HOD 3270 emulation	175
10.5	IMS Web	176
10.6	IMS WWW Templates	179
10.7	IMS Client for Java	182
10.8	IMS Object Connector	183
10.9	Net.Data	185
10.10	MQSeries Link LotusScript Extension (MQLSX)	187
10.11	MQSeries Enterprise Integrator for Lotus Notes (MQEI)	189
10.12	AS/IMS, Java, and DCE/RPC	191
10.12.1	DCE/RPC	191
10.12.2	Web access to IMS through Application Support/IMS	192
10.12.3	Encina DE-Light	192
Chapter 11.	Solutions for accessing DB2 from the Web	195
11.1	Programming interfaces to DB2	197
11.1.1	Java Database Connectivity (JDBC)	197
11.1.2	SQLJ	200
11.1.3	Open Database Connectivity (ODBC)	201
11.1.4	Java Stored Procedures	202
11.2	DB2 connectivity	202
11.3	Host On-Demand	204
11.3.1	Accessing DB2 using HOD 3270 or 5250 emulation	204
11.3.2	Database On-Demand	205
11.4	Host Publisher	206
11.5	Net.Data	207
11.6	References and where to find more information	209
Appendix A.	Special notices	211
Appendix B.	Related publications	213
B.1	International Technical Support Organization publications	213
B.2	Redbooks on CD-ROMs	214
B.3	Other publications	214
11.7	Referenced Web sites	215
How to get IBM Redbooks		217
IBM Redbooks fax order form		218
Index		219
IBM Redbooks evaluation		223

Preface

This redbook concentrates information on available Web-to-Host solutions in one book. It summarizes the products and gives you an idea of the circumstances that would lead you to choose each product. It is designed to be a general source of information for anyone considering integrating enterprise data into the Web. Information used in this book has been collected from a variety of sources, including IBM product publications, IBM Redbooks, and personal experience of people in the field.

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, Raleigh Center.

Carla Sadtler is a Senior Software Engineer at the International Technical Support Organization, Raleigh Center. She writes extensively in many areas including SecureWay Communications Servers, network integration, Web-to-host integration products, and the IBM Network Station. Before joining the ITSO 14 years ago, Carla worked in the Raleigh branch office as a Program Support Representative. She holds a degree in mathematics from the University of North Carolina at Greensboro.

Manda Lackey is a System 390 IT Specialist in Tallahassee, Florida. She is a graduate of IBM's Enterprise University, an extensive S/390 education program. Prior to joining IBM, Manda worked in customer software support at Wachovia Bank. She holds a degree in Business Administration with a concentration in Management from Meredith College.

Daniel Raisch is a Certified Senior I/T Specialist in IBM Brazil. He has 18 years of experience in the mainframe field, including operating systems and DB/DC subsystems. He has worked at IBM for six years and for the last four years has been dedicated to managing, implementing and supporting client/server and Web environments on S/390. He has written extensively on ITSO residencies and held presentations across all Latin America countries. Daniel Raisch can be reached by e-mail at *raisch@br.ibm.com*

Satoshi Takahashi is an I/T Specialist in IBM Japan. He has 7 years of experience in the client/server and networking fields. He is in charge of technical support for the Communications Server family and Host On-Demand for S/390 in Japan. Before moving to his current organization, he worked with healthcare industry customers. His areas of expertise include AIX and Windows NT system management, Internet services and applications as well as SNA and TCP/IP networking.

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

Julian Over
Juan Rodriguez
International Technical Support Organization, Raleigh Center

Phil Wakelin
Maria Sueli Almeida
International Technical Support Organization, San Jose Center

Rick Zuburg
Yih-Shin Tan
Ed McCabe
Chip Mason
IBM, Raleigh

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 evaluation” on page 223 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

Chapter 1. Introduction

Today's business environment is rapidly changing and very competitive. In order to survive, companies must meet two basic objectives: growth and cost reduction. How can you achieve these objectives in a competitive environment? Technology, and more specifically the Internet, are key factors in helping companies achieve these objectives. We are living in a time when technology has become a definitive element of succeeding or not succeeding in the business arena.

Since the Internet (especially the World Wide Web (WWW)) has become an important medium for businesses, customers and developers need ways to connect existing host applications to the Web for easy access. Many solutions have been developed. Some solutions are general solutions for making multiple data sources accessible to the Web. Other solutions are specific to products, allowing easy access to the data.

This book is about enabling existing applications for the Web and developing new applications deploying Web technology. We will take a high-level look at many of the solutions available. The intent is to give you enough information to determine the possible solutions that best fit your needs and to guide you to places where you can get more details on each before finalizing your choice.

1.1 Why the Internet?

Before we start describing solutions and products, let's take a look at a primary question: Why Web-enable your business at all?

The Internet is rapidly becoming the core foundation for corporations doing business, and the Internet is redesigning the way business is being done. Through Internet deployment, companies can access more customers and at the same time reduce the cost of business transactions. On the Internet, a company does not need to be physically big in order to do big business. Customers never come in contact with the facilities and many customers can be handled by computer transactions, reducing the need for sales and service employees.

The really big thing the Internet provides is a very popular and inexpensive user interface to get connected, namely, the Web browser. Almost anyone with a computer and a phone line can have a Web browser and access a Web site. For business with Web sites, this means a very large potential customer base.

But the Internet is going beyond the business area; it is changing the way people communicate and relate to each other, making the world smaller. It is a social revolution and the technology involved is evolving very rapidly.

1.2 Designing your business for the Internet

When venturing into the world of the Internet, you should keep a few things in mind when considering your business design:

- Your design should be open. An open system has three major characteristics: interoperability, portability and connectivity. An open design allows the possibility of expansion into new and existing technologies not considered before.

- Your implementation should be scaleable. When you open your systems to the Internet, you never know how many people will access them. You should be able to handle all the incoming requests, but may not need to begin with the largest mainframe that exists. Design your systems so you start with an appropriately sized configuration that has growth potential. The growth potential should apply to the hardware, operating systems, and applications.
- Consider availability. In a “normal” physical business environment, the business closes at the end of the day. People know the business is only open at certain hours. Business on the Internet is different. Your customers exist around the world in different time zones and people do not need to leave their homes to get to your place of business. Business hours are extended to 24 X 7. Your systems must support these business hours to provide full availability.
- Ensure security. With the physical walls gone, anyone with a browser can conceivably access your business. You will want to monitor who is getting in and what is being done, to avoid damages. You will want to restrict access to intranet data. Security is a key consideration in the Internet environment.
- Your solution should be cost effective. Cost reduction is one of the objectives companies are looking for. There are many factors that contribute to cost reduction, among them are protection of the investments that have been made. When planning to deploy Web technology, consider the applications, the software, the hardware and the skills you have in your shop. They represent investments and they are often a good starting point to build new technology.

1.3 SecureWay software products

SecureWay software products offer a complete solution for networking, Web-to-host integration, and security for the enterprise.

Web-to-host integration is made easy through the use of the following products:

- SecureWay Host On-Demand provides host emulation for Web-to-host connections in intranet, extranet, or Internet environments. SecureWay Personal Communications provides host emulation for SNA and TCP/IP networks.
- SecureWay Screen Customizer provides host application rejuvenation using custom graphical front-ends to host applications. Screen Customizer works with Personal Communications and Host On-Demand.
- SecureWay Host Publisher provides Web delivery of host data, providing access from any standard HTML Web browser to multiple back-end resources, including 3270, 5250, VT, Java, and JDBC through a single front-end. Host Publisher allows customers to combine multiple application and database sources into one application. The foundation of the applications is Integration Objects, which are actually Java beans, meaning they are also available for application development outside of Host Publisher. The applications are fully integrated with WebSphere.
- SecureWay software host integration APIs and reusable components provide programmable host integration with e-business applications.
- SecureWay Communications Servers provide SNA gateway connections across SNA and IP networks.

The SecureWay host integration products offer integrated security features for secure connectivity. Transport of sensitive information is secured with Secure Sockets Layer (SSL) technology, providing both server and client authentication as well as data encryption. The SecureWay software infrastructure facilitates integrating the SSL client authentication with IBM Vault Registry, allowing enterprises to benefit from public key infrastructure (PKI) methods. In addition, the host integration client and server products work with existing host security systems (for example, RACF) so that a Web user cannot bypass any host controls when requesting access through a browser interface.

SecureWay FirstSecure is a security framework that has been designed and developed to provide a comprehensive, yet flexible, solution for enterprise security.

The SecureWay host integration products also offer integrated directories based on the Lightweight Directory Access Protocol (LDAP) industry standard. SecureWay Directory and the IBM Common Schema supply a central repository and a standardized information format. Therefore rather than having to build and maintain unique directories for each host application accessible from the Web, enterprises can take advantage of an existing centralized directory.

1.3.1 SecureWay Host Integration Solution

As part of the SecureWay family of products, the SecureWay Host Integration Solution provides intelligent connectivity between users and applications in an overall e-business support structure that reduces the complexity and risk. It delivers a complete infrastructure for connecting host resources across any network to any user, without requiring host application changes. This solution offers a complete set of integration functions including terminal emulation, application integration, and application customization. Information about the IBM SecureWay Host Integration Solution can be found at:

<http://www.ibm.com/software/network/hostintegration/>

The IBM SecureWay Host Integration Solution includes the following products:

- IBM SecureWay Host On-Demand
- IBM SecureWay Personal Communications
- IBM SecureWay Host Publisher
- IBM SecureWay Communications Server for Windows NT
- IBM SecureWay Communications Server for OS/2
- IBM SecureWay Communications Server for AIX
- IBM SecureWay Communications Server for UnixWare 7

In some configurations it includes the following firewall software:

- IBM SecureWay Firewall for Windows NT
- IBM SecureWay Firewall for AIX

The SecureWay Host Integration Solution requires no modifications to existing host software. Not only does this eliminate the risk of breaking legacy code, it also speeds up the integration process. It is one of the ways that the SecureWay Host Integration Solution protects a company's investments in host application and data.

The design of the SecureWay Host Integration Solution with five independent but compatible host connectivity options allows companies to increasingly tighten the affinity between host systems and the Web. Host emulation, host rejuvenation, unconstrained Web delivery, programmable host integration for e-business applications, and SNA gateways are all based on the same infrastructure. This ensures that companies can start simple and build upon prior Web-to-host connectivity as they move further down the e-business path, without having to redo any past integration work.

1.4 The WebSphere product family

The WebSphere product family offers a comprehensive set of middleware for designing, implementing, deploying and managing e-business applications. It allows you to develop, execute, and manage a Web environment.

The WebSphere family consists of:

- WebSphere Application Server
- WebSphere Studio
- WebSphere Performance Pack
- WebSphere Site Analysis
- Individual back-end connectors

The individual connectors are key for developing e-business applications that integrate enterprise applications and data. Connectors exist for Host On-Demand, CICS, IMS, DB2, and MQ that are easily integrated with the WebSphere environment.

WebSphere Application Server V3.0 provides three editions to meet the needs of customers:

- WebSphere Application Server Standard Edition - includes support for Java servlets, Java Server Pages (JSP), Database Connection Manager and performance tools.
- WebSphere Application Server Advanced Edition - includes the Standard Edition plus supports the Enterprise JavaBeans (EJB), providing a runtime environment for deploying enterprise beans.
- WebSphere Application Server Enterprise Edition - includes the WebSphere Application Server Advanced Edition plus TXSeries, Component Broker (CB) and MQSeries support. It is suitable for managing enterprise-level distributed transaction application environments provided by TXSeries combined with the fully distributed processing and distributed object applications provided by CB.

More information on the WebSphere family and how you can use WebSphere to integrate host data into the Web can be found in Chapter 7, "WebSphere" on page 111.

1.5 Web-to-host integration positioning

IBM has aligned its strategic Web-to-host e-business solutions on a common technological foundation, centered around Java technology and the Application Framework for e-business. The result is leading edge solutions for integrating existing applications into the Web using the SecureWay host integration products

that can be leveraged by new Web-based application and business logic development and deployment within an IBM WebSphere software implementation.

The transition for an enterprise to go to a Web-based solution for applications will typically go through three stages of development. IBM has products to provide the solution for each step of the way.

1. Stage 1 involves implementing Web-to-host application integration solutions that open up applications to intranets, then extranets, and last to the Internet, without requiring changes to the host applications. The SecureWay host integration products provide the quickest way to Web-enable existing host applications requiring little or no new business logic. They provide solutions for different needs and complement each other to provide a total solution.

- Host On-Demand is IBM's answer for Java-based host access primarily designed to meet the needs of intranet and extranet users. These users are familiar with the original host application screens and can be considered power users who require a full function emulator. User desktop software is typically well controlled and can include a Java-enabled browser. Users typically connect for extended periods of time, and fast response times are important to maximize productivity.

The addition of Screen Customizer to Host On-Demand transform traditional 3270, 5250, and VT application interfaces to more user-friendly graphical user interfaces.

- Host Publisher provides the capability to transform existing 3270, 5250, VT, JDBC, and Java business processes into Web applications with the advantage of having the capability to build composite applications. Host Publisher is designed primarily for Internet users (but can also be used for intranet and extranet) who are not familiar with typical host screens or how to navigate through legacy applications. For these users, a new, easy-to-use graphical interface is critical. Because the Internet user base is uncontrolled, an HTML front-end is required, rather than relying on the users having Java-enabled browsers. The users typically connect infrequently and for short periods of time.
 - APIs and reusable components provide a programming capability for integrating existing host applications into the Web environment. They provide the same powerful building blocks for developing applications that Host On-Demand and Host Publisher are built upon. They are for use in situations where the other host integration products do not provide the precise solution required.
2. Stage 2 begins by adding new business logic to existing Web-to-host application implementations. Host Publisher applications can be enhanced using products like WebSphere Studio and VisualAge for Java.
 3. Stage 3 involves developing new e-business applications. WebSphere provides a Java infrastructure for the development and execution of Java applications and servlets. It focuses on adding new business logic to existing applications or deploying totally new Web applications.

Host Publisher and WebSphere software complement each other. WebSphere software focuses on providing infrastructure for developing and deploying server-side Java applications as well as tools to develop, deploy and manage Web sites efficiently. Host Publisher focuses on consolidating multiple existing

applications, without modification, to present a single, integrated view in a Web browser, leveraging security, scalability and availability features that are essential for the deployment of e-business applications. Host Publisher uses WebSphere Application Server Standard Edition as a key component of its runtime environment.

With WebSphere you can leverage earlier host integration implementations within the new Web applications to provide host access for the new application, providing investment protection. Integration Objects created by Host Publisher can be reused within WebSphere-developed applications while Java objects created with WebSphere can be imported and used in Host Publisher, delivering synergy between the two products.

Together, these initiatives deliver a leading edge solution for both the integration of existing applications and the deployment of new applications on the Internet.

1.6 Solutions for multiple data sources

Enterprise resources typically include terminal-based applications, program-to-program based applications, message-based applications, file systems, and database data. Although some resources may be further managed by a transaction subsystem like CICS or IMS, the ways to access these resources in a network environment follow several basic approaches:

- Standard terminal data stream architecture, such as 3270
- Standard database languages, such as SQL
- Standard file transfer protocols, such as FTP and IND\$FILE
- Application-specific client/server protocols, such as CICS and IMS non-3270 based transactions
- Application-specific messaging protocols, such as MQ messages

Most enterprises are familiar with the communications software used in their network environment to provide such access, like 3270 emulators or CICS Client. These software programs have most likely provided Web-enabling capabilities so enterprises can start from where they are, in environments they are familiar with. The next step for an enterprise is to determine an overall strategy for making resources available to the Web.

For enterprises that have more than one data source they would like to make available to the Web, products offering a variety of data integration options may be the answer. Instead of modifying each data system for Web access, they can install one product with an interface to each. Doing this provides flexibility in the enterprise, allowing them to expand their Web access to new data and applications while minimizing the cost and disruption.

Host On-Demand gives fast and easy access to 3270, 5250, or VT applications by emulating a terminal. Host On-Demand also provides Host Access Beans for Java and Host Access Controls for ActiveX, which are reusable components for the direct manipulation of application data. As a terminal emulator, Host On-Demand gives Web users access to applications using the old and familiar "green screen" interface. By using the Screen Customizer/LE functions included with Host On-Demand or with the addition of the full Screen Customizer product, the "green screen" can be presented as a graphical interface. The reusable components provide the quickest and easiest way to program host access applications and are used by Host Publisher and Screen Customizer. They

support many prevailing IDEs, Java and non-Java, for developing client or server applications. No changes are required to the existing applications. Host On-Demand also provides the Database On-Demand feature giving SQL access to AS/400 databases through a JDBC driver.

Host Publisher provides a quick and easy way to create interactive Web pages from existing resources of 3270, 5250, VT, and Java applications, whether transactional or not, as well as JDBC accessible databases. Web pages can be custom designed to include data from one or more sources. The Host Publisher Studio is provided to build the applications. Host Publisher applications access host applications and data by using Integration Objects, which are actually beans written to Sun's JavaBeans specification, and as such can be used outside of Host Publisher in other Java applications. Java Server Pages (JSPs) are built to invoke the Integration Objects. The resulting JSPs can be modified to add more function or enhancements. Host Publisher was designed to also allow integration with IBM's Java connectors to provide even broader back-end application support.

Host Publisher and Host On-Demand are not rival products, but rather complementary to each other as seen in 1.5, "Web-to-host integration positioning" on page 4. Table 1 shows a quick comparison of the two products.

Table 1. Host On-Demand and Host Publisher comparison

	Host On-Demand	Host Publisher
Technology	Java applets	Java servlets
Browser requirements	JVM 1.1 support	No requirements
Host interfaces	<ul style="list-style-type: none"> - 3270 - 5250 - VT (VT52, VT100, VT220) - JDBC to AS/400 - CICS Gateway applet 	<ul style="list-style-type: none"> - 3270 - 5250 - VT (VT52, VT100, VT220) - DB2 - JDBC interface - Oracle, Sybase, other - Java applications
Programming effort	No programming necessary; may add customization with Screen Customizer	<ul style="list-style-type: none"> -Build Integration Objects and JSPs with the Host Publisher Studio -Use Java to add new business logic
Presentation	TN3270 / TN5250 green screen or GUI with the Screen Customizer functions, VT	HTML (use HTML editors to enhance output pages)

Component Broker provides a development and run time environment to access data from a variety of data sources, including IMS, CICS, and DB2 using the standard CORBA Object Request Broker.

Net.Data enables Web access to relational data on a variety of platforms. The most prevalent databases can be data sources for your Web application: DB2, Oracle, Sybase, DRDA-enabled data sources, ODBC data sources as well as flat file and Web registry data. With the new IMS Web language environment, Net.Data provides an end-to-end solution for running IMS transactions on the

Web. You can submit IMS transactions and display the results to the end user quickly and efficiently. Net.Data is available with the DB2 product family.

Lotus Domino Web servers can access applications accessible from MQSeries through LotusScript applications using the MQSeries LotusScript Extension (MQLSX) or the MQSeries Enterprise Extension (MQEI). ODBC databases can also be accessed from LotusScript applications using the LotusScript Data Object, the DB2 LSX, or the JDBC-ODBC bridge. Relational databases can be accessed using JDBC.

Lotus Enterprise Integrator (LEI), formerly known as NotesPump, acts as a data distribution server, able to transfer large volumes of data between data sources. It provides both forms-based and programmatic access to many data sources, including Notes, DB2, and ODBC data sources.

1.7 Specific solutions

Many legacy application servers have designed specific interfaces to provide Web access to data. For enterprises with only one type of data source to consider, this may be a simpler approach, and in some cases, a more efficient approach. We will discuss these approaches in the last three chapters when we discuss three major IBM host applications: CICS, DB2, and IMS.

Chapter 2. IBM SecureWay Host On-Demand

IBM SecureWay Host On-Demand (HOD) is an IBM product that gives Web users access to host programs through a Web browser and gives application developers a set of self-contained and portable core host access reusable components for direct manipulation of host application data. Browser access to host systems is provided through the use of Java applets downloaded into the client Web browsers. The latest release, Host On-Demand Version 4, supports TN3270, TN5250, VT52, VT100, VT220, and CICS Java Gateway displays. 3287 and 5250 printer emulation is also provided. Database On-Demand, included with HOD, provides access to DB2 information stored on AS/400 computers using a JDBC driver.

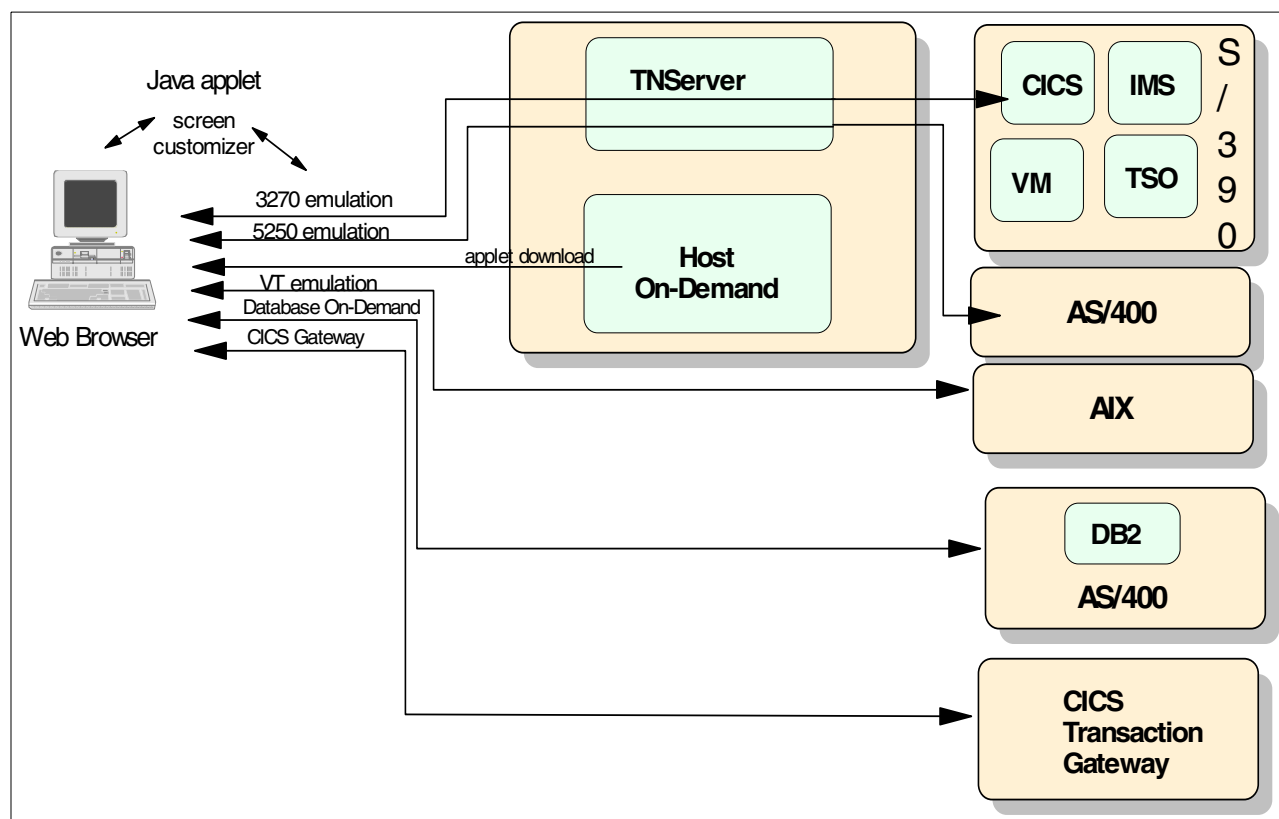


Figure 1. Host On-Demand overview

For 3270, 5250, and VT emulation, an applet running in the Web browser Java runtime environment establishes a connection with a telnet server that manages access to the host application. The user interface is identical to the host terminal, so the traditional 24x80 "green screen" is displayed. In addition, icons for PF keys, copy/paste, macro recording, file transfer, printing, and other basic operations are provided.

Host On-Demand also ships an Express Server and Express Client. These work on the server and the client computer, respectively, to optimize the data communication between client and server. They can reduce the amount of data transferred by up to 80%, thanks to caching, data compression and protocol reduction. This is especially useful with slow connections.

The Database On-Demand feature allows users to access AS/400 databases by running SQL statements online. It uses the OS/400 JDBC client which has been integrated into Host On-Demand.

An alternative to the traditional host-session "green screen" is provided by Screen Customizer/LE, which is included in the Host On-Demand clients. Screen Customizer/LE is a thin Java client. It interprets the host data stream that Host On-Demand sends to it and provides a default graphical presentation of the host screen.

A separate product, IBM SecureWay Screen Customizer, provides the ability to create customized GUIs for host screens. For more information on Screen Customizer, see Chapter 4, "IBM SecureWay Screen Customizer" on page 69.

Host On-Demand includes an application development environment that includes Host Access Beans for Java, Host Access Class Library (HACL) for Java, Host Access Controls for ActiveX, ECLApplets, and Open Host Interface Objects (OHIO).

HACL is an application programming interface that provides access to 3270, 5250, and VT data streams. Host Access Beans for Java are beans written using HACL, providing the most advanced programming technology. HACL can be used to supplement the beans if the programmer wants to build applications from bits and bytes. The Host Access Beans for Java provide self-contained reusable components that provide core host access functions to all types of applications and servlets.

OHIO is IBM's initiative to standardize HACL through IETF. The draft is still changing and the current OHIO function is a subset of that provided by HACL. IBM's intent is to make them interchangeable.

More information on the Host On-Demand programming API's can be found in Chapter 5, "Host access APIs and reusable components" on page 83.

2.1 Host On-Demand server

The server provides management of the Host On-Demand environment and supplies the client applets for download. A Host On-Demand server must always be installed in the same machine as a Web server so that the applets can be downloaded to remote clients.

The following functions and facilities are provided by the server:

- Client code that can be downloaded to browsers on remote workstations
- Group and user management
- License management
- Redirector
 - Provides redirection of sessions between clients and telnet servers
 - Supports SSL security - on Windows NT and AIX
- Express Server (NT only)
 - Provides compression and hence improved performance (with Express Client)
- Database On-Demand provides SQL access to AS/400 (and other) databases
- Host Access Toolkit including:

- Host Access Class Library (HACL)
- Host Access Beans for Java
- Open Host Interface Objects (OHIO)
- SSL V3 support
- LDAP support for storing configurations
- SLP support
- Certificate Wizard for managing client and server certificates
- The ability to import and export session configurations, providing the ability to back up sessions, share sessions, debug session problems, and to import Personal Communications sessions for migration to HOD.
- Publications

A Host On-Demand server must reside in the same machine as a Web server; it can also reside in the same machine as a communications server.

The administration component is used to create and store configuration data. The configuration data can be stored in a local data store or can be stored using LDAP.

The administrator is used to:

- Manage users, groups, and sessions
- Manage and trace the Express (Windows NT only) and Redirector services
- Configure the Redirector
- Configure Express (Windows NT only)
- Configure Database On-Demand
- Enable security
- View trace and message logs

The Host On-Demand server is available for the following operating systems:

- Windows NT 4.0 with SP3 or later
- AIX Version 4.2 or later
- OS/2 Warp Version 4 and Warp Server
- Novell NetWare Version 4.x
- Sun SolarisOS Release 2.5.1
- HP/UX 10.20
- RedHat Linux Version 6
- OS/400 Version 4 Release 2 or later
- OS/390 Version 2 Release 5 or later

2.1.1 Users, groups, and sessions

Host On-Demand users are defined either individually or in groups. User parameters include the user ID, password (optional), description, and whether they are a member of a group. Once a user or group is defined, sessions defining host connections can be configured and made available to the user or group.

Does this mean you have to define every user that will access the system? Not at all. The password is optional. If you have an application you want to make widely available, one technique to do this would be to define a user ID, for example GUEST, and leave the password blank. Users connecting to the HOD server would only need to know to enter GUEST in the user ID field.

Suppose you do want to define individual “real” users. To simplify the administration, you can configure groups of users and define session information for groups instead of each user individually, reducing configuration and administration work.

Users can be allowed to save their session configurations, macros, and keyboard and color mappings on the Host On-Demand server and retain those settings whenever they log on. Session definitions can also be made available to users by exporting the sessions to a network drive. Users can easily import them into their configured sessions window. Note that if you are using Host On-Demand for On-Demand Server or an LDAP server, users cannot be members of more than one group.

2.1.2 Redirector

Java has a security policy that does not allow downloaded code running in a Web browser to establish connections with servers other than the server from which the applet has been downloaded unless the code is known to originate from a trusted source. In this case the applet has to be signed by a trusted entity (which the Host On-Demand applets are). In addition, the browser must be enabled to recognize trusted applets.

For Host On-Demand, this would mean that the Web server, Host On-Demand server and the telnet server must all reside on one machine, as shown in Figure 2, unless you know that all Web browsers accessing the host through your HOD server support signed applets.

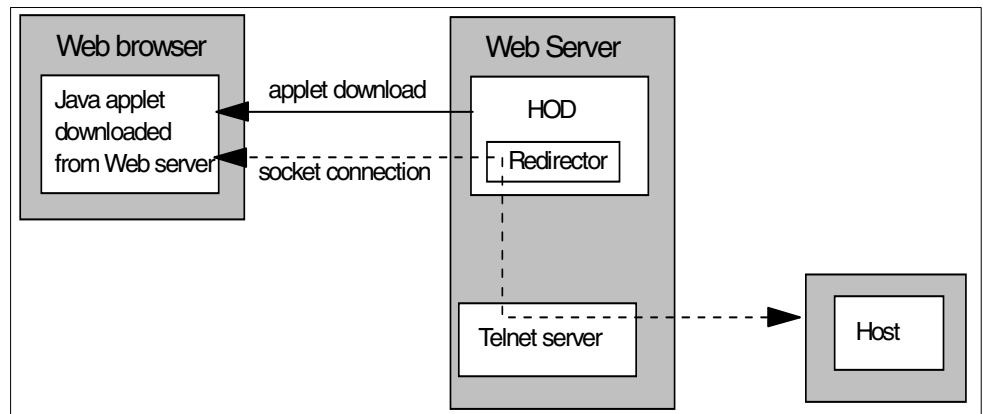


Figure 2. HOD server, telnet server, and Web server on the same machine

Almost all browsers available today support signed applets. However, if there is a possibility that your clients may be using browsers that do not and you want to use more than a single system for HOD and the telnet server, the Redirector is designed to allow you to do this.

To allow the Host On-Demand server and the telnet server to reside on separate machines, the Host On-Demand Redirector can be configured to act as a gateway for clients to telnet servers, as shown in Figure 3. This eliminates the need for the Web browsers to support signed applets since the connection is actually being made to the Redirector which resides on the Web server machine.

Note: You cannot use the Redirector for 5250 file transfer or Database On-Demand since they do not use telnet.

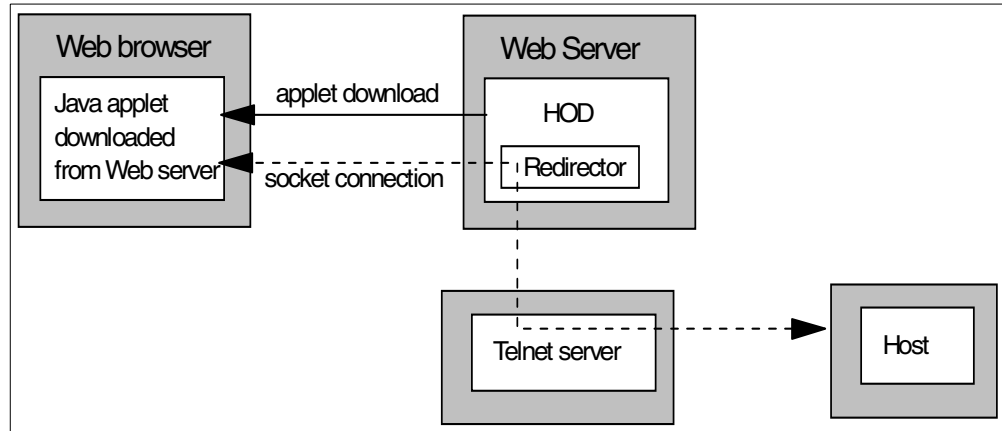


Figure 3. Server configuration using the Redirector

2.2 Host On-Demand clients

Clients connecting to the Host On-Demand main URL will see the following client options:

- Download client
- Cached client
- Function On-Demand client
- Locally installed client
- Debug client (includes trace)
- Database On-Demand client

The emulator download clients include the Screen Customizer capability.

2.2.1 Download client

Perhaps the simplest configuration of Host On-Demand involves users that connect to HOD and request the download client. In this configuration, the client downloads the applet from the server and establishes a connection to the host using a TN3270, or TN5250 server for 3270 or 5250 emulation, directly to AIX using the VT emulator, or to a CICS gateway. Screen Customizer/LE is included in the download client.

With HOD V4 the client must be a Java-enabled Web browser supporting JVM 1.1.

2.2.2 Cached client

The cached client provides all Host On-Demand function, including problem determination and the Screen Customizer (optional). It is cached on your local disk the first time you download it. The next time you start a session, the applet does not need to be downloaded from the server but does check the server to see if the code has been updated. The Screen Customizer/LE function is optional and may be eliminated to reduce the size of the client.

The cached client is persistent across operating system restarts and browser reloads. If you want to remove it, you must remove it manually.

Supported browsers and operating systems:

- Netscape Navigator 4.08 or 4.5 and later (Windows 95, Windows 98, Windows NT, AIX, HP/UX, Linux)
- Microsoft Internet Explorer 4.0.1 with SP1 or 5.0 (Windows 95, Windows 98, Windows NT)

2.2.3 Locally-installed clients

Host On-Demand is shipped with a client component that can be installed individually on client computers. This would be used primarily in situations where users are connected to the network over a slow link and downloading the Host On-Demand client would take longer than desired.

In the client configuration the user loads the applet from the local file system and establishes a direct connection to the server. Session configuration and other option settings are performed locally.

The local client is available for Windows NT, Windows 95, and Windows 98. It includes Screen Customizer/LE functions.

This client has the option of using the Express Client/Server to improve the performance of host-session connections.

2.2.4 Function On-Demand

Function On-Demand downloads the basic files needed initially to the client machine. Other files are downloaded when required, for example, if a 3270 session is started, the files needed only for this type of session are downloaded.

2.2.5 Host On-Demand Express

Host On-Demand Express consists of two components:

- Host On-Demand Express Server for Windows NT
- Host On-Demand Express Client for Windows 95, Windows 98, and Windows NT

The server and client work together to reduce the amount of data exchanged between themselves by as much as 80%. Techniques used during data reduction are:

- Caching - eliminating the need to retransmit unchanged data
- Data compression - reducing data to a fraction of the original data stream by applying arithmetic compression algorithms
- Protocol reduction - reducing the standard telnet connection protocols
- Security - using the secure sockets layer (SSL) protocol to encrypt data transmitted between the Express Server and Express Client

2.3 How Host On-Demand works

Figure 4 illustrates how the components of Host On-Demand interact with one another.

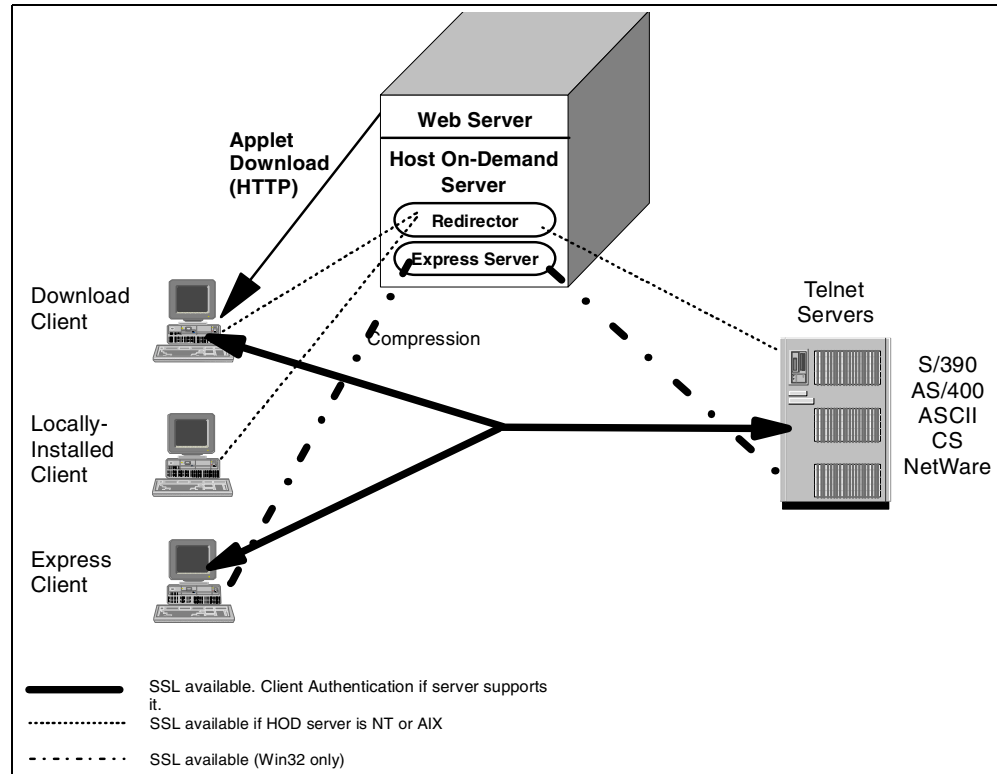


Figure 4. Host On-Demand clients

On the left, three types of client are shown: a download client, which does not have any Host On-Demand code installed; a locally installed client, which has the code installed on its hard-disk; and a locally installed client that also has the Host On-Demand Express Client installed. The latter two are running Windows; the first could have almost any operating system as long as it has a supported Web browser.

The server in the center could be running on any of the platforms supported for Host On-Demand, from a PC to a System/390. Regardless of the platform, a Web server must be active in the same machine.

On the right there are various telnet servers for the emulator sessions, which could be gateways - like Communications Server for Windows NT or NetWare for SAA - or host systems such as System/390 or AS/400.

The user of a download client loads one of the client applets from the Web server through a browser. The user logs on and can start a session that connects directly to a telnet server or that connects to the Redirector, which passes the session on to a telnet server.

The user of a locally installed client loads the client applet from the local file system through a browser (using the Windows Start menu) and starts a session. The session can be configured to connect to a telnet server directly or through the Redirector.

Sessions started on either of these clients can be encrypted if the telnet server to which they connect supports SSL.

With the Express Client, two items of software are loaded - the Express Client itself (which is usually started automatically from the Startup folder) and the Host On-Demand client applet, which is started from the Start menu. These clients are configured in such a way that the Host On-Demand emulator sessions connect to the Express Client (in the same machine), which in turn connects to the Express Server (which is part of the Host On-Demand server on Windows NT). The data flowing between the Express Client and Express Server is compressed and cached and it can also be encrypted.

Any of the clients could be using the Screen Customizer graphical interface.

2.4 Security

Host On-Demand clients can ensure the privacy of their communications through the use of the Secure Sockets Layer (SSL) protocol when connecting to SSL-capable telnet servers. The following servers support the SSL-capable telnet protocol:

- Communications Server for Windows NT Version 6
- Communications Server for OS/2 Version 6
- Communications Server for AIX Version 5.04
- Communications Server for OS/390 Version 2 Release 6 or later
- OS/400 Version 4 Release 4
- Host On-Demand Server

Host On-Demand servers provide support for SSL on both the Redirector and the Express Service.

With SSL you get:

- Reliable message privacy and integrity through encryption and hashing.
- Some assurance for the client that the server is who he says he is. Only some, because:
 - For download clients, the information used to authenticate the server, the CustomizedCAs.class, is itself downloaded from the server.
 - For local clients the CustomizedCAs.class is distributed separately. Therefore, they may be more secure.
 - Express clients use their own key database, but will accept any self-signed certificate.

Note: Host On-Demand V4 supports a new feature - client authentication. With this function, the administrator can prevent a client from seeing and getting past the initial VTAM screen without an installation approved certificate. Currently this function is supported by only Communications Server for OS/390 IP Version 2 Release 8 as the server side.

In addition to using SSL for privacy and integrity, other measures are available to limit access to authorized clients:

- Access security at the host.

This is most likely to be already in place, and the implementation of SSL can be an enhancement on an already secure environment, particularly on an intranet.

- Host On-Demand user security.

By careful implementation of Host On-Demand's own access controls, users are forced to supply a user ID and password before gaining access to the SSL secured connection.

Note: A local client user who knows the IP address of the Host On-Demand server and the incoming port that is used for the redirection, can bypass this security if they have an updated CustomizedCAs.class.

- Use the facilities of the Web server to restrict access to the Host On-Demand applet through, for example, additional user IDs and passwords.
- Use HTTPS, a protocol that combines SSL and HTTP, and which can be used to protect the applet and authenticate the client.
- Use a smartcard or similar technology to authenticate the client.

2.5 Load balancing and failover support

Host On-Demand V4 provides load balancing and failover support for TN3270 and TN5250 connectivity by providing Service Location Protocol (SLP) support in the clients. Network resources, such as TN3270 servers, can be collected together into administrative domains called *scopes*. To benefit from SLP, both the telnet server and clients must support it and be configured to participate. IBM Communications Server for Windows NT Version 6 (CS/NT) and intraNetWare for SAA (NWSAA) support SLP.

2.5.1 SLP overview

SLP is defined in Request for Comments (RFC) 2165. It is a service-discovery method for TCP/IP-based communications, providing a simple and lightweight protocol for automatic advertisement and maintenance of intranet services, while minimizing the use of broadcast and multicast in the network. SLP uses multicast, which targets a group of nodes, unlike broadcast, which targets all nodes. The benefit of multicast is that it sends one packet that all members of the group receive but that only the intended recipients read. A multicast packet is not isolated to a local segment. Routers can forward it to whatever subnets are attached.

Specialized components called agents perform tasks and support services:

- | | |
|------------------------------|--|
| User Agent (UA): | Supports service query functions. It acquires/requests service information for user applications. |
| Service Agent (SA): | Service registration and service advertisement. |
| Directory Agent (DA): | Collects service information from Service Agents which is later requested by User Agents in intranets. |

Services are described by configuring attributes associated with a type of service. A User Agent can select an appropriate service by specifying the attribute that it needs in a service request. When the service request is returned, it contains a URL (Uniform Resource Locator) pointing to the service desired, and other information needed by the User Agent.

The Host On-Demand client is the User Agent. CS/NT or NWSAA is the Service Agent.

SLP can reduce overall network traffic by using scopes to manage client service requests. A scope is essentially a grouping method to organize servers into named groups. Scope values are defined by a network administrator, and may represent departments, regions or organizations. Different scopes can be assigned for different services provided on the server if desired.

2.5.2 Load balancing

CS/NT and NWSAA provide information on the server load using SLP by calculating the percentage of available resources. For traditional LUA sessions, such as 3270 sessions, the load percentage is the number of active application connections divided by the total number of LUs available.

The Host On-Demand client gets the load percentage using SLP, determines which server is the least loaded, and attempts a connection to that server.

In Figure 5 the client has three servers available for the client connection in the named scope, "hodsscope". The HOD client will attempt to connect to Server A because according to the load values returned it is the least loaded of the three. If the connection is successful the load of Server A has increased. If this increase means that Server A is no longer the least loaded, the next SLP client will determine this and connect to a different server.

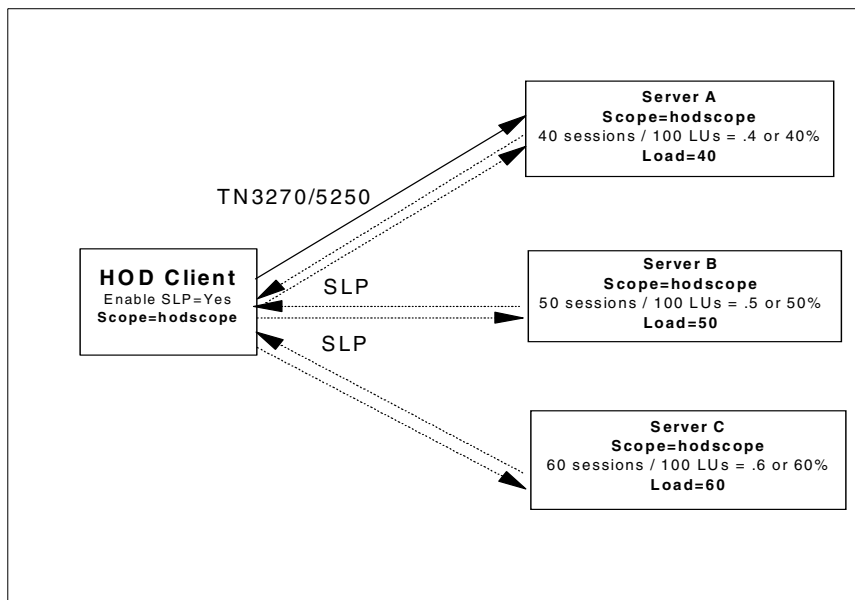


Figure 5. Host On-Demand load balancing with SLP

2.5.3 Failover

A failover capability is provided for TN3270 and TN5250 sessions by using SLP and the auto-reconnect option in the HOD client session. This can be done whether or not you are implementing a load balancing scheme. If the current connection fails, the Host On-Demand client queries the servers again. The server which is alive for standby responds to the client and the client establishes a connection to the standby server.

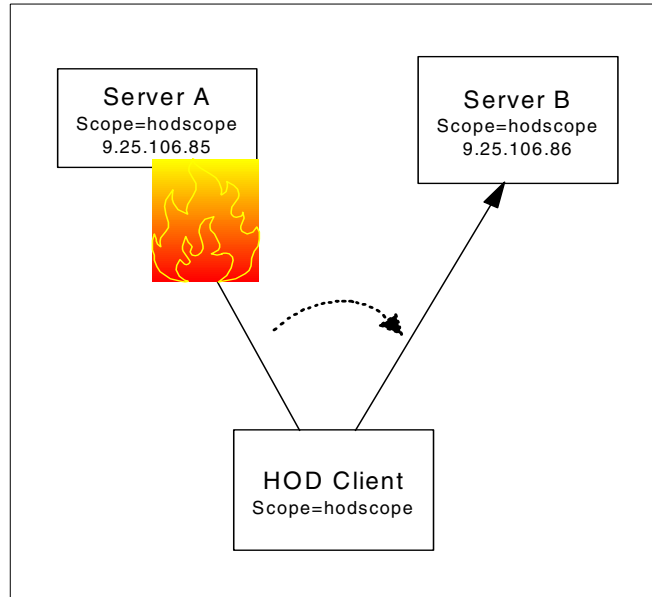


Figure 6. Host On-Demand warm standby with SLP

2.6 3270 host print

Host On-Demand supports local copy and host directed print. A Host On-Demand printer session emulates an IBM 3287 printer, allowing you to print from a host application to a printer connected to your workstation or to a network printer.

Printer sessions run through a browser and use a Java interface in the same way as display sessions. As a result, they cannot use the drivers provided by the operating system. Instead, Host On-Demand printer sessions use Printer Definition Tables (PDTs) to format the data and send it directly to the printer as text and printer control codes. Customized PDTs can be created by creating or customizing Printer Definition Files (PDFs). PDFs created for Personal Communications can be used for Host On-Demand by copying them and running them through the Host On-Demand PDT compiler.

The following PDFs and PDTs are provided specifically for Host On-Demand:

- Basic: Basic ASCII text mode.

A simple PDF that should produce readable output from most printers. Basic ASCII text mode may work if your printer does not support one of the other modes supported by Host On-Demand. However, if you use this mode, the commands that are unique to your printer will not be available.

- LaserPCL: HP PCL Level 3 (laser printers)

Suitable for most laser printers in PCL mode. Level-3 commands are understood by printers that support later levels.

- LaserPPDS: IBM PPDS Level 2

Suitable for most laser printers in PPDS level-2 mode. Printers that support later levels of PPDS (3 or 4) will work with the LaserPPDS PDT.

- Proprinter: IBM PPDS Level 1 (Proprinter XL, X24, XL24)

Suitable for most printers in PPDS level-1 mode. All printers supporting the PPDS mode support PPDS level 1. Proprinter-compatible printers and printers that support only PPDS level 1 cannot do code-page switching and must use the Proprinter PDT.

PostScript printers are not supported for printer sessions. They can be used by local copy from a display session only.

2.7 5250 host print

Host On-Demand V4 supports 5250 host print through the Host Print key and through Host Print Transform (HPT).

With HPT the host print data stream is converted from SCS or AFP to ASCII on the AS/400 and sent to the client. HPT is capable of supporting many types of ASCII data streams. Using HPT has the following advantages in an HOD environment:

1. The client code is much smaller. This is significant for browser-based applets since less code is needed at the client, therefore less code to download.
2. The transform of the data stream is performed by OS/400. Since all transforms are performed by OS/400, the output is consistent across various emulators. Therefore, jobs printed through Host On-Demand should look exactly like jobs printed through Client Access or Personal Communications emulators using HPT.

The Host Print key is available in a 5250 display session. It tells the host to print the contents of the presentation space as text. The Host Print key is not available in Screen Customizer sessions.

2.8 Host Access Beans for Java

The Host Access Beans for Java are beans written using the HACL libraries, providing a powerful programming interface for host access. They can be used with visual development tools, such as VisualAge for Java, to rapidly develop new applications from existing products and applications.

At the core of the technology is a Session bean that secures a connection to a host, and a Macro bean that utilizes a "Record and Play" technology to "mine" application data on the host. The Macro bean is very sophisticated and easy to drive which can quickly turn multiple 3270 application screens into a new business object. By dropping and wiring the beans in a visual development IDE of the customer's choice, an e-business application can be developed in days. No special tools are required because the Macro bean supports both development and runtime environments.

For more information on Host Access Beans for Java, see Chapter 5, "Host access APIs and reusable components" on page 83.

2.9 Host Access Class Library (HACL)

At a lower programming level than the Host Access Beans for Java, you have the Host Access Class Libraries (HACL) for Java. HACL is the basis for the Host

Access Beans and provides a core set of classes and methods that allow the development of platform-independent applications that can access host information without the need for a graphical display. The library represents an object-oriented abstraction of a host connection that includes reading and writing the host presentation space, enumerating the fields in the presentation space, reading the operator information area (OIA) for status information, transferring files, and performing asynchronous notification of significant events.

For more information on HACL, see Chapter 5, “Host access APIs and reusable components” on page 83.

2.10 LDAP and Host On-Demand

Host On-Demand V3 and the default implementation of V4 utilize a private data store model that does not provide sharing across servers. Host On-Demand V4 has added a new function which allows you to store the configuration using an LDAP server. Host On-Demand V4 can place all of its preferences on the LDAP server and these preferences can be configured from the Host On-Demand administration client. In addition, this function allows you to share the information stored on the LDAP server among multiple Host On-Demand servers. The addition of LDAP support in Host On-Demand is essential for enterprises that manage high volumes of user information.

Note

At the time of writing, HOD LDAP support was limited to only the AIX and Windows NT server platforms.

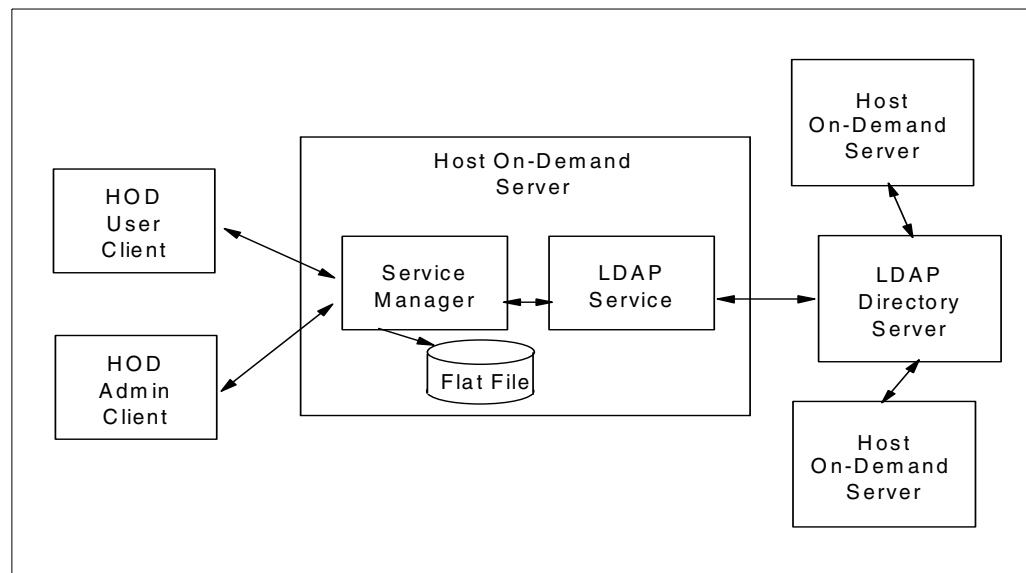


Figure 7. LDAP and HOD interaction

Host On-Demand supports the following LDAP servers:

- Netscape Directory Server V3.X and later
- IBM SecureWay LDAP Server V2.1 (V3 was not supported at the time of writing)

The IBM standard schema is required by Host On-Demand. This schema is provided in several files that are located in the LDAP subdirectory of the Host On-Demand product when it is installed. These files contain extensions to the shipped LDAP schema, and are stored in standard slapd format. These schema extensions must be incorporated into the LDAP server and must be in effect before Host On-Demand can contact and store configuration information into the LDAP server.

2.10.1 Host On-Demand directory operations

The default operational mode for Host On-Demand V4 is to use the private data store. HOD V4 is also capable of using an LDAP directory server to manage and share definitions across multiple Host On-Demand servers.

LDAP directory support is enabled by the HOD administrator under the **Directory** tab of the Administration panel. It is here that you have fields that identify the directory server and suffix that you wish to use, and optionally the ability to migrate your existing Host On-Demand server configuration data to the LDAP server.

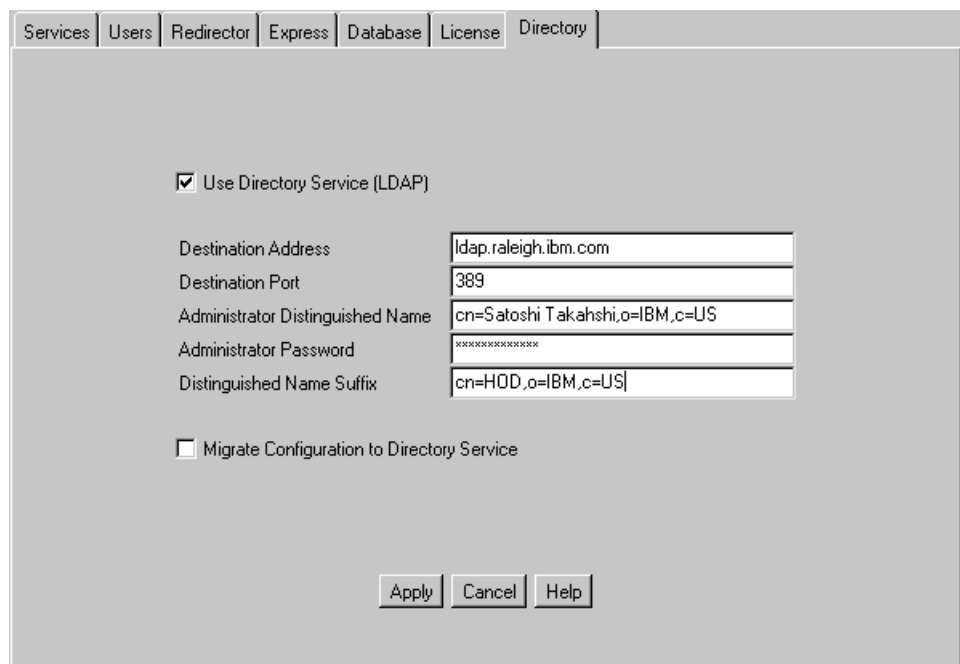


Figure 8. HOD Directory Services panel

Use Directory Service (LDAP)

Placing a check in this box enables the remaining fields on this panel and allows you to specify the directory service that you wish to use.

Destination Address

Enter the address of the LDAP directory. Use either the host name or dotted decimal format. The default is the Host On-Demand server.

Destination Port

Enter the TCP/IP port on which the LDAP server will accept a connection from an LDAP client. The default port is 389.

Administrator Distinguished Name

In LDAP, the distinguished name is a unique name that identifies a single directory entry. Enter the distinguished name (DN) of the directory administrator that allows Host On-Demand to update information. You must use the LDAP string representation for distinguished names (for example, `cn=SatoshiTakahashi,o=IBM,c=US`).

Administrator Password

Enter the directory administrator's password.

Distinguished Name Suffix

Enter the distinguished name (DN) of the highest entry in the directory information tree (DIT) for which information will be saved. Host On-Demand will store all of its configuration information below this suffix in the DIT. You must use the LDAP string representation for distinguished names (for example, `cn=HOD,o=IBM,c=US`).

Migrate Configuration to Directory Service

To migrate users and groups from the private data store to the LDAP directory select the check box and press **apply**. You can check this box either when you switch to the directory server, or after you have made the switch.

Changes made on this panel are effective immediately. Once you have migrated to the LDAP server, subsequent user or group related changes will be made only on the LDAP server, including administrative changes to groups, users, or sessions, and changes such as new passwords, macros, keyboard changes, etc., by either the administrator or a user.

Note: The Redirector and Express Server configurations are not migrated to the directory server.

Considerations for migration to LDAP

Migrating to LDAP has some implications for your group and user configuration information. LDAP enables you to manage Host On-Demand configuration information by arranging those users into a hierarchical tree of groups. The Host On-Demand private data store is not arranged hierarchically; therefore, migrating your configuration information to an LDAP directory may change the relationship of your users and groups. Specifically, all groups and their users become members of one of the groups they were members of before migration. Users that are members of multiple groups will not lose configuration information as a result of migration. However, users are not allowed to belong to multiple groups using LDAP so they will be assigned to one group.

For more information on LDAP see:

- *LDAP Implementation Cookbook*, SG24-5110
- *Understanding IBM SecureWay FirstSecure*, SG24-5498
- *Lotus Notes and Domino R5.0 Security Infrastructure Revealed*, SG24-5341
- *IBM SecureWay Host On-Demand 4.0: Enterprise Communications in the Era of Network Computing*, SG24-2149-01, available at <http://www.redbooks.ibm.com>

2.11 Host On-Demand administration

Host On-Demand administration is done from a Web page. The page can be reached from any browser, meaning you can run both client and administration functions remotely. When initially connecting to the Host On-Demand main menu URL, shown in Figure 9, you will see both administrator and client options.

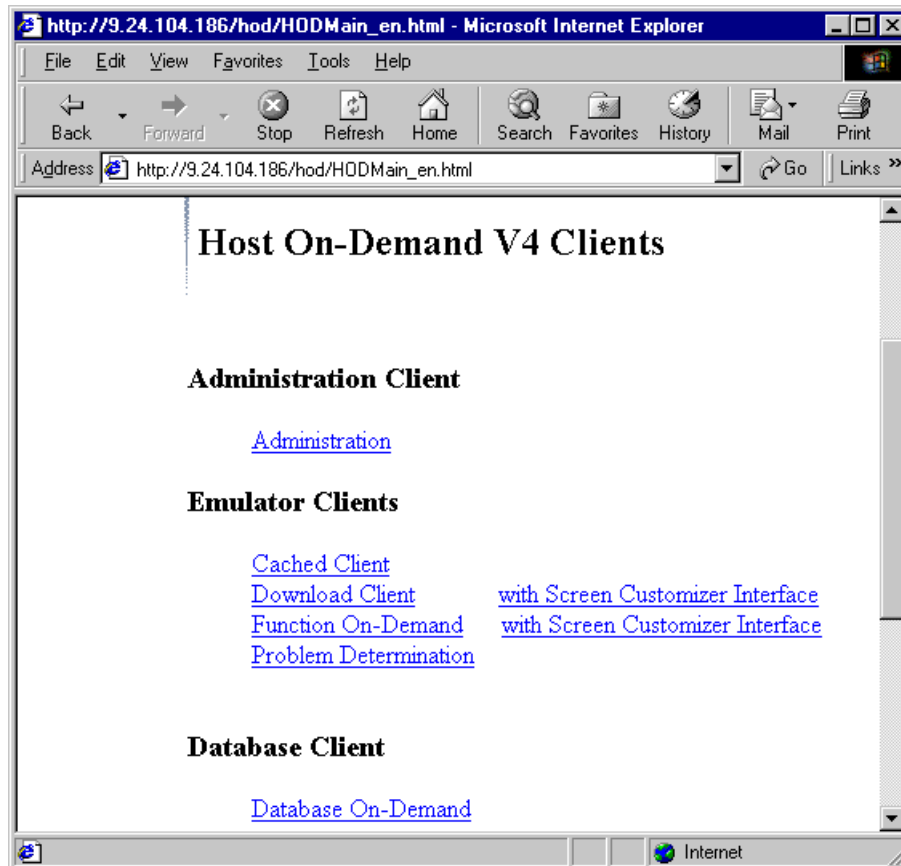


Figure 9. HOD main menu

Clicking on the **Administration** link will cause the administration applet to be downloaded into the browser environment. The applet is signed by IBM so the browser recognizes it as originating from a trusted source and shows you a dialog window asking your permission to install and run it.

A user ID and password are required to logon to the administrator. Seven tab panels allow the administrator to perform basic administration tasks:

- **Services** - start/stop the Express and Redirector services.
- **Users** - create new users and groups and set default sessions for them.
- **Redirector** - create new Redirector instances and set their configurations.
- **Express** - set the Express server configuration.
- **Database** - manage SQL statements and set database options for users and groups.
- **License** - show license count and manage License Usage Management (LUM).
- **Directory** - set the LDAP server configuration and invoke migration to LDAP.

2.11.0.1 User configuration

Users must be defined to Host On-Demand. The concept of user management is that groups are created and sessions defined for the group. The sessions contain specific information required for connection. Users are defined and added to the groups, giving them access to the sessions. Specific sessions can also be defined for individual users.

Users are not required to have a password. A general user can be defined, for example GUEST, with no password, for general access.

The structure and options for users and groups in the administrator are shown in Figure 10.

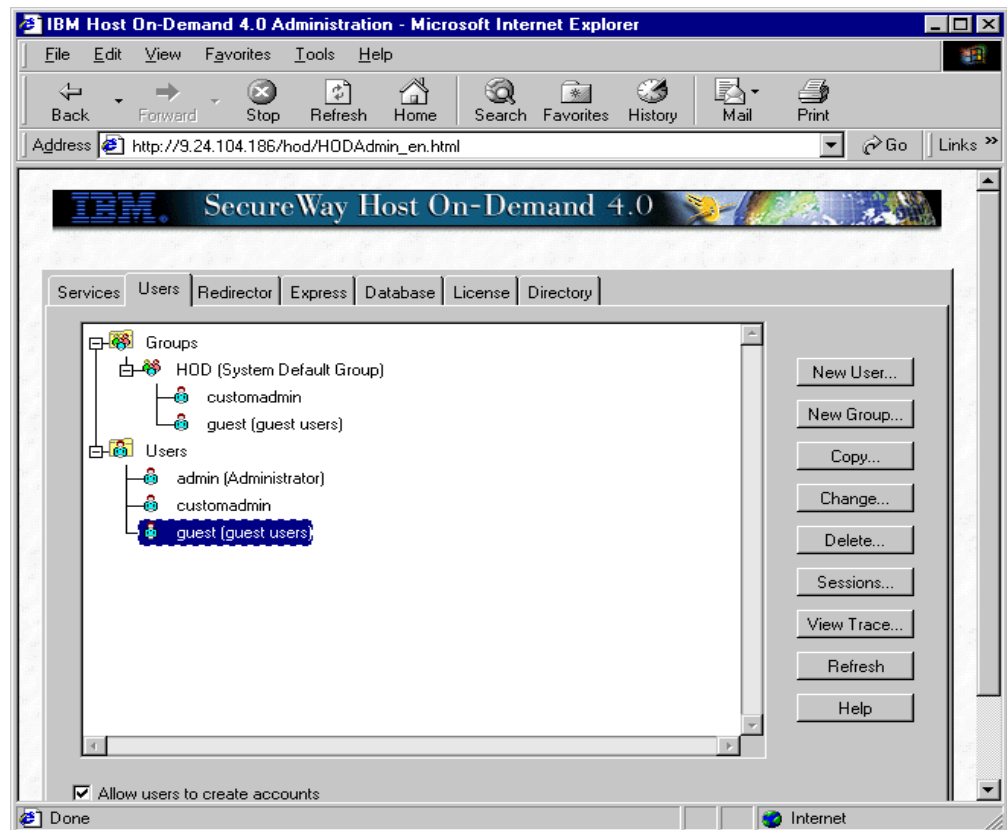


Figure 10. User and group administration

When adding a new user, the administrator specifies a user ID and optionally, a description, password, group membership, and options determining whether the user can save preferences or change the password.

Since the password is optional, it is possible to create an ID that many people can use without a password. In the case of creating a general ID for more than one person, it is probably best to prevent the user from saving preferences.

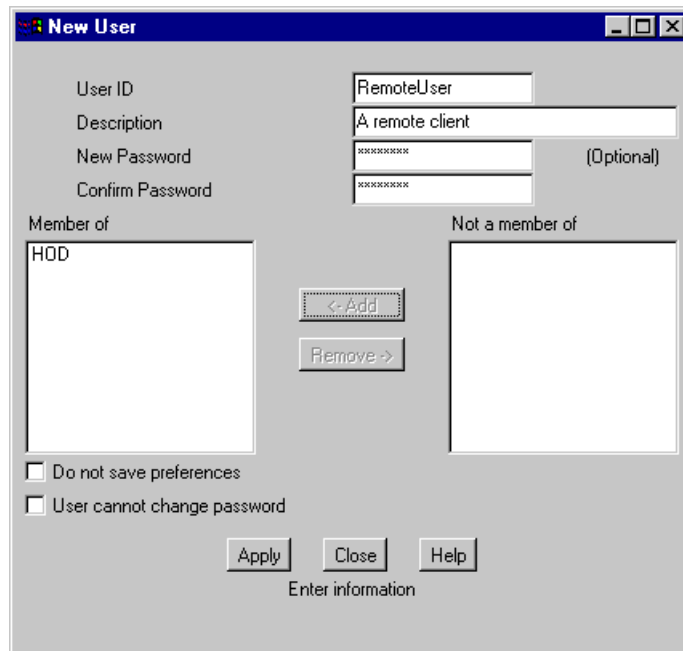


Figure 11. Administration New User window

2.11.0.2 Session configuration

Session definitions define the host connectivity and can be created for a user or for a group. Defining the sessions at the group level reduces the amount of administrative work involved in setting up the HOD environment. Once the user or group has been created, the option to add a session is enabled. A dialog window will be displayed that shows existing sessions and allows you to configure new sessions for the selected user. Figure 12 shows this window and the options available. In this case, one 3270 display session has already been defined.

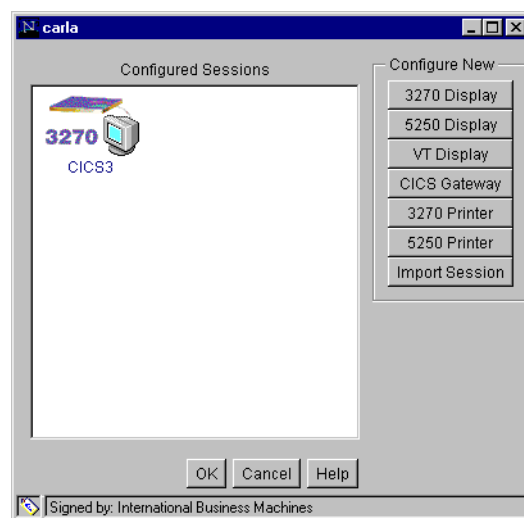


Figure 12. Administration session definition window

Clicking the 3270 Display button in Figure 12, for example, will bring up a panel allowing you to enter the details required for a connection.

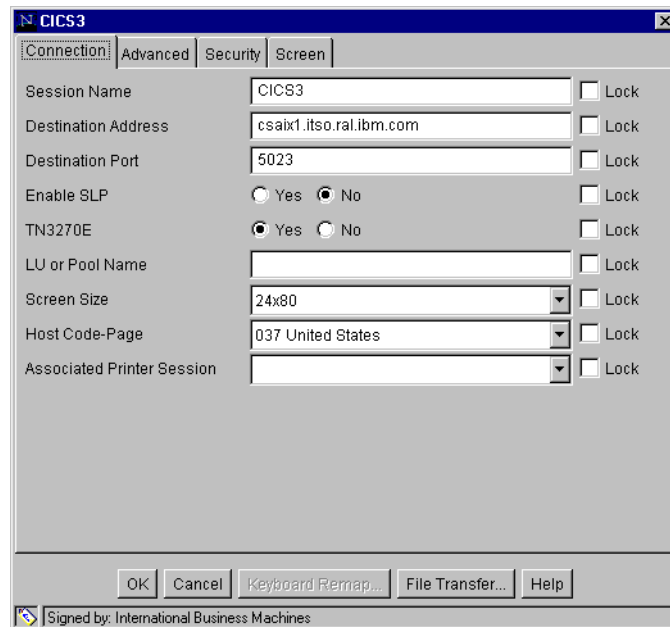


Figure 13. HOD 3270 session definition

It is enough, for this example, to specify the TN3270 server IP address. The rest of the information has been supplied by default.

Once the session is defined, the clients logging on with the user ID or group that has this session available, will find this icon on their HOD menu.

2.11.0.3 SLP

If using SLP, the IP address in Figure 13 would be left blank and the Enable SLP box would be selected. When using SLP support, the LU or Pool Name field can only contain a pool name. The scope name can be specified under the Advanced tab.

Figure 14 on page 28 shows the SLP options for a session definition. Using SLP also requires that the telnet server have SLP implemented. Communications Server for Windows NT Version 6 and intraNetWare for SAA as telnet server provide SLP support.

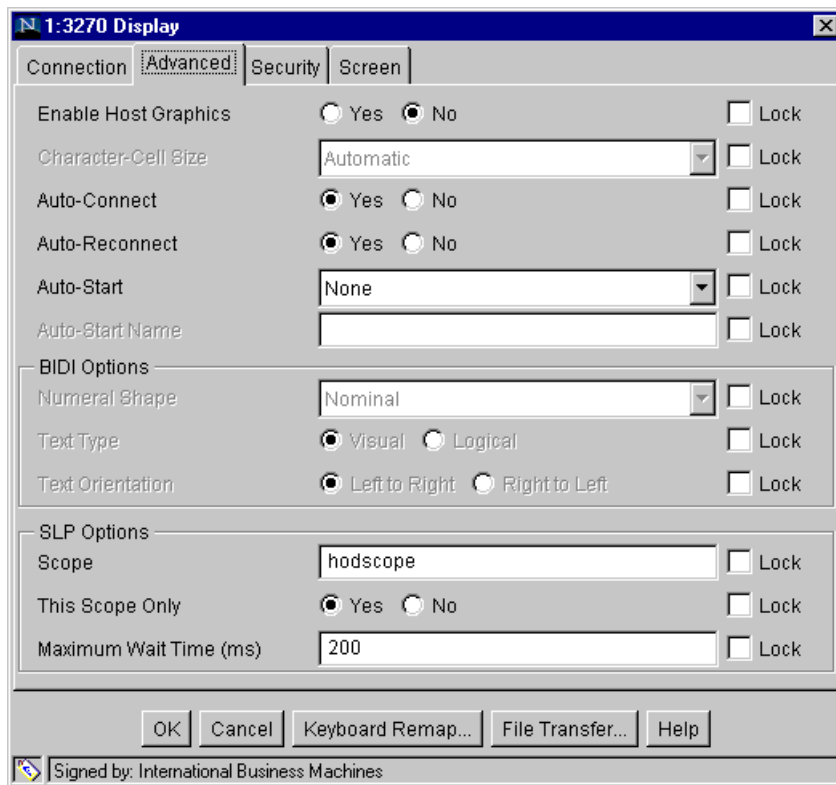


Figure 14. SLP configuration example

2.11.1 Database On-Demand configuration

Database On-Demand provides SQL access to AS/400 databases. The AS/400 JDBC driver is shipped with Database On-Demand. To illustrate how this is configured, we will take you through the administrator panels used to define Database On-Demand for a user.

Database On-Demand is configured under the Database tab of the administrator screen. The administrator mainly determines what the user is allowed to do. The actual SQL requests are defined by the user. As shown in Figure 15, the Database panel lists defined users and groups.

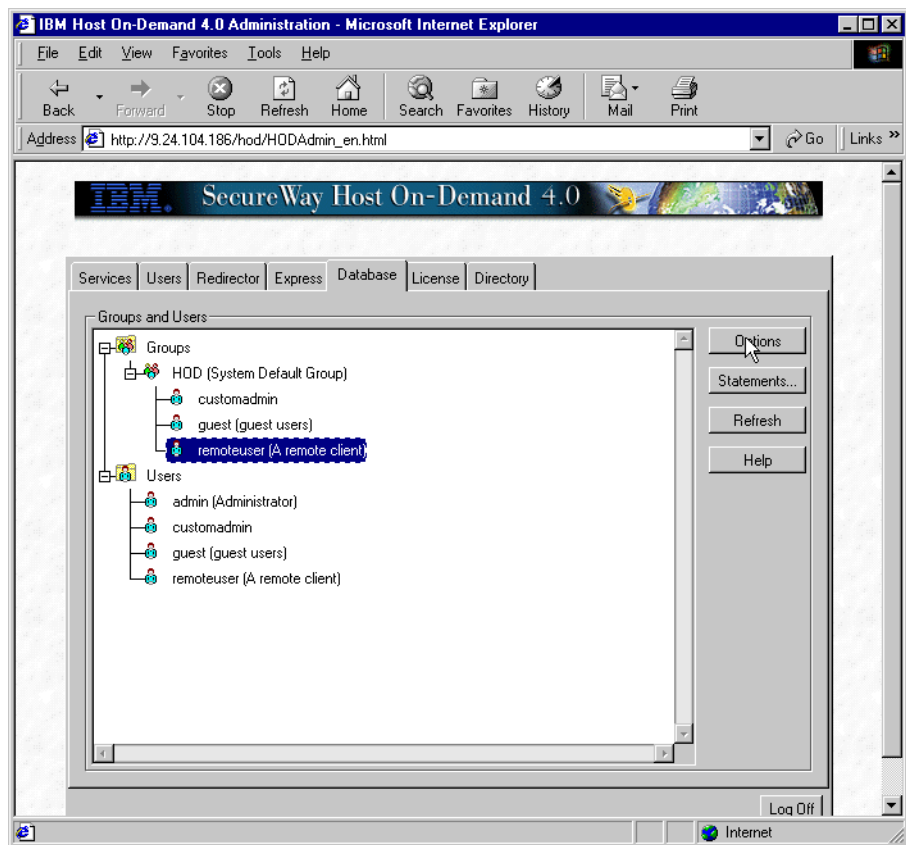


Figure 15. Database On-Demand configuration

Expand the **Groups** or **Users** tree and select the appropriate user or group and click the **Options** button. The database options dialog window will be opened for the selected user.

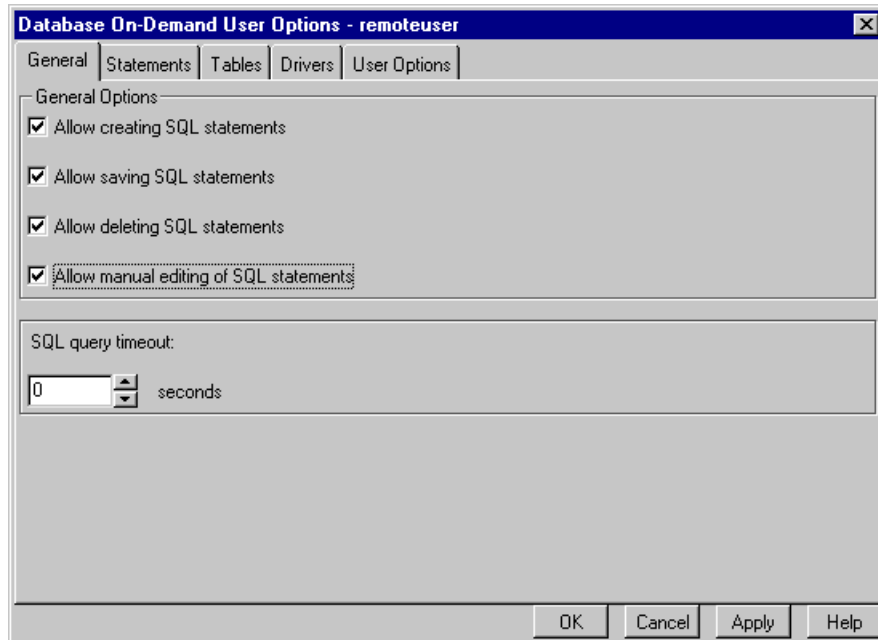


Figure 16. Database On-Demand general configuration

Notice it has five tabs:

- **General** - user access control to SQL statement creation and update.
- **Statements** - user access control to specific select/update/insert/delete statements.
- **Tables** - control which kinds of tables to show.
- **Drivers** - JDBC driver registration. You may register drivers other than the AS/400 JDBC driver, but this is not recommended or supported.
- **User Options** - minor database options for users.

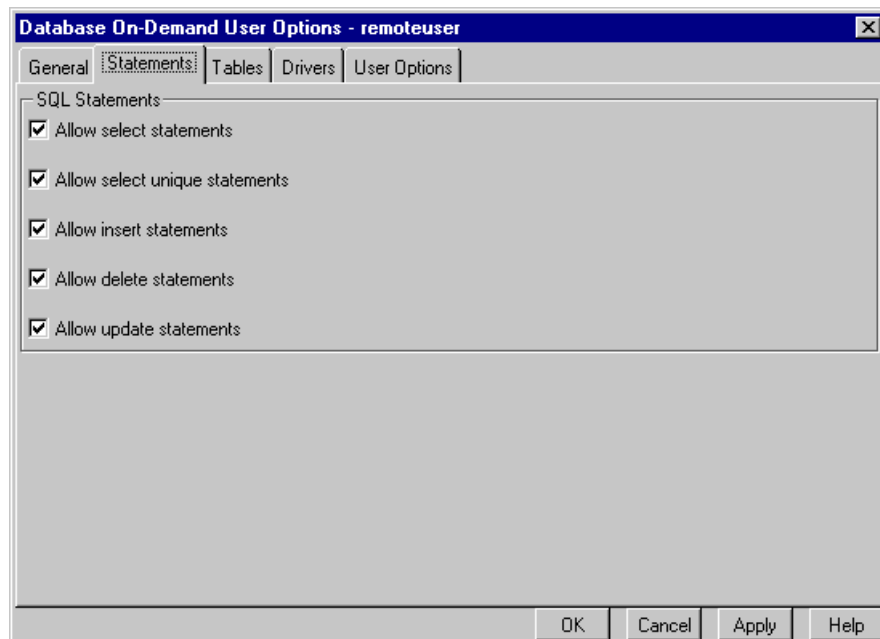


Figure 17. Database On-Demand Statements panel

Use the Tables tab to specify the types of tables the user is allowed to access.

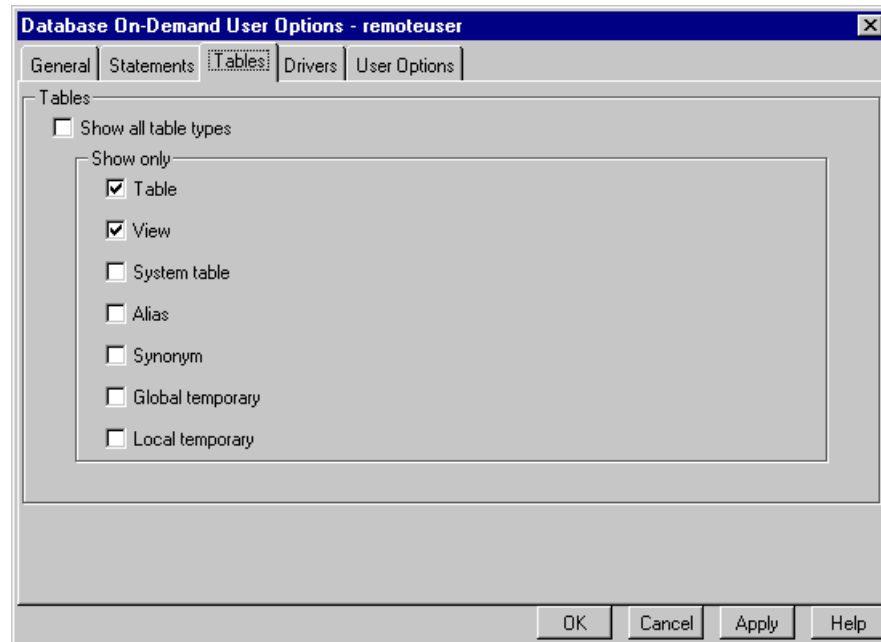


Figure 18. Database On-Demand Tables panel

The User Options tab allows the administrator to restrict what the user can configure.

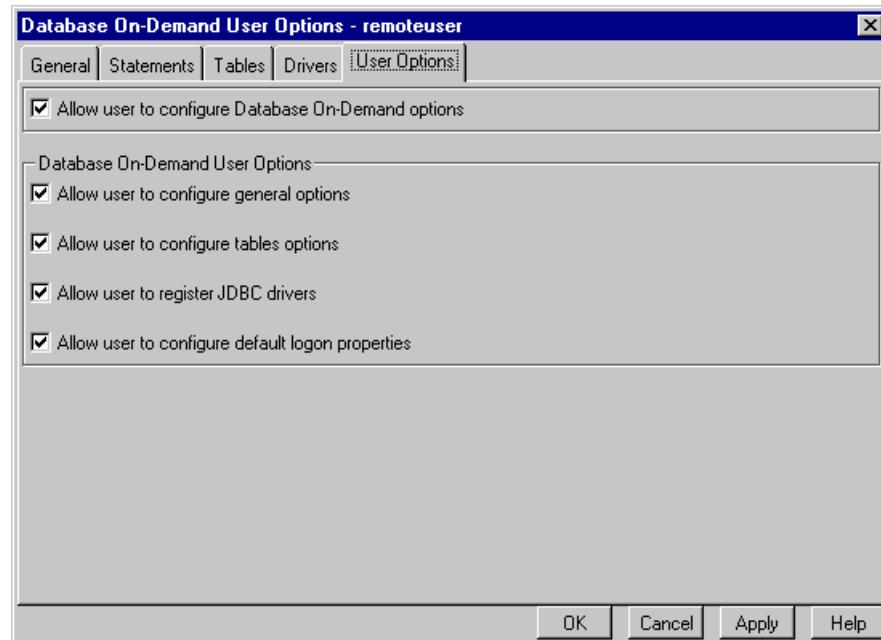


Figure 19. Database On-Demand User Options

2.12 Using the client

Now we are going to consider the client perspective. Once the administrator has defined users, the users can log on to the host using Host On-Demand clients.

The Web browser can be pointed directly to the URL, for example, `http://<server>/hod/HOD.html`. Or, by going to the main page at `http://<server>/hod/HODMain_en.html`, you can choose the client function you want without remembering each specific URL.

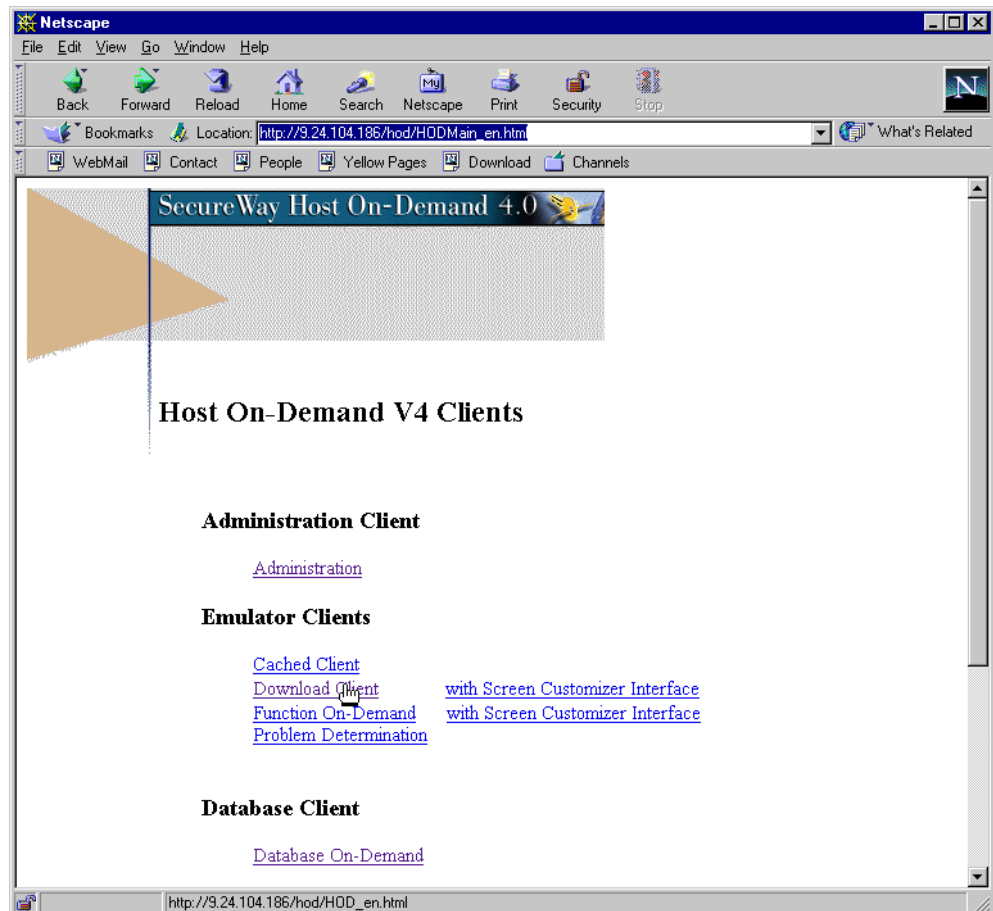


Figure 20. Host On-Demand main page

The main page gives you several emulator clients to choose from. Information on these clients can be found in 2.2, "Host On-Demand clients" on page 13.

Clicking on the desired client starts the applet download, or in the case of the cache client, the code is checked to see if a download is necessary. You will need to log on with the user ID and optionally a password defined earlier by the administrator.

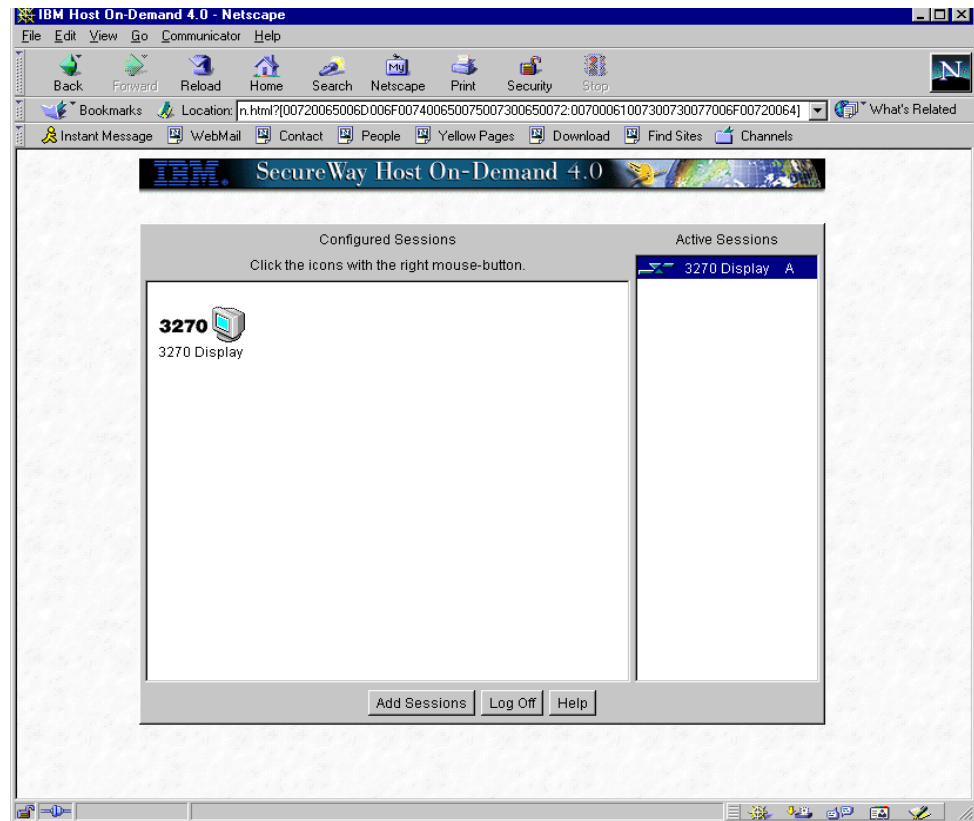


Figure 21. Host On-Demand client main panel

Once the applet is downloaded, the user will see the sessions that have been defined by the administrator for the user or group in the left panel. In this case there is one session defined. The administrator used the default name (3270 Display) but could have used something more descriptive. Double clicking the session icon will initiate the connection to the host. Active sessions are displayed in the right panel.

The user may also define new sessions by right-clicking the icon, selecting the pop up menu **copy** option, and changing the properties of the new session.

Starting an emulator session will give you a window that looks much like host sessions. Figure 22 shows a 3270 session. Host On-Demand has added a menu and toolbar providing basic operations such as copy and paste, macro recording, file transfer, etc.

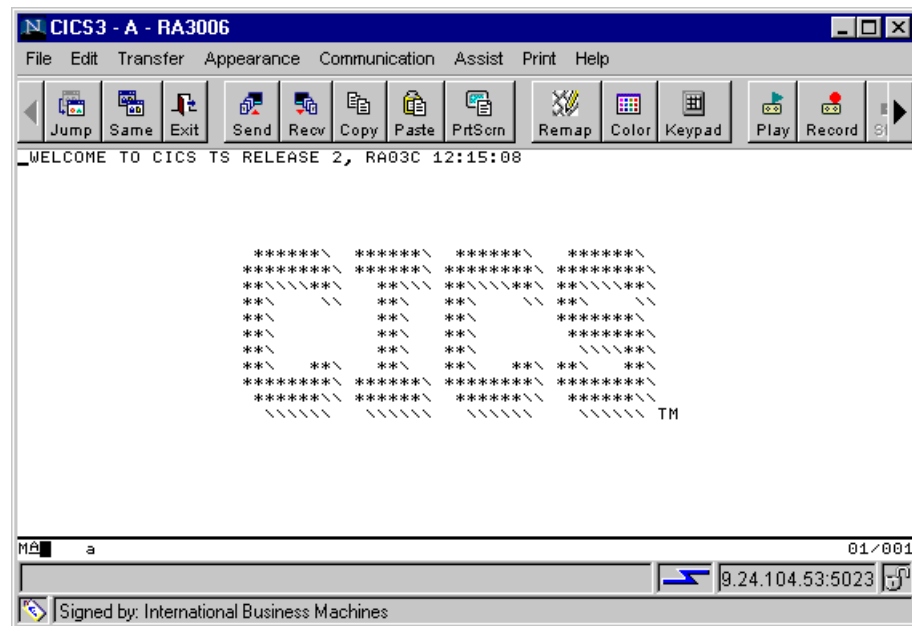


Figure 22. Host On-Demand terminal emulation window

Macro recording is of particular interest. Based on the Host On-Demand Macro bean, this feature allows you to build macros consisting of command sequences that perform actions on the host. Macros are stored using XML script. This is a great way to store frequently used actions for repeated use. Macros can be created using the **Assist->Macro** pull-down on the toolbar, or by starting the Macro Manager from the toolbar. The Macro Manager allows you to add functions other than recorded keystrokes to the macro, including:

- Smart waits which cause a macro to wait during playback until it recognizes a screen according to set conditions.
- Prompts that allow you to type in information that varies or which you do not want displayed during playback.
- Data extraction that will retrieve data from a host application and put it into an applet or graphical user interface by using the MacroExtractEvent method of HACL. A macro with data extraction in it is particularly useful for application developers who want to retrieve data from the host without knowing the structure. The data extraction function has no significance with regard to playback through a Host On-Demand emulator.

2.12.1 Using Database On-Demand

The database client for Database On-Demand is also available from the main page shown in Figure 20 on page 32. Clicking the Database On-Demand link begins the applet download.

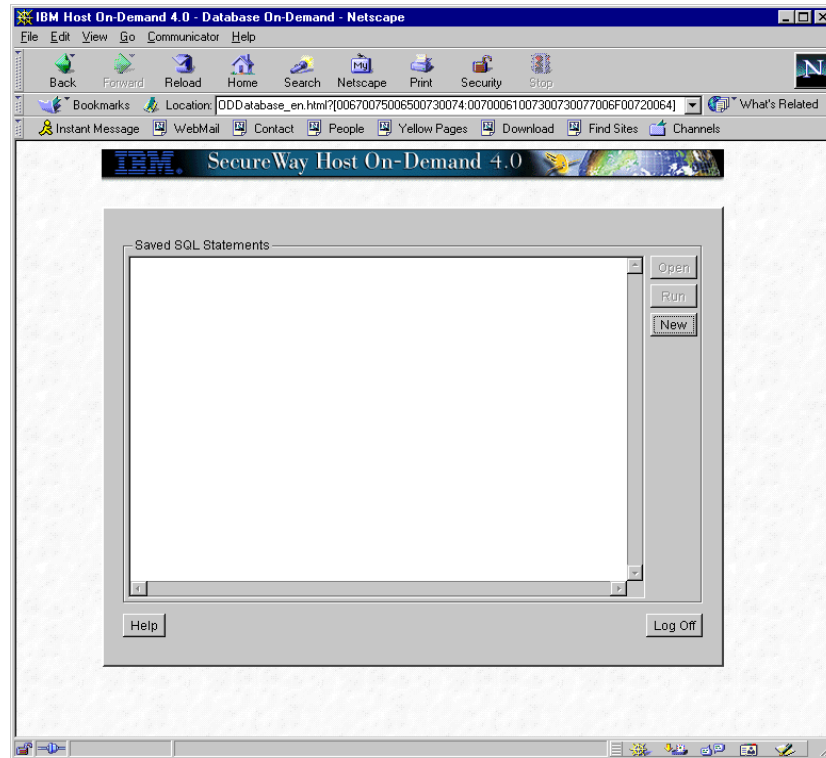


Figure 23. Database On-Demand main panel

Initially, the main panel shows an empty list of SQL statements, and only the **New...** button is enabled. Clicking on it will bring up the database **Logon** panel:

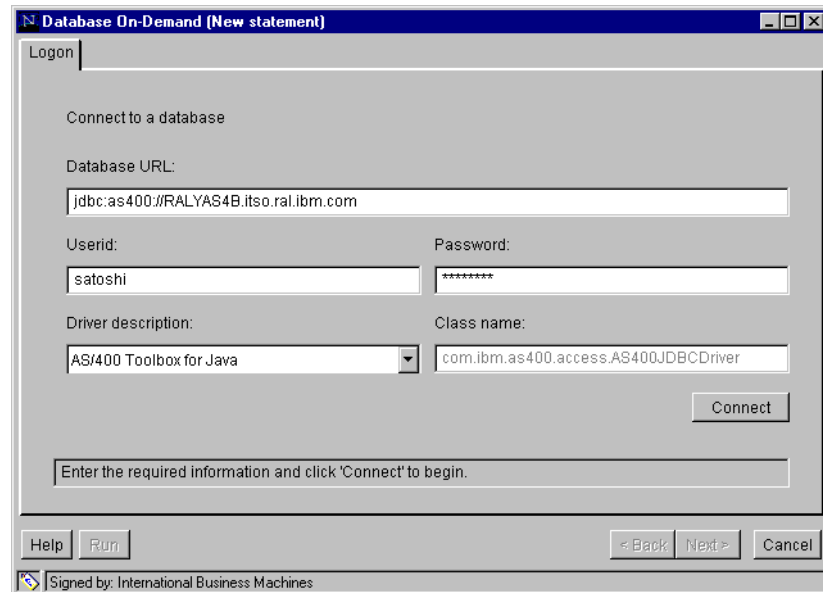


Figure 24. New Statement wizard - logon panel

This panel allows you to define the database and logon information for the target database, including the database name, account information and the desired

driver. The AS/400 driver is supplied with HOD and is the only supported driver. Others may be added by the administrator.

The database name field is the address of the database you want to work with. The AS/400 Toolbox for Java JDBC driver requires the format:

```
jdbc:as400://systemname
```

where systemname is the fully-qualified network name of the database host.

The Connect button will establish a connection to the database. If the connection succeeds, information about the database will be available to help you build the SQL statements. For example, the Tables tab, shown in Figure 25, will show a list of the database tables.

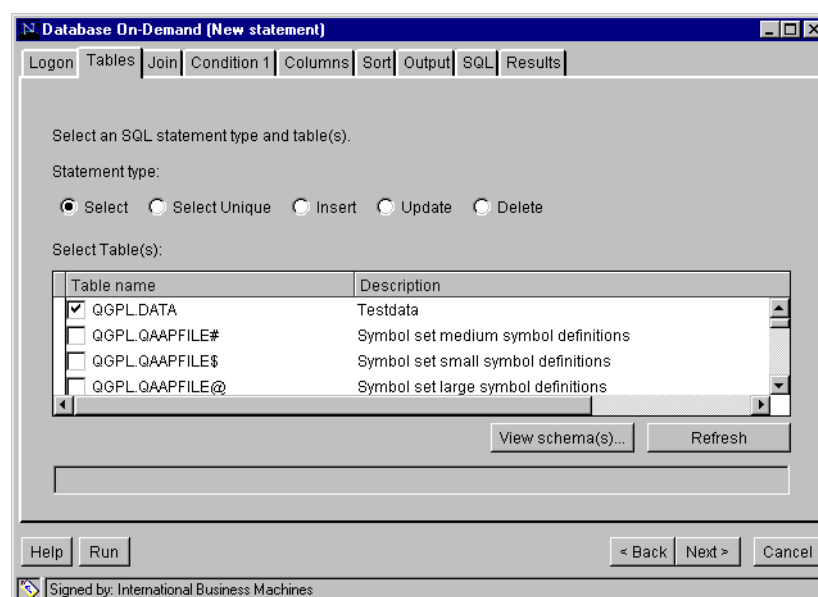


Figure 25. New Statement wizard - Tables panel

To build an SQL statement, select the tables needed in the Tables tab. In Figure 25, a table has been selected. Clicking **Next** will bring you to the Condition panel, shown in Figure 26, where you specify the condition statement (the `WHERE` clause in the `SELECT` statement).

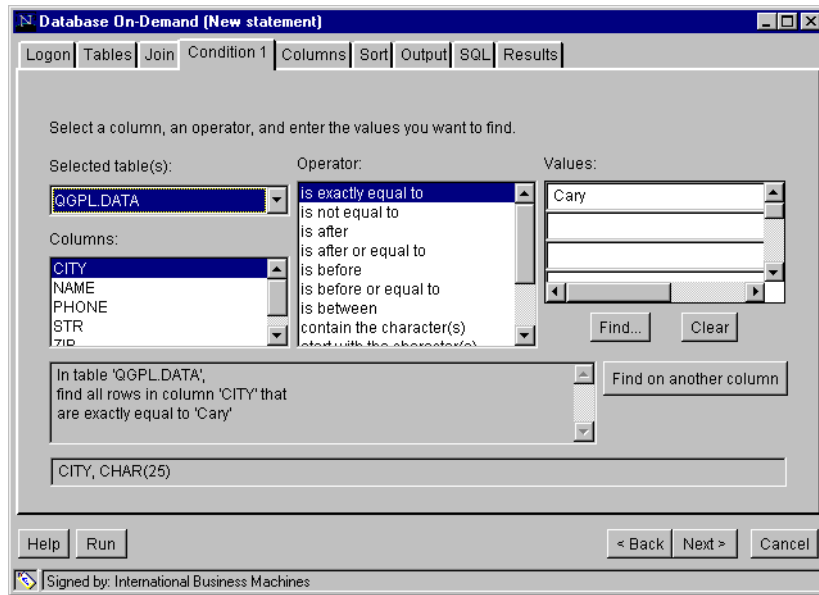


Figure 26. New Statement wizard - Condition panel

For example, in Figure 26, we have specified we want all the records in the QGPL.DATA table where the value of the CITY column is exactly equal to Cary. Clicking **Next** will take you to the Columns panel shown in Figure 27.

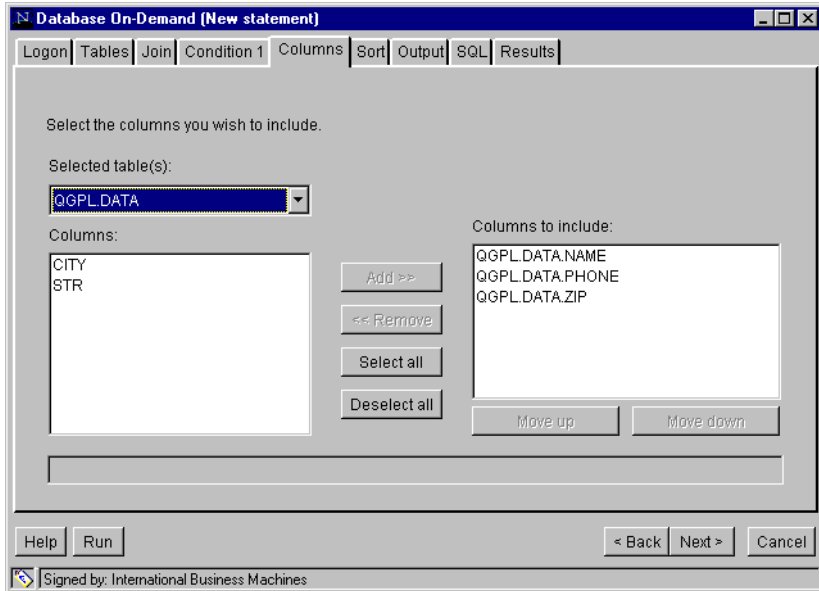


Figure 27. New Statement wizard - Columns panel

The Columns panel allows you to choose the fields that will be displayed in the query results. In Figure 27, the data in columns QGPL_DATA.NAME, QGPL_DATA.PHONE, and QGPL_DATA.ZIP will be included in the results. Clicking the **Next** button will take you to the Sort panel shown in Figure 28.

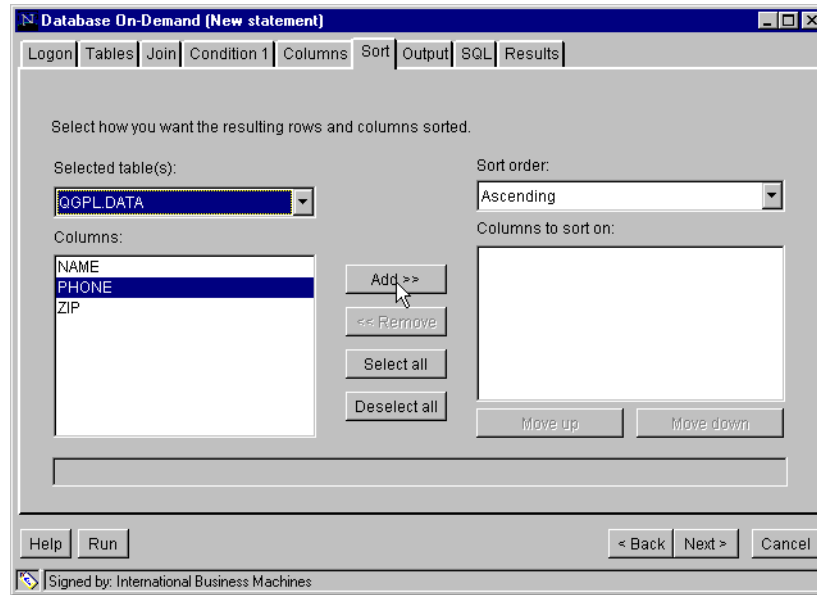


Figure 28. New Statement wizard - Sort panel

Here you can set the order you want the data sorted in. For example, in Figure 28, to sort the data by phone number, **PHONE** would be selected and added with the **Add>>** button. This defines the SQL **ORDER BY** clause of the **SELECT** statement. Clicking **Next** will take you to the Output panel.

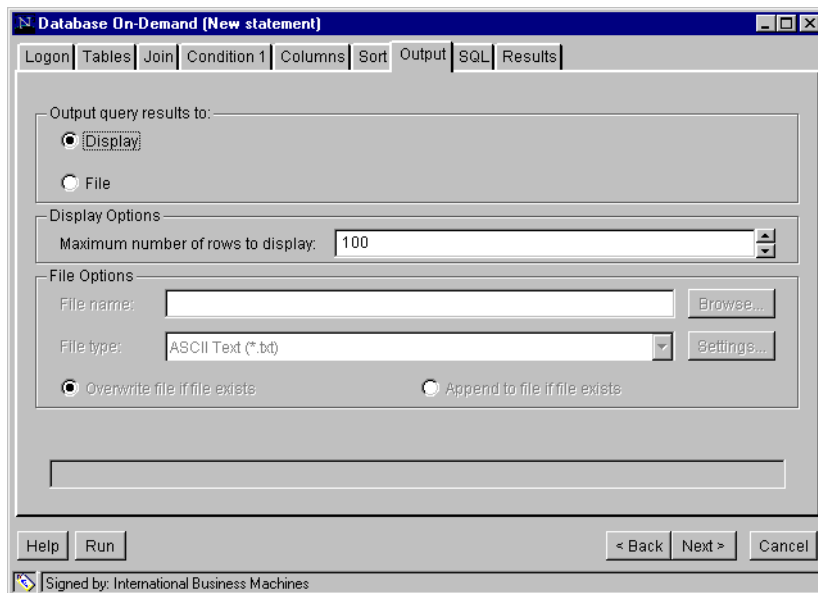


Figure 29. New Statement wizard - output panel

The Output panel, shown in Figure 29, allows you to specify whether the output goes to the display or a file and can be used to limit the amount of output. Clicking **Next** will take you to the last definitions panel.

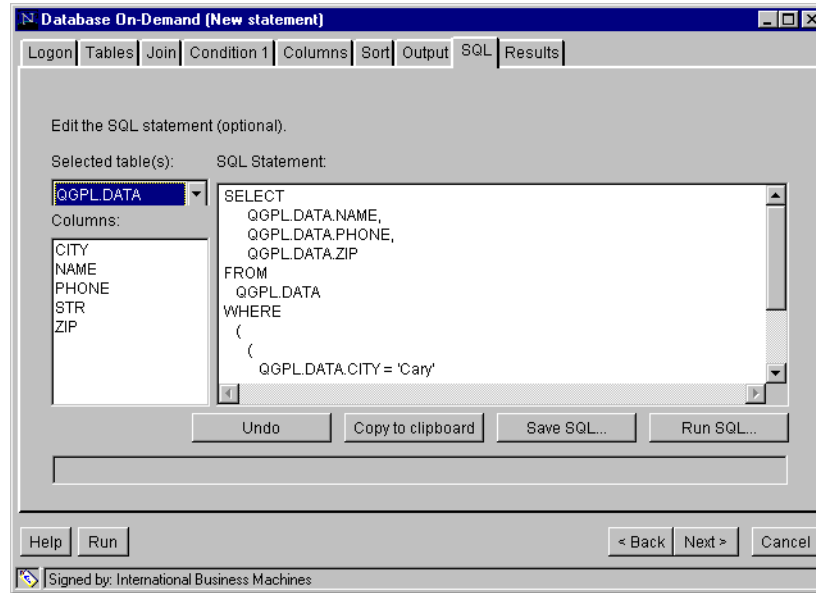


Figure 30. New Statement wizard - SQL panel

The SQL panel, shown in Figure 30, will show you the SQL `SELECT` statement resulting from the selected options. The options can be corrected by using the **Back** button to return to the previous panels. Clicking **Run SQL...** will execute the statement.

You can now view the query results in the Results panel, by clicking **Next**.

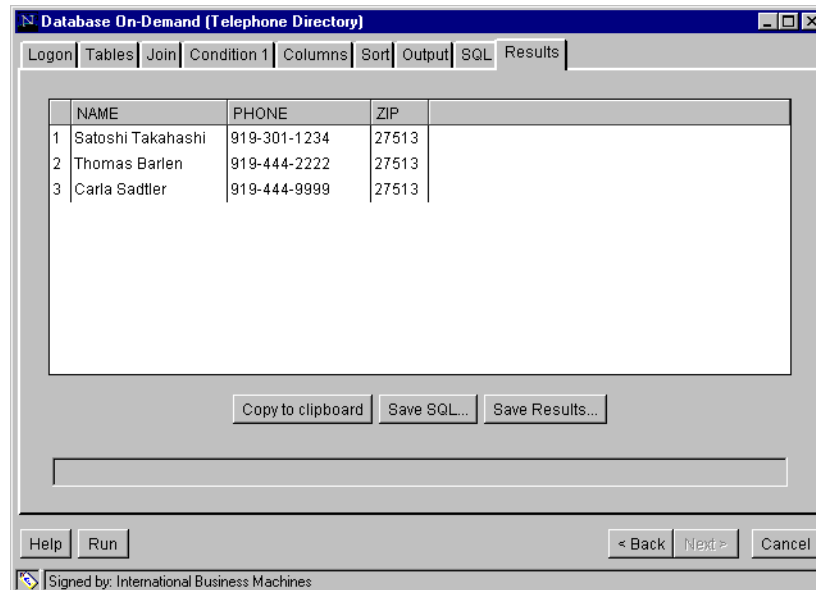


Figure 31. New Statement Wizard - Results Panel

Once the SQL statement has been built it can be saved in the list of the Database On-Demand main panel, so that it can be run at a later time by clicking the **Save SQL...** button. An administrator can copy SQL statements from one user for the use of another.

The Database On-Demand main panel now shows the statement just created in the list.

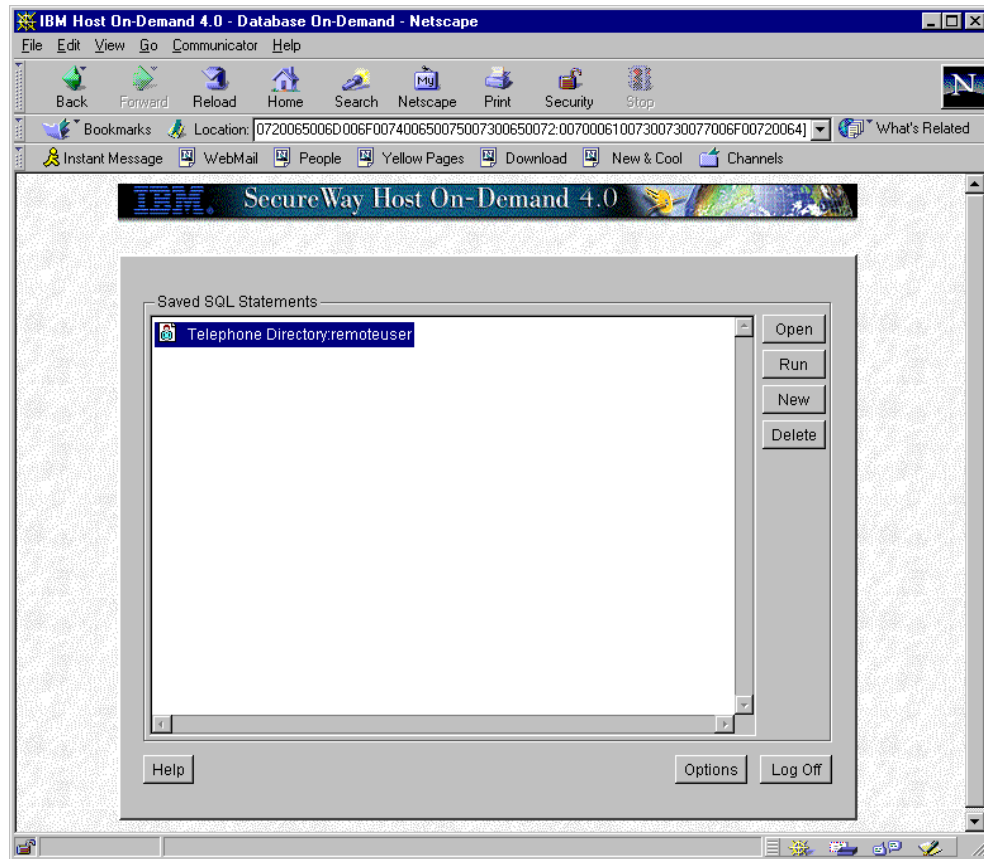


Figure 32. Database On-Demand main panel

This SQL statement can now be run by selecting it in the list and clicking the **Run** button.

2.13 Where to find more information

- *IBM SecureWay Host On-Demand 4.0: Enterprise Communications in the Era of Network Computing*, SG24-2149-01, available at <http://www.redbooks.ibm.com>
- *Programming with the Host Access APIs*, SG24-5856
- <http://www.ibm.com/software/network/hostondemand/>

Chapter 3. IBM SecureWay Host Publisher

IBM SecureWay Host Publisher Version 2.1 is a key component of IBM's Host Integration software portfolio and provides significant changes from the previous releases of Host Publisher. In this book when we talk about Host Publisher we are referring to IBM SecureWay Host Publisher Version 2.1. Host Publisher provides a Web-to-host solution designed to address the unique characteristics of the Internet. It enables Web integration with existing 3270, 5250, VT, JDBC, and Java host applications, without requiring any changes to those existing applications and using industry-standard HTML Web pages to support end users running non-Java browsers.

Host Publisher is designed to enable enterprises to jump-start e-business applications from existing host applications and data. It delivers host information via HTML to standard Web browsers. It is Java-based and supports AIX, OS/390, Windows NT and Sun Solaris operating systems. Support for OS/400 will be added in a future release.

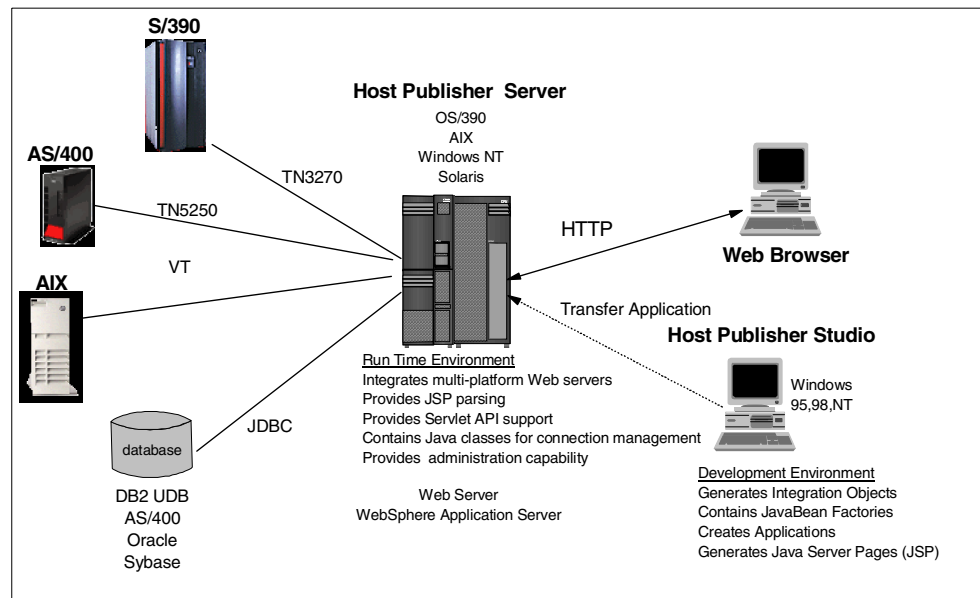


Figure 33. Host Publisher

As illustrated in Figure 33, there are two major components in IBM SecureWay Host Publisher:

- Host Publisher Studio, which provides easy-to-use tools to assist in the creation of Web-to-host integration projects.
- Host Publisher Server, which consists of IBM WebSphere application server and Host Publisher run time components.

Host Publisher Studio tools allow you to create Integration Objects that encapsulate the interaction and data retrieval with host applications. Integration Objects are reusable objects that can be used in Java applications created outside Host Publisher. Likewise, interactive development environment (IDE) tools can be used to add new business logic to the applications Host Publisher creates.

The Host Publisher Studio generates fully customizable HTML output with embedded Java Server Pages tags to invoke the Integration Objects.

Other new features included in IBM SecureWay Host Publisher Version 2.1 are:

- IBM SecureWay Network Dispatcher is shipped with Host Publisher to provide load balancing and failover capabilities for Host Publisher Servers.
- Fully customizable HTML Web pages are generated that the application builder can enhance using his or her favorite HTML editor.
- User-defined connection pools are used during run time to cache connected, logged on, and ready connections to improve response time.
- Integration Object chaining enables you to break the application into steps to provide greater flexibility and performance.
- You can integrate multiple data sources, including host and database applications, into a single Web page on a client's browser, making host access transparent to the user.
- Inclusion of a run time version of IBM WebSphere Standard Edition (Host Publisher does not include WebSphere for the S/390 or the AS/400 versions. The WebSphere Standard Edition that comes with the operating system is a prerequisite).
- Works with IBM's connector solutions, such as MQSeries which significantly extends the breadth of existing applications that can be supported.
- Host Publisher supports Secure Sockets Layer (SSL) encryption and authentication, as well as DES-encrypted passwords, to provide a high level of security. Individual application security at the Web server can be provided with the addition of On-Demand Server.

Host Publisher provides backend connectivity for the following applications:

- JDBC-compliant database applications
- 3270 applications via Telnet 3270
- 5250 applications via Telnet 5250
- VT (VT52, VT100, VT220) applications
- Java applications

3.1 Packaging and software requirements

IBM SecureWay Host Publisher is available via the following product offerings:

- IBM SecureWay Host Publisher V2.1 for AIX, Windows NT and Solaris
- IBM SecureWay Host Publisher V2.1 for S/390 (available 1/21/2000)
- IBM SecureWay Communications Server for Windows NT Host Publisher feature

Additionally Host Publisher capability is available as part of IBM SecureWay Host Integration Solution.

In addition to the Studio and Server, Host Publisher includes the following products from IBM on the installation media:

- IBM HTTP Server 1.3.3.3
- IBM WebSphere Application Server Standard Edition 2.0.3.1 (AIX, Windows NT, Solaris)

- JDK 1.1.8 (AIX and NT)
- IBM SecureWay Network Dispatcher

In general the following software is required:

- A supported Web server. Host Publisher includes IBM HTTP Server on the CDROM.
- JDK 1.1.8
- IBM SecureWay WebSphere Application Server (V2 only) Standard, Advanced, or Enterprise Edition. Host Publisher includes and integrates the WebSphere Standard Edition V2.0.3 in the AIX, Windows NT, and Solaris versions.
- Host Publisher Studio
- Host Publisher Server
- Database support such as IBM DB2 Universal Database and databases from Oracle and Sybase (not provided with Host Publisher). Database support is only required if you are planning to work with database Integration Objects.

Host Publisher Studio

One of the following operating systems:

- Windows NT 4.0 with ServicePack 4 or later
- Windows 95
- Windows 98

Host Publisher Server

A machine with at least 128 MB RAM and one of the following operating systems:

- Windows NT 4.0 with Service Pack 4 or later
- AIX V4.2.1 or AIX V4.3.2 or later
- OS/390 Version 2 Release 7 or later with WebSphere Application Server Standard Edition Version 1.1
- SUN Solaris V2.6

One of following Web servers:

- IBM HTTP Server V1.3.3.3 for Windows NT, AIX, Sun Solaris(on Host Publisher CD)
- Apache Server V1.3.2 for Windows NT, AIX, Sun Solaris
- Domino V5.0 for Windows NT, AIX
- Lotus Domino Go Webserver V4.6.2.5 for NT, AIX, Sun Solaris
- Microsoft Internet Information Server V3.x or V4.0 for Windows NT
- Netscape Enterprise Server V3.01 and V3.51 for Windows NT, AIX, Sun Solaris (V3.51 recommended)
- Netscape FastTrack Server V3.01 for Windows NT, AIX, Sun Solaris
- Lotus Domino Go Webserver V5.1 for OS/390

3.2 Components of Host Publisher

Figure 34 shows Host Publisher components and how they interact with each other. Host Publisher consists of two major components. Host Publisher Studio provides the development environment. Host Publisher Server is the run time environment.

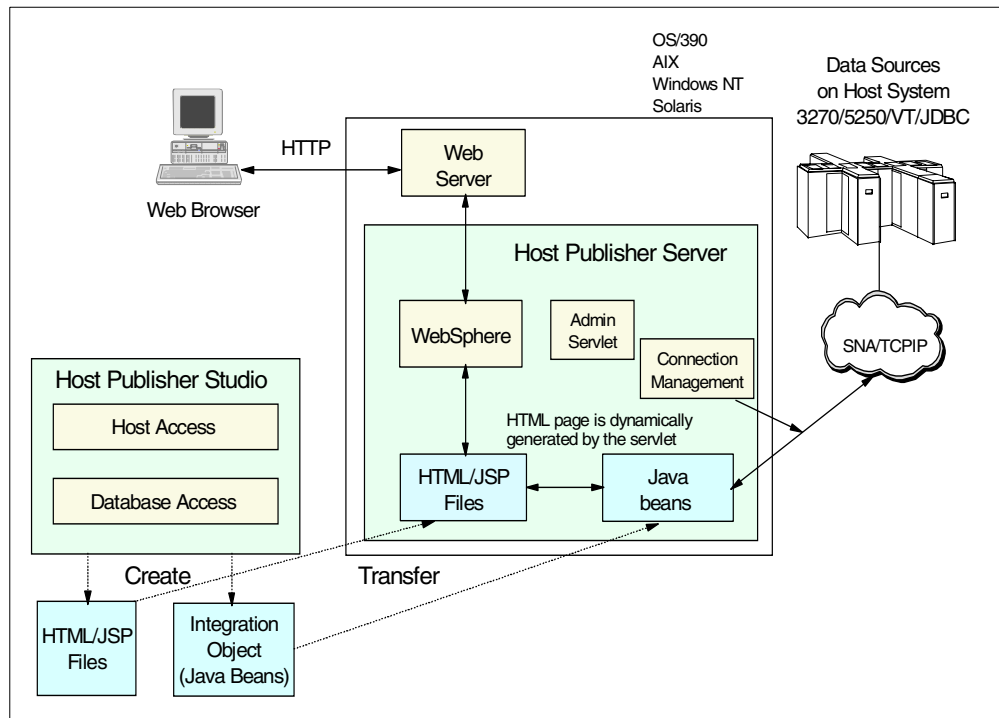


Figure 34. Host Publisher components

3.2.1 Host Publisher Studio

Host Publisher Studio is a collection of tools that assist the Web application builder in managing and creating Web-to-host projects. It uses task-oriented prompts to guide the user through the creation process, including recording host and database interactions, identifying desired data, and labeling that data for retrieval.

Host Publisher Studio consists of three tools:

1. Host Access

Host Access allows you to build Integration Objects that can access 3270, 5250, and VT applications.

2. Database Access

Database Access allows you to build Integration Objects accessing JDBC-capable databases.

The Host Access and Database Access tools generate beans, called Integration Objects, which encapsulate the interactions and data retrieval logic.

3. Host Publisher Studio

Host Publisher Studio provides the capability to create applications using JSPs to invoke Integration Objects created by the Host Access or Database Access tools. Host Publisher Studio generates a fully customizable HTML Web page. You can enhance the generated HTML with your favorite Web authoring tool, such as NetObjects Fusion or Microsoft Front Page, to meet corporate guidelines on style and image. Once the Web page is completed,

you use the Host Publisher Studio to publish it to a Host Publisher Server for production access by end users.

Host Publisher Studio can be installed on the Windows 95, Windows 98, and Windows NT operating systems. It does not have to be installed in the same machine as the Host Publisher Server.

3.2.2 Host Publisher Server

The Host Publisher Server provides the run-time environment for supporting Web applications created with Host Publisher Studio. It consists of the IBM WebSphere Standard Edition Application Server and other run-time components such as connection management, license monitoring, run-time administration, and log and trace management.

The Host Publisher Server components are:

- IBM WebSphere Standard Edition Application Server

WebSphere Application Server is mandatory for the Host Publisher Server. Applications built by the Host Publisher Studio consist of JSPs and Integration Objects. WebSphere parses the JSP pages and executes the Integration Objects.

- Connection management

Applications accessing hosts require a connection with the host machines. By default, Host Publisher applications connect to and disconnect from the host each time they are invoked. However, sometimes for performance reasons, it is better to keep a connection and reuse it. The connection manager manages this pooled connection.

- License monitoring

This component tracks the number of requests per minute and automatically logs a message when the value exceeds the number of licenses purchased. One license is equivalent to the right to execute one Integration Object per minute.

Host Publisher Server can optionally track license usage history over time. This history maintains the maximum number of licenses used (or Integration Objects invoked) during a one-hour period, logging this information to a file each hour.

- Run time administration

This component allows you to manage your Host Publisher Server by using a standard Web browser. The component itself is a servlet and is invoked from WebSphere.

- Log and trace management

This component makes log files and trace files for problem determination. You can control tracing and logging from the run time administration panel.

3.3 Load balancing with IBM SecureWay Network Dispatcher

The IBM SecureWay Network Dispatcher is shipped with Host Publisher. Network Dispatcher enables you to balance client request loads across multiple Host Publisher Servers. Network Dispatcher forwards page requests from clients to

available Host Publisher Servers in either a round-robin fashion, meaning that each Host Publisher Server is dispatched a client request in turn, or based on actual server load.

Network Dispatcher provides two techniques for load balancing that can be used separately or together to provide superior load balancing results, the Dispatcher and Interactive Session Support.

Dispatcher

The Dispatcher function balances traffic among your servers through a unique combination of load balancing and management software. The Dispatcher can also detect a failed server and forward traffic around it.

All client requests sent to the Dispatcher machine are directed to the server selected by the Dispatcher as optimal according to certain dynamically set weights. You can use the default values for these weights or change the values during the configuration process. You can also choose to have advisors monitor the response time of each server and dynamically adjust the weights.

The server sends a response back to the client without any involvement of the Dispatcher. No additional code is required on your servers to communicate with the Dispatcher.

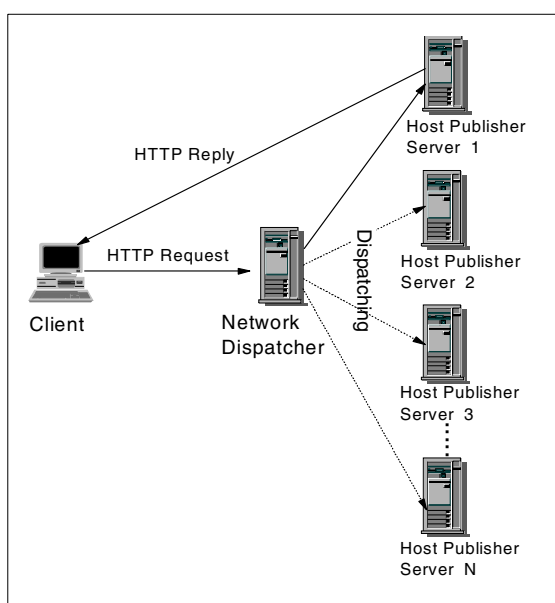


Figure 35. Network Dispatcher function

Interactive Session Support (ISS)

ISS periodically monitors the level of activity on a group of servers and detects which server is the least heavily loaded. It can also detect a failed server and forward traffic around it. Once every monitoring period, ISS ensures that the information used by the TCP/IP domain name services (DNS) server or the Dispatcher accurately reflects the load on the servers. The load is a measure of how hard each server is working. The system administrator controls both the types of measurement used to measure the load and length of the load monitoring period. You can configure ISS to suit your environment, taking into account such factors as frequency of access, the total number of users, and types

of access (for example, short queries, long-running queries, or CPU-intensive loads). Load information is provided to the ISS monitor by ISS daemons running on the target servers.

You can use the ISS function with or without a DNS server. ISS can balance server loads based on a round-robin scheme or it can collect server load information from agents running in each server and determine the least loaded server based on that information. ISS can do the load balancing itself by using DNS or it can collect load information and forward it to the Network Dispatcher for load balancing.

If you are using ISS for load balancing, a DNS server is required. This can either be an actual DNS server or, if you set up a small, separate subdomain for a new name server, a replacement name server provided by ISS. The DNS server has a generic entry that represents the Host Publisher Servers. ISS determines the server with the least load or the next to be used in a round-robin scheme and dynamically modifies the IP address associated with the generic host name in the DNS server. Clients using the generic name for connection and that DNS for name resolution will automatically use the least loaded server.

If you are using ISS to collect server load information, you can use a DNS for load balancing or you can have that information forwarded to the Network Dispatcher. The Dispatcher uses this load information, along with other sources of information, to perform load balancing.

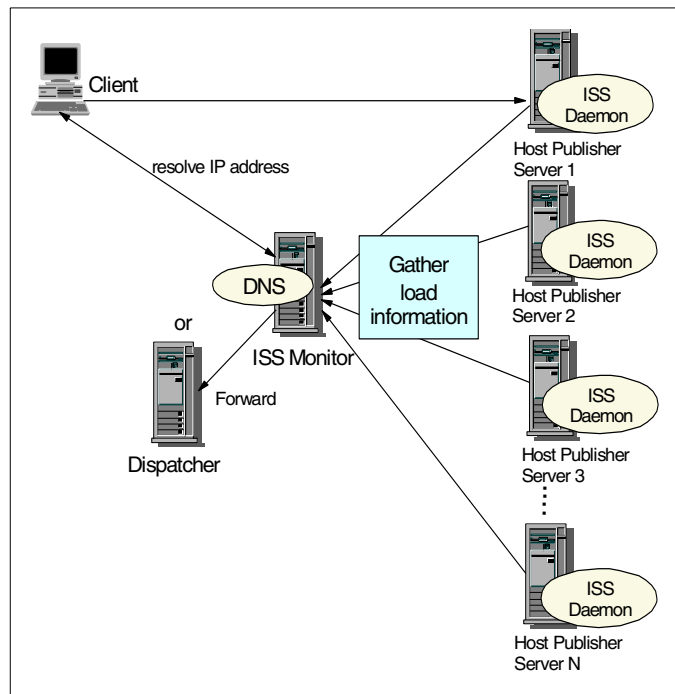


Figure 36. How ISS works

More information about the IBM SecureWay Network Dispatcher can be found in *IBM WebSphere Performance Pack Usage and Administration*, SG24-5233.

3.4 Host Publisher administration

The Host Publisher Server provides the run time environment for supporting Web applications created with the Host Publisher Studio. The run time environment includes Web-based server administration for controlling the run time and the applications it serves. The administration features are accessed from a Web browser.

The following functions can be executed from the Host Publisher Server administration:

- Check server status: You can start, stop or refresh the server.
- Administration: Under administration you can work with license management, connection pools, pool definitions, host and database connections, and user pools. Application administration allows you to list, deploy, remove and delete applications.
- Problem determination: You can view the log, set log options, view traces and set trace options.

Figure 37 shows the administration panel.

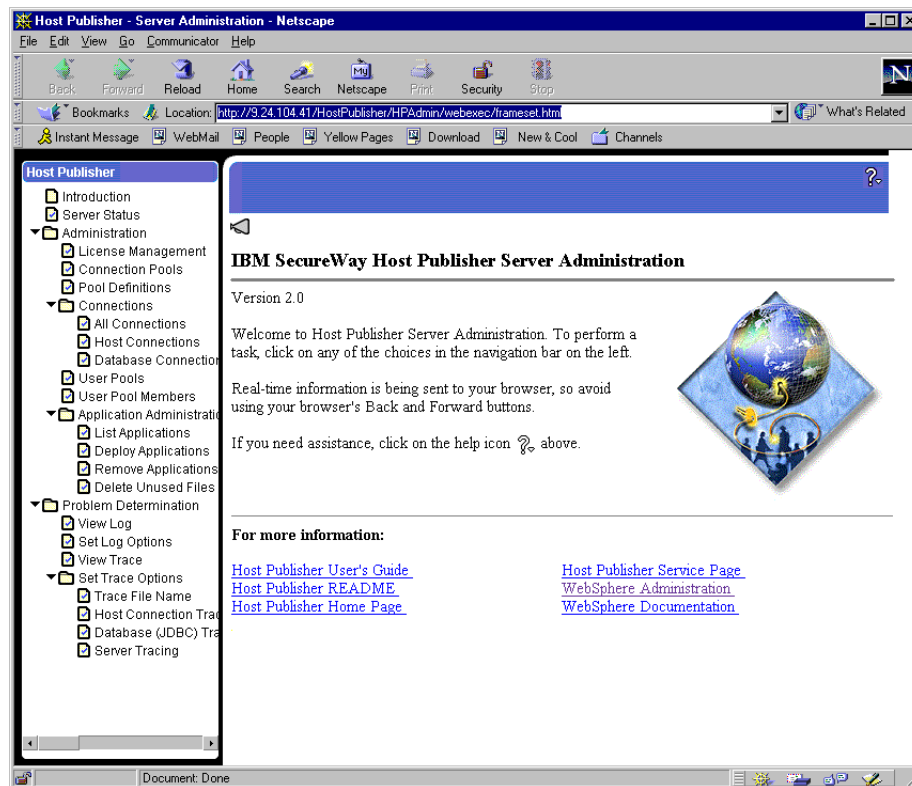


Figure 37. Host Publisher Administration panel

3.5 Creating Host Publisher applications

Applications created by Host Publisher consist of two components:

Integration Objects are Java beans created by the Host Publisher Studio tools used to access existing backend resources. They encapsulate interactions with data sources and return the desired data as JavaBean properties. Host Publisher provides the ability to develop two types of Integration Objects: Host Access Integration Objects and Database Access Integration Objects. Host Access Integration Objects are used for developing applications which interact with 3270, 5250, and VT type host applications. Database Access Integration Objects are used for developing applications which interact with JDBC-capable databases. As Java beans, Integration Objects can be enhanced by using other IDE tools and are capable of being used in Java applications created outside Host Publisher.

Java Server Pages (JSP) is a relatively new technology developed by Sun Microsystems. JSPs look like HTML files, but in JSP you can embed the results from servlets and beans. By using JSPs, the user is actually using Java applications on a server, as opposed to applets which run on the browser. This means there is no Java requirement on the browser. This enables you to separate the HTML coding from the business logic in your Web applications. WebSphere Application Server (included with Host Publisher V2.1) provides the support for JSP.

Because Integration Objects are beans, they can be invoked (instantiated) from Java servlets as illustrated in Figure 38 on page 49. They can also be instantiated from JSPs, Java applets and Java applications.

Note

The easiest way to invoke Integration Objects is using JSPs since it does not actually require you to code a Java program. When using the Host Publisher Studio to create applications, it generates JSPs for you.

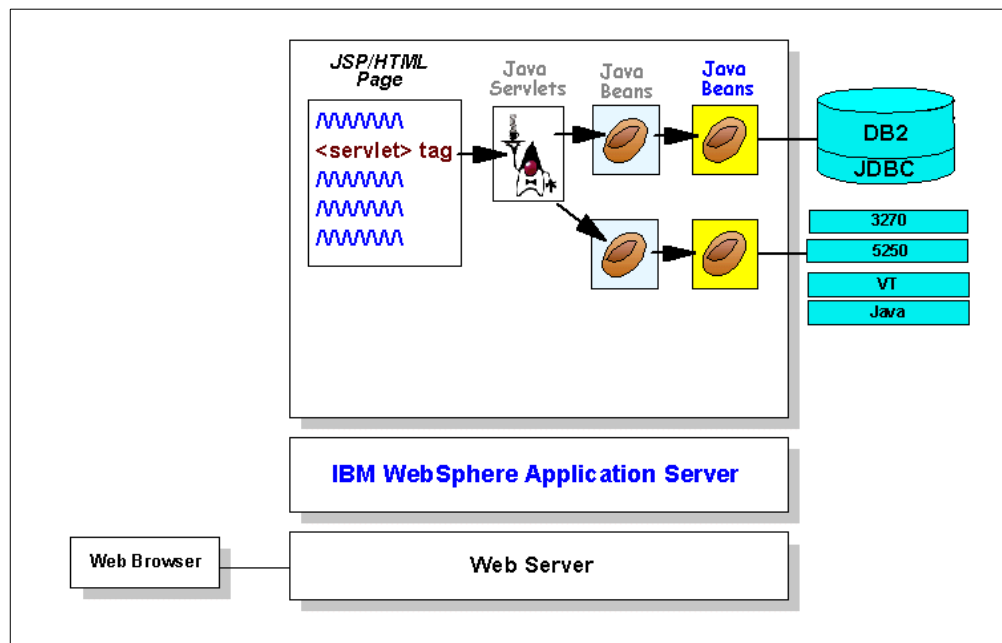


Figure 38. Host Publisher backend resources

3.5.1 Composite applications

Composite applications combine multiple Integration Objects to produce a single stream of output information to the user. Composite applications enable you to use the output of one Integration Object as input to another or fill a table with data from several different Integration Objects that access different data sources.

3.5.2 Integration Object chaining

Host Access Integration Object chaining enables multiple Integration Objects to execute in sequence, each using the same connection. You can use chaining to break up a complex application into multiple tasks, each task represented by an Integration Object. Host Publisher Studio ensures that the order in which the Integration Objects are invoked is correct. Integration Object chaining requires the Host Publisher Server and is not supported when invoking Integration Objects from custom Java applications.

There are three types of Integration Objects in an object chain: first, middle, and last. An application can have multiple middle objects.

3.6 Creating host Integration Objects

Host Publisher Version 2.1 includes a tool that allows you to easily create Integration Objects for 3270, 5250, and VT applications. This tool is called Host Access. To develop a new host Integration Object you record macros that contain information about the way you connect to the host and how you navigate to the information you want to publish. The macros themselves become part of the Integration Object you create.

3.6.1 Using Host Access

Generally, the Host Publisher Studio records the keystrokes you use and lets you define the host application screens that contain information. To show you the basics we will take you briefly through the steps involved to create a simple application. This is a high level overview only.

3.6.1.1 Configuring the host connection

The first step in creating a host access object is to define and configure a host connection. You can either select the single connection option or alternatively, you can configure a connection pool to be used by this and other Integration Objects. Figure 39 shows the connection definition screen. The normal telnet parameters like terminal type, host name, port, LU or pool name for TN3270 or TN5250, screen size, etc. The terminal types to choose from are:

- TN3270
- TN5250
- VT
- TN3270E
- TN5250E

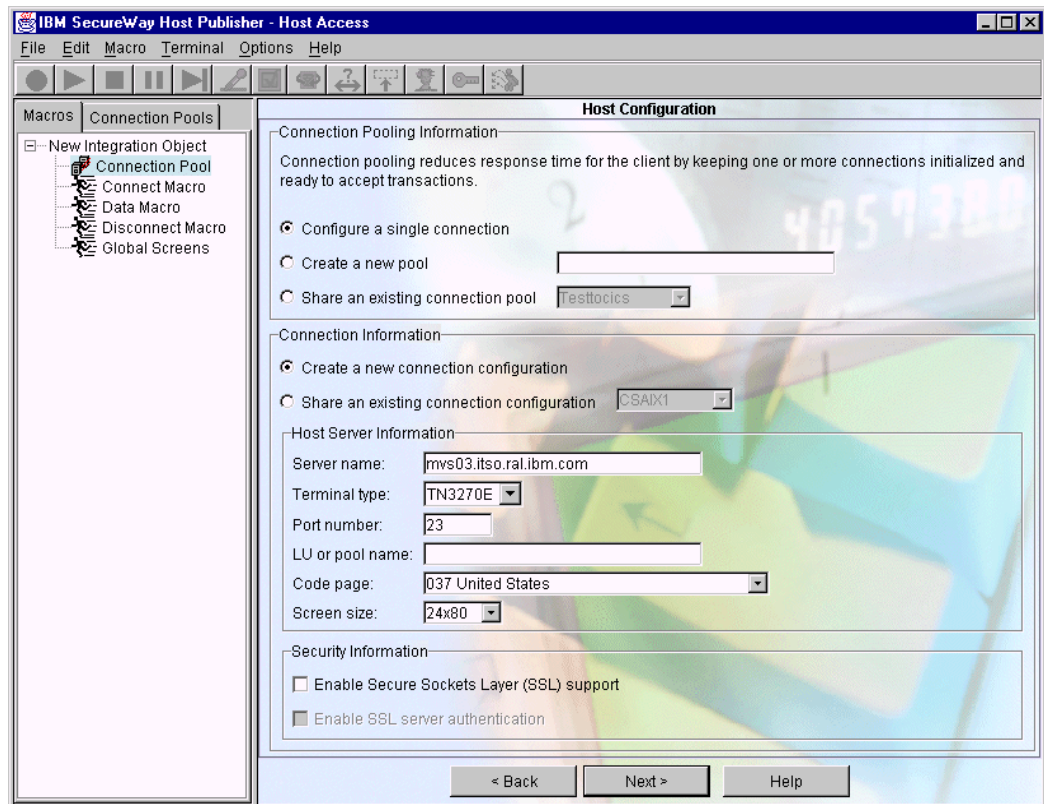


Figure 39. TN3270E connection configuration

3.6.1.2 Connection pooling

The default for Host Publisher is to establish a connection for the execution of an Integration Object, then disconnect at the end. Depending on the transaction, this could mean that a significant portion of the user's time is spent establishing the connection and disconnecting compared to the actual gathering of data.

Connection pooling allows you to maintain the connection once an application has requested a connection. When the Integration Object that opened the connection terminates, the connection is kept open and released to a pool, ready to be reused by any Integration Object requesting a connection from the same pool. Subsequent users of the same or other applications requesting a connection from that pool can then use it without having to go through the connect phase of the host Integration Object, thus reducing the overall application time.

Since each host connection, particularly in the case of 3270 applications, may require a unique user ID, the pooling of user IDs in addition to the pooling of connections is supported.

The Connection Manager of the Host Publisher Server establishes, maintains, and deletes host and database connections for an application's Integration Object.

3.6.1.3 Creating macros

After you configure a connection, Host Publisher will establish a connection to the host for the next step. Once the connection is made the host connection you configured appears in the terminal at the lower right.

Host Access guides you as you interact with the host to navigate the screens that log on to the application, get to the data you want to extract, and define the way it should be presented.

Macros are recorded as you navigate through the host application using the open connection.

In Host Access, the Integration Object includes the following types of macros:

- **Connect macro**
Includes the information Host Publisher needs to connect to the host. The connect macro should contain steps necessary for logging on to a system from as many initial states as possible.
- **Data macro**
Includes the information about how to navigate to and extract the data you want to publish.
- **Disconnect macro**
Includes the information about how and when to disconnect from the host. Typically, this means tearing down the network connection.

As the macros are built, they are displayed in the screen on the left. (See Figure 40 on page 53.)

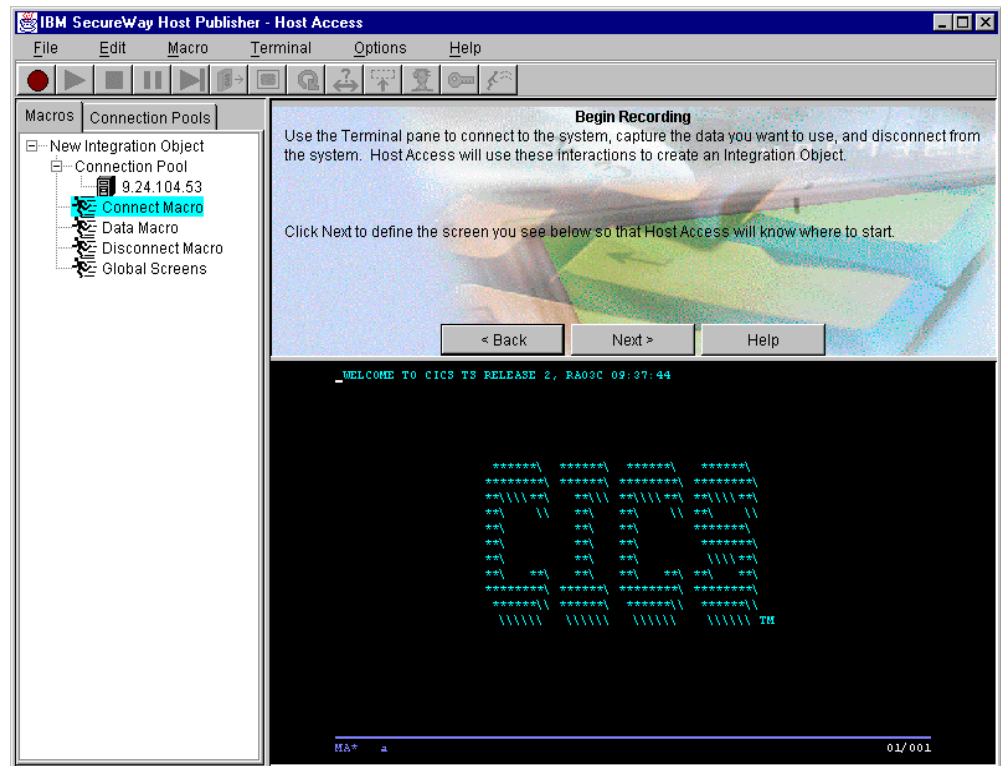


Figure 40. Host Access - Begin Recording screen

3.6.1.4 Screen recognition

In order for Host Publisher to navigate through the host connection, screen definitions are needed. As you create the macro, you will be given the opportunity to define each screen by giving it a name and specifying a trait that can be used to recognize that screen.

There are four ways to define characteristics of a screen for recognition. You can choose multiple distinguishing characteristics:

- Text on the screen

Select this choice if there is specific text that identifies this screen. For example, the screen in Figure 41 will be recognized by the characters that spell out CICS.

- Number of input fields
- Number of total fields
- Cursor position

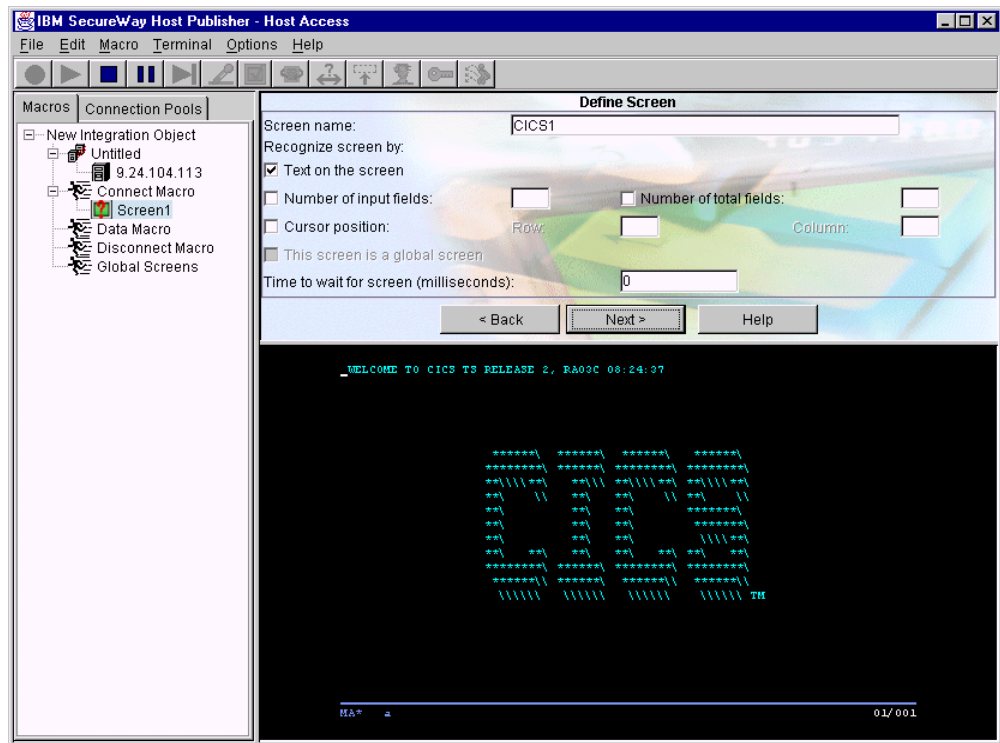


Figure 41. Define Screen panel

3.6.1.5 Global screens

Screens that are intermittently displayed and always require the same action, but appear randomly or multiple times can be defined as global screens. For example, a screen that appears multiple times but only requires clearing can be marked as a global screen. When a global screen is encountered during the execution of an Integration Object, the actions defined for it are performed even though this screen is not on the previous screen's next screen list. The macro continues recognition of the next screen, essentially ignoring the global screen.

3.6.1.6 Input variables

You can use input variables for information you want the user (or another Integration Object) to provide. This information is not coded into the macro and is provided when a user makes a page request. For example, if your application enables the user to search for information about a person, you could use an input variable to contain the name to search on.

Figure 42 shows an example of defining a variable. The variable field will be defined at the cursor position. A temporary value is defined so Host Access can continue with the macro definition.

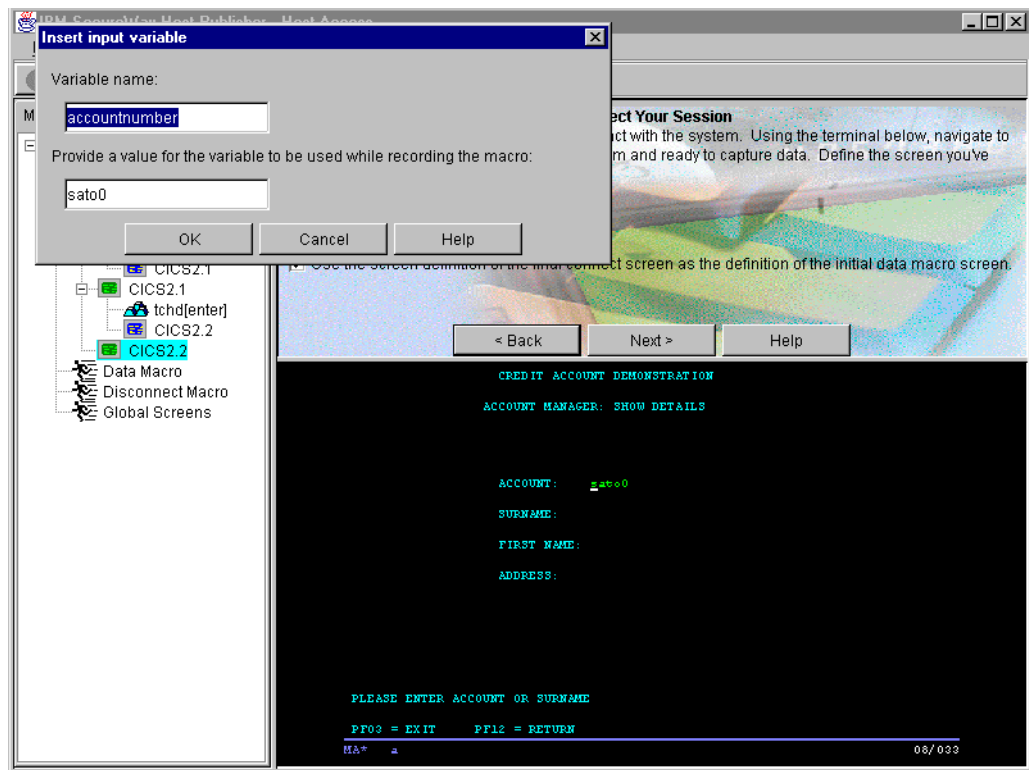


Figure 42. Inserting an input variable

3.6.1.7 Gathering the data results

Once you reach the screen with the results of the inquiry, you need to define which part of the screen contains the data to present to the user. Figure 43 shows an example of defining the area of the screen containing the data results. The data is selected by drawing a box around the data areas. You can also use a data extract area that you have already defined. The name you assigned to that data will appear in a drop-down menu and you may choose to use it instead of selecting data from the screen.

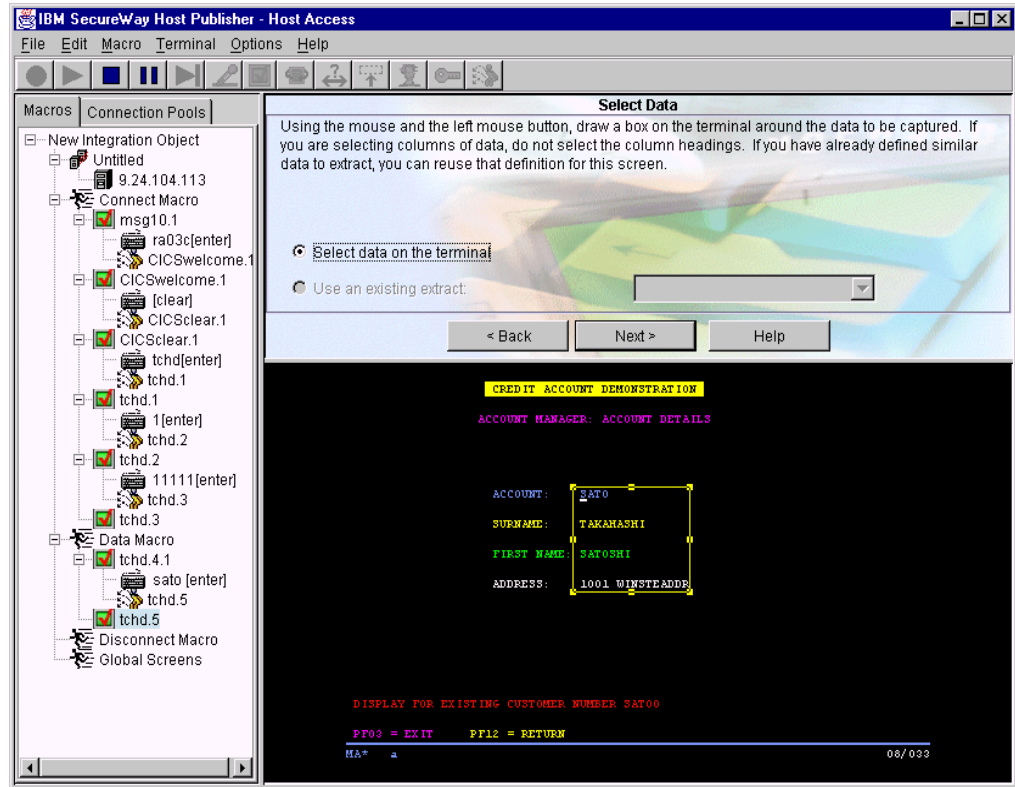


Figure 43. Gathering output data

3.6.1.8 Data looping

Looping enables you to define an action that should be repeated until a condition is met. For example, the data resulting from a query may take up more than one screen. The macro can loop through the screens, collecting data, until the data ends.

3.6.1.9 Conditional statements

Conditions can be built into the macro to handle the situation where more than one screen may result from a host command or keystroke.

3.6.1.10 Presentation of the data

You can specify how you want the results presented to the user. The choices are:

- Table format
- Plain text

If you select table format, the Host Access assists you in defining the table. You have the option of selecting data in the host connection to use as column headers or defining them yourself. You can modify the width of columns, merge columns, or other necessary operations. Figure 44 shows the Host Publisher Studio interface for defining and modifying table attributes.

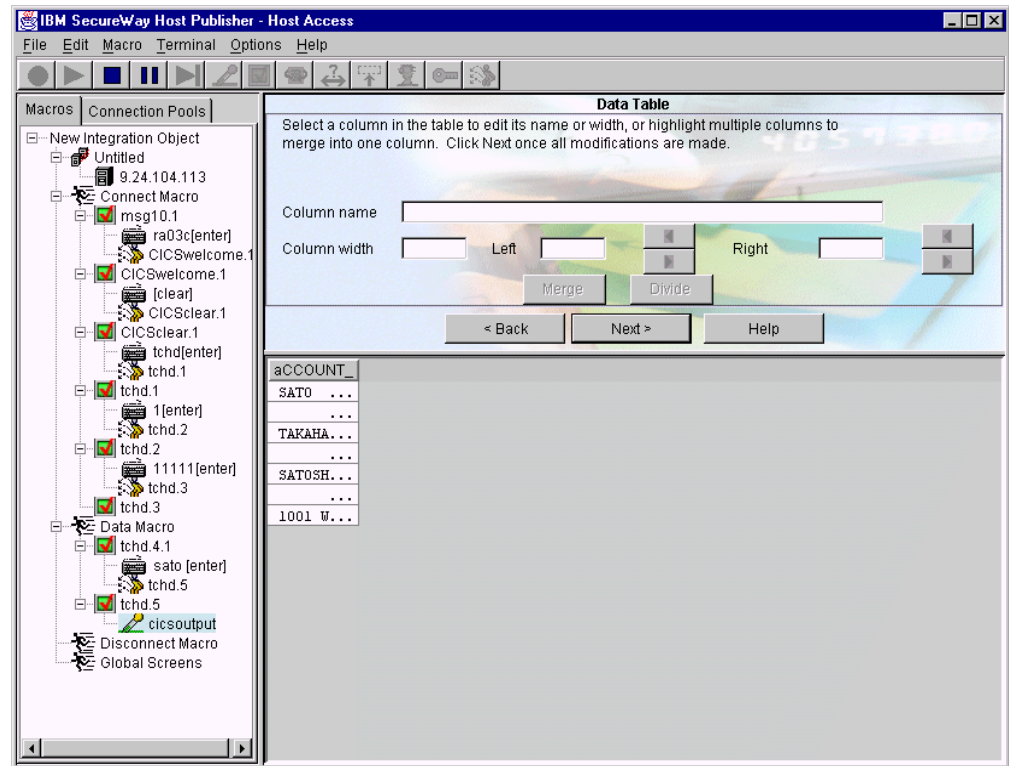


Figure 44. Defining table attributes

3.6.1.11 Disconnecting from the host

The last step is to record a disconnect macro showing Host Publisher how to disconnect from the connection.

3.6.2 Creating the application

The next step is to build the application by creating the JSPs for the application, transferring them to the Host Publisher Server and deploying the application. To do this you use the Host Publisher Studio. These steps are the same for both the host and database Integration Objects and are covered in 3.8, “Using the Host Publisher Studio to build the application” on page 65.

3.6.3 Application results

After deploying the application you can start the application from a Web browser. In the example we have been building, we would use the following URL to access the application:

`http://<server>/HostPublisher/Cics3270/inputform.jsp`

Where <server> is the Host Publisher Server’s host name or IP address. HostPublisher is the URL alias defined at installation time. The application is called Cics3270 and inputform.jsp is our entry Web page.

Figure 45 and Figure 46 show the screens we have built. Our application accesses a simple CICS transaction. Keep in mind that this is a very primitive example designed to show you the basics. If you want a more attractive interface, you can use a general HTML editor to improve the pages.

Figure 45 shows the input page. The input fields are passed to the Integration Object to search the database served by the CICS transaction.

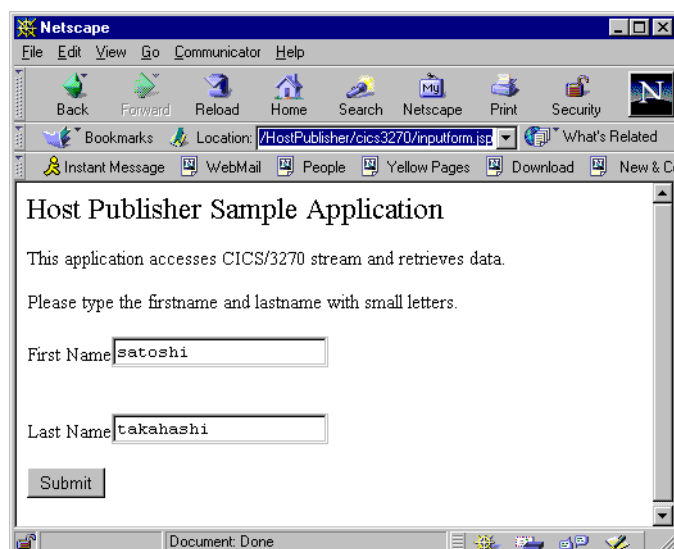


Figure 45. Sample application input screen

Figure 46 shows the resulting output.

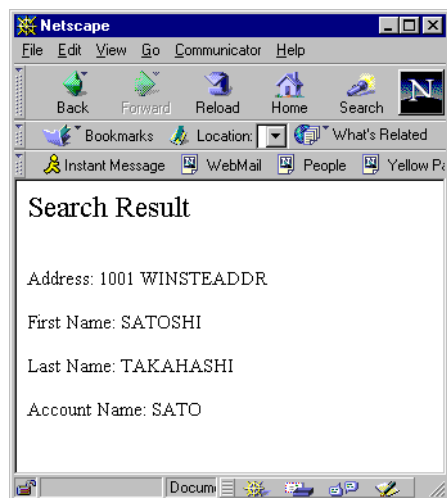


Figure 46. Sample application output screen

As a comparison, take a look at the original CICS transaction “green screen” in Figure 47. You can see what a difference there is between the Web interface and the original screen.

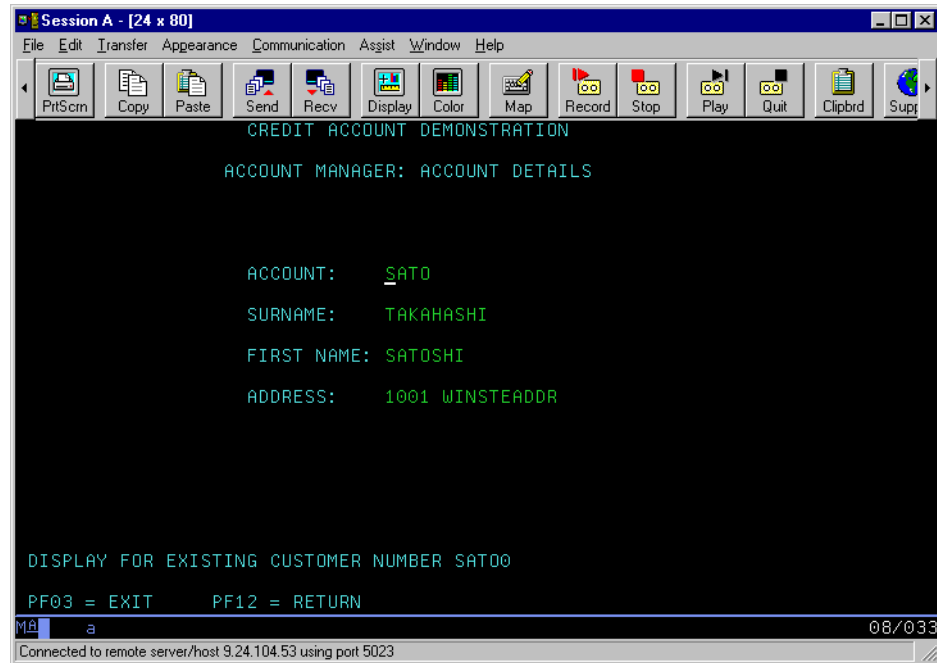


Figure 47. Original 3270 screen

3.7 Creating Database Access Integration Objects

IBM SecureWay Host Publisher Version 2.1 includes a tool that allows you to easily create Integration Objects for database applications. The tool is called Database Access, which uses the Java JDBC interface to communicate with the target database.

Java Database Connectivity (JDBC) is an object interface that allows Java applications, applets, servlets, and agents to retrieve and manipulate data in database management systems using SQL. The interface allows a single application to connect to many different types of databases through a standard protocol. JDBC handles details such as connecting to a database, fetching query results, committing or rolling back transactions, and converting SQL types to and from Java program variables. JDBC is implemented as a driver manager for multiple drivers. Each driver links the application to a specific type of database.

JDBC was first introduced in the JDK 1.1 from Sun Microsystems. The JDBC classes and interfaces are part of the `java.sql` package. The major components of JDBC are the JDBC driver manager and the underlying drivers. JDBC uses the driver manager to handle finding and connecting to a driver.

A JDBC data source consists of the data the user application wants to access and its associated parameters. Each JDBC driver processes JDBC method invocations, sends SQL statements to a specific data source, and returns results to the application.

IBM SecureWay Host Publisher Version 2.1 supports the following JDBC drivers:

- IBM DB2 UDB local
- IBM DB2 UDB remote
- AS/400 Toolbox for Java

- JDBC-ODBC bridge
- Oracle
- Sybase
- Others that you may install

Note: The JDBC-ODBC bridge for gaining JDBC access to a database with an ODBC interface is a productivity aid provided by Sun Microsystems. They do not officially support this tool, and consequently neither can IBM. If you decide to use the JDBC-ODBC bridge with an ODBC-enabled database, be aware that there are limitations and problems with the tool. For example, when using the JDBC-ODBC bridge to connect to a Microsoft Access database, columns which are of type "Date" will not show up on the Condition panel of Database Access application. This is due to the fact that the JDBC-ODBC bridge does not report the column type correctly.

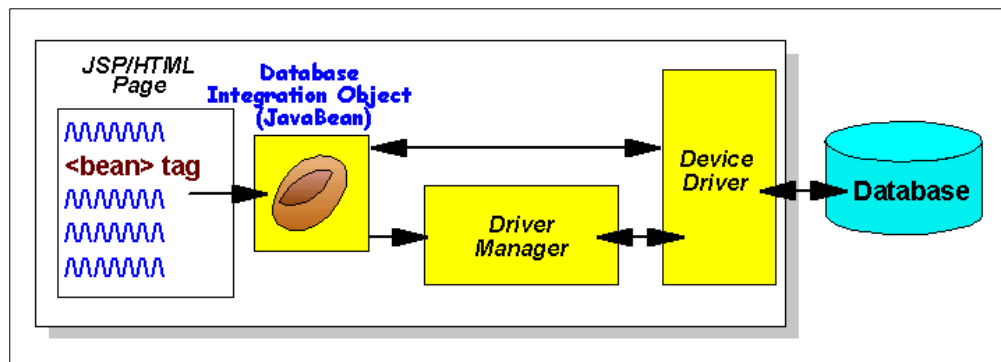


Figure 48. JDBC architecture

Host Publisher applications using JDBC use the following basic scenario:

- An input page (JSP) is created to request input from the user at the Web browser using an HTML form.
- The user submits the request (post) to an execution page (JSP) that invokes the database Integration Object (bean).
- The Integration Object processes the SQL statement and returns a response.
- The execution page displays the results.

3.7.1 Creating the Database Access Integration Object

When you use the Database Access, it guides you to specify the JDBC URL for the database you want to access. Using tabs, you create an SQL statement to retrieve data from the tables in the database. Actually this step is very similar to the panel in Database On-Demand.

After entering the connection information for the database, you can connect to the database by clicking the **Connect** button. (See Figure 49.)

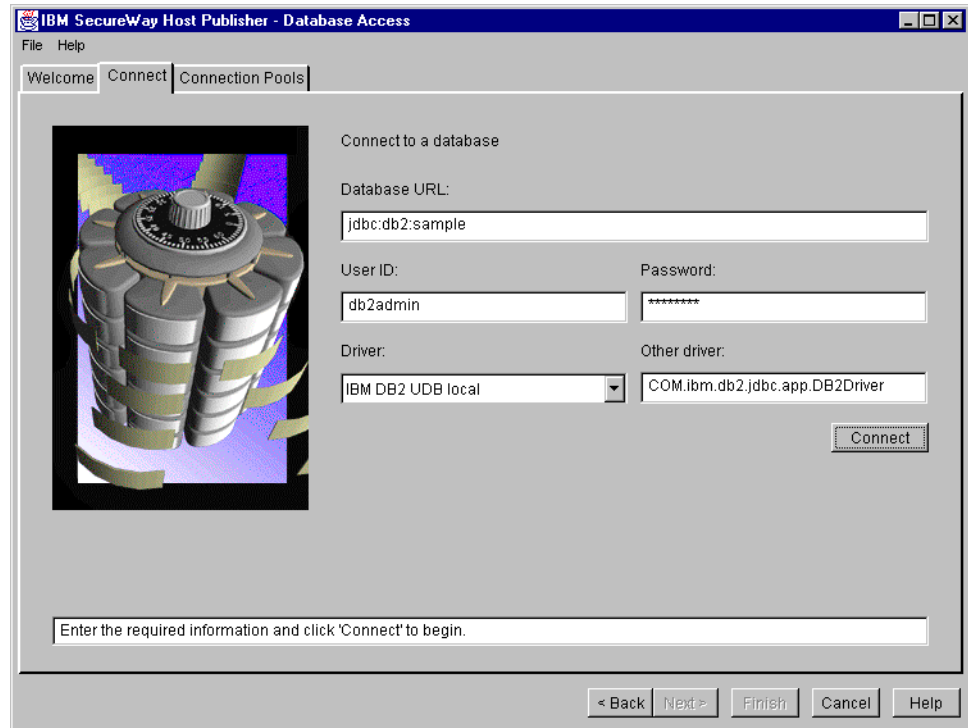


Figure 49. Welcome panel for creating a Database Access Integration Object

If the database connection succeeds, additional tabs (panel options) will be added to the applet to allow you to continue defining the database Integration Object.

From here you navigate through each definition panel, building the SQL statement. Optionally, you can enable and configure connection pools for better performance.

Figure 50 shows the Tables panel, which is the first step in defining the SQL statement. In this panel you define the type of statement (select, insert, etc.) and specify which tables to extract the data from.

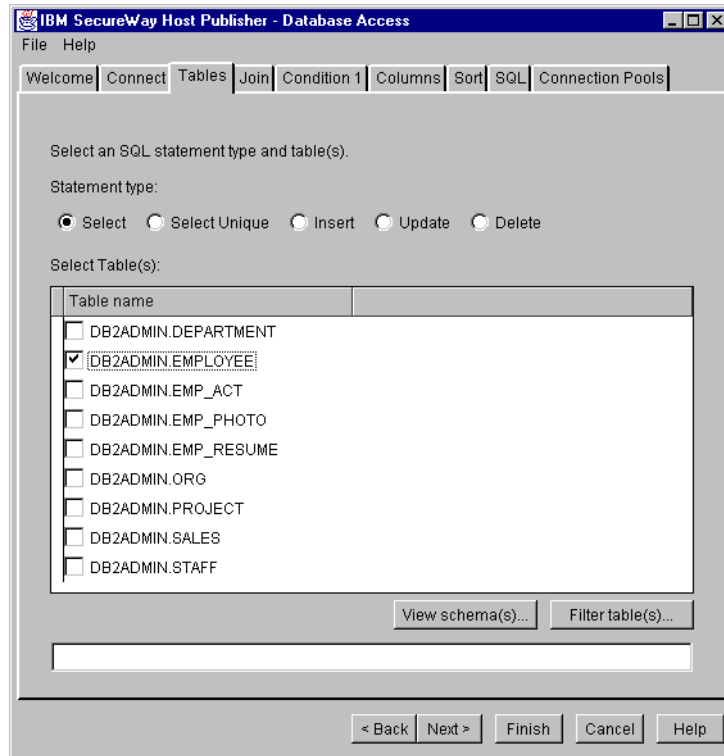


Figure 50. Table definition

Figure 51 shows the the Condition panel. This assists you in building the SQL statement by allowing you to specify the data in the table you want to find. You can enter multiple conditions for selecting data and you can use input variables for information you want the user to provide.

In this example a variable is used as the value for the column to match. When the application is running, this variable is initialized with input from the user and the actual SQL is generated dynamically on the server.

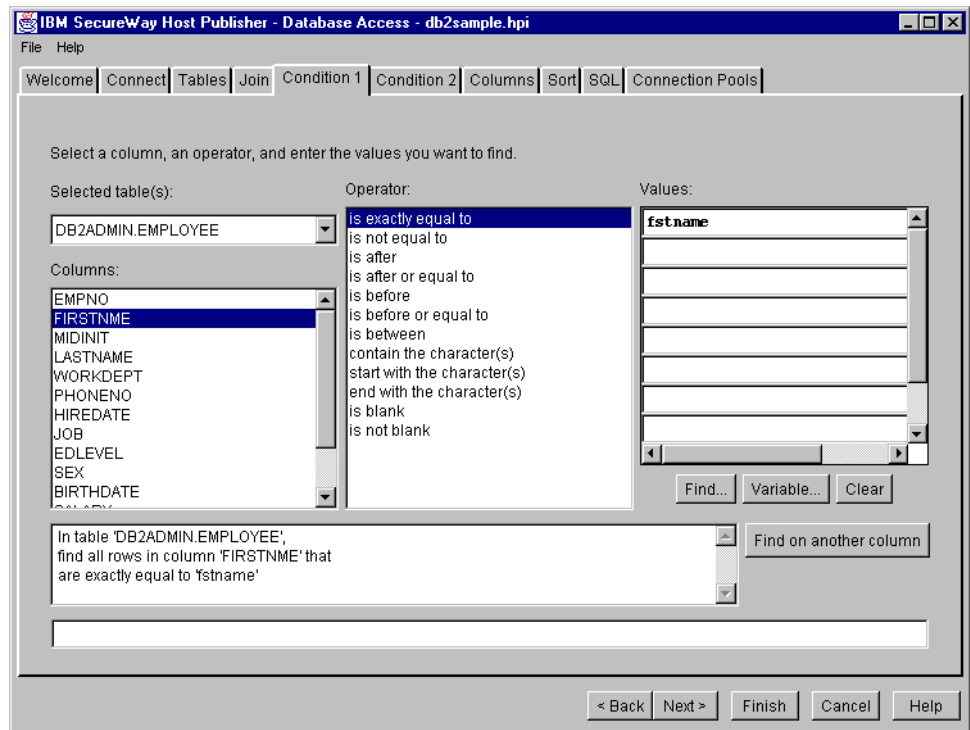


Figure 51. Define the condition for selecting data

The next step takes you to the Columns tab where you specify the columns to include in the results. The Sort tab allows you to specify the sort order for the data returned.

3.7.2 Creating the application

The next step is to create the JSPs for the application and integrate the Integration Object just created. Once the application is created it must be transferred to the Host Publisher Server and deployed. The steps are much the same as for host integration applications and are described in 3.8, “Using the Host Publisher Studio to build the application” on page 65.

3.7.3 Application results

The results of the finished application we have been building can be seen in Figure 52 and Figure 53. After deployment our application can be accessed from a browser using the following URL:

`http://<server>/HostPublisher/db2sample/db2input.jsp`

Figure 52 shows application from a Web browser.

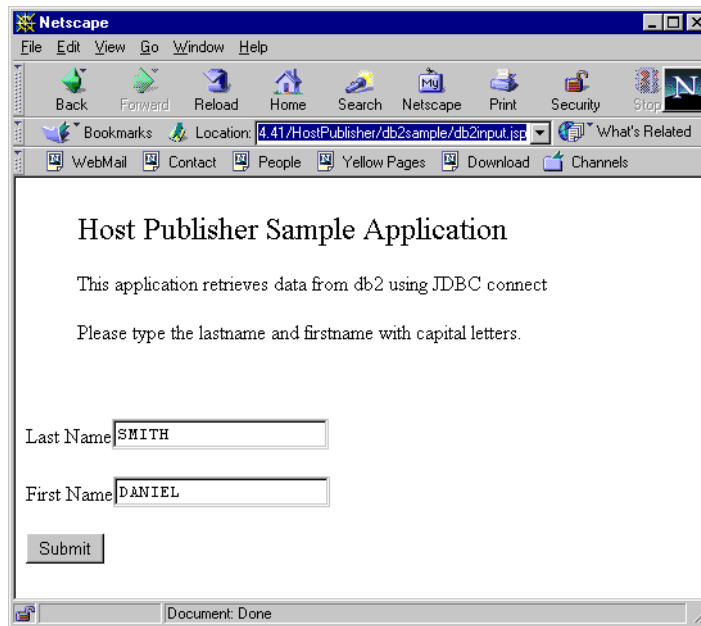


Figure 52. Input page example

The user fills in the Last Name and First Name fields, and pushes the **Submit** button. The Integration Object uses the information submitted in the previous page to access the database and retrieve the result. The output of the Integration Object is dynamically converted into HTML and displayed. Figure 53 shows the resulting page.

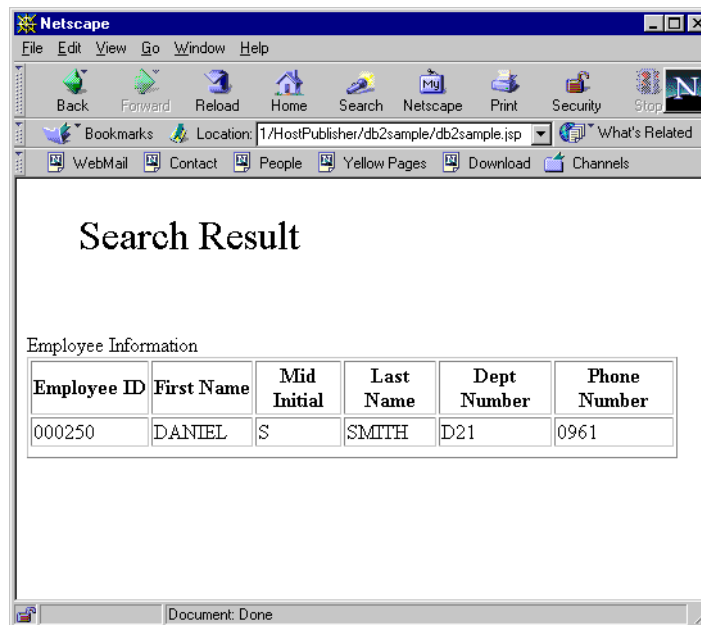


Figure 53. Output page example

3.8 Using the Host Publisher Studio to build the application

The Host Publisher Studio provides a development environment for you to create Web applications and manipulate the Integration Objects you created in the Host Access application. There are two phases to this development: creating the pages, and the customization of those pages. You are not required to use the Studio to create your application. However, it is recommended for a quick and easy way to initiate the development of applications.

The Studio allows you to make multiple pages for the application. For example, you could create an initial page that asks for input and a second page that executes an Integration Object (with the input from the first page) and presents the data. A wizard assists in building the pages and allows you to preview the results.

After building and previewing the JSPs, you have the raw HTML data available for editing. Figure 54 shows the results of defining multiple pages. The pages are listed at the bottom left. Clicking on a page brings up the HTML source. The tools at the top assist in modifications such as adding Integration Objects, input fields, etc.

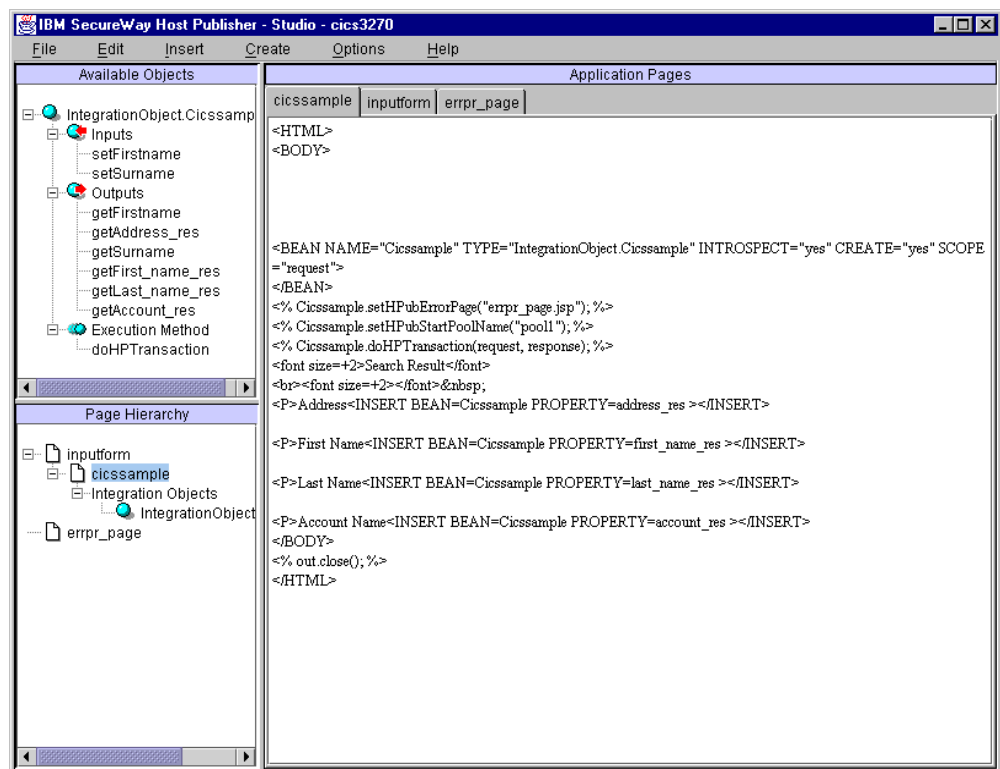


Figure 54. Host Publisher Studio

Once your application is finished you can save it.

3.8.1 Transferring the application to the server

The Host Publisher Studio provides you the ability to transfer your Web application to Host Publisher Servers, making the application ready for

deployment. From the same screen shown in Figure 54, you have the option under the File pull-down to transfer the application to the server.

The transfer option will give you a list of defined Host Publisher Servers to transfer the application to. You may transfer it to multiple servers. The transfer uses FTP to put the application on the Host Publisher Server in a staging subdirectory. Later, during deployment, the Host Publisher Server validates the application and then promotes it to a production area. Figure 55 shows the transfer screen and the messages generated during the transfer.

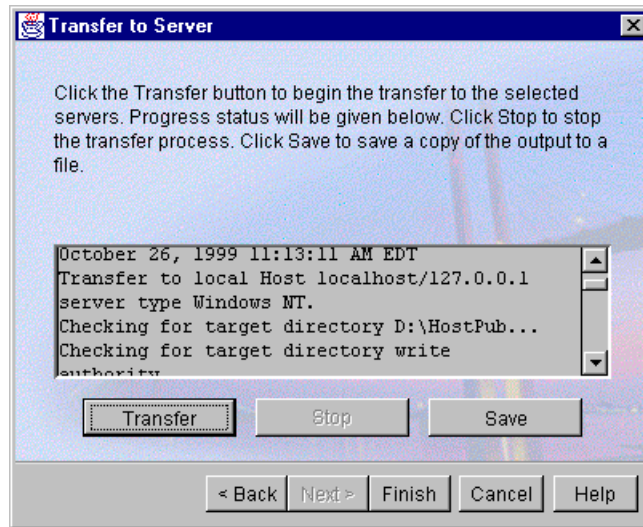


Figure 55. Transfer application to server

3.8.2 Deploying the application

Once the application is transferred it is ready for deployment. By default, applications must be deployed manually. If you want an application automatically deployed when the server starts, you must change the server.properties file.

Once you choose the applications you want to deploy, Host Publisher Server scans the staging directory for the application files and, if no problems are found, moves the files over to the production directory for immediate use. Some examples of problems are: error in connection pool configuration files, missing files, conflicting information, or applications already deployed. To deploy your application, you would use the Host Publisher Server Administration panel shown in Figure 56.

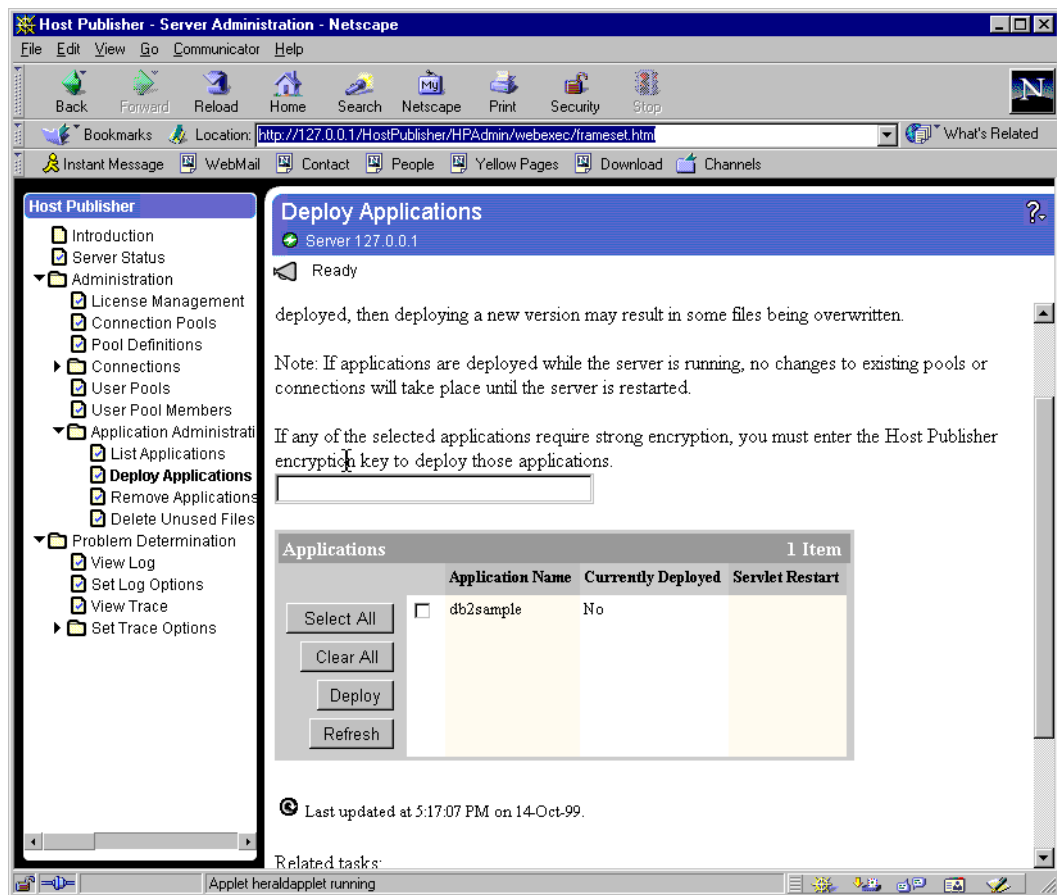


Figure 56. Deploy application

3.9 Where to find more information

- <http://www.ibm.com/software/network/hostpublisher/>

Chapter 4. IBM SecureWay Screen Customizer

IBM SecureWay Screen Customizer V1.0 was developed to integrate smoothly with IBM SecureWay Host On-Demand (HOD) and IBM SecureWay Personal Communications (PCOMM) connectivity software to form a complete Web enabling solution. While Host On-Demand and Personal Communications manage the TN3270 and TN5250 connectivity, Screen Customizer delivers a graphical presentation of host application screens. It is the follow on to ResQ!Net for Host On-Demand V3 from ResQ!Net.com, Inc. (formerly Advanced Transition Technology), offering additional enhancements. Customization for 3270, 5250, and CICS Gateway display sessions is available.

Screen Customizer is a thin Java client that automatically converts host screens into a graphical presentation that is easily customizable without any programming, making it a cost-effective, quick solution that allows you to leverage your investment in legacy applications. With Host On-Demand, the Screen Customizer components run as Java applets in a browser. With Personal Communications, they run as Java applications.

Screen Customizer requires no access to source code and generates no scripting or coding. Therefore, 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, with no degradation in host response time.

Screen Customizer's default graphical user interface is automatically presented for each host screen at the client. It features automatic menu buttons, defined input fields, and hot spots. The application administrator can customize screens by adding new objects, changing fonts or colors, combining data, or automatically navigating the user through screens.

There are three components of Screen Customizer:

1. **Administrator** - The Administrator launches and uses active host sessions to let you capture host screens for customization. The screens are identified by a Screen ID and saved as maps. These maps can then be modified in the Customization Studio. The Administrator is also used to set global defaults for font, colors, button styles, and other attributes, and to save the defaults in a profile. The Administrator can be installed and run only on a Windows 95, 98 or NT workstation. It cannot be downloaded from a Web server.
2. **Client** - The Client resides on a Web server or a client workstation and presents a default or customized graphical interface for host sessions. The Client is usually integrated with a Host On-Demand or Personal Communications session, but can be started independently. A limited edition version of Screen Customizer (Screen Customizer/LE) is automatically included in most Host On-Demand clients.
3. **Customization Studio** - The Customization Studio allows you to create customized versions of host screens captured by the Administrator. The Studio can be installed and run only on a Windows 95, 98 or NT workstation and does not require a host connection.

To install the Administrator or Client, you must have either Host On-Demand or Personal Communications installed. The installation program detects which product you have and installs the appropriate files.

4.1 Screen Customizer features

Screen Customizer offers the following features:

- Ability to customize green screens into a modern graphical interface with radio push buttons, images, Web links, check and choice boxes, colors, etc.
- Create context sensitive field help for host applications.
- Create a list of values that are valid for a field, so that the user can choose one; this minimizes errors.
- Create buttons that launch a browser and link host applications to the Internet.
- Hide unnecessary screens so that users see only the screens that they need to see.
- Global variables allow you to save data from one screen and use it as information for another screen or to perform calculations.
- Create unique customizations for different groups of users.
- Start tracing on the client to help determine the cause of a problem.
- Supports multiple customizations of the same screen.
- Customization Studio requires no programming.

New, enhanced features not available on ResQ!Net:

- File transfer now possible between your workstation and S/390 or AS/400 host systems.
- Startup/run applet - Customers can launch their own Java applets at startup or from any Host On-Demand macro feature such as a button or link.
- Enhanced session management allows users to jump between active sessions with Host On-Demand, and run a new instance of the current session, or exit a session with either product.
- Enhanced edit functions allow users to access cut, copy, and paste buttons from the menu bar.
- Consist with Host On-Demand menu, toolbar, show/hide status bar, and keypad.
- Associated printer - Allows a user to associate a printer session with a display session.

4.2 Requirements for Screen Customizer

To install the Administrator or Client, you must have one of the following:

- Host On-Demand 4 CSD1
- Host On-Demand 4 CSD1 for On-Demand Server
- PCOMM 4.3 CSD2

The installation program detects which product you have and installs the appropriate files. You can only install Screen Customizer for either Host On-Demand or Personal Communications, not both. If both products are installed, you are given an option. If neither product is installed, you can only install the Studio.

Some VTAM installations have an optimizer installed that can merge fields to enhance performance. This option must be turned off. A field merge effect may also be caused by CICS and IMS applications that write the data stream directly, rather than utilizing BMS or MFS mapping systems.

4.3 Migrating from ResQ!Net

On Windows workstations, migration is automatic if you are installing into the HOD directory that already has ResQ!Net installed. The setup program copies the contents of the ResQ!Net directories to the equivalent Screen Customizer directories. However, if you have customizations saved in any other directories, you must copy them to the new path manually. On non-Windows platforms, you must copy all customizations manually.

4.4 Supported operating systems (minimum levels)

Client

If installing the client on a Host On-Demand (HOD) workstation the following operating systems are supported:

- Windows 95, 98
- Windows NT 4.0 with SP4

Note: HOD comes with a limited edition Screen Customizer that is included in many of the HOD clients.

If installing the Client on a HOD server, the following operating systems are supported:

- Windows NT 4.0 with SP4
- AIX Version 4.2
- OS/2 Warp Version 4 and Warp Server
- Novell NetWare Version 5.0 with SP2
- Sun Solaris OS Release 2.6
- HP/UX 10.20
- RedHat Linux Version 5 Release 2
- OS/400 Version 4 Release 2
- OS/390 Version 2 Release 1

If installing the Client on HOD *specifically developed for On-Demand Server* the following operating systems are supported:

- Windows NT 4.0 with SP4
- AIX Version 4.3.2
- Sun Solaris Version 2.6
- OS/390 Version 2 Release 5

If installing the Client on a PCOMM workstation the following operating systems are supported:

- Windows 95 and Windows 98
- Windows NT Version 4.0 with SP4
- Windows 2000

Administrator

If installing the Administrator on either a HOD server, HOD client, or PCOMM workstation the following operating systems are supported:

- Windows 95 and Windows 98
- Windows NT Version 4.0 with SP4
- Windows 2000 (Personal Communications only)

Customization Studio

The Customization Studio is supported on the following operating systems:

- Windows 95 and Windows 98
- Windows NT Version 4.0 with SP4
- Windows 2000

4.5 Supported browsers

The following browsers can be used to run the Screen Customizer Client with Host On-Demand:

- Netscape Navigator 4.08 or 4.5 (Windows 95, 98, NT, 2000, UNIX)
- Netscape Navigator 4.04 (OS/2) (remote server only)
- Microsoft Internet Explorer 4.01 with SP1, or 5.0 (Windows 95, 98, 2000, NT)

The following browsers can be used to run the Screen Customizer Client from a remote server with HOD for On-Demand Server:

- Netscape Navigator 4.5 (Windows 95, 98, NT 4, 2000, UNIX)
- Microsoft Internet Explorer 5.0 with JVM build 3167 (Windows 95, 98, 2000, NT 4).

4.6 Screen Customizer RunTime

Screen Customizer RunTime is an add-on product to either HOD V4 (with CSD1) or Personal Communications V4.3 (CSD2).

Screen Customizer RunTime for Host On-Demand includes the Screen Customizer client for HOD and an installation guide. The RunTime for Personal Communications includes the Screen Customizer client for PCOMM (Windows only), the Screen Customizer client for HOD, and an installation guide.

RunTime is the client-only function - that is, it is what gets installed on a HOD server or locally installed client, or on a PCOMM client, so that a HOD or PCOMM session can use customized screens. It is priced separately from the Administrator or Studio, which capture and customize the screens.

4.7 Sample Screen Customizer screens

To give you an idea of the advantages of “dressing” up your green screens for users, take a look at the following example. With very little effort, the original screen presentation can be modified for a much more user-friendly look. The following screen is the typical “green screen” prior to any customization by Screen Customizer.

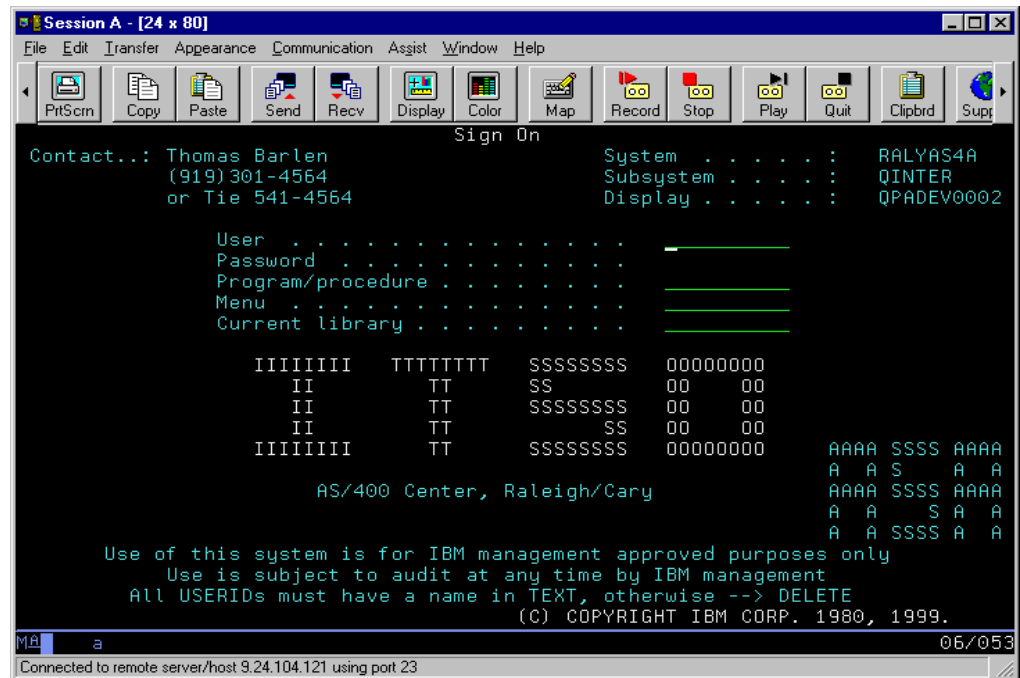


Figure 57. Host On-Demand 5250 emulation session "green screen"

The following figure is a representation of the same screen, but presented with the Screen Customizer default GUI. No manual customization has been performed. Note that any input field in the 3270/5250 data stream becomes an input field in the Screen Customizer's representation of the screen.

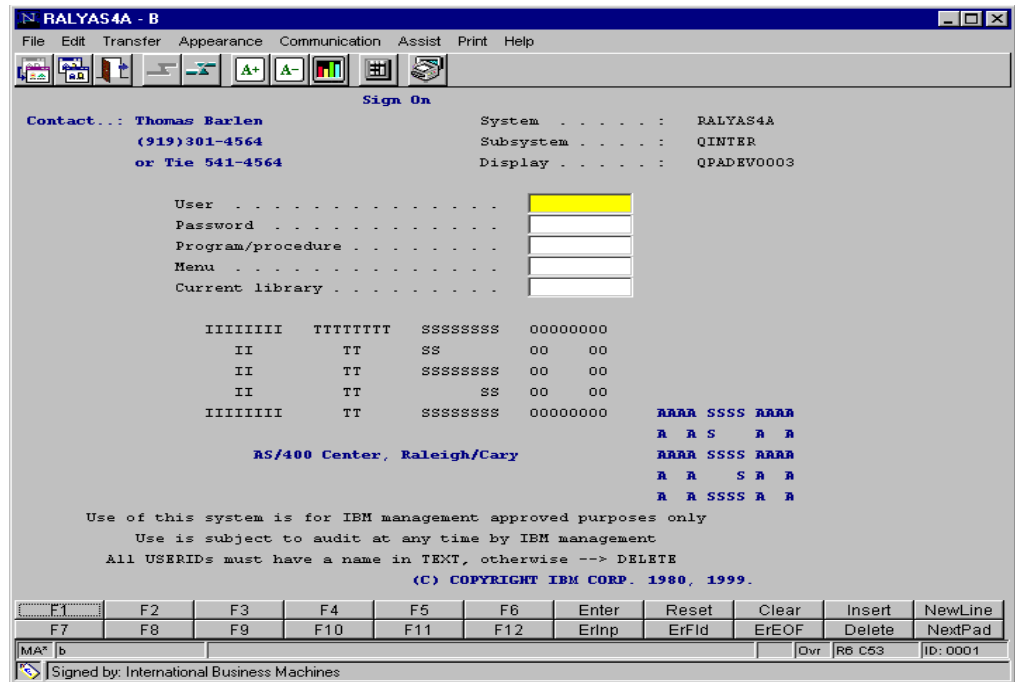


Figure 58. Same Host On-Demand screen using the Screen Customizer default GUI

Additional modifications can be performed with the Screen Customization Studio to make the screen in Figure 58 more user friendly. Figure 59 represents the newly customized screen. Some input fields were changed to check boxes and lists. The “ITSO” logo and a radio button were added. Fonts and colors were changed, as well.

The screenshot shows a customized Host On-Demand screen. At the top left is the ITS O logo in large blue letters, with ASA below it in smaller blue letters, and AS/400 Center, Raleigh/Cary in even smaller blue letters. To the right of the logo is a 'Sign On' section with the following fields: System (RALLYAS4A), Subsystem (QINTER), and Display (QPADEV0002). Below the logo are five red labels for input fields: 'Userid', 'Enter your Password', 'What Program or Procedure?', 'Choose a Menu', and 'Enter the Current Library'. The 'Userid' field has a yellow background. The 'What Program or Procedure?' field has a dropdown menu showing 'Program 1'. The 'Choose a Menu' field has two radio buttons labeled 'Menu A' and 'Menu B'. The 'Enter the Current Library' field is a text box. At the bottom left is a button labeled 'Need sign on assistance?' and a button labeled 'Contact: Thomas Barlen'. At the bottom right is a copyright notice: '(C) COPYRIGHT IBM CORP. 1980, 1999.' The bottom of the screen has a status bar with the text 'MA a' on the left, 'Ovr R6 C53' in the middle, and 'ID: 0003' on the right.

Figure 59. Same Host On-Demand screen customized by the Screen Customizer

4.8 Screen Customizer Administrator and Customization Studio

The Screen Customizer Administrator function allows you to add, delete, modify, and select sessions. The Screen Customizer Administrator can work together with Host On-Demand and Personal Communications in order to use existing session definitions. The Screen Customizer Administrator interface is shown in Figure 60.

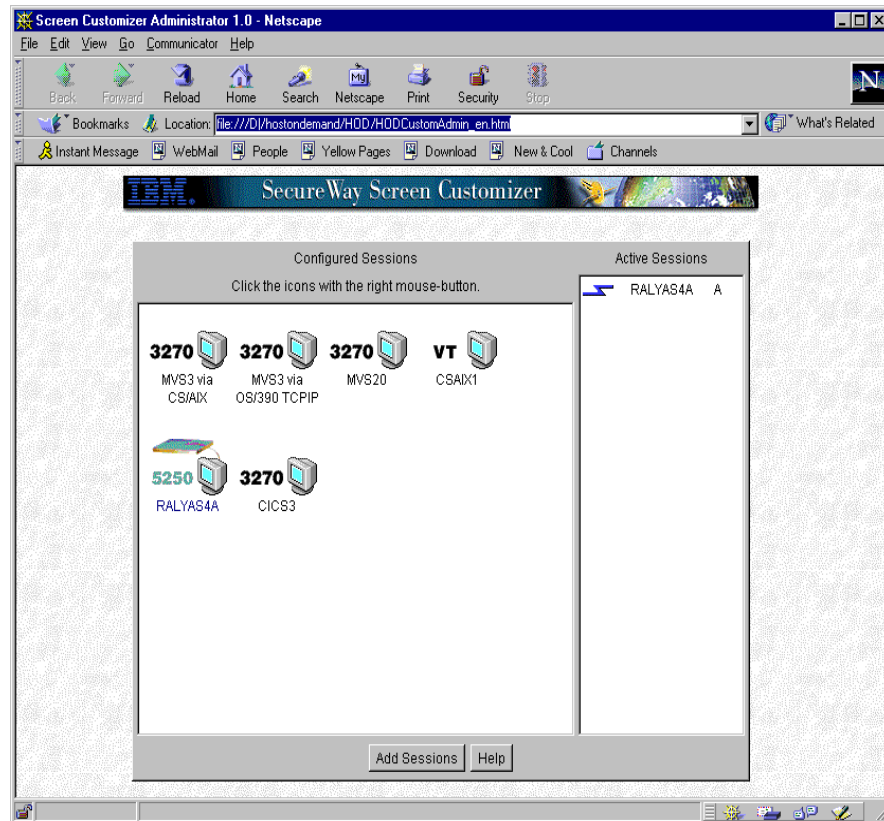


Figure 60. Screen Customizer Administrator

By default, the sessions are in “client” mode when you first open the Administrator. To view a particular session in administrator mode, you must change from Screen Customizer mode to Administrator mode. This is located in the session properties, as shown in Figure 61.

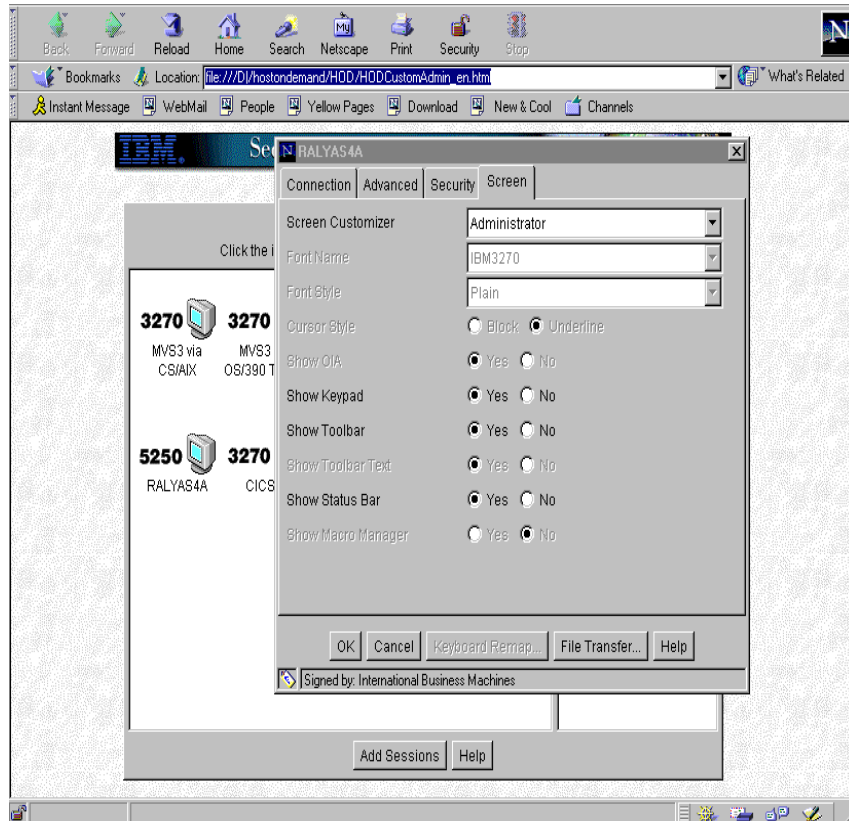


Figure 61. Session Properties window

The Administrator allows you to set basic global defaults for font, colors, and other attributes. You can create a profile to define global customization and button appearance. For example, if you wanted the default screens for all your users to use the Courier font, to have a certain color scheme, and no keypad displayed, you could set up a global profile with these preferences. This profile would then need to be defined as the default for the Screen Customizer clients by setting a parameter in the HTML that calls the applet.

Figure 62 shows a host green screen that is the result of a global profile definition, specifying new default background and foreground colors for all screens in this host application. Changes for individual screens are done in the Screen Customization Studio.

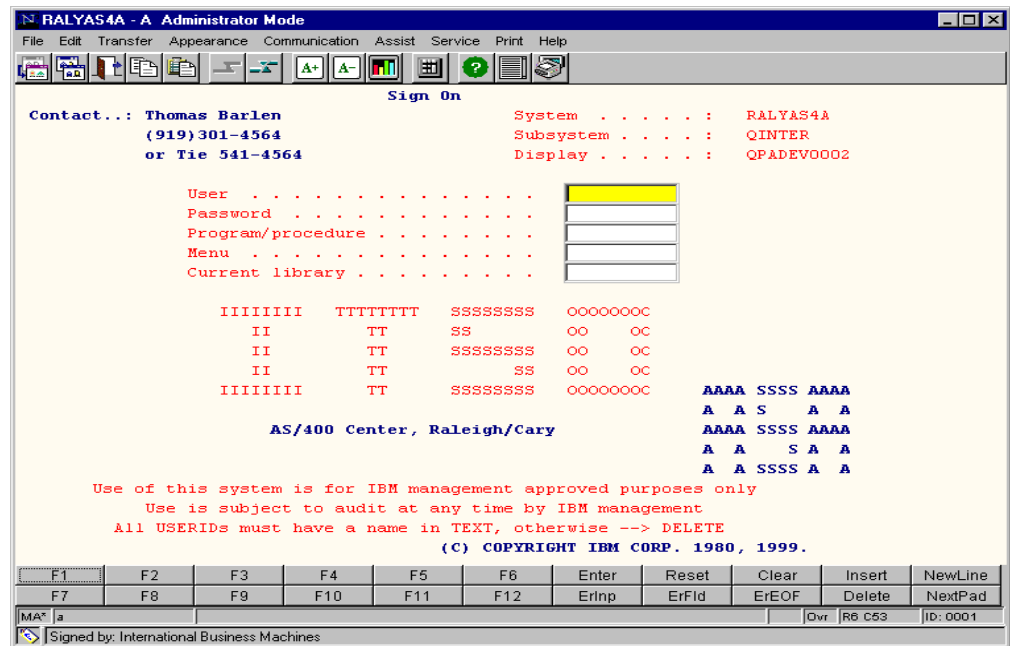


Figure 62. Host screen modifications made by Screen Customizer Administrator

The Screen Customizer Administrator also lets you capture host screens and save them as maps that can be modified in the Customization Studio.

Once screens have been saved as maps, the Customization Studio allows you to create modified versions of the captured screens. For example, you can change the colors, font, tab order, and keyboard mapping for a particular screen. Figure 63 shows the screen originally captured by the Administrator (Figure 62), after modifications have been done in the Screen Customization Studio. It illustrates the options available from the Screen pull-down.

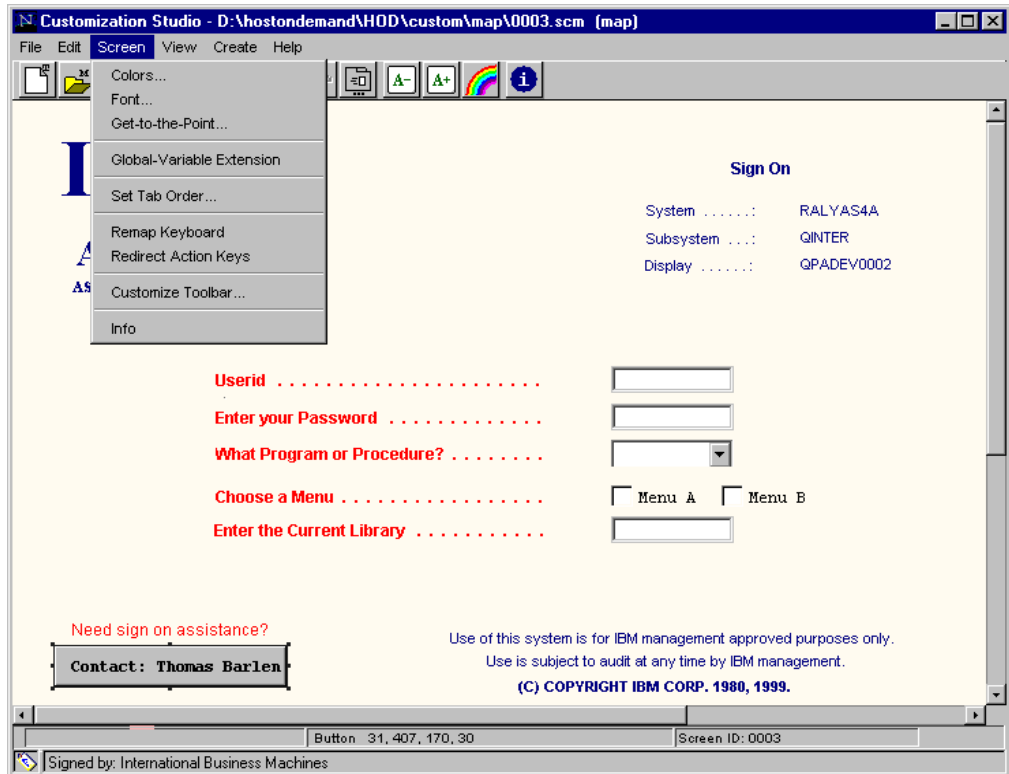


Figure 63. Screen Customizer Customization Studio

Figure 64 shows the Create options. The Studio allows you to create buttons, check boxes, labels, and lists to name a few.

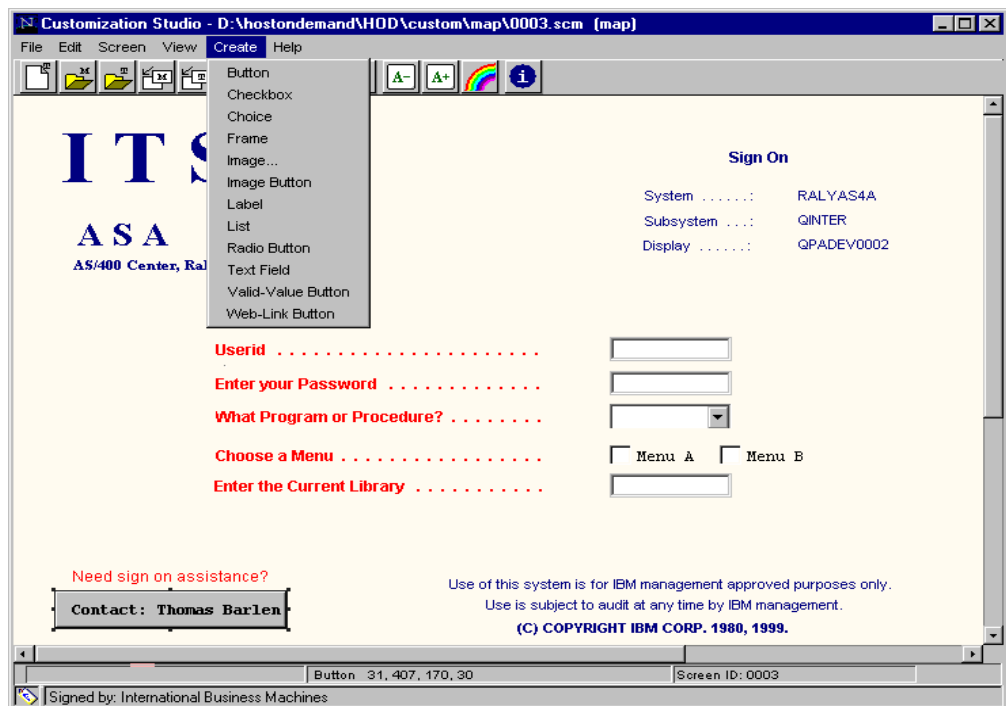


Figure 64. Screen Customizer Customization Studio

Within the Customization Studio you can change the properties of each field. The Set Label Properties screen allows you to alter the caption, font, color, size, and read/write abilities of that text field. Figure 65 is an example of how the wording of a particular field can be changed.

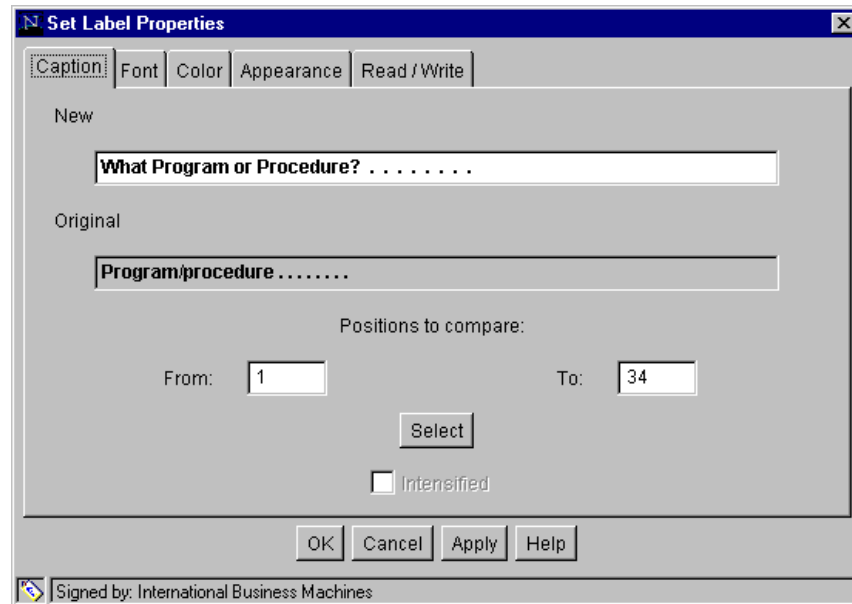


Figure 65. Changing field properties

Figure 66 shows the screen after customization within the Customization Studio. Once complete and saved, the screen is ready to be accessed by the Host On-Demand clients.

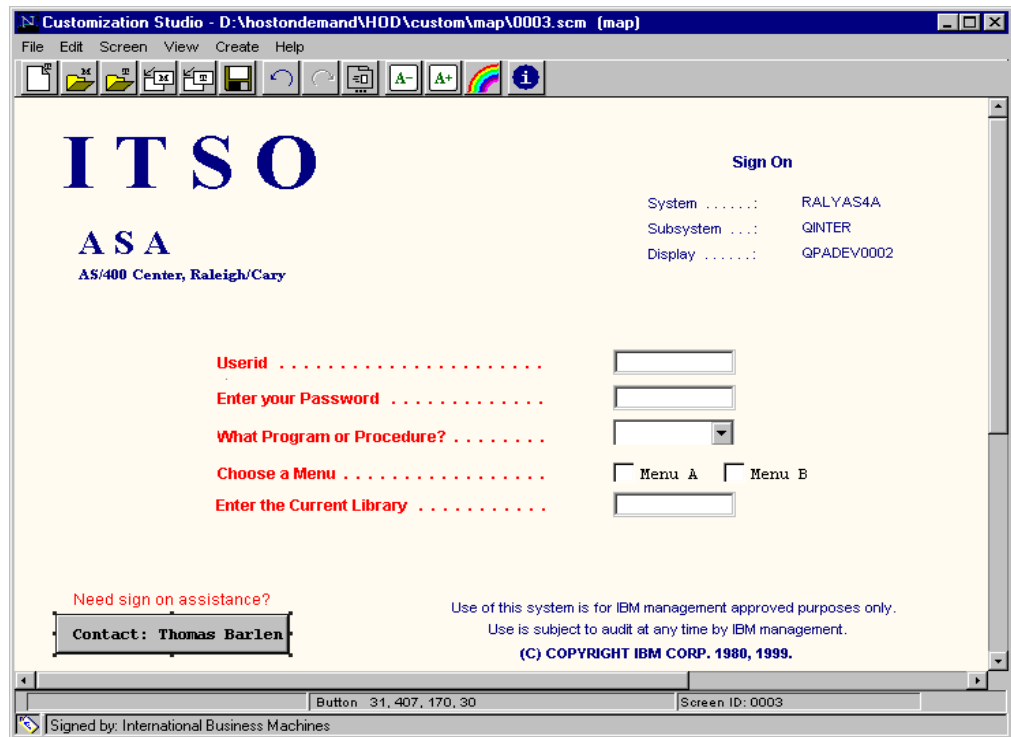


Figure 66. Customization Studio: the finished screen

The customized screen must be enabled from Host On-Demand. This change to Enabled is made in the particular session's Properties display, as shown in Figure 67.

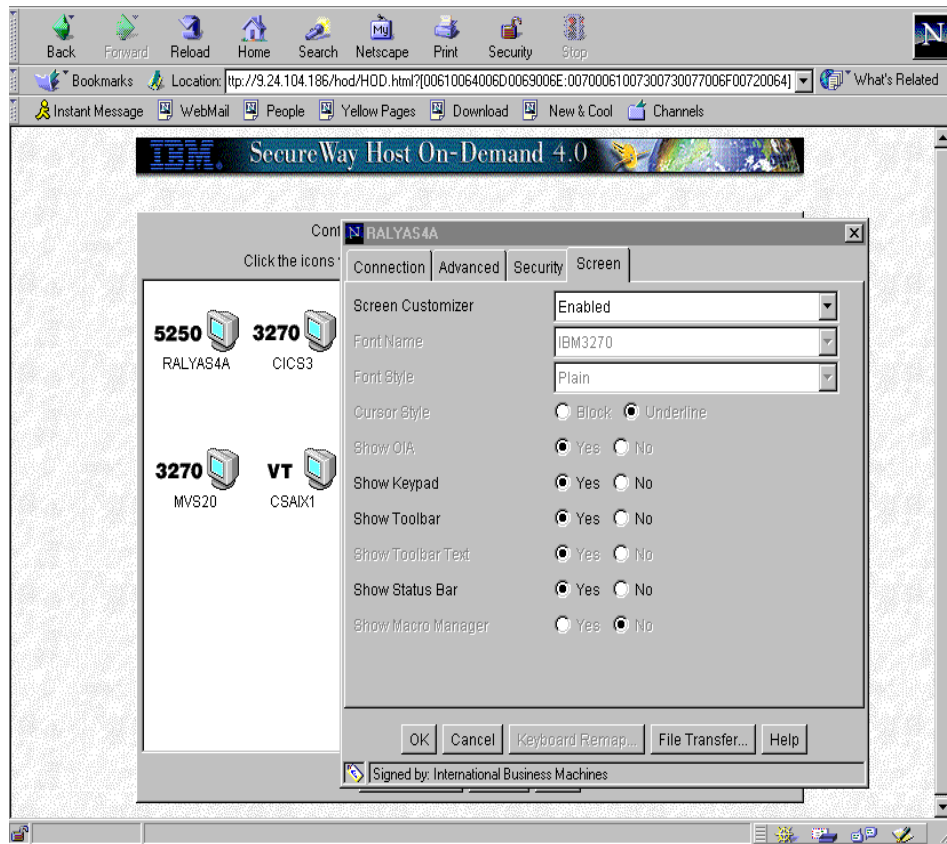


Figure 67. Host On-Demand session properties: enabling Screen Customizer

Once Screen Customizer is enabled, the newly customized screen will be displayed to the users accessing that session.

Chapter 5. Host access APIs and reusable components

Re-engineering existing processes by developing new applications to integrate host applications with the Web requires the most work but also can provide the greatest return on investment. The SecureWay product line provides common host access APIs and reusable components, based on industry standards, to enable businesses to update their operations to take advantage of the latest Internet technologies. These APIs and components include Host Access Beans for Java, Host Access Controls for ActiveX and the Host Access Class Libraries. Integration using these APIs is simple and requires no changes to be made to the existing applications.

Host Access Beans for Java and Host Access Controls for ActiveX, included with both Host On-Demand and Personal Communications, jump-start development of custom e-business applications with pre-built emulator functions. They connect backend information to Web application servers. The Host Access Beans can be used with any Integrated Development Environment (IDE), including IBM's VisualAge for Java and Symantec's VisualCafe. And, because they are object-oriented, businesses can reuse what they develop.

There are additional flavors of the APIs and components known as Host Integration Objects, the Host On-Demand Connector, and Open Host Interface Objects (OHIO). IBM has made the host access programming capability available across its entire portfolio of Application Framework for e-business servers and clients, thus ensuring universal availability of this exciting new capability, delivering maximum flexibility to the customer's platform choice, and ensuring customer investment protection as their computing platform needs grow or change over time.

5.1 Host Access Beans for Java

The Host Access Beans for Java is a set of beans that provide direct manipulation of legacy application data by Java servlets and server applications as well as Java applets and applications. The legacy applications include 3270-based CICS, IMS, or OS/390 applications, 5250-based OS/400 applications, and VT100/220-based applications from other host systems such as UNIX and Windows NT. The beans are the quickest and easiest way to program host access applications from IDEs of the customer's choice, such as VisualAge for Java and VisualCafe. Because they are properly designed discrete functional units, they ensure that typically tedious and error prone host access functions can be implemented easily utilizing component software development technology.

5.1.1 What is a JavaBean?

"JavaBeans" is Sun's trademarked term for their architecture. To quote from Sun's JavaBeans API specification, "A JavaBean is a reusable software component that can be manipulated visually in a builder tool." When programmers write to the JavaBeans specification, they are developing "beans".

A bean written to the JavaBean specification has:

- Properties, which can be read and/or modified
- Methods, which are the functions that the bean can perform
- Events, which are the means by which beans communicate

Beans are intended for use in visual development environments, such as VisualAge for Java, and similar full-functioned development tools from other vendors. However, you can gain valuable experience with beans from simpler builder tools such as SunSoft's JavaBeans Development Kit (BDK), and the Lotus Bean Machine. Beans can also connect, through bridges, into other component models such as ActiveX.

The JavaBeans specification, which can be obtained from the following site, was developed by Sun with participation from IBM and other leading companies:

<http://www.javasoft.com/beans/docs/spec.html>

Host Access Beans for Java are written to the JavaBeans specification and are developed on top of Host Access Class Library. Application development is significantly simpler with the beans. Host Access Class Library can supplement the beans which may return HACL objects upon request. A typical application may be assembled out of:

- Host Access Beans for Java
- Beans from other sources
- Beans built specially for the project using Host Access Class Library and other libraries

Note: JavaBeans 1.0 requires a Java Virtual Machine at Version 1.1. If the beans are used to create a Web-served applet, a browser capable of fully supporting JVM 1.1 is also required.

5.1.2 What are the Host Access Beans for Java?

The nine Host Access Beans for Java are listed in Table 2 with the icons used to represent them in a typical visual-development environment.










icon	Description
	The Session bean is a non-visual bean that provides methods and properties for setting up and establishing communications with a host system.
	The Screen bean provides the graphical interface for displaying the host data from a Session bean.
	The Terminal bean combines the Session and Screen beans to provide a composite bean that encompasses both communication with the host and the graphical user interface for displaying the host data.
	The KeyPad bean provides a set of function keys which can be invoked by clicking with a mouse.
	The KeyRemap bean allows users to remap keystrokes to other keys including host function or aid keys.
	The FileTransfer bean provides a toolbar interface with the host file transfer facilities.
	The Macro bean is a non-visual bean that records and plays a single macro. It supports automatic screen recognition, variable prompt and extraction data areas, conditional branching and looping of screens, and dynamic loading of Host Access scripts. The Host Access script is described in XML document format; therefore, it is portable and easily interchangeable with other data sources.
	The MacroManager bean provides a toolbar interface for managing multiple macros. It allows users to record, play, load, delete and edit macros.
	The Color Remap bean provides an interface for modifying the colors displayed by the Screen or Terminal beans.

Table 2. The Host Access Beans for Java

Note: ColorRemap bean is only available with Host Access Beans provided with Host On-Demand.

For additional details, please refer to the Host Access Beans for Java book, which is supplied with Host On-Demand as beanReference.html in the \hostondemand\hod\[language]\doc\beans\ directory.

If you are using Personal Communications, you can find it in the \Program Files\Personal Communications\doc\beans\ directory.

5.1.3 Application development

The way developers work with beans is centered around a Session bean that secures a connection to a host, and a Macro bean that utilizes a "Record and Play" technology to "mine" application data on the host. Mining means navigating a series of host application screens and extracting data along the way to create an object that contains the extracted data. The navigation encapsulates the data with the commands, function keys, and input values fed to the screens, with a marking capability that allows variable input and output areas to be defined, as well as a looping and conditional branching capability from screen to screen.

This experience is captured in an XML based Host Access script document, which now represents a piece of business logic sliced from the existing application. To incorporate the business logic, a developer simply connects the Session bean and Macro bean, sets the host properties in the Session bean, sets the script in the Macro bean, and starts the Session to play the Macro bean. If this were to be done using Host Access Class Library, every screen recognition, field manipulation, key interaction, and error handling, etc. would need to be coded manually. Host Access Beans for Java have automated most of these tasks.

The Macro bean is very sophisticated but easy to drive through a Macro Manager bean which can turn multiple host application screens in minutes into a new business object. A visual development IDE is not necessary but it can speed up application development. By dropping and wiring the beans in a visual development IDE of the customer's choice, an e-business application can be developed in days. No special tools are required because the beans can be imported to the IDE and added to its component palette which then becomes a natural extension of the IDE.

One example of usage would be to write an application that combines the output of multiple application screens, that may or may not be from the same application, into a single consolidated screen to improve the productivity of existing application users. But remember, Host Publisher can also generate these kinds of applications.

5.2 Host Access Class Library (HACL)

IBM SecureWay Host Access Class Library (HACL) delivers a cross-platform application programming interface (API) for customer and independent software vendor (ISV) use in creating graphical user interfaces for existing host applications, without requiring changes to the existing applications. This API provides access to 3270, 5250, and VT data streams. IBM has made the Host Access Class Library available across its entire portfolio of SecureWay communication servers and clients at no additional cost, thus ensuring universal availability of this exciting new capability, delivering maximum flexibility to the customer's platform choice, and ensuring customer investment protection as their computing platform needs grow or change over time.

HACL is not the simplest way to program Web applications for host integration, but in the case of special programming requirements not provided for in other programming APIs, HACL provides a flexible solution.

An example of using HACL would be for a customer or ISV to write an application that replaces the native interface of an existing application with a more graphically enriched Web-like interface. The new interface can improve the productivity of end users who are not experienced with using native emulation screens. Screen Customizer, however, would be an easier way to generate this type of application.

5.2.1 HACL programming options

HACL supports all of the major development and client environments. New applications can be written in Java, C/C++, Visual Basic, PowerBuilder, or LotusScript.

HACL comes in several flavors:

- HACL for Java

HACL for Java provides the full function of emulators such as Host On-Demand and Personal Communications. The developer can create a HACL application or applet that provides emulator functions without a separate emulator program. Applications or applets written by HACL for Java are independent of client platform.

- HACL for C++

The C++ API allows you to interact with Personal Communications sessions from programs using the C++ language.

- Automation Objects

The Automation Objects API attaches to Personal Communications. With this API, the developer can use development tools that have the ability manage and implement OLE automation objects.

- LotusScript Extension

This API allows you to interact with Personal Communications sessions from LotusScript programs. These classes of API are similar to the C++ API.

5.2.2 HACL for Java

HACL for Java is a set of classes/objects that developers can use to represent a host-based session. As far as the end user is concerned, all the session work is done internally. Communication with the host is achieved without ever seeing a “green screen”. No memory or CPU is required to maintain a visible screen object. This means HACL has a significant advantage in reducing the overhead of running applications.

Although the HACL versions provided with Host On-Demand and Personal Communications are assumed to be 100% portable, there are some small differences. The HACL version provided by Host On-Demand is a pure Java implementation which only supports TCP/IP connections. The HACL version provided by PCOMM maps internally to native interfaces. Only the HACL version provided by Host On-Demand supports the creation of applets which can run in a Web browser. HACL for PCOMM is intended to write Java applications and to develop ECLApplets for PCOMM's run applet feature.

5.2.2.1 Sample program

Host On-Demand provides a rich sample application called LaunchPad, which you will find in the \hostondemand\HOD\toolkit\hac1\samples\LAUNCHPAD\ directory. It logs on to a VM host system, collects a list of host-file names, and presents the information to the user in a workstation dialog box.

The major elements of most HACL applications are illustrated in this sample and the source code contains ample comments. It provides an excellent model for Java HACL development, with a distinct object defined for each unique host screen. LaunchPad also illustrates a strategy for handling host-screen responses. Look in particular at the ScreenAction.java class, which is the parent class for all of the screen objects.

Note: Although Personal Communications does not include the LaunchPad sample, you can download it from the following URL:

<http://www.ibm.com/software/network/technology/hac1/samples.html>

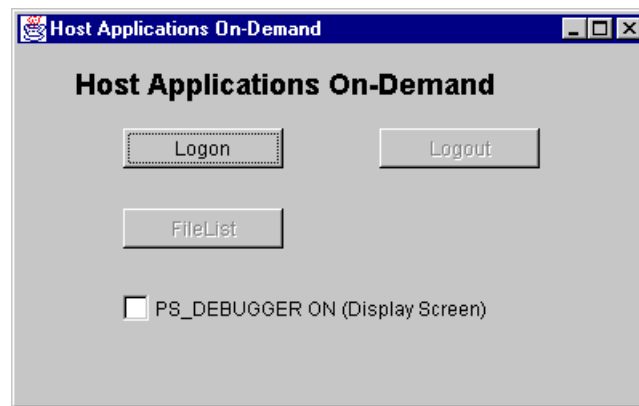


Figure 68. LaunchPad logon screen

5.2.2.2 Application development

Developing HACL for Java applications for Host On-Demand is relatively straightforward for a programmer familiar with Java and object-oriented application development. Any emulator API application is typically composed of the following major logical elements:

1. Reading the presentation space

The application needs to read the “virtual screen” into application variables and have logic to identify and validate that the appropriate host screen is displayed.

2. Synchronize with the host

The application must be able to detect when the host has finished sending data to the screen. The data may be sent to the general output section of the screen and/or to the bottom line where the Operator Information Area (OIA) is displayed. OIA information is one way that a host sends information to a user about a host processing delay error. Therefore, the HACL program needs to be able to detect information arriving and potentially act upon this information.

3. Sending data to the host

The application must send “virtual keystrokes” to the host in the same way that a user would type data in the emulator session window. After typing data into the entry fields, the application needs to simulate pressing a key to transmit the data, for example the Enter key or a PF (Program Function) key. This also requires that the application recognize which fields are input fields in which data can be entered.

HACL accommodates these functions and more with the incredible amount of functionality built into its objects. There are five basic functions that will almost always be an integral part of your application. The methods are:

- StartCommunication - Establishes a session to the host
- SearchString - Searches for text
- SendKeys - Sends text
- GetString- Copies text to a string
- StopCommunication - Disconnects from the host

Beyond these basics, HACL includes a wealth of functionality, including screen recognition technology that fires events when a screen is loaded into the presentation space that meets a pre-defined set of criteria.

5.3 Host Integration Objects

Host Integration Objects are provided by Host Publisher as a byproduct of its Studio component. They are beans made from Host Access Beans for Java's Session and Macro beans, as well as Java's JDBC database access class library. They represent discrete units of business logic extracted via the Studio from existing legacy applications and data, for example, the customer purchase order processing logic from a CICS green screen application. The difference between Host Integration Objects and Host Access Beans is that the former include the business logic as an integral part of the bean, but the latter keeps it as a separate Macro script. For example, using a purchase order Host Integration Object, a programmer needs only to instantiate the object, set input property values for the order such as item name and quantity, invoke a common transaction method, and check the output property values. On the other hand, when using the Session and Macro beans, the programmer needs to instantiate both beans, wire them together, set the Macro context to the script, implement Macro event handlers, feed input values at proper Macro prompt events, and extract output values from proper Macro extract events. Using Host Access Beans is a lot simpler than coding from scratch using the Host Access Class Library, but requires more programming effort than when using Host Integration Objects.

Since the main purpose for Host Publisher is to allow customers to combine their existing application and database sources quickly to generate e-business applications using HTML Web pages as the front-end, Host Integration Objects are most powerful when used together with Host Publisher for e-business application development.

Developers get a quick jump-start with Host Publisher. It helps host application administrators and developers who may not be that familiar with the Web to create applications for Internet users who are familiar with standard HTML browsers and are accustomed to Web response time. The browsers need not be Java-enabled. Developers can take their familiar "green screen" applications all the way to the Web users without writing a single line of code initially. They can

then manipulate the Host Integration Objects, adding other data sources, and enhance the generated applications over time with minimum disruptions to the users.

Host Integration Objects support Host Publisher's connection pooling which boosts the performance for general Web user requests that do not require special authorizations for using backend resources.

5.4 Host On-Demand Connector

The Host On-Demand Connector is Host Access Beans for Java in the context of the WebSphere application development environment for developing Web server-based servlets or applications. It supports the Common Connector Framework (CCF) interface required by Visual Age for a common tooling methodology across all connectors. Connector is the term used in WebSphere to represent a software component that accesses backend resources such as 3270 and 5250 applications.

VisualAge for Java 3.0 Enterprise Edition ships with the HOD Connector. Additionally, the HOD Connector can be purchased and subsequently downloaded for Visual Age 2.0 from the IBM VisualAge Developers Domain page at <http://www7.software.ibm.com/vad.nsf/>. You will also need to purchase a license to “deploy” your applications, even if you received the HOD Connector in the 3.0 Edition of VisualAge. The deployment license is obtained by purchasing Host On-Demand.

5.5 Host Access Controls for ActiveX

The developer who is familiar with ActiveX technology may use the Host Access Controls for ActiveX, which are actually an implementation of Host Access Beans, allowing the developer to access all functions which are available on Host Access Beans.

5.6 OHIO

Open Host Interface Objects(OHIO) is an industry-standard version of HACL. It is an attempt to standardize the proprietary advanced Object Oriented APIs provided by the following vendors:

- IBM
- Attachmate
- WallData

IBM and Attachmate co-authored the OHIO Internet Draft.

Host On-Demand 4.0 provides OHIO APIs. HACL classes implement OHIO interfaces. Host On-Demand provides an online reference of OHIO which is located in the \hostondemand\HOD\doc\ohio\ directory.

Note

The definition of the OHIO API has not been completed and the final version might differ from what is presented in HOD V4.

5.7 System requirements

Requirements for Host Access Controls for ActiveX, HACL, and Host Access Beans for Java can be divided into two parts. One is the run time environment which allows you to run applications. The other is the development environment which is needed for developing applications.

Host On-Demand (HOD), Personal Communication (PCOMM), and Communications Server include a core set of APIs.

Note: The HACL shipped with Communications Server is a limited implementation which actually only works on Communications Server. If you want to use the full function of HACL, you need to purchase Host On-Demand or Personal Communications.

There are some differences between HOD and PCOMM in what is provided. The differences are shown in Table 3.

Table 3. Host On-Demand vs Personal Communications

	Host On-Demand	Personal Communications
Java applications	X	X
Java applets	X	
ECLApplets	X	X
Host Access Beans for Java	X	X
HACL Automation Objects		X
Host Access Controls for ActiveX	X	X
Support for LotusScript Extension (LSX)		X
C++		X
Java servlets	X	
Server applications	X	

5.7.1 Requirements for the run time environment

The following are the run time requirements for each type of programming:

HACL for Java and Host Access Beans for Java:

HACL for Java and Host Access Beans for Java can do emulation internally, so no emulator software is required to run the application.

HACL for Java applets and Host Access Bean applets are downloadable and executable on client workstations using standard Web and Java technology. The only client requirement is a Java-enabled Web browser that supports the Java 1.1 specification.

HACL for C++:

HACL for C++ requires a separate 3270/5250 emulator. You must install IBM SecureWay Personal Communications to run applications using HACL for C++.

Host Access Controls for ActiveX:

Host Access Controls for ActiveX has an emulator function built in, so no emulator software is required to be installed on client. ActiveX controls only run with Microsoft Internet Explorer.

HACL Automation Objects:

HACL Automation Objects need an associated emulator session. You must install Personal Communications to run the application using HACL Automation Objects.

LotusScript Extensions:

- Lotus Notes V4.5.1 or later
- Personal Communications V4.2 or later

5.7.2 Requirements for development environment

This section describes the tools for developing applications.

HACL for Java and Host Access Beans for Java:

At least one of the following Java 1.1 environments must be installed to build Java class files:

- Java Development Kit, V1.1.8 by Sun Microsystems. It can be obtained from:
<http://java.sun.com>
- IBM VisualAge for Java, V2.0 or 3.0 Enterprise Edition is needed specifically for Host on-Demand's HACL.
- Symantec Visual Cafe, V3.0.
- **Only for Host Access Beans:** Other prevalent Java IDEs.

HACL for C++:

One of the following compilers must be installed to develop C++ applications using HACL:

- IBM VisualAge C++
- Microsoft Visual C++

Host Access Controls for ActiveX:

Microsoft has fully implemented ActiveX technology into a variety of its programs including the Win32 operating systems and Microsoft Office. ActiveX controls can be implemented in Microsoft IDEs, such as Visual Basic and Visual C++, by inserting them as new components.

Host Access Class Library Automation Objects:

A development environment which can handle COM-automation objects is needed. For example, Microsoft Visual C++ and Visual Basic can use HACL Automation Objects.

LotusScript Extentions:

Lotus Notes V4.5.1 or later

5.8 Where to find more information

- *IBM SecureWay Host On-Demand 4.0: Enterprise Communications in the Era of Network Computing*, SG24-2149-01, available at <http://www.redbooks.ibm.com>
- *Programming with the Host Access APIs*, SG24-5856
- <http://www.ibm.com/software/network/hostondemand/library/publications/beans.html>

Chapter 6. Lotus Domino

Lotus Domino is an integrated solution geared to the new Web-oriented society we are living in. Domino is fully Web-enabled and addresses the requirements of modern society where the computer is used not only for business purposes but also for education, health-care, communication, and many other activities.

Lotus Domino implements a client/server architecture. It contains databases, e-mail applications, a Web server component and several connectors that allow connections from Domino resources to non-Domino resources.

Lotus Domino is the server side of this environment. Lotus Notes client, and all Web browsers available in the market can be considered clients of a Domino application.

Lotus Domino servers are available for AS/400, S/390, Windows NT, LINUX, HP-UX, AIX, Sun Solaris, and OS/2.

6.1 Lotus Domino and IBM WebSphere

Although Domino and WebSphere both provide HTTP servers, they do not compete with each other. WebSphere is well suited for large transaction applications, with thousands of transactions per second accessing large databases. More information on WebSphere can be found in Chapter 6, "Lotus Domino" on page 95. Domino is suitable for workflow applications, messaging, user-group and discussion databases, departmental applications triggering e-mails and collaborative work. Domino is very powerful for managing mobile workers who need global access to applications and data, working connected to or disconnected from the server.

Domino and WebSphere are complementary when used in a collaborative way, integrating high transaction volumes with collaboration tasks, either running together on the same machine, like an S/390, or on different machines.

Customers can use WebSphere Studio tools to design Domino applications. Domino R5 incorporates the WebSphere Standard Edition as part of the Domino run-time environment. VisualAge for Java, a WebSphere Application Server component, includes Domino Java class descriptions that WebSphere can use to invoke Domino object services.

The unique point where WebSphere and Domino conflict is at the HTTP server level when both are running on the same machine. Each product has its own HTTP Server component. The IBM HTTP Server is a general purpose Web server, with all the capabilities and function a Web server should have, including most advanced features. The Lotus Domino HTTP server component is a Web server oriented toward supporting Domino applications. It does not support many of the advanced features you would find in the IBM HTTP Server.

HTTP servers usually listen on TCP/IP port 80, the well-known port for HTTP services. When running both the WebSphere and Domino HTTP servers in one machine, one server (most likely the Domino HTTP server) would need to be set up to listen on a different port in order to avoid conflict, or some other alternative

used to avoid conflict. In the future, this restriction will be removed. Lotus Domino will deploy the WebSphere HTTP server and not its own.

6.2 Lotus Domino and Java

Domino R5 is a complete Web application server which fully supports the Java environment. In addition to Formula, LotusScript, and JavaScript, objects in Domino applications can have Java code attached. The Domino Designer R5 includes a Java editor and Java Virtual Machine (JVM) for developing applications. This allows you to create and compile Java agents, and edit all scripts and formulas.

Domino provides a programming interface for Java applications and applets, which can operate locally by accessing installed Domino software or remotely by connecting to a Domino server using IIOP protocols.

Domino supports CORBA to build distributed applications, as well the IIOP protocol. Domino also supports JDBC calls to allow Java programs to access Domino data. The Domino JDBC driver makes Domino databases look like relational databases, allowing an application to perform a JOIN operation with data from other relational databases, like DB2.

6.3 Lotus Domino integration to enterprise

Lotus provides several utilities for connecting Domino R5 to enterprise resources. These utilities can be divided by the four groups of services they provide:

- Access to enterprise relational database systems
- Access to ERP applications systems
- Access to unstructured data
- Access to enterprise transactions systems

Access to enterprise relational databases, ERP applications and access to unstructured data are provided by the following Lotus utilities:

- Domino Enterprise Connection Services (DECS) - provides a real-time forms-based interface to enterprise data. DECS is shipped with Domino and has been available since Domino R4.6.3.
- Lotus Enterprise Integrator (LEI) - provides scheduled and event-driven high-speed data transfer and synchronization capabilities between Domino and enterprise systems. LEI is a separate product available for Domino R5. It is the follow-on of NotesPump.
- Lotus Connector LotusScript Extension (LC LSX) - provides LotusScript access to enterprise systems. The LC LSX is shipped with Domino R4.6.3 and above and with LEI 3.0.
- Lotus Connector Java classes (LC Java) - provide programmatic access to enterprise data through a common set of Java classes. The LC Java classes are available for download at <http://www.edge.lotus.com> and also shipped with LEI 3.0.
- Lotus Connector API (LC API) - provides C/C++ access to enterprise systems.
- @DB Functions - The Domino functions @DBCommand, @DBLookup and @DBCColumn enable you to access RDBMSs that use the underlying ODBC

interface. The @DB formulas are read-only , but provide easy-to-use access to ODBC-compliant databases.

Access to enterprise transactions systems are based on MQSeries technology and are provided by the following Lotus utilities:

- MQSeries Enterprise Integrator (MQEI)
- MQSeries link LotusScript Extension (MQLSX)

6.4 Lotus Domino access to enterprise data and ERP systems

Enterprises that have a Lotus Domino environment have several choices when determining how to connect Domino to enterprise data and make it available to the Web. These choices primarily depend on the amount of programming they want to do to control the presentation of the data on the Web.

The Lotus Domino Server acts as both a Notes server and a Web server. It allows access from Web browsers to Notes applications by dynamically translating Notes applications into HTML.

6.4.1 Options that require programming efforts

The following Domino options require programming effort:

- @DB functions
- Lotus Connector LotusScript Extensions (LC LSX)
- LotusScript Data Object (LS:DO)
- DB2LSX
- Java classes
- Lotus Connector API (LC API)

6.4.1.1 @DB functions

Notes applications have long had the @DB functions available in the Notes formula language that allow access to external data. @DBCcolumn, @DBLookup, and @DBCommand are simple ways to get read access to ODBC databases.

6.4.1.2 LC LSX

Lotus Connector LotusScript Extensions (LC LSX) gives a high level of programmatic control over data access. This option requires knowledge of LotusScript and agents. The connections are non-persistent. LC LSX is best for low volume/low performance application requirements.

LC LSX has extensions to most relational databases and directory services like LDAP (Lightweight Directory Access Protocol), Novell Directory Services (NDS), Lotus Domino Directory as well as to EDA/SQL, file systems and text files.

6.4.1.3 LS:DO

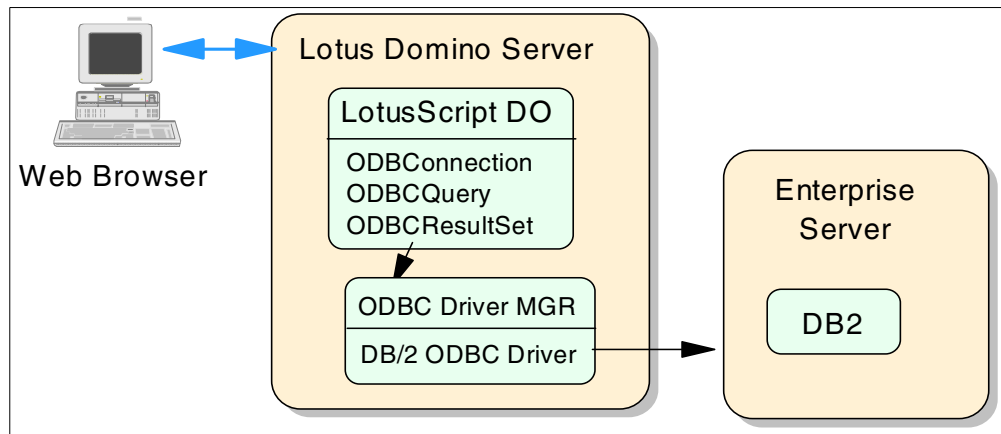


Figure 69. LS:DO

LotusScript Data Object (LS:DO) is a LotusScript extension that provides full read and write access to ODBC-compliant databases from Notes forms and agents using LotusScript. It provides functions like on-the-fly lookup, immediate updates, input validations, duplicate entry warnings, and mobile user queries. LS:DO is automatically available in Lotus Domino.

Three object classes are used to establish the connection, execute an SQL statement, and get the result set:

- ODBCConnection
- ODBCQuery
- ODBCResultSet

LS:DO is better suited for low-volume data transfer from a performance perspective. Sessions are non-persistent. LS:DO allows multiple query/result set objects against the same connection and bidirectional scrolling over the result set.

For connections to DB2 databases, DB2 Connect can provide both the ODBC driver and the connectivity to the DB2 database. For more information on DB2 Connect see 11.2, “DB2 connectivity” on page 202.

LS:DO and DB2LSX are similar in the way they process data, but different in the way they are programmed. LS:DO, by default, autocommits when an update operation occurs, while DB2LSX requires explicit commands. LS:DO accesses data through ODBC drivers, while DB2LSX uses the Call Level Interface (CLI) mechanism.

6.4.1.4 DB2LSX

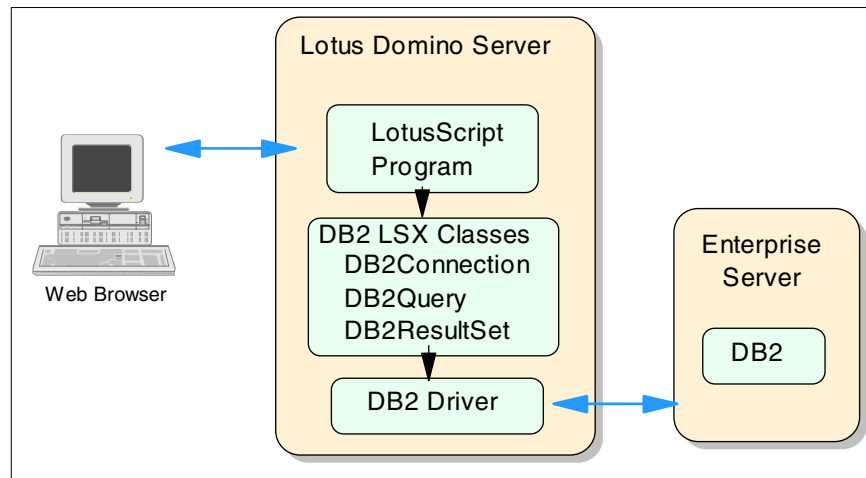


Figure 70. DB2LSX

DB2LSX provides native access to DB2 through the DB2 Call Level Interface (DB2 CLI). It is a set of three LotusScript object classes designed specifically for DB2 data access:

- DB2Connection
- DB2Query
- DB2Result

DB2 is accessed natively, bypassing ODBC and extending functions not supported by ODBC, including large object data and BLOBS, user-defined types and functions, and transaction commit and rollback.

DB2LSX provides non-persistent sessions and supports multithreading. It differs from LSX LC in that DB2LSX is based on LS:DO and is specific to DB2. The LSX LC, provides a common API across connectors. In addition when used with a connector like the Lotus Domino connector for DB2, you have the ability to access data without the need for DB2-specific commands, such as SQL statements.

Requirements

- DB2LSX Release 2 requires Notes 4.6.1
- DB2 CAE
- Even though DB2LSX accesses DB2 natively, you are still required to register the database as an ODBC resource if using DB2LSX V1.1. DB2LSX V1.2 does not have this requirement.

6.4.1.5 Java classes

The JDBC API defines Java classes to represent database connections, SQL statements, result sets, etc. It allows a Java programmer to issue SQL statements and process the results. JDBC is the primary API for database access in Java. The JDBC API is implemented through a driver manager that can support multiple drivers connecting to different databases.

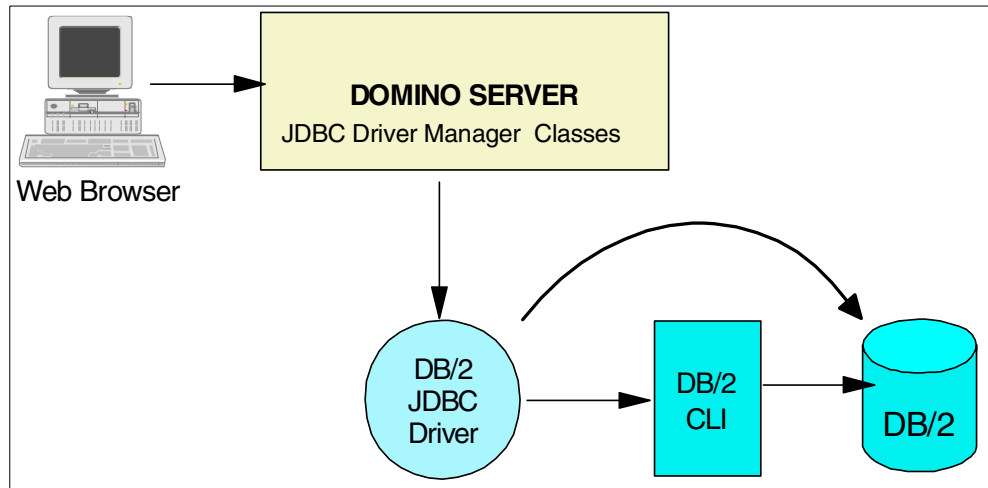


Figure 71. Java access to DB2

These JDBC drivers are implemented as wrappers to the DB2 implementation of CLI. Just like CLI, JDBC is a dynamic SQL interface, where all SQL statements in transactions are evaluated on-the-fly. No prep or bind steps are needed to run a JDBC program. However, JDBC presents a convenient object-oriented version of CLI that makes a JDBC program's structure resemble classical embedded SQL programs.

Domino introduces several major enhancements related to Java:

- Java applets to a Domino application

By adding Java applets, you can create a richer, more interactive Web browser and Lotus Notes client user experience. Java applets are treated like any other Notes object. They can be stored in the Domino object store, they are easily accessible from the menu, and their properties can be easily modified. By storing Java applets within the Domino object store, you can take advantage of Domino's replication technology for keeping Java applets synchronized in multiple locations. To prevent hostile applets from modifying or destroying information stored in Notes databases, Java applets cannot directly access the Notes Java classes at this time, unless the Lotus Notes client is installed on the Web browser machine. Lotus Notes Designer is not a Java programming environment; therefore you have to develop your Java code outside the IDE.

- Java applets in Lotus Notes Designer for Domino and Lotus Notes clients

Lotus Notes Designer for Domino and Lotus Notes clients contain the JVM. Therefore when you place a Java applet in an application that is being developed, the applet automatically runs in place. Thus you can view and test the applet while designing the application.

- Domino server agents in Java, using the Java interface to the Domino object services

With Domino's Java server agent support, you can use Java to develop server agents to use along with the in-the-box simple agents, formula agents, and LotusScript agents. These agents can perform a range of tasks from e-mail filtering to knowledge management to automated server administration. The agents can be triggered by server events or run as scheduled tasks. The

ability to write agents in Java allows you to get the most out of your Java programming resources while still utilizing the power of the Domino environment for your Web needs.

JDBC-ODBC Bridge

The JDBC-ODBC bridge provides JDBC access through most ODBC drivers. Because the bridge requires access to the ODBC driver and the database client, its usage is most appropriate for application server code written in Java in a three-tier architecture. Domino provides such a bridge.

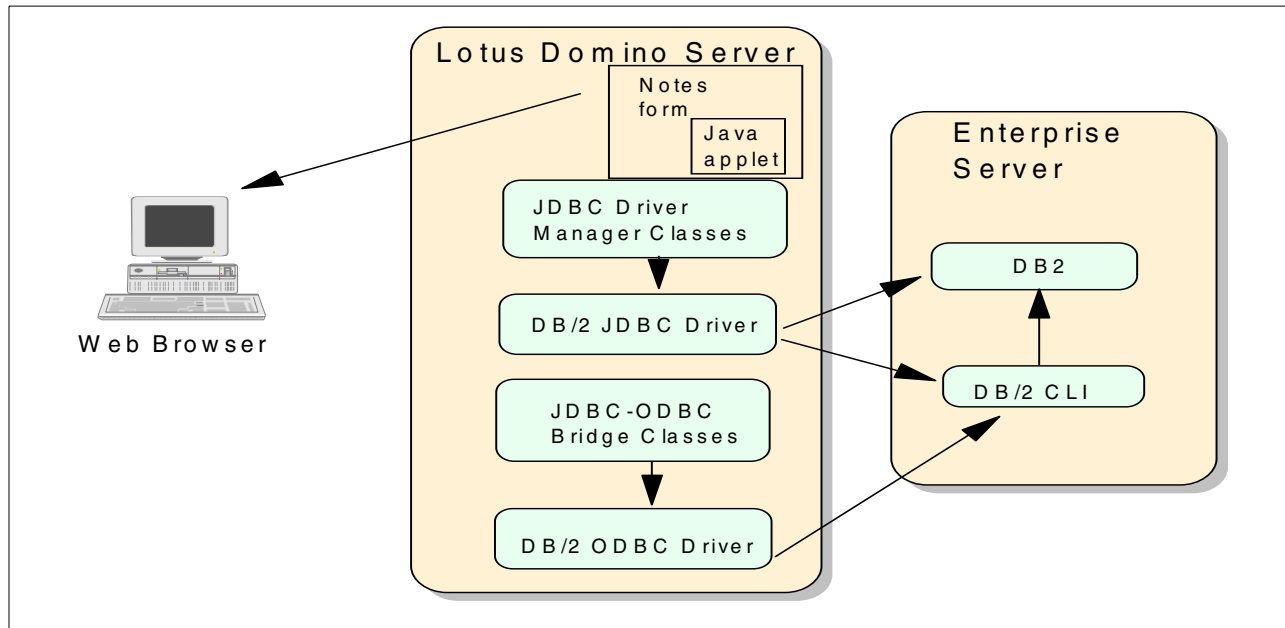


Figure 72. JDBC connectivity

The JDBC-ODBC bridge makes the transition from JDBC to ODBC. DB2 has both JDBC and ODBC interfaces. Java Database Connectivity (JDBC) Java classes allow access from Java to a DB2 data source and the ability to retrieve results.

Features

- Allows multithreading
- Work-around for 64 KB LotusScript barrier (removed in 5.0)

Skills Needed

- Java programming
- LotusScript programming

6.4.1.6 LC API

Lotus Enterprise Integration architecture divides products into two areas: Connectors and Tools. Connectors deliver general connectivity as components that provide a consistent interface to a variety of systems. Tools are products built to use the Connectors to provide higher-level functionality.

The Connectors offer a consistent interface to a wide variety of systems. This LC (Lotus Connector) API is a more general interface than one like ODBC, which is RDBMS-oriented. The LC API uses more generic terminology, and rather than

operating on SQL queries and columns, works on the more general principle of transferring named fields to and from various systems. The Connectors have a unified object model and a multi-language API (C, LotusScript, and Java) which enable programmatic development. The LC API Toolkit allows third parties to extend the set of Connectors available to other systems.

The EI Tools use the Connectors, which allows any Tool to work with any Connector without requiring any specific knowledge of the corresponding external system. They support codeless (non-programmatic) application development - replacing common and complex program development with a consistent forms-based user interface, and adding Enterprise functionality such as scheduling, logging, remote management, etc.

The LC Toolkit documentation is installed as lc30api.nsf in the Domino or Notes doc directory.

6.4.2 Options that to not require programming

In this section we describe the non-programming options for enterprise integration with Lotus Domino:

- Lotus Enterprise Integrator (LEI)
- Domino Enterprise Connection Services (DECS)
- ActiveX Data Object (ADO)

6.4.2.1 Lotus Enterprise Integrator (LEI) and NotesPump

LEI, formerly called NotesPump, is used to exchange large volumes of data between Lotus Notes and relational database systems on an event-driven, scheduled, or ad-hoc basis and can be used for enterprise-wide connectivity. LEI can access Lotus Notes, IBM DB2, Oracle, Sybase, and ODBC-compliant databases on a wide range of mainframe and client/server platforms.

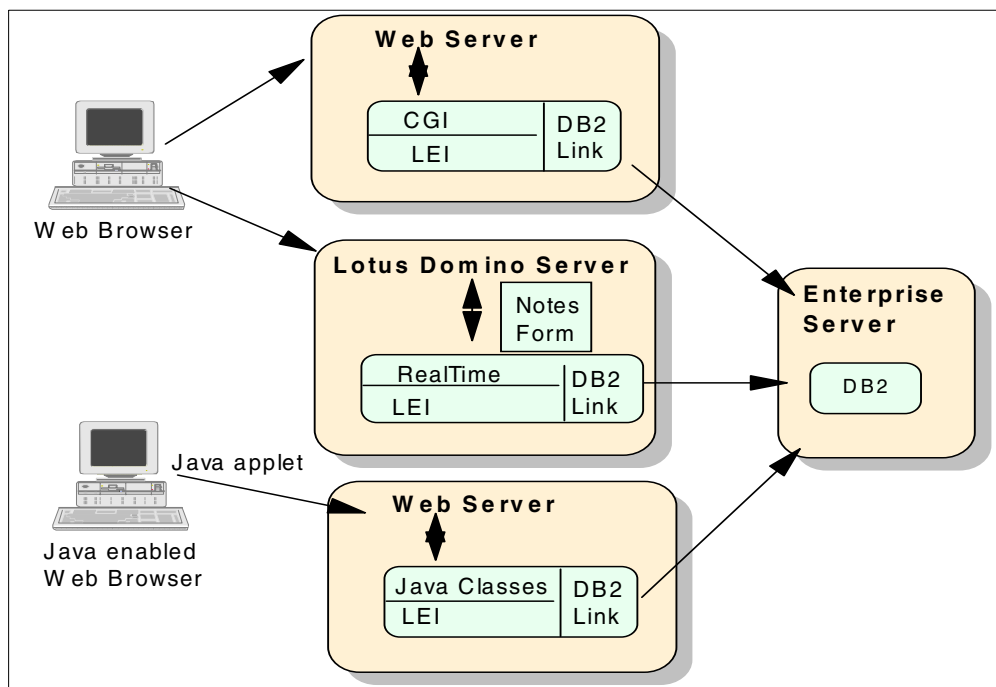


Figure 73. Lotus Enterprise Integrator

LEI offers three interfaces for Web access to DB2 data:

- LEI RealTime to access DB2 data to populate Notes forms
- Java classes for Java applet access to DB2 data
- Web server CGI for HTML access to DB2 data

LEI acts as a data distribution server, able to transfer large volumes of data between data sources. It provides both forms-based and programmatic access to many data sources, including Notes, DB2, ODBC, and many others. It allows sources that usually don't communicate to interface with each other.

LEI administration is Notes based, consisting of Notes documents that define links (data connections) to data sources and activities (transfer instructions). Transfer operations are created using Activity types, including things like RealTime Notes, Direct Transfer, LotusScript, Java, and others.

LEI comes with the Common Gateway Interface (CGI) that can be installed on a Web server and which allows Web browser clients to submit and receive data queries.

When used in conjunction with a Lotus Domino server, LEI RealTime Notes Activity can be used to access DB2 data to populate Notes forms. LEI catches and handles Notes events as they occur. Users can open or update Notes forms, causing real-time queries or updates to externally supported LEI sources (for example, DB2).

Domino translates the Notes form for HTML presentation to a Web client. LEI RealTime Notes Activity is functionally equivalent to DECS.

Java applets can also use the LEI Java classes to access DB2 or other data.

Other features of LEI are:

- Provides persistent parallel connections.
- Allows full text searches to backend data.
- Offers replication between dissimilar databases.

Requirements

- LEI runs on Windows NT, OS/2, HP-UX, Sun Solaris on SPARC, Sun Solaris Intel Edition, IBM AIX, and OS/400. NotesPump runs on the same systems, plus S/390.
- The LEI Server engine must be installed on either a Notes Client or Lotus Domino Server machine.
- The LEI Administrator requires Lotus Notes 4.5 or above.
- DB2 Connect (follow-on for DDCS) or DB2 CAE (if the Notes server is not on the S/390).

Availability

- NotesPump 2.0 is available in Domino.Connect.
- LEI is sold as a separate product.

6.4.2.2 DECS

DECS is a new feature that was added to Domino R4.6.3 and R5. It runs as a server task within Domino and provides real-time access to enterprise data. DECS administration is forms-based.

DECS can access any relational database as well interoperate with ERP systems like SAP R/3.

DECS extends the Domino application to query and modify enterprise data. When a Notes or Internet client accesses data through a Domino form connected to an enterprise system, DECS transparently retrieves and updates the enterprise data. The client needs no additional software. DECS requires no programming.

DECS features:

- Multi-value support for one-to-many data relationships. The expand/collapse metaconnector from Lotus Enterprise Integrator was added to DECS.
- Automatic reconnect if a connection is dropped by the enterprise system.
- Stored procedure support.
- DECS can be used with the SAP R/3, PeopleSoft, BEA Tuxedo, JD Edwards and Oracle Applications Lotus Connectors.
- DECS provides real-time access to DB2, EDA/SQL, ODBC, Oracle, Sybase, File System and Notes data types.
- LC LSX and RDBMS connectors support connection pooling when using LotusScript.

6.4.2.3 ActiveX Data Object (ADO)

ADO was designed by Microsoft to access databases of all types. ADO is the interface for OLE DB and is one component of Microsoft Universal Data Access.

ADO is a COM-based (Common Object Model) component and any application that can work with COM can use ADO.

You can access enterprise resources using Microsoft's ActiveX Data Object. You must install and register the ADO objects on a Microsoft Windows system. LotusScript will invoke OLE automation to interact with the ADO objects.

6.4.3 Data access roadmap

As you can see, there are several options to access enterprise data from a Domino application. Some of the options overlap, causing some confusion about when to use one or the other. Some scenarios could have more than one option that fits as the best solution. Below, you can find a summary guideline to help you decide which option to select for each scenario:

@DB Functions - Use this set of programming functions when you need to get data from several records from the host to create keyword lists, to run stored procedures, or when you want to look up a single field value.

DECS - Use DECS when your application accesses individual records on the host and they can be identified through a simple keyed lookup.

DB2LSX - If you are skilled in LotusScript and your database is on DB2, this is the right option. You could also consider using LS:DO, as it is very similar to DB2LSX.

LC LSX - If you are skilled in LotusScript and your database is *NOT* on DB2, this is the right option.

LEI - Use LEI when you have a high volume of data to transfer among servers, when you want to schedule a data transfer or you need to synchronize data between Domino and the relational database.

Java Classes - If your programming preference is Java and you want flexibility to manage data access. You could use JDBC drivers if native connectivity is not required.

6.5 Access to enterprise transaction systems

Lotus connectors for transaction systems are based primarily on MQSeries solutions. MQSeries controls the flow of data between the Notes application and the numerous operating systems and their different host transaction systems, as well as the translation, connection, delivery and reply from the target system(s).

MQSeries supports more than 25 different platforms, including Windows, Windows NT, OS/390, CICS, IMS, AIX, OS/2, OS/400, Tandem, Solaris, HP-UX, and Digital VMS VAX.

IBM MQSeries allows Notes to participate in transaction systems by allowing Notes to initiate transactions and by acting as a store-and-forward data repository for data from transaction systems. The enterprise-critical data and the associated business rules continue to be managed by the transaction system. This allows Notes applications to take advantage of the data storage and processing logic of very large, distributed transaction systems.

6.5.0.1 MQSeries Enterprise Integrator (MQEI)

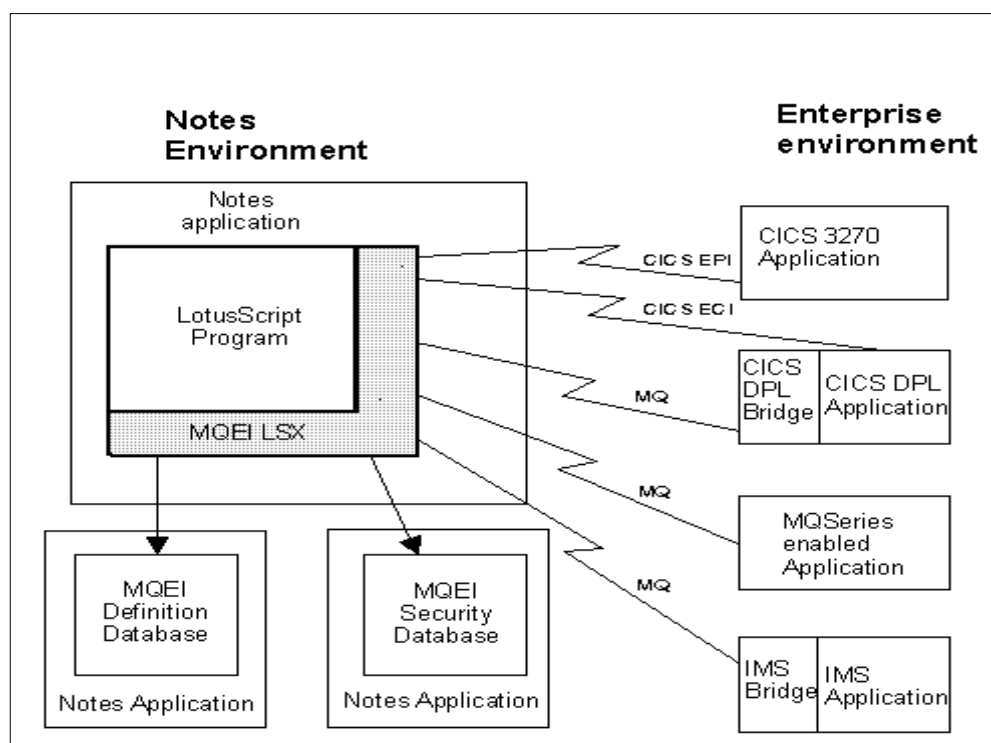


Figure 74. Using MQEI to access the enterprise environment

The MQEI is a LotusScript extension that provides a simple, common API for enterprise system access. This new API gives your Web browser or Notes client application, via LotusScript, connectivity to MQSeries, IMS (via MQSeries), CICS DPL (via MQSeries or CICS Client) and CICS 3270 (via CICS Client). LotusScript applications using the MQEI have independence from both network configurations and message formats, with integrated security features enabling your users to seamlessly sign-on to multiple enterprise systems from one common easy to use front end.

Direct access to CICS transactional applications running off any of the CICS family of transaction monitors is possible using the MQEI, which uses the CICS Client ECI and EPI programming interfaces internally. The family includes IBM Transaction Server for MVS/ESA, IBM Transaction Server for Windows NT, and TXSeries. MQEI also provides access to the MQSeries family of message-queueing middleware, enabling Notes to interact with most MQSeries-enabled applications on more than a dozen different enterprise computing platforms, including the MQSeries bridges to both IMS and CICS on OS/390.

The MQEI also enables Lotus Notes clients and servers to work with multiple systems simultaneously, such as accessing a 3270 transaction monitor “green screen” running on IMS or CICS. The MQEI interface allows Notes application developers to set up message and service definitions in Notes databases, minimizing the need for developers to code network-specific application calls.

Technical advantages

The technical advantages of using the MQEI interface are:

- Simplified access from any Web browser or Notes client application via LotusScript to MQSeries, CICS, and IMS.
- Web access to more than 25 different mainframe or host system platforms such as HP-UX, Tandem, DEC, Sun Solaris, Windows NT, OS/400, and OS/390 MQSeries.
- Direct access to the CICS family of transaction monitors via a CICS client.
- A configuration Domino database containing definitions of the messages that can be sent to enterprise systems, and definitions of the applications that run on these systems that minimize programming. Rapid Application Development (RAD) is possible.
- Seamless security: Integration of the identification and authentication services of Lotus Domino with the access controls provided by CICS, IMS and MQSeries, provides a secure, seamless sign-on capability to your enterprise systems.
- The ability to populate the configuration database from CICS BMS maps.
- Easy interface: Faster host system and mainframe integration deployment are achieved using this RAD middleware tool. The MQEI provides a set of classes to a Notes programmer with a simple interface to communicate with enterprise applications. Details of any transaction management system being used by the enterprise application are hidden.
- Native access: Integration tasks are performed through an optimized connection with built-in capabilities to access unmodified enterprise applications running under CICS and IMS, as well as MQSeries-enabled

applications on other enterprise server platforms. This includes CICS Dynamic Program Link (DPL) programs plus CICS and IMS 3270 transactions.

- Enhanced message building facilities: Notes programmers can build and interpret messages by reading and writing named fields within the message, without having to know precise details of the message format.
- Automatic data conversion of both character and numeric data.

6.5.0.2 MQSeries Link LotusScript Extension (MQLSX)

The MQSeries Link LotusScript Extension is fully integrated with the Domino application development environment and can be used in any application supporting LotusScript. It incorporates the full power of the IBM MQSeries Message Queuing Interface (MQI), giving your LotusScript applications the ability to interact with any MQSeries application throughout your enterprise.

MQLSX enables Notes applications to use message queuing to participate in message-driven processing, and to access any data or proprietary processing logic accessible by an MQSeries application on a remote system. The MQSeries API, called the Message Queueing Interface (MQI), gives Notes access to any logic or data available on the target system, including RDBMS, and non-relational file stores like VSAM and sequential files.

MQLSX also enables Lotus Notes clients and servers to work with multiple systems simultaneously.

MQLSX provides access to store-and-forward message queues so that Notes developers can use queues in Notes applications. Notes application integration with MQSeries eliminates the necessity for developers to code network-specific application calls. MQSeries shields developers from network complexities.

Technical advantages

The MQLSX functions as a special link from Notes to transactions and other systems. It has the following advantages over other middleware products:

- Integration with the Notes application development environment: All the definition, design and testing take place in the Notes development environment. MQLSX is a LotusScript Extension (LSX). LotusScript extensions expose their functionality and classes to LotusScript in exactly the same way as Notes itself does. Notes developers therefore have seamless access to Notes and MQSeries functionality.
- Application location transparency for developers: MQLSX shields the Notes application developer from the multi-vendor, multi-protocol complexity of today's business networks and provides application-location transparency. MQLSX provides a programming interface for computers and networks from multiple vendors and offers a simple, reliable means of building distributed and client/server applications.
- Integration into Notes application interface: Notes applications can transparently integrate transaction system data, and LotusScript allows the data returned from host transactions to be posted directly to the Notes user interface.
- Time-independent (asynchronous) processing: MQLSX allows time independent (asynchronous) processing, which means that when a message is created to initiate a transaction, that message might not be delivered

immediately (if, for example, the system the transaction runs on is not available at the time the message is created). MQSeries will ensure that the message is kept until the transaction can process it. Using MQSeries and the agent capabilities of the Notes server, it is possible to develop sophisticated applications, such as future point-in-time workflow applications for scheduling production runs for customer orders received in Notes.

- Communication through queues: All communication using the MQLSX occurs through queues only. MQLSX couples queued, store-and-forward messaging with Notes' powerful integrated client/server messaging.
- Data integrity protection: MQLSX allows access to enterprise processes, whereas the other middleware products allow access to enterprise data only. Using MQSeries, the only way to access data is through a transaction, never directly. MQLSX programs protect data by not allowing access to raw data directly from external applications that could put data integrity at risk.
- Automatic data conversion of both character and numeric data.

6.5.0.3 Comparison between MQLSX and MQEI

It is important to understand the differences in functionality between the two MQSeries products (MQLSX and MQEI). The distinctions are in the programmability of the interface and level of native access to host data.

MQEI differs from MQLSX by providing a simplified interface for the application programmer, and by enabling connectivity to CICS without the need for MQSeries. MQLSX, on the other hand, provides the full flexibility of the MQI, while the MQEI provides only the most commonly used features of the MQI. The differences are in the programmability of the interface and level of native access to host data.

The following table is a quick overview of the basic differences in MQLSX and MQEI.

	MQLSX	MQEI
Application	-Native MQSeries application -IMS through the MQ-IMS bridge -CICS DPL through the MQ-CICS/ESA DPL bridge -SAP R/3 through MQ	-Native MQSeries application -IMS through the MQ-IMS bridge -CICS DPL through the MQ-CICS/ESA DPL bridge -CICS DPL direct -CICS 3270 direct
Programming on Lotus Notes	Full LotusScript environment with all the functions for the MQSeries MQI.	Full LotusScript environment with most commonly used functions of the MQSeries MQI, plus most functions of the CICS ECI and EPI.
Security	To be implemented in the LotusScript program.	Implemented through the security database, or in the LotusScript program.
Messages	To be implemented in the LotusScript program. Field offsets must be known in the LotusScript program.	Implemented through the Definition database. No need for the LotusScript program to know the offsets.

	MQLSX	MQEI
Abstraction for the enterprise system	You need to understand the enterprise system to which you connect	Provides a higher level of abstraction, and easier to program to.

6.6 LDAP and Lotus Domino R5.0

Lotus Domino supports a variety of business applications including an LDAP-based directory service. The Domino Directory is designed to serve as a key enabling technology for a directory-enabled infrastructure. To leverage the inherent value of this potentially rich and useful store of corporate information beyond e-mail addresses and certificates, Domino/Notes customers can exploit this directory architecture, which they already have in place.

Today's Domino Directory can be part of a general purpose directory infrastructure for the enterprise and for multi-enterprise extranets. On the other hand, in a heterogeneous networking environment with other directory systems, the Domino Directory can also serve as the integration point for directory synchronization, administration, and authentication.

Directory features in Domino R5 include:

- Support for X.500 naming conventions, including hierarchical naming and extensible attributes, for maximum flexibility in configuring the namespace.
- LDAP protocol support in both the client and the server providing lookup (read), add, delete, and modify (write) support for non-Notes clients (for example Web browsers) and servers (for example NDS and Four11).
- Rule-based domain relationships for faster lookups across large namespaces.
- Hierarchical naming and trust between domains to support the relationship of entries across domains.
- Support for a public key infrastructure.
- A dynamically extensible directory schema ideal for customizing the directory to meet specific business requirements.
- Multi-master replication, a key element for reliable directory synchronization and maximum availability.
- An open architecture that can easily incorporate support for emerging standards.

Directory service features

In addition to the Domino Directory itself, Domino provides three directory service features: the directory catalog, directory assistance, and the LDAP service. These features help users find user names, e-mail addresses, and other information in the Domino Directory.

- The directory catalog consolidates key information about users and groups from one or more Domino directories into a small, lightweight database. Notes users who use a local copy of the directory catalog - a mobile directory catalog - can quickly address mail to users throughout the organization, even if the organization uses a large directory and/or multiple directories. In organizations with multiple Domino directories, a directory catalog on a server

combines these directories into a single database so that a server can look up names in one database rather than in multiple Domino directories.

- Directory assistance is a feature that helps manage name lookups in organizations that use multiple Domino directories and/or third-party LDAP directories. A directory assistance database associates each Domino directory/ LDAP directory with specific hierarchical names. So when looking up a hierarchical name, Domino first searches the directory that contains names in that hierarchy.
- You can set up a Domino server to run the Lightweight Directory Access Protocol (LDAP) service to enable LDAP clients, like Web browsers and Notes clients, to search for and modify information in the Domino Directory. The Domino LDAP service is LDAP V2 and LDAP V3 compliant.

For more information on LDAP see:

- *LDAP Implementation Cookbook*, SG24-5110
- *Understanding IBM SecureWay FirstSecure*, SG24-5498
- *Lotus Notes and Domino R5.0 Security Infrastructure Revealed*, SG24-5341
- *IBM SecureWay Host On-Demand 4.0: Enterprise Communications in the Era of Network Computing*, SG24-2149-01, available at <http://www.redbooks.ibm.com>

6.7 Where to find more information

- *Connecting Domino to the Enterprise Using Java* - SG24-5425
- *Lotus Domino Enterprise Integration: Architecture and Products* - SG24-5593
- *Lotus Solutions for the Enterprise, Volume 2 , Using DB2 in a Domino Environment* , SG24-4918
- *Lotus Solutions for the Enterprise, Volume 5 NotesPump: The Enterprise Data Mover*, SG24-5255
- *Lotus Solutions for the Enterprise, Volume 4 Lotus Notes and the MQSeries Enterprise Integrator*, SG24-2217
- *Lotus Solutions for The Enterprise, Volume 1 Lotus Notes: An Enterprise Application Platform*, SG24-4837
- *Lotus Solutions for the Enterprise, Volume 3 Using the IBM CICS Gateway for Lotus Notes* , SG24-4512
- <http://www.edge.lotus.com>
- <http://www.lotus.com/enterpriseintegration>
- <http://www.lotus.com/home.nsf/welcome/ei>
- <http://www.s390.ibm.com/products/domino/>
- <http://www.software.ibm.com/ts/mqseries/>

Chapter 7. WebSphere

IBM WebSphere is a family of cross-platform products, that allows customers to develop, execute, connect to legacy, and manage the whole Web environment. WebSphere works closely with other host integration solutions we have already discussed. WebSphere is an integral part of the Host Publisher solution. WebSphere also provides the Host On-Demand Connector in the VisualAge for Java and Component Broker components.

WebSphere is highly scalable, horizontally and vertically. You can start on a Windows NT server and scale up to an S/390 server, including multiple servers, without changing the code. IBM WebSphere provides a load balancing tool among multiple servers.

WebSphere provides a robust and portable application deployment environment as well as application execution management. It is supported on many popular operating system platforms, including Windows NT, IBM AIX/6000, Linux, Novell NetWare, Sun Solaris, OS/390, OS/400, and OS/2 Warp.

IBM WebSphere products follow the open standards such as HTTP, HTML, XML, SSL, CORBA, LDAP and are strongly based on the Enterprise Java APIs.

The WebSphere strategy is defined by four principles:

- Build on what you have while integrating your business system with the Web.
- Leverage existing skills and existing information technology (IT) systems.
- Utilize open standards for portability and productivity.
- Deploy secure, scalable, and manageable applications.

7.1 Java terms

Considering that WebSphere Application Server is based on Java concepts and terminology, we decided to include a brief description of the terms used in this chapter in order to make sure they are commonly understood and for completeness of this technical material. We do not intend to go in depth in these explanations. Java is much more complex than a few words, but it could help for a basic understanding.

Keep in mind, Java always runs in a Java Virtual Machine (JVM), no matter if it is on the server side or on the client side.

Servlets

Servlets are Java programs that use the Java Servlet Application Programming Interface (API) and the associated classes and methods. They run on a Java-enabled Web server and extend the server's capabilities. In addition to the Java Servlet API, servlets can use Java class packages that extend and add to the API. Servlets extend server capabilities by creating a framework for providing request and response services over the Web.

When a client request arrives at the Web server, the Web server creates a Request object and a Response object that are specific to the request. The server invokes the `servlet service()` method, passing the Request and Response objects. The `service()` method gets information about the request from the Request object, processes the request, and uses methods of the Response

object to pass the response back to the client. Servlets perform a wide range of functions. For example, a servlet can:

- Create and return an entire HTML Web page containing dynamic content based on the nature of the client request.
- Communicate with other server resources, including databases and Java-based applications.
- Handle connections with multiple clients, accepting input from and broadcasting results to multiple clients.

Servlets communicate in the same fashion as other Java programs, through method calls. Servlets can interact with C and C++ programs via the JNI (Java Native Interface).

Applets

An applet is a program written in the Java language that can be included in an HTML page. When you use a Java-enabled browser to view a page that contains an applet, the applet's code is transferred to your system and executed by the browser.

Java Server Pages

Java Server Pages (JSP) is an easy-to-use solution for generating HTML pages with dynamic content. A JSP file contains combinations of HTML tags, NCSA tags (special tags that were the first method of implementing server-side includes), <SERVLET> tags, and JSP syntax.

One of the many advantages of JSP is that it enables you to effectively separate the HTML coding from the business logic in your Web pages. Use JSP to access reusable components, such as servlets, JavaBeans, and Java-based Web applications. JSP also supports embedding in-line Java code within Web pages.

Beans and JavaBeans

Sun's JavaBeans architecture describes Java components designed to be used on client systems. Programs written to the JavaBeans architecture are called beans. Beans are Java classes that conform to certain coding standards. They can be described in terms of their properties, methods and events. Beans may be packaged with a special descriptor class called a Beaninfo class and special property editor classes in a JAR file.

Enterprise JavaBeans

Enterprise JavaBeans (EJBs) is Sun's trademarked term for their EJB architecture (or "component model"). When writing to the EJB architecture, developers are creating enterprise beans. Enterprise beans are server-side Java components designed for distributed environments. They do not exist in isolation but are deployed in containers that provide services such as security, naming and directory services and persistent storage. WebSphere Application Server is just such a container.

RMI

Remote Method Invocation (RMI) is a distributed object protocol that allows Java objects to call each other across a network. It is part of the core Java specifications.

IIOp

Internet Inter ORB protocol (IIOp) is an Internet protocol used for CORBA object communication.

7.2 IBM WebSphere components

The IBM WebSphere family includes the following:

- WebSphere Application Server
- WebSphere Studio
- WebSphere Performance Pack
- WebSphere Site Analysis
- Individual backend connectors

Figure 75 shows the WebSphere components.

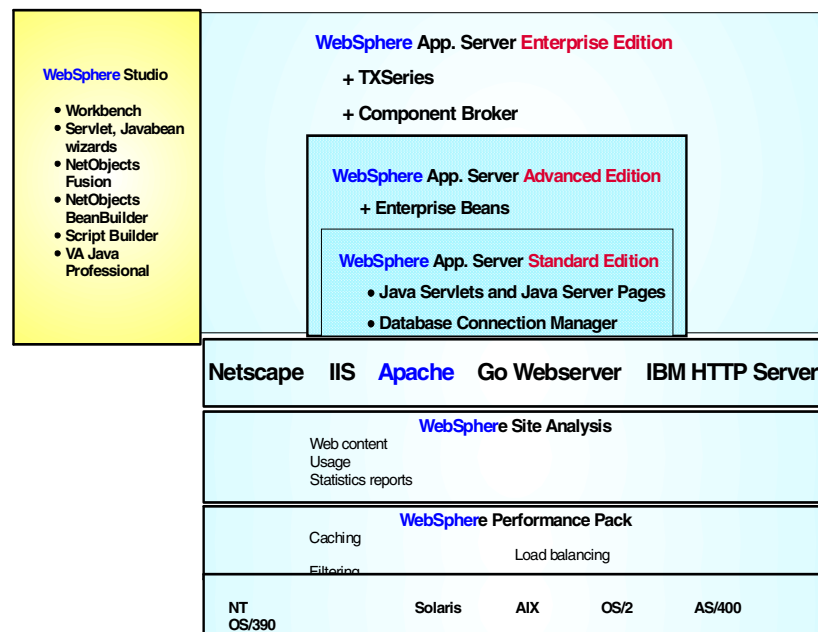


Figure 75. WebSphere Application Server building blocks

7.2.1 WebSphere Application Server

WebSphere Application Server is the IBM Web-deployment environment and the core architecture for e-business. It is a Java-based environment for running Java applications on the Web. Those applications are typically composed of either enterprise beans, Java Server Pages (JSPs) and Java servlets.

Servlets are secure, portable, high-performance alternatives to CGI (Common Gateway Interface). They are multi-threaded, maintain sessions across HTTP requests and can be preloaded, remaining in memory as needed. They run within the servlet engine process. No servlet can run with access rights beyond those of that process.

Clients running Java applets can communicate with WebSphere either through the HTTP protocol used by the Web browser, using the Web server as an

intermediary, or can use IIOP to communicate directly to Java servlets and enterprise beans running within WebSphere Application Server.

WebSphere Application Server is available in three different editions:

WebSphere Application Server Standard Edition Version 3.0

The standard application server combines the portability of server-side business applications with the performance and manageability of Java technologies to offer a comprehensive platform for designing Java-based Web applications. It enables powerful interactions with enterprise databases and transaction systems.

The standard application server is used to build active Web sites and Web applications. It is a single system Web application server, typically supporting a Web site. To create applications, developers implement extended HTML content using the WebSphere Studio, build and test the Web site, then publish updates to the active site.

The standard edition provides:

- Support for static and active HTML
- Support for the Java servlet 2.1 specification
- Support for JSPs
- Database support using JDBC
- Enterprise beans (development tools are not provided in the standard edition)
- XML server tools
- A Web site analysis tool
- IBM HTTP Server
- IBM VisualAge for Java
- Tivoli ready modules
- LDAP client for security management

WebSphere Application Server Advanced Edition Version 3.0

The advanced application server builds on the standard application server. It introduces capabilities for applications to be built to the EJB specification and provides support for integrating Web applications with existing business systems.

The advanced application server includes the features provided with the standard application server plus:

- Full support for the Enterprise JavaBeans specification
- Deployment support for EJBs, Java servlets, and JSPs with performance and scale improvements
- Enhanced support for distributed transactions and transaction processing
- Improved management and security controls
- CORBA support
- SecureWay Directory Server as an optional directory supporting LDAP

WebSphere Application Server Enterprise Edition Version 3.0

The WebSphere Application Server Enterprise Edition offers a solution for creating enterprise-wide distributed applications. The enterprise edition includes the WebSphere Application Server Advanced Edition. Plus, it combines TXSeries, IBM's world-class transactional application environment, with the full distributed object and business process integration capabilities of Component Broker.

The enterprise edition includes the following products for the AIX, Windows NT and Solaris platforms (unless otherwise noted):

- Component Broker run time V3.0
- TXSeries V4.3
- WebSphere Application Server V3.0 Advanced Edition
- DB2 Universal Database V5.2
- MQSeries V5.1
- DCE V2
- CICS Transaction Gateway V3.0.2
- CICS Universal Clients V3.0.2
- Component Broker tools (AIX and Windows NT)
- DB2 Software Developers Kit V5.2
- C++ Compiler and Libraries (AIX and Windows NT)

TXSeries consists of three popular middleware packages that are used to create distributed transactional applications:

- TXSeries CICS for AIX, Solaris, and Windows NT
- The Encina family of products used to develop and manage open distributed systems
- DCE-Encina Lightweight Client (DE-Light)

7.2.1.1 Supported Web servers

WebSphere Application Server is not a Web server or an HTTP server.

WebSphere is a Web application server and works in close cooperation with Web servers to handle requests from Web clients.

WebSphere Application Server is a Java engine (servlet engine) that runs on top of the Web server as a server plug-in to the HTTP server. In order to have the Java environment ready to execute servlets, the HTTP server should be first up and running.

WebSphere Application Server can work with the following Web servers:

- Lotus Domino Go Webserver Version 4.6.1 and beyond for OS/390 V2.5, Windows NT, Sun Solaris, and AIX
- IBM HTTP Server (supplied with WebSphere Application Server)
- Microsoft Internet Information Server 2.x, 3.x, or 4.0
- Netscape Enterprise Version 2.01 or higher for Windows NT, Sun Solaris, and AIX
- Netscape FastTrack Server Version 2.01 or higher for Windows NT, Sun Solaris, and AIX
- Apache Version 1.2.x for Sun Solaris and AIX

7.2.1.2 IBM HTTP Server

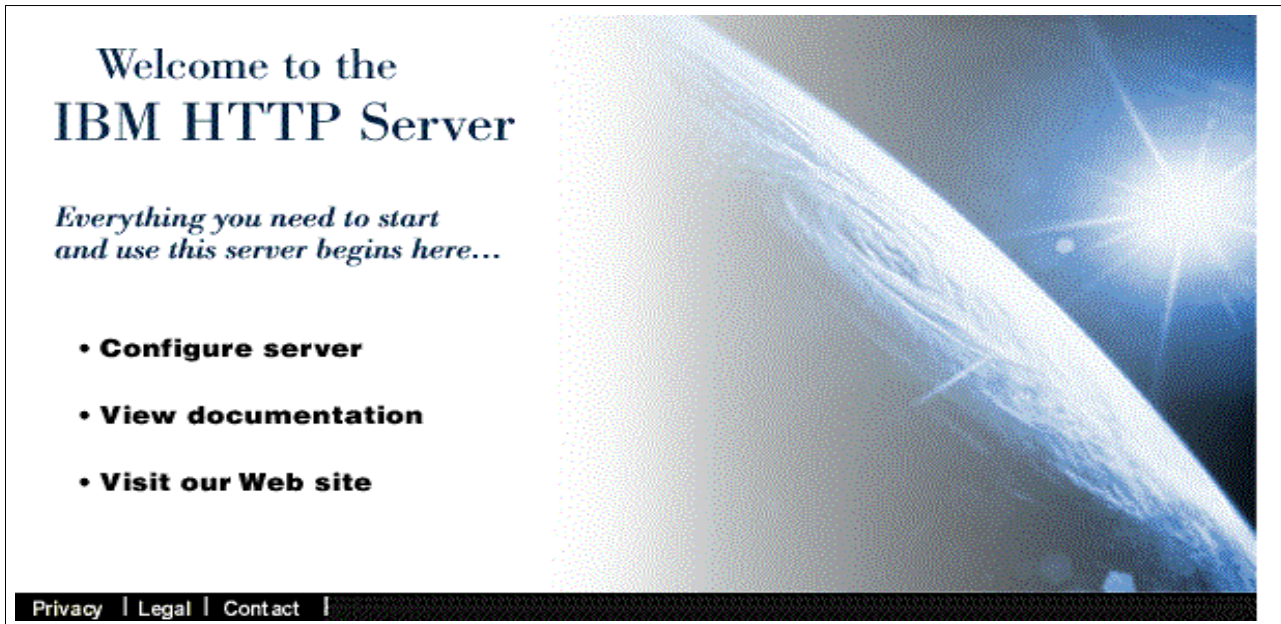


Figure 76. IBM HTTP Server welcome page

WebSphere Application Server includes a copy of IBM HTTP Server that can be installed automatically as part of the WebSphere installation process for customers who do not already have a Web server.

Note for OS/390

On the S/390 platform, WebSphere Application Server is the follow-on of ServletExpress available in previous releases, an entry-level support for servlets. Customers running servlets under ServletExpress can migrate to WebSphere Application Server with minor customization changes.

7.2.1.3 CORBA support

WebSphere Application Server CORBA support consists of an Object Request Broker (ORB) and an extensive feature set that enables the development of Web-based Java applications employing distributed CORBA objects and object servers embedded within servlets. WebSphere Application Server CORBA support provides traditional ORB services and features that directly support the IBM Network Computing Framework (NCF). It is a robust development tool and run time environment that can distribute Java Web-centric applications over the Object Management Group's open, standard Internet Inter-ORB Protocol (IIOP).

Servlets can communicate with each other across different WebSphere instances through Java Remote Method Invocation (RMI). WebSphere also supports servlet communication across WebSphere instances via RMI over CORBA IIOP. RMI over IIOP is used for object-to-object communication where those objects are CORBA-compliant.

7.2.1.4 IBM WebSphere Application Server Manager

The WebSphere Application Server Manager provides a Java-based console for servlet management. It is a Java applet that allows you to configure and manage the WebSphere Application Server environment.

The WebSphere Application Server Manager allows you to:

- Configure setup parameters for session tracking and connection management.
- Monitor servlet activity, for example, view the list of the currently loaded servlets and the resources used by them.
- Establish and maintain security by defining users, groups, resources and access control lists.
- Manage servlets by loading or unloading them and by defining initialization parameters.

Once installed and properly configured, the Application Server Manager listens, by default, on port 9090. Figure 77 gives you an idea of what the servlet management environment is all about.

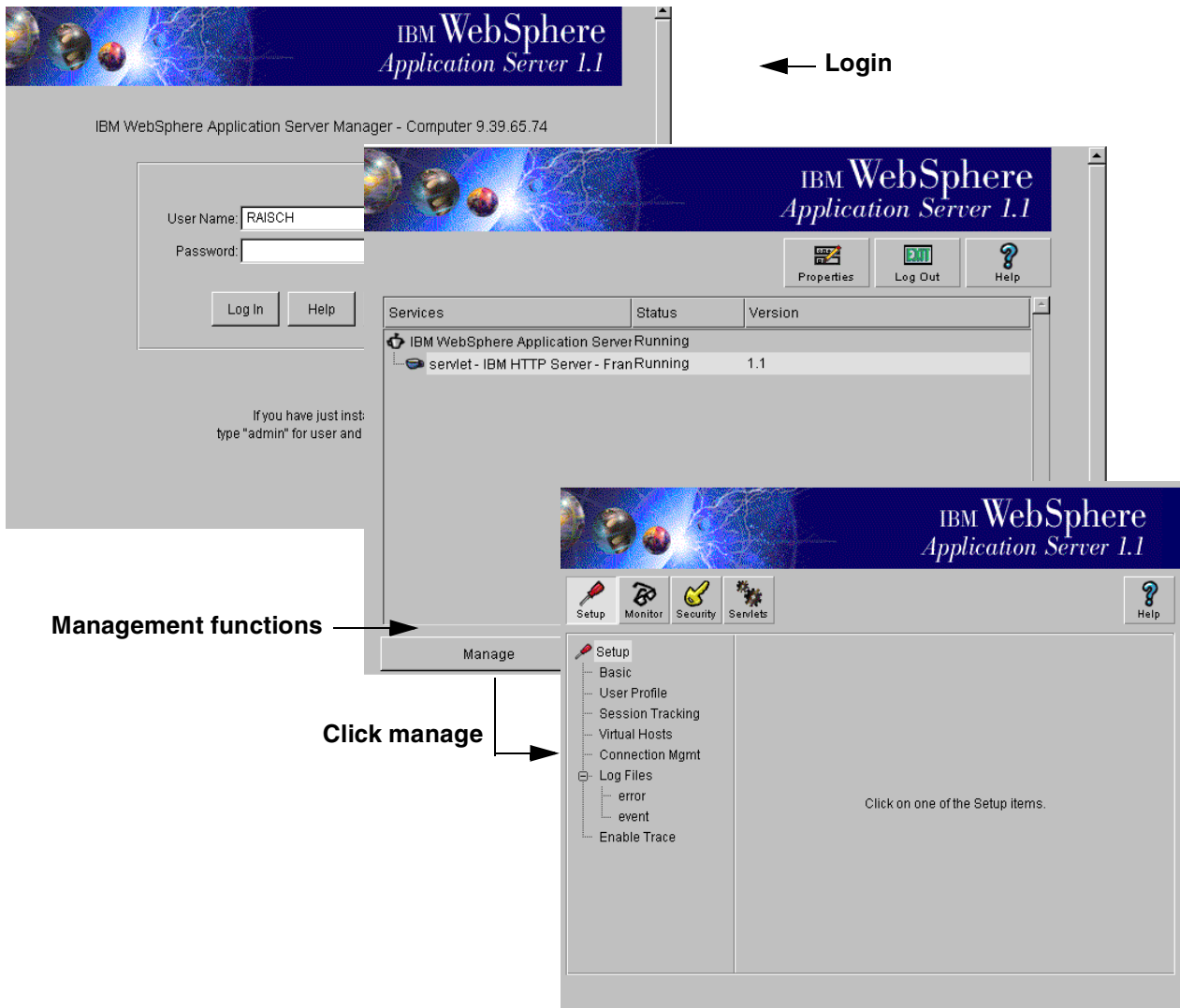


Figure 77. Application Server Manager

Session tracking

Session tracking is one of the most important WebSphere added values with respect to a simple servlet Java run time environment. The HTTP protocol is stateless in nature. The client issues a servlet request, the servlet provides a response, the connection is closed and no relation can be set with an eventual following request. Because of this stateless nature, it is difficult to implement applications in which a complex conversation must be held between the client and the server. The server cannot distinguish, between subsequent requests, the ones originating from the same client.

WebSphere provides the `com.ibm.servlet.personalization.sessiontracking` package to manage sessions. Servlet `service()` executions requested by the same user can access the associated session object by calling the `HttpSession` class `getSession()` method of the package. The session object stores data that can be accessed by subsequent requests of the client.

Connection management

The connection manager lets you control and reduce the resources used by your Web-based applications. The connection manager spreads the connection overhead across several user requests by establishing a pool of connections, which servlets can use. Put another way, each user request incurs the overhead of only a fraction of the cost of a connect/disconnect. After the initial resources are spent to establish the connections in the pool, additional connect/disconnect overhead is insignificant because the existing connections are reused repeatedly. The servlets use the connection pool as follows: When a user makes a request over the Web to a servlet, the servlet uses an existing connection from the pool, meaning the user request does not incur the overhead of a data server connect. When the request is satisfied, the servlet returns the connection to the connection manager pool for use by other servlets. The user request therefore does not incur the overhead of a data server disconnect.

7.2.2 WebSphere Studio

The WebSphere Studio is a suite of tools that can be used by all the people involved in creating and maintaining Web sites. It allows your team to:

- Group your Web site files into projects and folders.
- Edit and update the files with your preferred tools.
- Easily create Java servlets and database queries using wizards.
- Publish all or part of the Web site on any of your WebSphere application servers.
- Maintain the files locally on individual workstations, or in a central location using a source control system.

Content authors, graphic artists, programmers, and Web masters can all work on the same projects, each having access to the files they need.

WebSphere Studio includes NetObjects Builder, NetObjects Fusion, VisualAge for Java, and some Wizard tools. This is the Web workbench for creating the site, Web pages, scripts and Java application editing and management of the project parts.

The VisualAge family is the visual development application environment and contains an editing facility, various compilers, run time libraries, debugging tools and connectors to databases and legacy applications. It can be integrated to the WebSphere Studio but is shipped separately from WebSphere.

7.2.3 WebSphere Performance Pack

IBM WebSphere Performance Pack is a set of products oriented to reduce Web server congestion, increase content availability and improve Web server performance.

It provides three basic functions:

- Cache and proxy services
- Load balancing
- Transarc's Andrew File System (AFS)

For more details on IBM WebSphere performance Pack, refer to *IBM WebSphere Performance Pack Usage and Administration*, SG24-5233.

7.2.3.1 Cache and proxy services

WebSphere Performance Pack includes Web Traffic Express for caching and proxy services. Web Traffic Express is supported by AIX, OS/2 Warp, Sun Solaris, or Windows NT. Web Traffic Express is integrated on IBM HTTP Server for OS/390.

With Web Traffic Express you can:

- Reduce the time it takes to retrieve information from the Internet.
- Prevent certain types of Web-based information from being forwarded to your clients using Platform for Internet Content Selection (PICS) filters.

For the S/390 platform, the cache function is implemented by the TCP/IP stack, or more explicitly, by the Fast Response Cache Accelerator (FRCA). The proxy function is implemented by the Web server itself, through proper configuration.

7.2.3.2 Load balancing

Load balancing is implemented by IBM Secureway Network Dispatcher, for AIX, Sun Solaris, or Windows NT, as well as on 2210 and 2216 routers. This scalable software solution load balances requests among your HTTP, FTP, or other TCP-based servers, offering customers improved site responsiveness, while you enjoy easy to use site management and productivity features.

7.2.3.3 Transarc AFS

Transarc's Andrew File System (AFS) is an enterprise file system that allows authorized users to share files in a Web environment, providing automatic, non-disruptive real-time replication of information across multiple servers, which guarantees data consistency and availability. It is supported on AIX, HP-UX, Digital UNIX, IRIX, Sun Solaris, Windows NT and Red Hat Linux.

For more information, see the Transarc Web site at

<http://www.transarc.com/Product/EFS/>.

7.2.4 WebSphere Site Analysis

Beginning with WebSphere Application Server V3.0, a new component, called site analysis, has been included. Site analysis technology is a tool for analyzing Web sites. It keeps track of what is occurring on the Web site, the users' behavior, the quality of services, and Web activity. It provides a detailed picture of the Web site by holding the data in a DB2 UDB database and providing report templates to manipulate and publish it.

Site analysis provides:

- Web content analysis
- Web usage analysis for monitoring resource usage
- Statistics using user-defined reports
- Usability

The site analysis component can be installed in a stand alone or client/server configuration. The server contains the analyzers whose responsibility it is to transform the raw data into valuable information and store it into your specified database. The client interface provides administrative, visualizing and report-generating functions. From the client, you can schedule analysis tasks to run at a specific time or time interval.

Regardless of the setup, reports may be made available by publishing them to a Web server using the provided publishing tool.

The site analysis component provides server support on IBM AIX, Sun Solaris and Windows NT. Client support is provided on all Windows 32 platforms. In addition, it supports the use of HTTP log files from IBM S/390 and IBM AS/400 servers.

WebSphere Site Analyzer for Windows NT and Solaris, requires JDK 1.1.7b. For AIX it requires JDK 1.1.6 with fix pack 8. Multiple sites and multiple servers are supported.

7.3 Connecting WebSphere Application Server to other subsystems

The WebSphere Application Server supports inclusion of a variety of different connectors, gateways, and application adapters to access backend servers and systems. Connectivity can be distinguished in two ways, by gateway or adapter.

Adapter connectivity establishes direct mapping to data and information contained within backend systems and servers, and encompasses the transactional nature of connections across dissimilar and remote systems. Enterprise JavaBeans components are examples of adapter connectivity. Adapter connectivity, due to its transactional nature, can dynamically accommodate changes within the backend systems. Greater interoperability exists when those systems, interactions and transactions are invoked via the WebSphere Application Server.

Gateway connectivity refers to Java-based applications that convert HTML requests into sets of parameters. These parameters are then passed to an associated client that executes in the same process as the gateway and communicates with the backend application server via native or Java-based APIs. Gateway connectivity is certainly not as scalable in nature as EJB adapters but provides a simple-to-use and easy-to-develop alternative which can also be called from servlets.

Both types of connectivity are used to help extend the enterprise systems to the Web and to the Web application server environments. Most of the time, new Web applications will require integration with existing applications and data. WebSphere Application Server provides connections to major database and transaction manager subsystems.

7.3.1 Host On-Demand Connector

The Host On-Demand Connector provides reusable Java beans and ActiveX controls for direct manipulation of 3270-based CICS, IMS, or other OS/390 applications. More information on the Host On-Demand Connector can be found in 5.4, "Host On-Demand Connector" on page 90.

7.3.2 Connecting to DB2

For relational database access, WebSphere Application Server supports the standard JDBC drivers for database access from servlets. WebSphere Application Server allows updates to multiple database managers through multiple connections. In addition, WebSphere Application Server supports

extensions to JDBC to make database programming easier and improve database query performance via data access beans and the connection manager.

Data access beans provide a higher level API for database access through JDBC. Using the IBM WebSphere Studio database access-servlet wizard or the VisualAge for Java Select Beans, a developer can visually create data access beans and use those beans to read and update relational databases such as DB2 Universal Database.

The connection manager feature allows many servlets to share "pools" of long-lived JDBC database connections, improving scalability and performance by eliminating the overhead required to open a new connection for each request. These database connection pools are defined in the WebSphere Application Server deployment manager.

In order to perform database access, the application requires the Java interface to the database provided by WebSphere Application Server. The Java application acts as a client to the DB2 server. If the Web server is running on a different machine than the database server, you need a DB2 client on the Web server machine in order to connect to the database server.

To set up this environment, the minimum required levels are DB2 V5 and JDK 1.1.6.

DB2 provides JDBC drivers and SQLJ support in its products for OS/390, OS/2, Windows NT, AIX, Solaris, HP/UX, SCO-UNIX, and Linux.

7.3.2.1 JDBC

JDBC is a set of APIs that are provided by the database vendor, usually called the JDBC driver for that vendor. JDBC enables Java applications to create sessions to the database, locally or remotely, execute SQL statements and retrieve the results.

JDBC supports dynamic SQL statements. In a dynamic SQL model, the application may change any statement during execution, and precompilation is executed during run time. The JDBC driver reads input data from an HTML form to build an SQL query, makes a JDBC connection to a DB2 database, processes the query result set and returns it to the Web page. Syntax checking and compiling is done during run time.

In the case of DB2, the database server should support DRDA with DDF and Call Level Interface (CLI) in order to be able to talk to the JDBC driver.

For more information on JDBC access to DB2 see Chapter 11, "Solutions for accessing DB2 from the Web" on page 195.

7.3.2.2 SQLJ

SQLJ is the embedded SQL statements in Java applications, similar to the embedded SQL statements in regular programming languages, which are precompiled and bound into databases.

SQLJ has two components: the translator and the run time components. Both are pure Java code.

SQLJ statements are static and based on ANSI standards. Static SQL statements require precompilation before run time and must be bound to the database, which means having database privilege and authorization control.

For more information, see *WebSphere Application Servers: Standard and Advanced Editions*, SG24-5460.

7.3.3 Connecting to CICS

Since WebSphere Application Server uses Java as its underlying language, the CICS Transaction Gateway and its prerequisite External CICS Interface (EXCI) are the only connectivity tools required to access CICS applications using WebSphere.

CICS Transaction Gateway is shipped by IBM with the TXSeries family of products (CICS and Encina). CICS Transaction Gateway is a set of Java applications and class libraries that enable Java programs to access CICS servers. For more information on CICS Transaction Gateway, see Chapter 9, “Solutions for accessing CICS from the Web” on page 135.

More information can be found in:

- *WebSphere Application Servers: Standard and Advanced Editions*, SG24-5460
- *OS/390 e-business Infrastructure: IBM WebSphere Application Server 1.1 for OS/390*, SG24-5604

And on the Web at

- <http://www.ibm.com/software/ts/cics/platforms/internet/cicsgw4j/announce/jgarn201.html>.

7.3.4 Connecting to Lotus Domino

Using the Java Notes interface that comes with Domino, you can connect to Domino applications from a servlet or a bean. For more information see Chapter 6, “Lotus Domino” on page 95.

7.3.5 Connecting to MQSeries

MQSeries provides Java support by shipping the MQSeries for Java component. It consists of a set of Java classes and native class implementation libraries. This MQSeries Java support allows a Java application to exchange messages with a backend system using MQI technology as the middleware.

The minimum software level required is MQSeries Server 5.0, MQSeries Client 2.0 and JDK 1.1.

7.3.5.1 MQSeries Internet Gateway

MQSeries Internet Gateway provides a bridge between the synchronous World Wide Web and asynchronous MQSeries applications. Running a Web server with MQSeries Internet Gateway and MQSeries server provides an Internet-connected Web browser with access to MQSeries applications.

The gateway can also be used to Web-enable enterprise applications for use in an intranet. The gateway takes a request originating from a Web browser and converts it into an MQSeries message. The message will be delivered by

MQSeries to a serving application running on any one of over 20 platforms that are supported by MQSeries. The browser can wait for a response to be returned from the application via MQSeries and the gateway, or alternatively an acknowledgment can be sent to the browser by the gateway as soon as it has received the request, confirming to the user that work is in progress and freeing that browser session to perform other tasks.

This product is shipped as part of the MQSeries V5 product.

7.3.6 Component Broker

Component Broker is IBM's implementation of the Object Management Group (OMG) Common Object Request Broker Architecture (CORBA) standards.

Component Broker allows you to develop distributed applications, according to the three-tier model, in which client machines run Java, ActiveX or C++ objects communicating via an Object Request Broker (ORB) with business objects residing on the middle-tier application server.

IBM's Component Broker Connector (CBCConnector) combines a run time, systems management, and an integrated development environment in one infrastructure. It provides TCP/IP, SNA, and LAN networking protocols and application adaptors for accessing CICS, IMS, Oracle, MQ, and DB2.

The Relational Database Application Adaptor for DB2 provides support for embedded SQL on DB2. Support for DB2 CLI and ODBC is targeted for future releases. The Procedural Application Adaptor provides support for IMS and CICS.

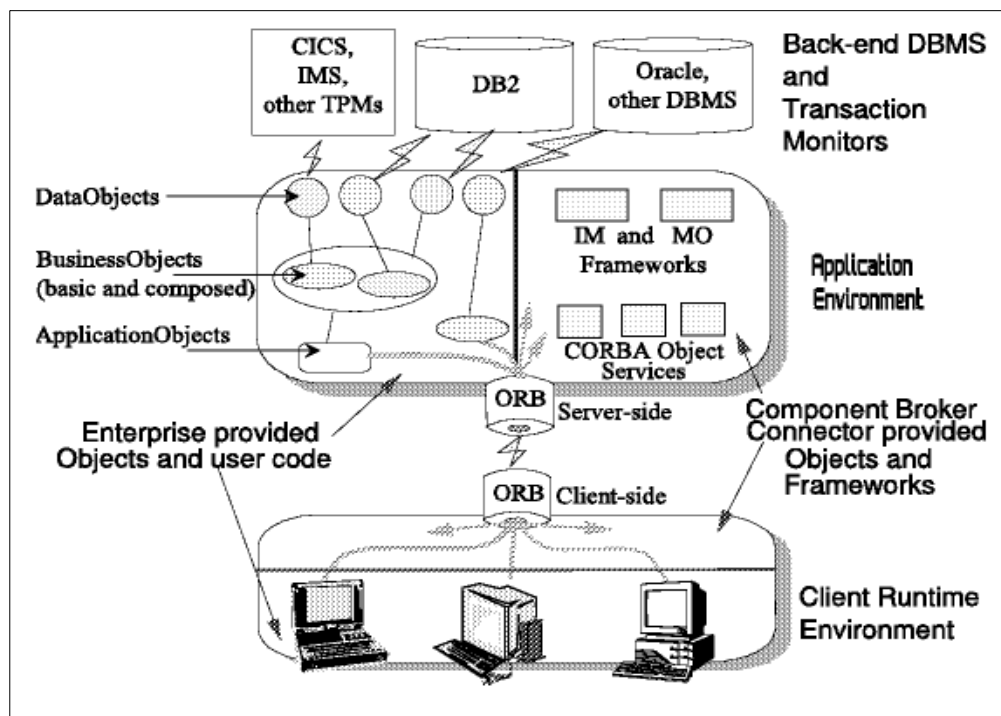


Figure 78. CBCConnector run time architecture building blocks

The ORB allows objects written in different languages and running on separate systems to communicate transparently with one another.

Business objects encapsulate the knowledge about real-world artifacts, such as customers, accounts, bank transactions, ledgers, and so forth. They deal with the business logic, rules, and constraints as defined by a specific domain. They do not deal directly with user-interface aspects of a particular usage scenario or application, nor are they concerned with data storage technology.

Persistence of business objects is provided by way of state objects on host database management systems or transaction systems.

For Internet access, Java code can be downloaded as an applet from a Web server into a browser. The downloaded Java code includes a proxy that stands in for an object managed by the CBCConnector server. The proxy provides a connection to the Java ORB. The Java ORB uses Internet Inter-ORB protocol (IIOP) to communicate with the CBCConnector server ORB. The server uses an application adaptor to access CICS, IMS and DB2.

7.4 LDAP and WebSphere

WebSphere supports authentication mechanisms based on validating credentials such as user ID and password, certificates or tokens. Credentials are verified against a user registry supporting such a schema. For example, user IDs and password-based authentication can be based on the LDAP user registry where authentication is performed using an LDAP bind. A certificate validation list may be used in cases where authentication of the user is based on the client certificate presented by the user over a mutual SSL connection.

WebSphere supports a three-party authentication schema, one in which the client principal and server principal are authenticated to a mutually trusted third-party. The third-party in this case is the authentication token server. An advantage of the three-party schema is that administration of the user registry can be controlled centrally.

WebSphere Application Server provides the capability to:

- Dynamically configure WebSphere to use a directory service.
- View directory service user and group information in the WebSphere Application Server Administration.
- Protect WebSphere resources with access control lists containing directory service user and group information.
- Use directory service authentication of user and group information.

WebSphere Application Server V3.0 supports the following list of LDAP directory servers:

- Netscape Directory Server Version 3.x and 4.x
- SecureWay Directory Server Version 2.1 and 3.1.1
- Lotus Domino Version 4.6 and 5.0

Additional attributes are available for customizing any of the default filters to fit a directory server not listed above.

For more information on LDAP see:

- *LDAP Implementation Cookbook*, SG24-5110
- *Understanding IBM SecureWay FirstSecure*, SG24-5498
- *Lotus Notes and Domino R5.0 Security Infrastructure Revealed*, SG24-5341
- *IBM SecureWay Host On-Demand 4.0: Enterprise Communications in the Era of Network Computing*, SG24-2149-01, available at <http://www.redbooks.ibm.com>

7.5 Where to find more information

For more information on WebSphere and related products, see the following:

- *WebSphere Application Servers: Standard and Advanced Editions*, SG24-5460
- *OS/390 e-business Infrastructure: IBM WebSphere Application Server 1.1 for OS/390*, SG24-5604
- *OS/390 e-business Infrastructure: IBM HTTP Server V5.1 for OS/390*, SG24-5603
- *E-Business Application Solutions on OS/390 Using Java: Volume 1*, SG24-5342
- *WebSphere Application Server - GC34-4757*
- *IBM Component Broker on System/390*, SG24-5127
- *IBM Component Broker Connector Overview*, SG24-2022
- *IBM WebSphere Performance Pack Usage and Administration*, SG24-5233

Web sites

- <http://www.ibm.com/software/ts/cics>
- <http://www.ibm.com/software/data>
- <http://www.ibm.com/s390>
- <http://www.ibm.com/software/websphere>
- <http://www.ibm.com/software/ebusiness>
- <http://java.sun.com/products>
- <http://www.ibm.com/software/developer/library>
- <http://www.transarc.com/Product/EFS/>
- <http://www.ibm.com/software/ts/cics/platforms/internet/cicsgw4j/announce/jgann201.html>
- White paper: IBM Component Broker Technical Overview (<http://www.software.ibm.com/ad/cb/litp.html>)

Chapter 8. IBM SecureWay FirstSecure

Security has always been an integral part of the planning and implementation of computing resources in every enterprise. Traditionally, companies were not accessible through the Internet and their network was limited to in-house connectivity. Security was certainly a concern but was not quite the challenge it is today. Today, with the trend toward Web-enabling applications for not only the intranet, but extranets and the Internet, security has to be a primary concern.

IBM SecureWay FirstSecure is a comprehensive security framework of protection, detection, infrastructure, and development capabilities. FirstSecure can provide scalable end-to-end security for your e-business environment.

FirstSecure consists of five major components:

1. Policy Director
2. Boundary Server
3. Intrusion Detection
4. Public Key Infrastructure
5. Toolbox

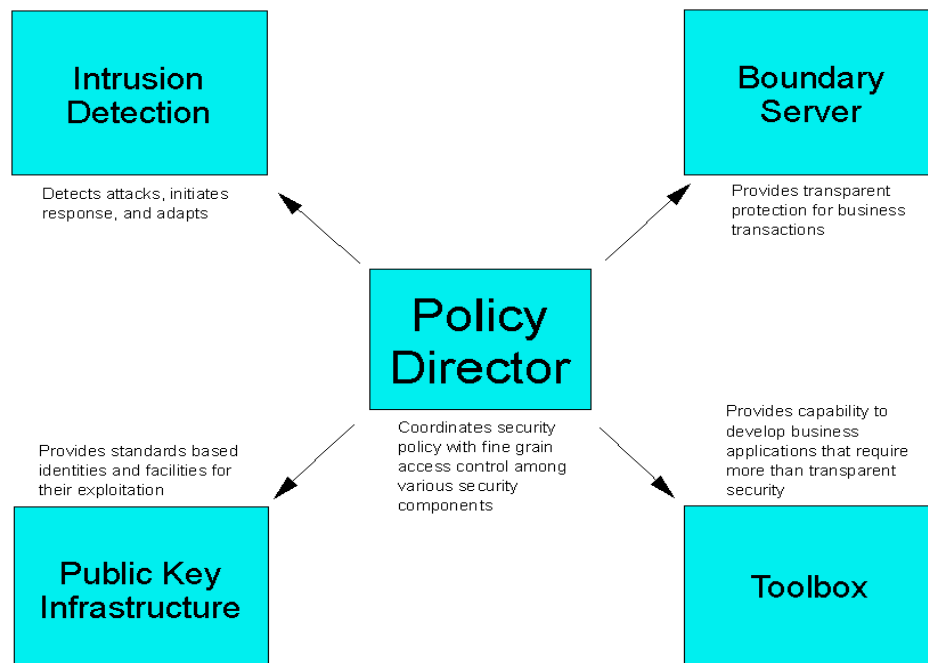


Figure 79. SecureWay FirstSecure Components

8.1 Policy Director

Policy Director is the central component that coordinates security policies and provides fine-grained access control across the network. Policy Director

integrates with existing legacy and emerging intranet infrastructures to provide a centralized policy management capability.

Components

- User Registry
- Management Server
- Security Manager with WebSEAL and NetSEAL
- Authorization Manager
- Management Console
- Directory Services Broker (part of Management Server)
- Credentials Acquisition Service
- NetSEAT

Features

- Central control point for FirstSecure components
- Provides access control to Web objects
- Enables replication and load balancing
- Provides a consistent, manageable access control policy
- Offers extensible authentication and authorization
- Delivers secure remote access and personalized access
- Offers one-time authentication capability with access to multiple Web resources
- Supports Public Key Infrastructure
- Auditing facility generates an audit trail

Architecture

In its simplest form, Policy Director can be considered a server on a network that is utilized to provide access control to other information repositories. Client computers access the Policy Director system transparently for authentication when requesting a service from a network server, such as a Web server. If they have the appropriate authorization to access the resource, they are connected to the server. The target resources may be a Web page, a CGI application, or another application that is accessed through a comprehensive framework like WebSphere.

In the case of multiple Policy Directors being used, the IBM SecureWay Network Dispatcher can load balance between the Policy Director servers to reduce potential bottlenecks.

8.1.0.1 Management Server

The Management Server maintains the master authorization policy database for the secure domain. The Management Server replicates this master authorization policy database throughout the secure domain and is responsible for updating the replica databases whenever a change is made to the master database.

The Management Server also maintains location information about the other Policy Director and non-Policy Director servers operating in the secure domain.

8.1.0.2 Security Manager

The Security Manager applies access control policy based on information from an authorization policy database. The Security Manager includes a NetSEAL component for coarse-grained TCP/IP access control and a WebSEAL component for fine-grained HTTP and HTTPS access control.

8.1.0.3 User Registry

The User Registry contains an account entry for all valid users (principals) who participate in the secure domain. This database is used in two important areas:

- Defining user information for authentication services.
- Defining the groups and organizations to which the principals belong and the roles the principals can assume. This information is considered during an authorization decision.

The User Registry is held in either an LDAP directory (provided by IBM SecureWay Directory Server) or on the DCE Security Server, called the DCE registry. The use of an LDAP-based registry has a number of advantages over a DCE registry, such as its scalability and better interoperability features. Also, it can be assumed that further integration work within IBM SecureWay FirstSecure will focus on LDAP rather than DCE. The use of LDAP, however, does not mean that DCE is not required. DCE is required as a security framework for the Policy Director components to work together.

8.1.0.4 Management Console

The Management Console is a graphical Java application used to securely manage security policy and all Policy Director components. From the Management Console, an administrator can add, delete, or modify users and groups, and apply access control lists. The Management Console also allows for managing the user registry and the primary authorization policy database. The Management Console uses either the NetSEAT client (on Windows NT, 95, and 98) or a DCE client to perform these management tasks through secure communications channels.

8.1.0.5 NetSEAT

NetSEAT is a set of Policy Director and DCE elements that are usually installed directly on a client system. It is lightweight and easy to install. It acts as a proxy between the client application and the Security Manager (WebSEAL or NetSEAL component). This capability allows for the inclusion of a client computer into a DCE cell, since NetSEAT functions as a DCE client, without installing DCE client code separately.

NetSEAT provides a secure tunnel endpoint for all network communications. The user's authenticated identity is passed over a GSSAPI or SSL created tunnel, along with the original protocol request. Such a tunnel is used, for example, in NetSEAT's communication with NetSEAL.

8.2 Boundary Server

Boundary Server is a perimeter security solution managing not only "who" accesses the network, but also "what" enters or leaves the network. Boundary Server enables a company to define and enforce Internet usage policies.

Components

- SecureWay Firewall
- ACE/Server
- MIMESweeper
- SurfinGate
- Norton AntiVirus for Internet Email Gateways

8.2.0.1 SecureWay Firewall

The SecureWay Firewall serves as the IP and application level access control on the dedicated firewall host. The firewall implements IP filters, SOCKS 5, and application proxies like HTTP, Telnet, and FTP.

The IBM SecureWay Firewall may be configured and administered using a Java based GUI from a Windows 95, Windows NT, or AIX workstation, or directly from the graphic display of the firewall. The firewall comes with a comprehensive set of predefined firewall services and SOCKS rules to simplify configuration of the connections.

8.2.0.2 ACE/Server

The Security Dynamics ACE/Server administers the user database and answers questions of the ACE/Clients about authentication success in the SecurID authentication system. Remote administration is provided for the ACE/Server with a GUI. It is only available for the Windows platforms.

In a typical authentication scheme, a user connects to the firewall where the ACE/Client controls access. The user has to enter his or her PIN code and the number generated by the token. The ACE/Client communicates over an encrypted connection with the ACE/Server to check if the number is correct. If so, access is granted.

8.2.0.3 MIMESweeper

Content Technology's MIMESweeper consists of MAILsweeper and WEBSweeper. MAILsweeper, which screens SMTP mail contents, consists of three components: the MAILsweeper itself, the MAILsweeper policy editor, and the MAILsweeper manager GUI. MAILsweeper checks mail and attachments for a configurable size limit, certain types of files, and certain keywords or phrases. Depending on the check results, the mail can be delivered, quarantined, or dropped.

WEBSweeper is a tool for filtering HTTP and FTP over HTTP traffic. It provides three filter mechanisms: anti-virus check, PICS rating, and code blocking. It also has a built-in caching proxy function. The anti-virus check filters the data portion of FTP and HTTP downloads for suspected file name extensions. PICS (Platform for Internet Content Selection) is a system for rating the contents of Web pages. WEBSweeper allows the blocking of Web pages. WEBSweeper can block the transfer of program code types like executables or Java, or if the code is not certified.

8.2.0.4 SurfinGate

Finjan Software Ltd.'s SurfinGate is a content screening tool for checking mobile code in HTTP, HTTPS, and FTP traffic for malicious actions. It can be used as a stand-alone tool with an HTTP proxy chain, or as a plug-in for the IBM SecureWay Firewall secure HTTP proxy.

SurfinGate consists of three components: a server, database, and console. The server performs the inspections. The database stores user settings and previous inspection results. The console is used to supervise the server either locally or remotely.

SurfinGate can perform content inspections and either block or allow Java, ActiveX, and JavaScript code. It can also block applets.

8.2.0.5 Norton AntiVirus for Internet Email Gateways

The NAVIEG works as an SMTP proxy. It accepts SMTP mail, checks it for attachments, splits and decompresses the attachments, and checks those with suspicious file types for viruses. If nothing bad is found, it delivers them to the target host or the next server in the SMTP proxy chain. **Note:** NAVIEG does not protect the machine it is installed on. Its only purpose is to scan passing mail.

8.3 Intrusion detection

Intrusion detection provides proactive detection and protection against potential or actual attacks over a network and alerts administrators of such attacks. It ensures that the network is actively secured from intrusions by communicating policy-based security alerts and events management. This intrusion detection is integrated with the Policy Director. When an alert comes in, the Policy Director will not only record the alert, but will provide the proper component with the information necessary to take action, such as shutting down access to a resource.

Components

- Tivoli Cross-Site for Security
- Norton AntiVirus Suite

8.3.0.1 Tivoli Cross-Site for Security

Tivoli Cross-Site (XSite) for Security is a network intrusion detection system that can detect attacks by monitoring network traffic. Its architecture consists of agents, a server, and consoles. The security agents are installed at strategic points on the network and monitor traffic. The agents act like smart IP packet sniffers that catch all packets, check them for abnormalities, and record the status of established connections and statistics. The server is the management and configuration center. The management console is a GUI for the administration and control of the server and the agents.

8.3.0.2 Norton AntiVirus Suite

The Norton AntiVirus Suite provides virus protection for clients and servers, as well as e-mail content checking. It provides administration tools to help manage virus definition files as well as functions for stopping the continued spread of a virus.

Norton AntiVirus for Windows 95/98 and NT are designed for use on workstations. They protect memory and disks from computer viruses. The Auto-protect component is installed in the computer's memory and monitors system activity and alerts the user of suspicious actions.

Norton AntiVirus for Windows NT Server provides some features of the workstation product as well as two sophisticated methods for detecting unknown and polymorphic viruses. Bloodhound heuristic technology performs code analysis to detect sequences indicative of viral behavior. Strike technology performs decryption of the polymorphic virus so it can be removed from the file.

Norton AntiVirus for Firewalls (NAVFW) scans passing HTTP traffic for suspicious files. The firewall secure proxy server invokes NAVFW whenever an HTTP transfer brings a suspicious file from the Internet.

8.4 Public Key Infrastructure

The FirstSecure Public Key Infrastructure (PKI) offers a high trust, flexible registration and certification solution which adheres to the Common Data Security Architecture and PKIX standards and supports Web applications, VPN, secure e-mail and custom applications.

FirstSecure PKI also provides a PKI application programming interface. This API is a collection of certificate services that gives developers basic PKI capabilities such as Certificate Authority and Registration Authority.

The PKI is built on SecureWay Trust Authority 3.1 which provides the use of certificates for authentication, secured communications, and validation of signed policy. Trust Authority provides organizations with all the tools necessary to issue, publish, and administer digital certificates. It supports the complete certificate life-cycle and provides GUI support for product installation, configuration, and administration.

8.5 SecureWay Toolbox

The SecureWay Toolbox is a set of tools and APIs available for developing secure middleware and applications. The Toolbox APIs allow users to utilize and modify the security features of FirstSecure based on their needs, allowing them to focus on the security of their e-business.

In order to provide customers with the ability to develop and test security applications, the Toolbox gives two samples of programming with the specific APIs. A secure mail client and a secure Web browser/server application are provided. The high-level APIs are implemented in C and/or Java language environments. The APIs available in the C environment are Keyworks, LDAP, PKI, and Authority API (Policy Director). The APIs available in the Java environment are LDAP and SSL.

The APIs described previously also use products and packages provided in the original form called Security Services.

Security Services Components

- IBM KeyWorks Toolkit is a cryptographic and trust services toolkit which enables applications to provide these functions. It works with the PKIX high-level API.
- Secure Socket Lite Java-based toolkit provides basic SSL protocol and utilities. It uses the SSL API.
- LDAP services provide access to LDAP V2 and V3 servers and Policy Director. It works with the LDAP API.

8.5.0.1 IBM KeyWorks Toolkit

The KeyWorks Toolkit provides a set of layered security services and associated programming interfaces designed to furnish an integrated set of information and common security capabilities. The four major layers are Application Domains, System Security Services, KeyWorks framework, and Service Providers. These layers start with fundamental components such as cryptographic algorithms and

random numbers, and build up to digital certificates, key management, and secure transaction protocols in higher layers.

The IBM Key Recovery Service Provider toolkit is a plug-in module for KeyWorks that creates key recovery blocks. This toolkit enables the recovery of stored and transmitted encrypted information without collecting and escrowing private keys and creating single points of cryptographic vulnerability.

The IBM Key Recovery Server is an NT-based application that is required for developing and testing application run time environments using the KeyWorks framework and Key Recovery Service provider.

8.5.0.2 SSL APIs

Secure Socket Lite uses SSL to control data access. It encrypts data using public and private keys. It does not function alone. SSL technology is incorporated in several other APIs for encrypting data and creating passwords. The Toolbox provides the SSLight Socket API for Java environments.

The SSLight Socket API is based on the regular Java Socket API. It provides SSLSocket and SSLServerSocket classes.

8.5.0.3 LDAP Services

SecureWay Directory uses LDAP technology to allow the user to organize, control, and access your directories. It is based on a client/server model that provides client access to an LDAP server. The Toolbox provides access to all LDAP API function calls for LDAP V3.

SecureWay Directory provides a means of maintaining directory information in a central location for storage, updating, retrieval, and exchange. It uses SSL support to create passwords and encrypt information, so you can control who has access to your directories.

8.6 References

- *Up and Running Guides* found at:
<http://www.ibm.com/software/security/firstsecure>
- *Understanding IBM SecureWay FirstSecure*, SG-5498
- README documentation that comes with the product

Chapter 9. Solutions for accessing CICS from the Web

CICS is one of the most widely used and popular transaction managers in the world with some of the largest companies around the world relying on CICS for their business transactions. CICS was introduced in the late 60's for mainframe platforms. Since then, CICS has been changing at the same pace as technology, going from centralized mainframes to distributed client/server architecture and now to the Internet, allowing people to continue to access their applications through Web browsers.

CICS is entering the new millennium with a lot of experience in managing business transactions and the capacity to follow the technology evolution.

This chapter will give you a high-level overview of the solutions available for Web-enabling CICS. There are several excellent redbooks available to discuss this topic, specifically for CICS, and they have been listed at the end of each section. For a good technical discussion of each of these options, planning information, security, and implementation considerations, we highly recommend *Revealed! Architecting Web Access to CICS*, SG24-5466.

9.1 Terminology

We will be using several acronyms in this chapter. The following terms will be used in explaining CICS Web-to-host options.

CICS External CICS Interface (EXCI) allows a non-CICS S/390 program to invoke a CICS program by distributed program link (DPL), and thus provides connectivity to CICS from an S/390 Web server program. The server program must execute on the same Sysplex as the CICS region or the same OS/390 image in a non-Sysplex environment. Data from the calling program is presented to the CICS program as a COMMAREA. EXCI has been available in all releases of CICS for OS/390 since CICS/ESA V4.1.

External Call Interface (ECI) enables a non-CICS client application to call an S/390 CICS program synchronously or asynchronously, as a subroutine. The client application communicates with the server CICS program, using the COMMAREA data area.

External Presentation Interface (EPI) enables client applications to start and converse with a 3270-CICS application running on CICS server. The CICS application sends and receives 3270 data streams to and from the client application as though it were conversing with a 3270 terminal. The client application captures these data streams and displays them in a graphical user interface (GUI) product.

CICS Universal Client provides an API for connectivity to CICS from non-S/390 server platforms, including OS/2, Windows NT, 95 and 98, AIX and Solaris. It can be used from a server program written in most languages supported by these platforms. It is also available to programs using object classes for C++, Microsoft COM interface, VisualAge, Visual Basic and PowerBuilder.

Distributed Program Link (DPL) is the component of CICS that enables a CICS client program to link a CICS server program and pass parameters.

LINK - LINK is a CICS operation for connecting to other applications.

Basic Mapping Support (BMS) is a CICS component responsible for generating, managing screens (maps) and the input/output operations to the screens.

COMMAREA - COMMAREA is the CICS applications input/output storage area used to communicate with other applications.

Alias - Alias is a CICS supplied transaction used by the CICS Web solution interface.

9.2 CICS Web solution overview

There are many options available to Web-enable a CICS application. CICS provides several solutions, designed exclusively for CICS. There are also several non-CICS solutions that are very well suited for Web-enabling CICS, such as Host-On-Demand and Host Publisher that can also be used to Web-enable other host applications.

9.2.1 Solutions for Web-enabling CICS and the rest of the enterprise

When considering how to Web-enable your CICS environment, you should consider solutions that will not only Web-enable your CICS applications, but other application data as well. Using a solution like Host On-Demand or Host Publisher, leaves your options wide open. They can address many applications and environments, where solutions designed exclusively for CICS will only solve your CICS Web-enablement problem.

Host-On-Demand (HOD) - Host-On-Demand provides terminal emulation in a Java environment. Access to CICS transactions and other S/390 applications can be done using a Web browser as a 3270 emulator. Setup is simple, requiring only a TN3270 server. The appearance to the user is much like a traditional 3270 screen. With the addition of the Screen Customizer function, the presentation of the CICS transaction interface can be enhanced to a graphical user interface (GUI).

Host-On-Demand connects to CICS by using a Java applet downloaded from the Web server. The Java applet connects to the OS/390 host using a TN3270 server. Network and Web server capacity can make a difference in the time it takes to download the client. HOD provides alternative clients to the download client that eliminate the need to download the applet for each user.

HOD and Screen Customizer are an alternative to using CWS and 3270 Web bridge, with the added advantage that HOD can connect to any 5250, 3270, or VT application. In addition, no changes need to be made to the host application. Often, the 3270 Web bridge will require that the CICS transaction be modified to handle unsupported 3270 commands, though there is no applet download time.

Because HOD uses a Java applet, the Web browsers must be Java enabled. This may make Host-On-Demand better suited for an intranet or extranet environment where there is some control over the client machines.

Host On-Demand also supports a direct connection to CICS through a CICS Transaction Gateway using the CICS Gateway applet. This type of session is in addition to the other emulation session types.

Host Publisher - Host Publisher is very suitable for developing new Web applications that get data from existing 3270-based, 5250-based, VT-based, and JDBC-based applications, providing dynamic HTML pages for display. Access to host applications and databases is provided by Host Publisher applications consisting of Integration Objects. Host Publisher uses an “inner-host-on-demand” technology to access existing applications. In doing so, Host Publisher delivers new applications based on existing ones without changing the existing ones. Host Publisher allows you to integrate data from several sources into the Web pages, giving more flexibility in building the data to be presented to the browser. Because the output of the applications is HTML there are no Java requirements for the client browsers.

Though based on the same technology for 3270 application access, Host On-Demand and Host Publisher differ in their capabilities and end user interaction. Host On-Demand provides a terminal front-end to host applications which can present either a traditional or GUI presentation of the host screens. Host Publisher provides a more sophisticated programmable interface, allowing logic, screen recognition, composite applications, and other application enhancements. The two products complement each other in providing a full range of host integration possibilities.

For more details about Host Publisher, see Chapter 3, “IBM SecureWay Host Publisher” on page 41.

Lotus Domino - For customers with a Lotus Domino environment, two LotusScript programming options exist for integrating your Domino environment with CICS via MQSeries:

- **MQSeries Link LotusScript Extension (MQLSX)** for programming Notes applications to access CICS or other applications via MQSeries, providing better performance than MQEI.
- **MQSeries Enterprise Integrator (MQEI)** is a more general LotusScript extension for MQSeries, CICS and IMS access, providing a simpler API than MQLSX. It provides access to CICS through MQSeries, CICS Client ECI, and CICS Client EPI.

For more details about Lotus Domino connections to CICS, see “Access to enterprise transaction systems” on page 105.

Host Access Class Library (HACL) -The Host Access Class Libraries (HACL) provide a core set of classes and methods that allow the development of platform-independent applications that can access host information without the need for a graphical display. The library represents an object-oriented abstraction of a host connection that includes reading and writing the host presentation space, enumerating the fields in the presentation space, reading the operator information area (OIA) for status information, transferring files, and performing asynchronous notification of significant events. For more information about HACL, see Chapter 5, “Host access APIs and reusable components” on page 83.

Component Broker provides a development toolkit and run time environment for distributed three-tier applications. Objects developed in different languages communicate with each other by way of the standard CORBA Object Request Broker and have a persistent image of traditional database and transaction

environments through Application Adaptors. IBM Component Broker Connector (CBCConnector) provides application adaptors to access CICS, IMS, and DB2.

More information on Component Broker can be found in 7.3.6, “Component Broker” on page 124.

9.2.2 Solutions designed specifically for CICS

There are several Web-enablement solutions designed exclusively for CICS. CICS Web support, 3270 Web Bridge, CICS CORBA Support and NetCICS are facilities that run in the CICS server to provide a Web interface to CICS transactions and programs.

CICS Transaction Gateway (CTG) is implemented as a separate system from CICS, functioning as a gateway between CICS and Web browsers or other clients. In this case, a CICS server is not aware of the Web environment and CICS applications do not need modification.

CICS Web support (CWS) and 3270 Web bridge - CICS Web support provides the means for Web users to access CICS programs and transactions running on a CICS server. It can be used as a call interface by non-CICS applications to access programs and services inside of CICS. When used in conjunction with the 3270 Web bridge and Web bridge exit, it can be used to access existing 3270 based transactions with little or no change to these transactions.

The 3270 Web bridge lets you execute a transaction written for 3270 terminal input/output without a 3270 (either real or emulated), without modifying the transaction, and without converting transaction data to and from 3270 data-stream format. Because the transactions run unchanged, the bridge is suited for Web-enabling CICS transactions while still being able to access it from the traditional 3270 terminals.

CWS, available in CICS Transaction Server V1.3, is the follow-on of CICS Web Interface (CWI). Functionally, CWS has added a new Web API with no 32 KB limit and some internal differences.

CICS CORBA Client Support - By introducing a new Java class library (JCICS), CICS Transaction Server for OS/390 V1.3 supports inbound requests to Java application objects, using the IIOP protocol. This feature allows a client on any platform, including a Web browser or a Web server, to access a CICS Java application, provided the request conforms to CORBA 2.0 standards. CICS Transaction Server V1.3 also provides support for writing new CICS applications in Java and for wrapping existing applications with this interface.

NetCICS - NetCICS is an alternative solution to CWS and the 3270 Web Bridge that is currently not available from IBM as a standard product but as a service offering only. NetCICS offers a quickly implementable solution to CICS/ESA V4.1 users to access a CICS 3270 environment from the Web without the need to migrate to OS/390 CICS Transaction Server. Full 3270 terminal support is provided, whereas the standard 3270 Web Bridge interface only supports a subset of the BMS. A higher degree of HTML customization is provided, allowing the HTML to be modified dynamically within CICS, as opposed to the standard HTML template customization, which permits only user-definable common headers and footers and static modifications to the generated HTML templates. A keyplay capability provides a one-to-many conversion opportunity, allowing

multiple BMS screen interactions to be handled before an HTML page is returned to the browser.

CICS Transaction Gateway (CTG) provides a set of Java-based Web server facilities for access to CICS applications from a Web browser. These facilities include Java classes and Java beans for writing servlets and applets. There are classes for access to both traditional and object-oriented CICS applications. The classes required for access to 3270-interface transactions are available only on non-S/390 platforms. The CICS Transaction Gateway can be used to connect Web clients via IIOP to a CICS Server.

CICS Transaction Gateway is available as part of IBM CICS Universal Clients Version 3.1. It is available for S/390, Windows NT, AIX, OS/2 and Solaris and replaces the CICS Gateway for Java and CICS Internet Gateway products. (The last two remain available for use with CICS Clients Version 2.0, but are not supported for use with CICS Universal Clients Version 3.1.)

Customers using the CICS Internet Gateway or the CICS Gateway for Java, which are not supported by CICS Universal Clients Version 3.1, should continue to use CICS Clients Version 2.0, but should consider migrating to the CICS Transaction Gateway Version 3.1 at the earliest opportunity.

9.2.3 Complementary products

9.2.3.1 CICS and VisualAge Generator

VisualAge Generator is a very powerful tool for developing applications for CICS, DB2, IMS, VSAM, Oracle, and more. It is an application tool that generates applications to all available user interfaces by automatically segregating the presentation logic from the business logic and allowing users to connect to the application with any client interface. Users can run the same application at the same time, through any client interface. VisualAge Generator automatically generates the client portion of the application for all of those interfaces.

If you have CSP (Cross System Product) applications in your shop, you can benefit from VisualAge Generator because it is the follow-on of CSP. It has the capability to migrate existing CSP applications to VisualAge Generator and uses CSP development skills.

VisualAge Generator is included in the WebSphere Application Server Enterprise Edition, bundled with VisualAge for Java. For more information about VisualAge generator see URL: <http://www.ibm.com/software>.

9.2.3.2 CICS and WebSphere

WebSphere is not an alternative solution for CICS but there are some specific components of WebSphere that fit into the Web-to-CICS solution picture.

IBM HTTP Server

The IBM HTTP Server is the official IBM Web Server for all IBM platforms. We mention several times in this chapter that you can access CICS applications from a browser client without involving a Web server, due to the CICS support for HTTP protocol. This does not mean CICS is a Web server and that it provides traditional Web server functions. CICS cannot perform as a Web server and supports only limited Web server functions. It should only be used for specific scenarios, such as CICS Web-enablement on an intranet.

In general, our recommendation is to access CICS applications through a traditional Web server when it is possible and does not add complexity to the existing environment. In particular, if your shop is S/390-based, you could have the Web server and the CICS server running on the same machine.

IBM WebSphere Application Server

IBM WebSphere Application Server (WAS) is a Java-based environment for Java applications and Java connectors for enterprise resources such as CICS .

WebSphere Application Server Enterprise Edition, in conjunction with Enterprise JavaBeans (EJB) and Component Broker (CB) has been developed to be a new transactional environment. Eventually, in the future, most transactional applications will likely run under WAS control.

CICS has been updated to support Java applications running under its internal Java Virtual Machine (JVM) and also to interoperate with WAS and CB. CICS Transaction Server for OS/390 V1.3 supports IIOP inbound connections and SSL.

This makes WAS and CICS complementary to each other. WAS is suited for Java technology applications and CICS is suited for managing transactions, thousands of transactions per second, with data integrity and session management.

9.3 Making the decision

Planning to make CICS programs and transactions available on the Web is much like planning for other legacy applications, though there are some unique considerations. There is an in-depth discussion of the considerations for Web-enabling CICS, including topics such as performance, scalability, security, networking, application design, and others, in *Revealed! Architecting Web Access to CICS*, SG24-5466. We highly recommend that you consult this book while designing your CICS Web-enablement. Here is a brief comparison of the products as related to design issues.

Application development

- CTG: You need to write Java applet or servlet applications using CTG classes. CTG is IBM's strategic direction for Internet integration with CICS applications.

There is strong support for this approach in products such as WebSphere Application Server. But high-end Web skills are required to implement CTG solutions. This will require appropriate existing skills or the ability to recruit the skills or time and budget for training.

- CICS Web support: You might need to write your own converter to convert data in the COMMAREA. When using CICS 3270 Web bridge or NetCICS to access 3270-based applications, no programming is necessary. Analyzer, Converter, bridge exit and HTML template customization is optional.

This is the simplest Web connection to CICS and the easiest way of invoking CICS programs from the Internet. Strong CICS skills are required. The HTML rendered by the 3270 Web bridge is fairly basic. The HTML templates provide only limited scope for static customization.

- CICS CORBA client support: You need to develop client and server objects in Java.

CICS CORBA client support only implements a subset of the CORBA version 2 services, but those excluded are already provided by CICS services, or they

are not used when running under CICS. CORBA knowledge, Java skills and CICS skills are required.

- HOD: No programming is needed. Customization using Host Access Class or Host Access Beans for Java is optional. This is the simplest means of getting at 3270 data from a browser. It only displays the normal text-based CICS screen. It is only suitable for use inside a firewall.
- Host Publisher: Very flexible for developing applications to access a variety of host applications and data. It can be used to combine information from multiple host applications into composite Web pages. You will need to develop Host Publisher applications consisting of Integration Objects and Java Server Pages (JSPs). Host Publisher comes with a development tool, Host Publisher Studio, that provides development wizards to guide you through the process of developing applications.

Changes to existing CICS applications

- CTG: There are no changes to COMMAREA-based applications. There are some limitations when using EPI to access 3270-based applications.
- CICS Web support: There are no changes to COMMAREA-based applications. There are some limitations when using CICS 3270 Web bridge and NetCICS to access 3270-based applications.
- CICS CORBA client support: No changes to COMMAREA-based applications. No changes when using CICS CORBA client support with FEPI to access 3270-based applications. There are some limitations when using CTG IIOP support to access 3270-based applications due to the limitation of EPI.
- HOD and Host Publisher: No changes to existing applications.

Security

- CTG: The CICS Transaction Gateway has support for secure Web interactions using SSL support. If using the servlet architecture, the SSL support is offered by the HTTPS support of the Web server and included in externally signed certificates and client authentication. If using the applet architecture, there is only support for self-signed certificates and no support for client certificates. Both ECI-based and EPI-based applications can invoke RACF security checking on the CICS server, although EPI applications cannot perform SIGNON operations.
- CICS Web support: With CICS Transaction Server V1.3 this offers high security based on the OS/390 platform. It uses CICS security for all components, it is RACF controlled, and SSL is supported. The OS/390 Web server can authenticate users against RACF.
- CICS CORBA client support: Similar to CICS Web support (OS/390, CICS security, RACF, SSL). However, the user authentication cannot be done within CICS, although OS/390 Web server can still authenticate users against RACF. CICS must "trust" that the user ID received is authentic. Also, note the limited availability of IIOP proxy server products.
- HOD: When using the HOD 3270 applet via an OS/390 Communications Server telnet server, there is support both for SSL-secured telnet sessions and direct SIGNON to a CICS system. HOD SSL support includes support for externally signed certificates, but communication is limited to telnet via SSL so HTTP proxy servers cannot be used, and a telnet application proxy such as the HOD redirector must be used.

- **Host Publisher:** SSL support for TN3270 sessions provides server authentication and data encryption for these connections. SSL can also be used between the Web browser and Web server for added security. In addition, encrypted passwords can also be used to protect user passwords when applications are transferred to the Host Publisher Server.

Administration

- **CTG:** Depending on which platform the CTG is implemented (OS/390 or distributed), there will be administration differences between the two environments. The OS/390 environment will require additional definitions within CICS; these will not be required for the implementation on the distributed platform. The distributed platform will need configuration for the CICS Universal Client which will not be required for the OS/390 implementation. Security will be based upon existing CICS security plus what is required to set up SSL or HTTPS security for the browser. Communications will need configuration for the TCP/IP address and port information.
- **CICS Web support:** Administration for this environment varies depending upon whether the user request is coming directly into CICS or via an indirect connection through WebSphere Application Server. Most of the additional administration relates to the extra definitions that have to be set up within CICS for the CICS-supplied programs and transactions. Security administration will involve setting up the appropriate authorities for the additional CICS resources and the configuration of HTTPS for the browser attachment. For communications, configuration will be required for TCP/IP through TCPIP SERVICE resource definition. If the user request comes in via the Web server, then there will be configuration required for the Web server environment.
- **CICS CORBA client support:** Most of the additional administration requirements for the CICS IIOP implementation involves installing the Java development environment. The server-side ORB is a set of CICS-supplied programs that have to be defined to the region in which they are running. Security is managed through normal CICS RACF security procedures, with SSL support for IIOP being provided through a PTF release to CICS Transaction Server V1.3. Communications will be via the TCP/IP server of OS/390 eNetwork Communications Server, and will require the CICS TCPIP SERVICE resource definition.
- **HOD:** Host On-Demand has the largest number of configurable options due to the number of different clients you can install and the number of different HOD server configurations you can set up. Since HOD is based upon telnet access to an existing CICS environment, most of the configuration requirements are for setting up the telnet TCP/IP environment and indicating whether you will use SSL or HTTPS security between the clients and the server. Despite the large number of configurable options, you would normally choose only one client environment and one server environment for configuration.
- **Host Publisher:** Administration tasks are simple. After building an application, you must transfer it to the Host Publisher server and deploy it. The transfer is done as a part of the Host Publisher Studio tasks. The server administration function is used to deploy (or remove) applications, monitor server status, and to manage other aspects of the Host Publisher environment (licenses, connection pools, etc.). No administrative tasks are needed on the host side.

9.3.0.1 Application interfaces

In the table below we have summarized the basic ways of accessing CICS resources from the Web and which solutions to use. Use this table to help you decide which solution to implement depending on the connection architecture you have in place and what type of application you have.

Table 4. CICS application interface support

Connection architecture	Application interface	CICS solution	Application type
Connection through S/390 Web server	COMMAREA-based	CTG for OS/390	CICS program
	COMMAREA-based	CWS	CICS program
	3270-based	CWS and 3270 Web bridge	CICS transaction
	3270-based	Host-On-Demand or Host Publisher	CICS transaction
Direct connection to CICS	COMMAREA-based	CWS	CICS program
	3270-based	CWS + 3270 Web bridge	CICS transaction
Connecting to non-S/390 Web server	COMMAREA-based or 3270-based	CTG	CICS programs or CICS transactions
	3270-based	Host-On-Demand or Host Publisher	CICS transaction

Figure 80 shows a graphical representation of this table:

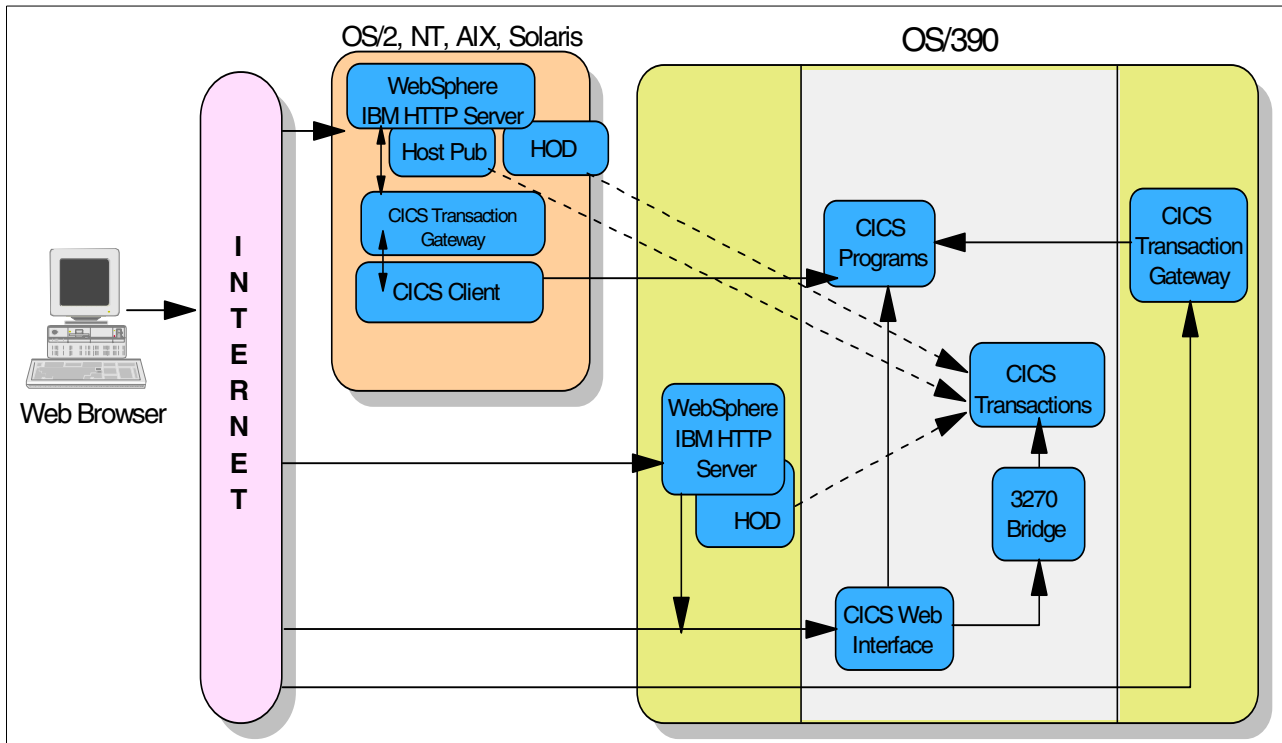


Figure 80. CICS application interface overview

9.4 Connectivity

Products that integrate the CICS environment into the Web use different ways to connect to CICS hosts. The most common CICS server is the OS/390 CICS Transaction Server. Connectivity into this server is summarized in Figure 81.

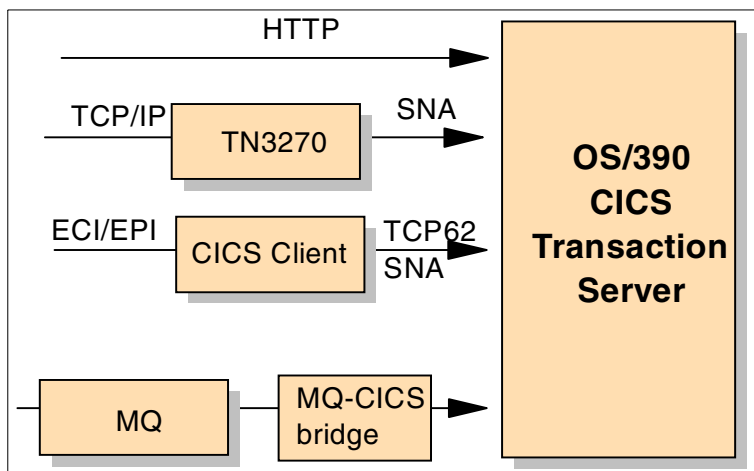


Figure 81. CICS connectivity

- **Direct TCP/IP connection to CICS** - This is the case for CICS Web support (CWS). Web browsers issue HTTP requests directly to CWS.

- **TCP/IP connection by way of TN3270 servers** - This is the case of Host Publisher and Host On-Demand. They use the TCP/IP telnet protocol to send a 3270 data stream, which is translated by a TN3270 server into an SNA connection to the CICS host.
- **ECI/EPI interface to a CICS client** - This is the case of the CICS Transaction Gateway. It invokes ECI/EPI APIs typically on the same machine where the CICS client is. The connection between CICS client and OS/390 CICS Transaction Server can be TCP62, which is LU 6.2 over TCP/IP, or SNA.
- **Direct ECI/EPI interface to the CICS server** - This is the case of LotusScript MQEI, and CICS CGI programs.
- **MQ-CICS bridge** - This is the case of the MQLSX and MQEI LotusScript options.

9.5 Connecting to CICS with Host On-Demand

Through the capabilities of a Host On-Demand environment, Web users can now access host CICS functions using the emulation properties of the Host On-Demand clients to manipulate the CICS application 3270 data stream. Web access to CICS through HOD is recommended only for intranet or extranet solutions for the following reasons:

1. Host On-Demand delivers to the browser a full function terminal emulator, which implies that any user accessing host functions through this method would need to understand the interface of the applications they are accessing. This is unlikely to be something that the general Internet user would be able to do. (The use of Screen Customizer can simplify this interface for novice users.)
2. It will require a Java-enabled browser, and sometimes running specific levels of Java, to support the applets you are downloading from the Host On-Demand server. This is easy to control on the intranet browser and also for terminals of extranet users (such as business partners) but not so easy to control on the terminals of general Internet users.

This does not exclude access through the Internet if you are using the Internet as an external network connection environment for trained or trusted users who can use this path for logging on to your host systems.

In addition to using the 3270 emulation functions, Host On-Demand provides the CICS Gateway applet which allows you to access CICS through a CICS Transaction Gateway.

9.5.1 Accessing CICS using HOD 3270 emulation

Users connect to the OS/390 host using a Web browser as a 3270 emulator, accessing CICS 3270 transactions. This does not differ from any other scenario where Host On-Demand is used as a 3270 emulator to access OS/390 programs.

9.5.1.1 Direct connection

The scenario shown in Figure 82 on page 146, illustrates the simplest configuration where all of the major server components are installed on the same machine. This also provides a fairly high performance solution since all of the components communicate via cross memory. Connections from the browser can

be established to both the telnet server and the Web server across a common TCP/IP connection but with each listening on different ports.

The HOD server can now download to the browser the appropriate client plus the Host Access Class Library via an HTTP request. Session classes from within the applet can now create a telnet session to the enterprise host machine over TCP/IP, which starts a telnet daemon on the host that is interfaced by VTAM. The user can now log on to a CICS region as though they were a locally attached 3270 terminal. Disadvantages of such a design are that all users have a direct connection to the host, which may have to support many telnet servers for each user connection.

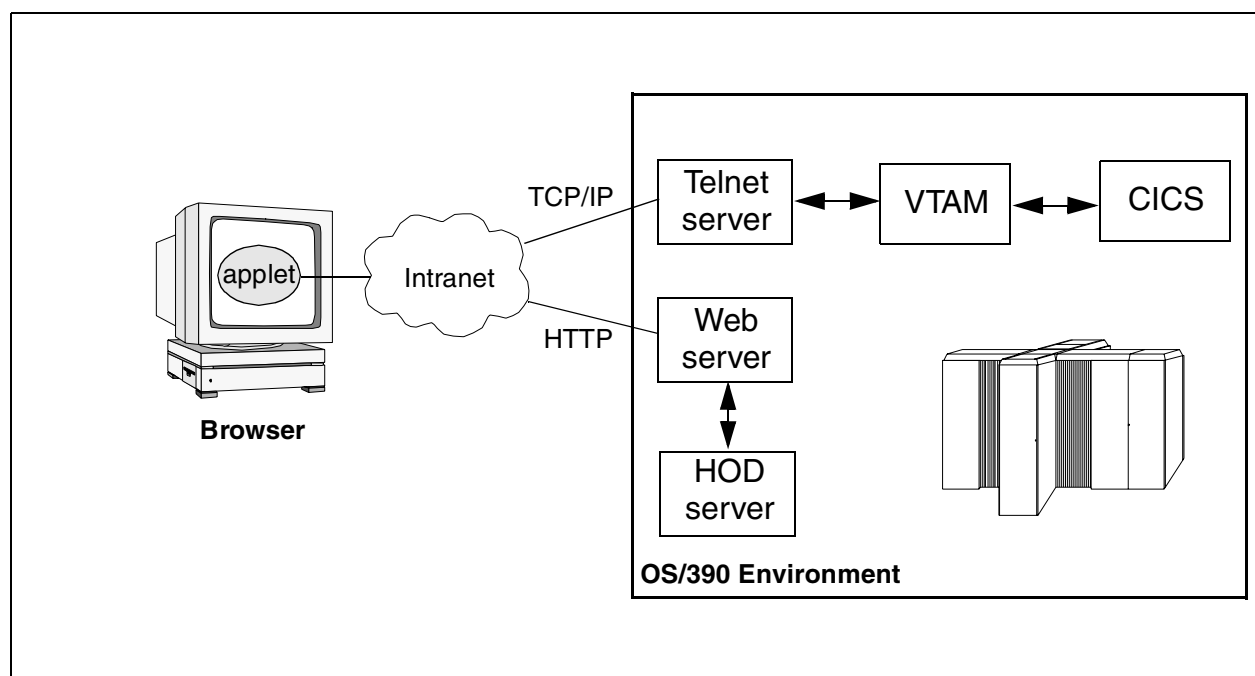


Figure 82. Host On-Demand - direct connection

9.5.1.2 Protocol isolation

In the scenario shown in Figure 83 on page 147, we have a similar connection through the Web to the Host On-Demand server; however, this time our browser sets up the telnet session with a machine designated as a protocol isolation server. The telnet server daemon running on this machine interfaces with the host CICS system through Communications Server and SNA. Communications Server can run multiple SNA sessions for each client connection request coming in through telnet.

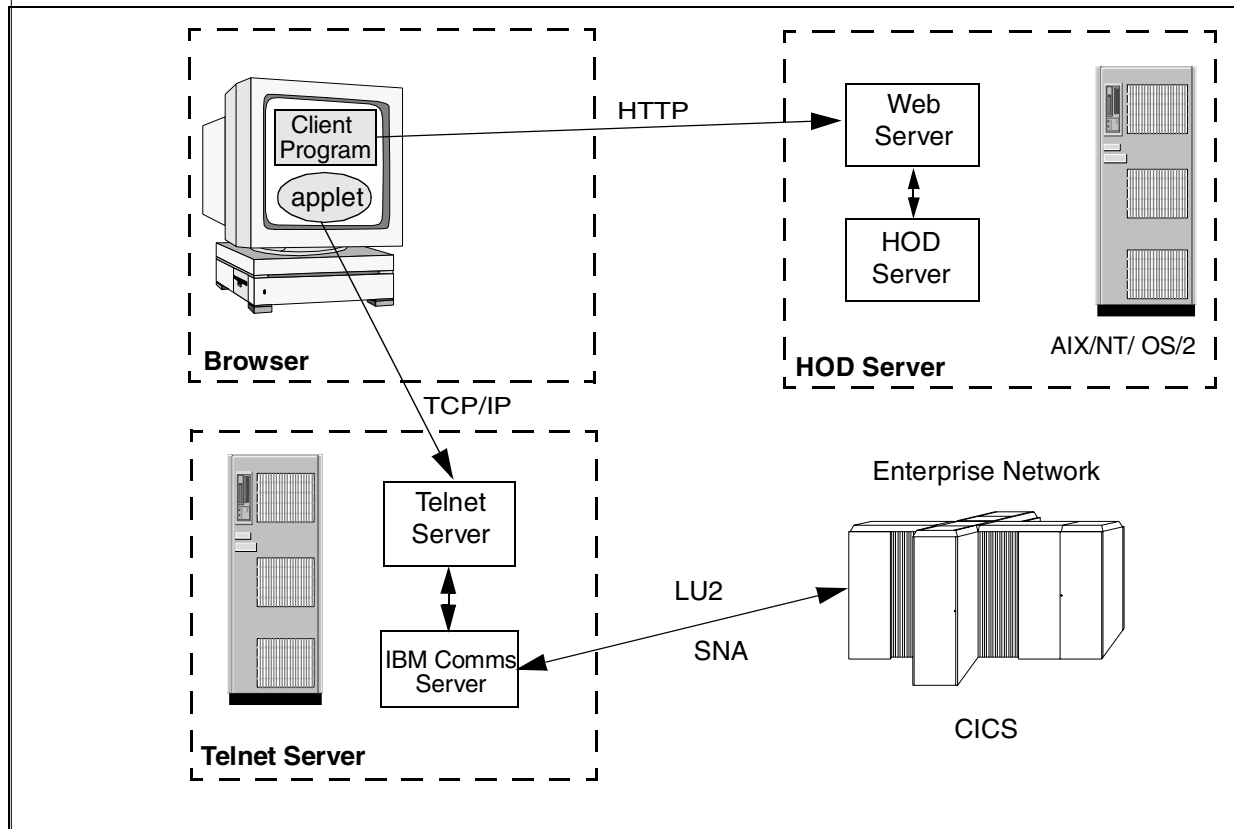


Figure 83. Host On-Demand - Protocol isolation

In the configuration shown in Figure 84 on page 148, the browser applet is setting up a telnet session using the CICS telnet daemon (cicsteld) which is a functional part of the CICS Universal Client. The Universal Client works with the Communications Server product for that platform to handle the telnet session with the CICS host system via SNA.

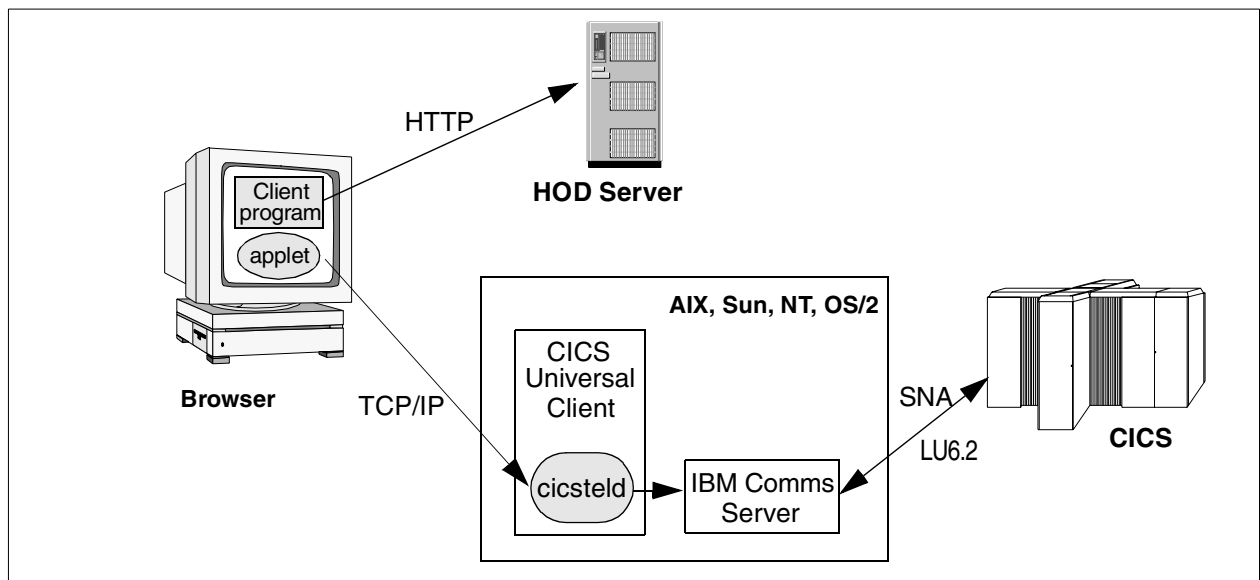


Figure 84. Protocol isolation using CICS Universal Client

9.5.1.3 Redirection

In this final scenario, shown in Figure 85 on page 149, we see an implementation that can be used for an Internet connection using both redirection and a security firewall. You might choose to use this configuration when it is necessary to connect a large number of users to a CICS host system, but because of security constraints, it would be sensible to hide the host address from the end user.

Here we are using two Host On-Demand servers within the firewall of the enterprise network: one is used purely for serving the client applets to the Internet-attached browsers, and the other acts as a redirector for incoming client requests. The redirector should use SSL and would therefore be limited to an AIX or Windows NT server; the connection between the client and the applet server should be in HTTPS.

In this configuration the initial client request will send the appropriate URL to the Host On-Demand applet server. If necessary, these requests can be filtered by a router on the firewall. This results in an applet being downloaded to the client machine along with the appropriate class library. The applet then sets up a telnet session with the redirector (which itself is a telnet server); this connection request is then passed on or redirected to another telnet server within the firewall through a different port. Communications between the client and CICS in the enterprise network are now managed through the telnet server. The role of the redirector is now that of a barrier, effectively hiding the address of the CICS host from the clients.

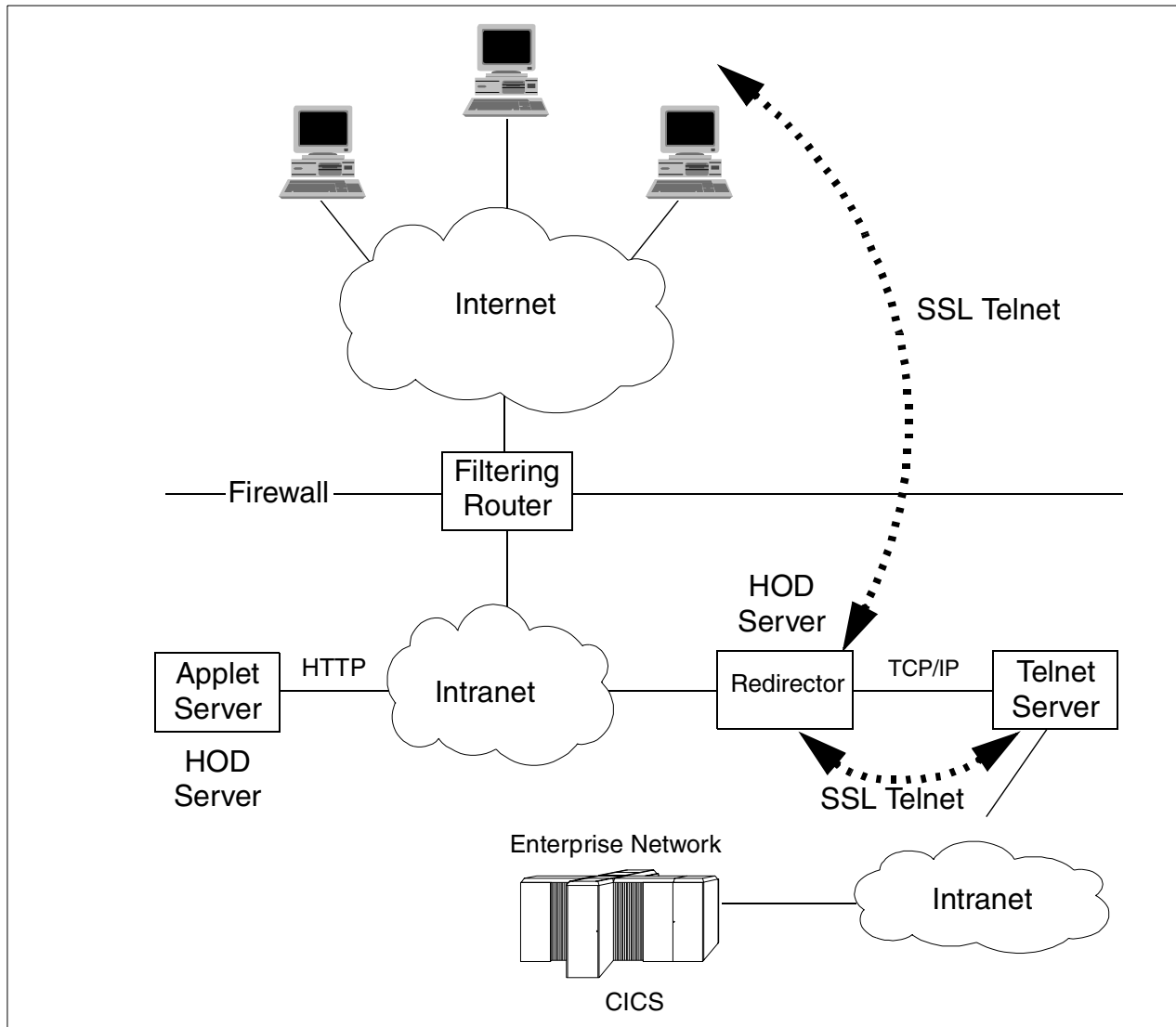


Figure 85. Host On-Demand - Redirection

9.5.2 Adding Screen Customizer functions

The Host On-Demand clients include Screen Customizer/LE. When enabled, Screen Customizer translates the screen presentation into a graphical user interface (GUI) interface, making it more user friendly.

The full version of Screen Customizer can be used to create customized GUI presentations by using screen recognition and saved map files. Screen Customizer Administrator allows you to capture and customize the host application screens by substituting Web page-like objects for application fields to create a true GUI. Some of the objects that are available for customizing your host screen are:

- Images
- Image buttons
- Frames
- Buttons
- Logos

- Check boxes
- Choice boxes
- Radio buttons
- Lists
- Valid-values buttons
- Web-link buttons

9.5.3 CICS Gateway applet

The CICS Gateway applet is shipped as part of the HOD Service Manager function. During configuration of your user sessions, you can set up a session called the CICS Gateway which will set up a TCP/IP gateway session with the CTG running on the distributed platform. Figure 86 on page 150 shows the configuration for using the CICS Gateway applet. In this example, the CICS Gateway applet sets up a TCP/IP gateway session with the CICS Transaction Gateway process. The CTG now sends an EPI request to the Universal CICS Client which will result in a terminal install in a region on the CICS server. The first screen that the user will see after the connection has been established is the CECI screen. From here the user can continue with a normal CICS session using the CICS default user ID.

Note: The HOD CICS Gateway applet is not written to use the SIGNON support available in the CICS Universal Client. Other methods may be needed to secure the data. Security is discussed in detail in *Revealed! Architecting Web Access to CICS*, SG24-5466.

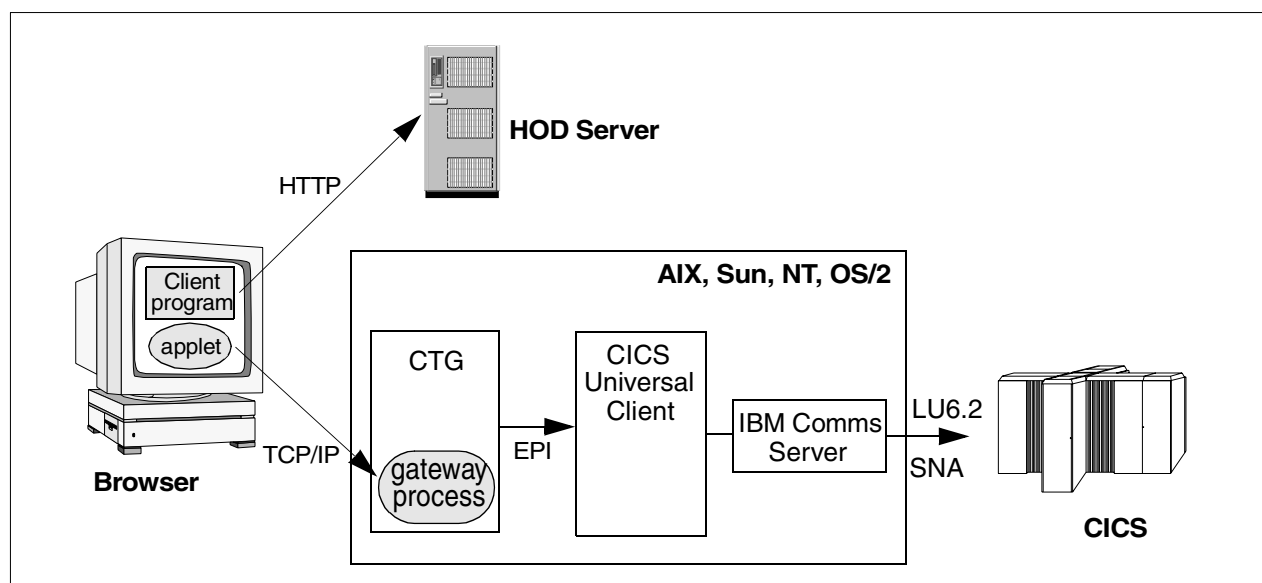


Figure 86. Protocol isolation using a CICS gateway session

For more information on Host On-Demand, see Chapter 2, “IBM SecureWay Host On-Demand” on page 9.

9.6 Connecting to CICS with Host Publisher

IBM SecureWay Host Publisher V2.1 provides access to 3270, 5250, VT, and JDBC backend data sources. As a 3270 application, CICS users can benefit

greatly from the implementation of Host Publisher applications. Like Host On-Demand, it provides access to 3270 applications without any change to the application. Unlike Host On-Demand, it does not place any specific requirements on the client Web browsers, and therefore, is appropriate for Internet use as well as intranet and extranet.

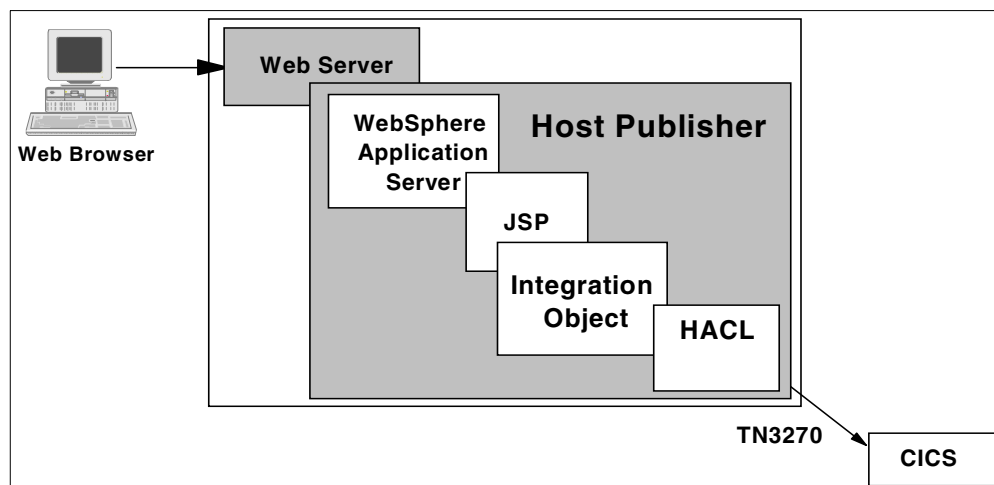


Figure 87. Using Host Publisher to access CICS

Host Publisher allows you to build an application consisting of JSPs that execute on the Web server. While Host On-Demand provides simple access to the 3270 application and with the enhancement of Screen Customizer functions can turn the presentation into a GUI with some enhancements, Host Publisher allows you to build Web applications that not only turn the presentation into a GUI, but adds enhancements such as session pooling for better performance and the ability to combine the output from multiple applications into one Web page.

Host Publisher applications consist of Integration Objects that are actually beans. The Integration Objects can be used not only in the JSPs built by the Host Publisher Studio, but in any Java application, applet, or servlet.

WebSphere Application Server Standard Edition is provided as an integral part of the Host Publisher server to support the Host Publisher applications.

Host Publisher comes with the IBM SecureWay Network Dispatcher for load balancing and failover capability among the Host Publisher servers.

More information about Host Publisher can be found in Chapter 3, "IBM SecureWay Host Publisher" on page 41. A brief overview of using Host Publisher to implement a CICS transaction can be seen in 3.6, "Creating host Integration Objects" on page 50.

9.7 CICS Web support (CWS)

CICS Web support (CWS) provides client Web browsers with direct access to CICS programs or transactions running on an OS/390 CICS region. The base

requirements for this function are provided in CICS/ESA V4.1, but significant enhancements are provided in CICS Transaction Server (CICS TS) V1.3.

Note

In CICS Transaction Server V1.3, the CICS Web functionality, previously known as the CICS Web Interface (CWI) was split into the listener support for TCP/IP, and the protocol support for HTTP, and was also internally redesigned. This book now refers to the HTTP protocol support as "CICS Web support" (CWS).

9.7.1 CICS Web support connection methods

CICS Web support (CWS) can be accessed by HTTP requests from Web browsers by one of two routes:

- Through a direct connection from a Web browser to CICS, using the facilities of the CICS Sockets listener.
- Through an indirect connection from a Web browser via the OS/390 Web server and then into CICS.

Both routes allow the use of the same facilities in CICS although the configuration of the two options is significantly different. For both of these routes CICS Web support can be used in two fundamentally different ways:

- To invoke a 3270 transaction using the 3270 Web bridge. The 3270 transaction remains unchanged and the 3270 output is dynamically converted to HTML.
- To call a "Web aware" program, which contains integrated HTTP presentation logic. The CICS program must be able to interpret and act upon the HTTP data stream passed to it and must be capable of building and returning an HTTP data stream back to the client Web browser. To create this HTTP-based presentation logic there are two different methods provided by CICS.
 - EXEC CICS Web API together with the Document API
 - COMMAREA manipulation and the CWS HTML template manager

We will only show a brief overview of CICS Web support. For further details please refer to *CICS Transaction Server for OS/390 Version 1 Release3: Web Support and 3270 Bridge*, SG24-5480 and the *CICS Internet Guide*, SC34-5445.

9.7.2 Calling a Web-aware program using a CWS direct connection

Figure 88 illustrates the major components of CICS Web support using a direct connection to CICS.

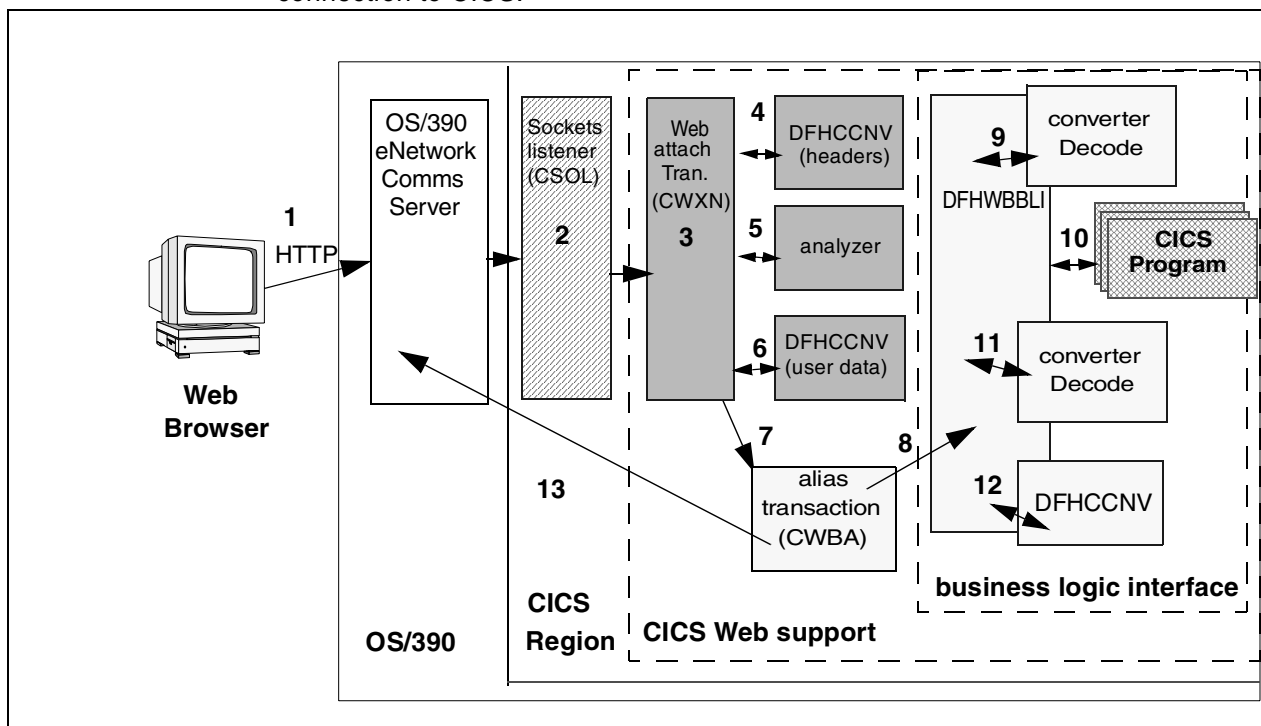


Figure 88. CICS Web support - direct connection

The typical flow of a Web browser calling a CICS program via a direct connection to CICS is as follows:

1. An HTTP request is sent by a Web browser to OS/390.
2. The long-running CICS Sockets listener task (CSOL) monitors the OS/390 eNetwork Communications Server for incoming HTTP requests
3. When the CICS Sockets listener task receives a request it attaches a Web attach transaction (CWYN), to perform the Web attach processing.
4. CWYN first calls the CICS data conversion program (DFHCCNV) to translate the HTTP request headers from ASCII to EBCDIC. If HTTP persistent connections have been configured in CICS, then CWYN will remain running until the client closes the connection or the connection is timed out.
5. The Web attach processing now calls the analyzer.
6. The analyzer calls the data conversion routines (DFHCCNV) to convert the body of the HTTP request from ASCII to EBCDIC.
7. The analyzer starts an instance of the alias transaction (CWBA) to handle all further processing of this HTTP request within CICS.
8. The alias transaction invokes the CICS Business Logic Interface (BLI) by linking to the program DFHWBBLI and passing it to the COMMAREA that contains the information from the HTTP request.
9. The BLI invokes the converter which runs the decode function. This will either retrieve the HTTP data stream using the EXEC CICS WEB API or modify the

HTTP data stream contained in the COMMAREA (for instance removing the HTTP headers).

10. The target CICS program is now invoked by the BLI.
11. On completion of the CICS program, the converter is invoked again and this time it runs the encode function. This will either build the HTTP data stream for output using the EXEC CICS WEB API, or will convert the supplied COMMAREA to the appropriate HTTP response format using the supplied CICS HTML template manager and environment variables program (DFHWBENV).
12. The alias transaction calls the data conversion routines to convert the COMMAREA or EXEC CICS WEB SEND output into the ASCII code page of the Web browser.
13. The alias transaction now returns the results to the OS/390 eNetwork Communications Server and ends. The HTTP response is returned to the browser by the TCP/IP support in the eNetwork Communications Server.

This is only a summary of a typical flow of requests. Many variations are possible but some of the more common possibilities are:

- To not invoke the converter at steps 9 and 11, and to perform manipulation of the HTTP data stream within the CICS application. This design of program is often referred to as a “Web aware” program
- To change the supplied analyzer processing, for instance to use the supplied security analyzer to verify a user ID and password.
- To invoke a user-defined alias transaction instead of the CICS-supplied alias transaction.

9.7.3 Calling a Web-aware program using an indirect connection

An alternative approach to accessing CICS Web support is through the services of the OS/390 Web server, using the CWS Web server plugin. In this implementation, some of the function previously handled through the CICS-supplied programs for CICS Web support is now replaced by function within the Web server. Figure 89 on page 155 illustrates the configuration using an indirect connection to CICS via the OS/390 Web server.

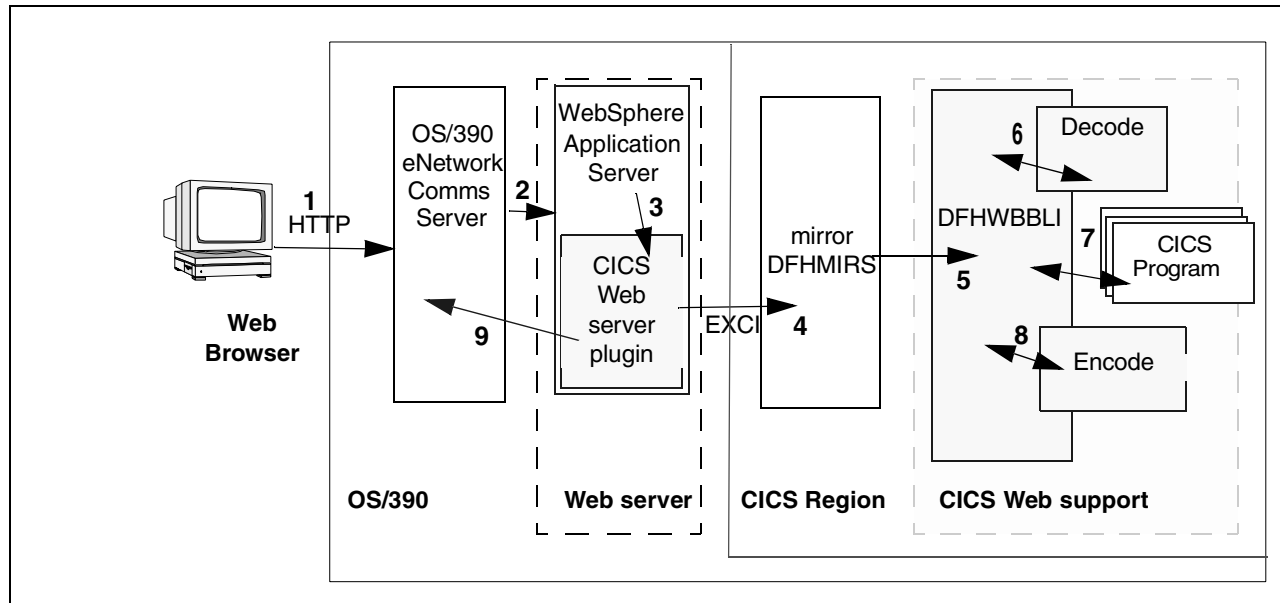


Figure 89. CICS Web support, indirect connection through the CICS Web server plugin

The typical flow for a Web browser calling a CICS program via the OS/390 Web server is as follows:

1. An HTTP request is sent by a Web browser to OS/390.
2. The Web browser's request is received by the OS/390 eNetwork Communications Server, which then passes it on to the OS/390 Web server.
3. The Web server interprets the HTTP request according to a *service* rule in the configuration files (`httpd.conf`). If it matches the rule it invokes the CICS Web server plugin (DFHWBAPI). This creates an EXCI request and sends it to the CICS region specified in the *service* directive. The Web server code will also handle the conversion of the HTTP headers from ASCII to EBCDIC.
4. The EXCI request causes a CICS mirror program (DFHMIRS) to be attached in the CICS region. The request HTTP data stream is passed as a COMMAREA via the mirror transaction.
5. The mirror program calls the BLI program, DFHWBBLI, and passes the COMMAREA to it.
6. The BLI invokes the converter to run the decode function. This will either retrieve the HTTP data stream using the EXEC CICS WEB API or modify the HTTP data stream contained in the COMMAREA (for instance removing the HTTP headers).
7. The target CICS program is now invoked by the BLI.
8. On completion of the CICS program, the converter is invoked again and this time it runs the encode function. This will either build the HTTP data stream for output using the EXEC CICS WEB API, or will convert the supplied COMMAREA to the appropriate HTTP response format using the supplied CICS HTML template manager and environment variables program (DFHWBENV).

The response from the CICS program is then passed back to the mirror program (DFHMIRS) which will return the results of the EXCI call back to the OS/390 Web server.

9. The CICS Web server plugin then converts the HTTP response from EBCDIC to ASCII and passes this to OS/390 eNetwork Communications Server for transmission back to the Web browser across the TCP/IP network.

The same facilities within CICS are available using the indirect connection via the OS/390 Web server but there are a few important differences, which are summarized below:

- The OS/390 Web server and the CICS region must be running within the same OS/390 image or Sysplex since they use the CICS EXCI communication mechanism.
- Only 32 KB of data can be passed to or from the CICS program when using the direct connection via the OS/390 Web server. This is because CICS EXCI communication uses the CICS COMMAREA which has a limit of 32 KB.
- Security processing can be performed in the OS/390 Web server if using the indirect connection.
- Data conversion is performed in the OS/390 Web server, not in CICS when using the OS/390 Web server.

9.7.4 CICS Web support and the 3270 Web bridge

If the CICS service that we now wish to access from the Web is a 3270 transaction as opposed to a callable program, then CICS Web support must use the facilities of the 3270 Web bridge and the Web bridge exit. We will refer to this function as the “3270 Web bridge”.

The 3270 Web bridge can be used with either the direct connection to CICS via the CICS Sockets listener, or the indirect connection via the CICS Web server plugin. Figure 90 on page 157 illustrates the data flow for a Web browser request using the facilities of the 3270 Web bridge to access a CICS 3270 transaction.

Note: the 3270 Web bridge feature is only available when using CICS TS V1.2 or a later release.

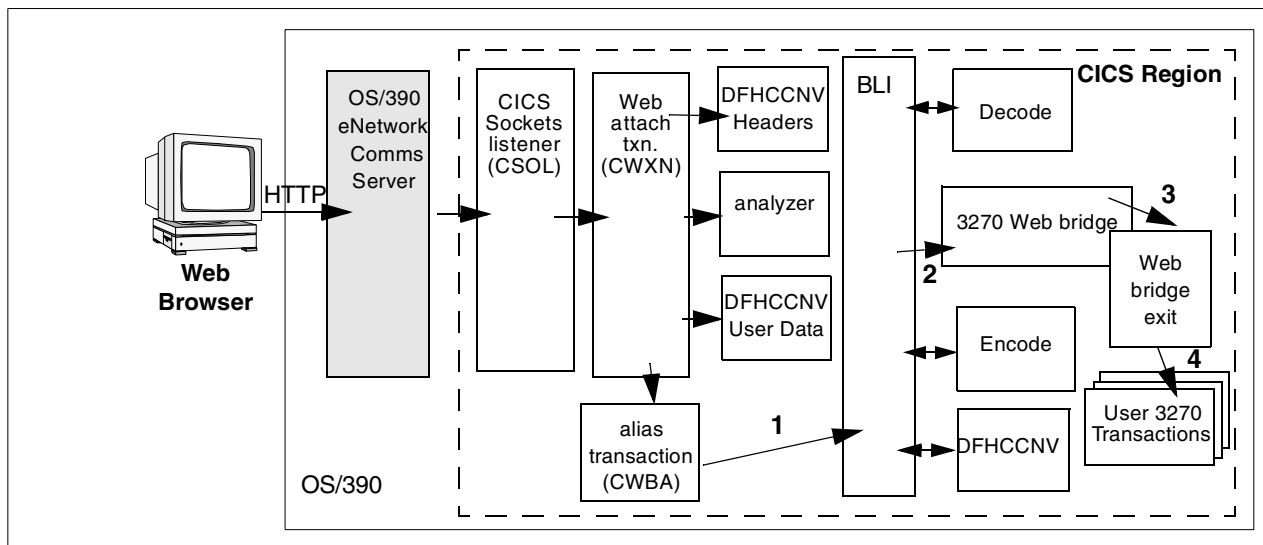


Figure 90. CICS Web support - 3270 Web bridge

The initial data flow is the same as that described in Figure 88 on page 153 for the description of CICS Web support and the BLI. However, instead of invoking a user program, the CICS-supplied 3270 Web bridge program (DFHWBTTA) is invoked by the BLI, and through which the 3270 transaction is run. The sequence of events within CICS Web support is as follows:

1. The alias invokes the BLI, which invokes the decode function as before.
2. The BLI then invokes the 3270 Web bridge (DFHWBTTA) as the user program.
3. The 3270 Web bridge finds the transaction ID requested by the user and starts it using DFHWBLT, the Web bridge exit.
4. The transaction runs under control of the 3270 Web bridge. CICS intercepts all 3270 terminal input and passes it to the Web bridge exit, which converts the request from HTTP to the form expected by the 3270 transaction. Similarly 3270 output from the user transaction is intercepted and converted by the Web bridge exit into HTML.

Once the 3270 transaction has completed, the final processing is the same as for a called program. Control returns to the alias transaction, which calls the converter for the encode function, and then the output is returned to the Web browser.

Bridge exits

Note several sample bridge exits are available for use with CICS TS V1.3 as follows:

- DFHWBLT** - This provides support for the invocation of CICS 3270 transactions from the Web, (this is the function we refer to as the “3270 Web bridge”).
- DFH0CBRE** - This provides support for the use of TS or TD queues for input or output from 3270 transactions.
- DFH0CBAE** - This provides support for the invocation of a CICS 3270 transaction as a CICS BTS (Business Transaction Services) activity.

CSQCB30 - This is supplied by MQSeries V2.1 and provides support for the invocation of a CICS 3270 transaction from an MQSeries message.

With these pre-written exits it should not be necessary to write a new bridge exit, although this is possible if you so desire.

9.8 CICS CORBA client support

CORBA is a specification created by the Object Management Group (OMG) that provides a language-neutral standard for interoperability between software objects. Client and server objects communicate indirectly through Object Request Brokers (ORBs) which use a “template” of each object’s interface (methods and parameters) that has been created using CORBA’s Interface Definition Language (IDL). IDL provides the language-neutrality. Client and server objects on the same machine communicate via a local ORB. Objects distributed across different interconnected machines require an ORB on each machine. The communication sequence is thus: local object to local ORB, to remote ORB, to remote object. IIOP provides the message formats and protocols used in this CORBA distributed environment.

CICS Transaction Server V1.3 introduces the CICS CORBA client support, which allows a client program or object to directly communicate with a Java program in a remote CICS server using the Internet Inter-Orb Protocol (IIOP).

CICS CORBA client support provides the functionality of a CORBA compliant ORB receiving inbound IIOP requests from a client object and invoking the required method on the server object, which in CICS must be a Java program. The CICS Java program can then link to other CICS programs or access CICS resources. Figure 91 on page 158 illustrates the control flow through CICS IIOP:

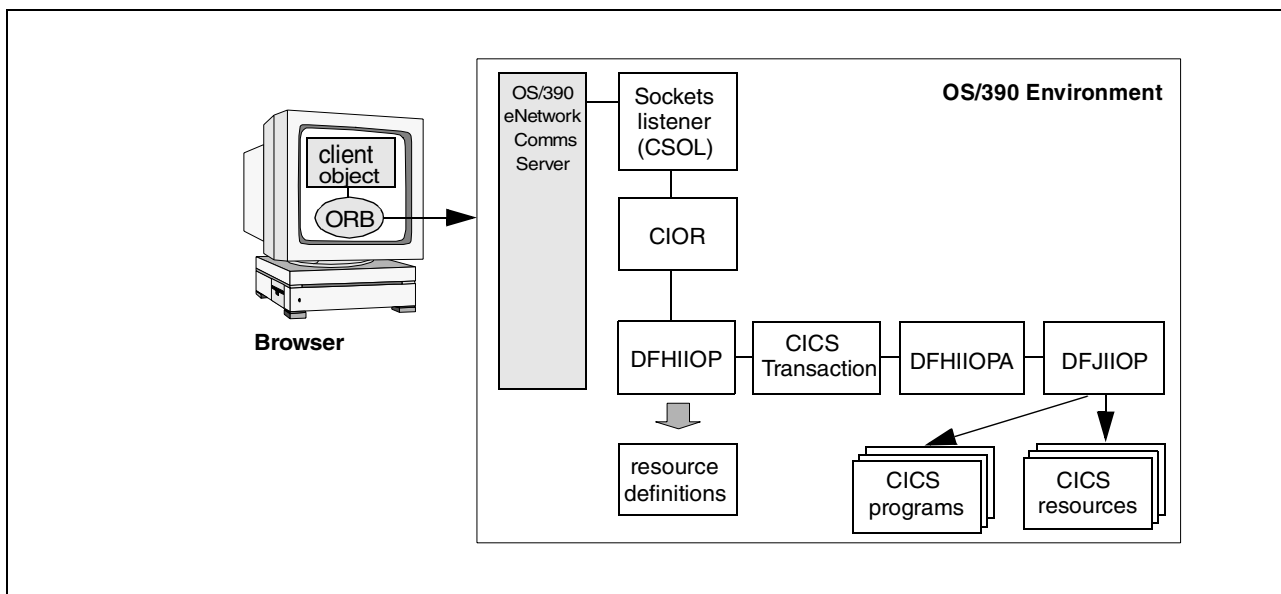


Figure 91. CICS CORBA client support

The control flow when using CORBA client support is as follows:

1. The CICS TCP/IP sockets listener (CSOL) monitors ports reserved for inbound IIOp requests. When a request is received, the listener starts the CICS CIOR transaction; CIOR then calls the CICS/IIOp receiver program, DFHIIOP.
2. The receiver program retrieves the incoming IIOp request and matches its class and method with templates defined by the CICS resource definition, REQUESTMODEL. A CICS transaction name specified in the resource definition must specify DFHIIOPA as the first program to be invoked. DFHIIOPA now issues a CICS LINK to a Java program, DFJIIOP, to handle the request.
3. DFJIIOP analyzes the contents of the IIOp request and instantiates the requested object, demarshals the input parameters, and then invokes the requested method on the target object (which must be a Java program). The Java program can in turn link to other CICS programs or access CICS resources; the Java program is responsible for building the contents of a COMMAREA if this is required by the linked program.
4. DFJIIOP now marshals the reply and returns it to DFHIIOPA for transmission back to the client object that sent the IIOp request.

9.8.0.1 The CICS-provided ORB

Unlike Component Broker which is the IBM full-function CORBA-compliant ORB, the CICS-provided ORB provides only a subset of the full CORBA-specified functions. This is because many of the functions within the CORBA specification are already efficiently provided as functions within CICS. The following are some of the reasons why the CICS-provided ORB is a better solution for CICS/IIOp than a product such as Component Broker:

- Component Broker links to other CICS programs through the EXCI or ECI, which has restrictions in the way that it uses DPL. With the CICS-provided ORB, these restrictions do not apply.
- CICS instantiates objects pseudo-conversationally, so high transaction throughput can be achieved.
- The CICS-provided ORB implementation uses existing and well-proven CICS security mechanisms; this includes security managers such as RACF.
- The server-side application runs in CICS, has access to all CICS facilities (including the 3270 interface), can communicate with other CICS systems, and can access all available data sources.

9.9 NetCICS

NetCICS is an alternative solution to CICS Web support (CWS) and the 3270 Web bridge, and is currently available from IBM as a service offering only and not as a standard product. However, it provides a number of functional enhancements over the standard CICS Web Support with 3270 Web bridge, and as a consequence, is worthy of discussion in this redbook. These enhancements are as follows:

- One of the primary benefits of NetCICS is that it offers a quickly implementable solution to CICS/ESA V4.1 users to access a CICS 3270 environment from the Web without the need to migrate to OS/390 CICS TS.

- Full 3270 terminal support is provided, whereas the standard 3270 Web bridge interface only supports a subset of BMS. For instance, the 3270 Web bridge does not support the ACCUM function or modified data tags.
- A higher degree of HTML customization is provided, allowing the HTML to be modified dynamically within CICS, as opposed to the standard HTML template customization, which permits only user-definable common headers and footers and static modifications to the generated HTML templates.
- A “keyplay” capability provides a one-to-many conversion opportunity, allowing multiple BMS screen interactions to be handled before an HTML page is returned to the browser.

Figure 92 on page 160 illustrates the data flow for an HTTP request to run a 3270 transaction in the backend CICS region.

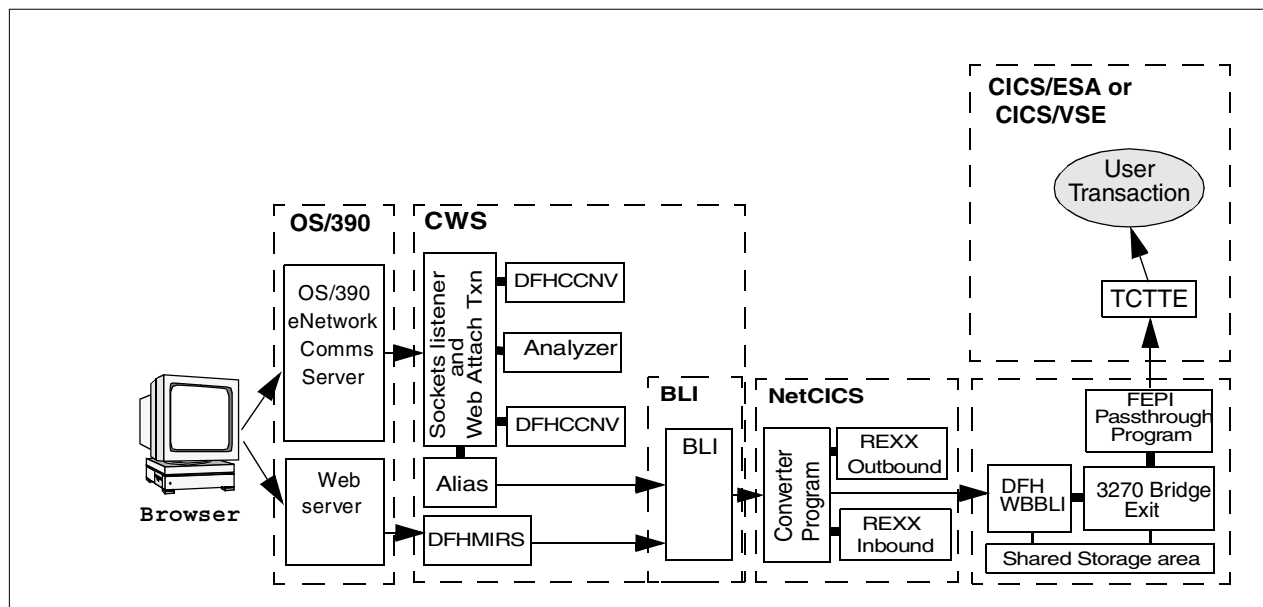


Figure 92. NetCICS

The data flow for the NetCICS implementation is very similar to the data flows described in the previous sections for CICS Web support and the 3270 Web bridge. The differences in the NetCICS implementation are described below:

1. The user sends an HTTP request from the browser with all of the CICS transaction request information imbedded within the URL. This request is passed through TCP/IP and then through either CICS Web support or a Web server. The conversion between the 3270 data stream and HTML is handled by function within CICS Web support or the Web server.
2. Web Attach Processing handles the conversion from ASCII to EBCDIC of the HTTP headers and user data and then starts the Alias transaction (CWBA). CWBA invokes DFHWBA1 which sets up a COMMAREA from the contents of the URL and then links to the Business Logic Interface (BLI) program, DFHWBBLI passing to it the COMMAREA data.
3. Alternatively, requests can come into the Business Logic Interface directly from the Web server and the CICS mirror program, DFHMIRS. In this situation, data conversion is handled by function within the Web server.

4. DFHWBBLI now calls the first component of NetCICS which is the NetCICS converter program (NCSWCONV). If decode/encode was specified in the URL request, then NCSWCONV will call a REXX for CICS EXEC to perform user-specified customization of the contents of the COMMAREA to a format that would normally be expected in the 3270 transaction request. If a request was made to decode the COMMAREA on the inbound request, then the COMMAREA contents will automatically be encoded on the outbound response.
5. The NetCICS converter program now calls DFHWBTTA, the 3270 Web bridge, which calls the Web bridge exit program, and places the contents of the COMMAREA in an area of storage that can be accessed by the Web bridge exit.
6. A Front End Programming Interface (FEPI) pass through program is now invoked by the Web bridge exit. The role of the FEPI program is to look like a 3270 terminal to the user transaction and handle the resultant LU2 flows that would occur as part of a normal transaction request. The FEPI program provides full 3270 terminal support managing the 3270 data stream as though it were receiving or sending it to BMS.

The service provided by IBM includes installing the NetCICS components and configuring CICS accordingly, then performing some initial REXX customization according to the customer requirements. This is an IBM service that would be of benefit to customers who are not familiar with the complexities of installing and configuring CICS Web support and 3270 Web bridge.

9.10 CICS Transaction Gateway

The CICS Transaction Gateway (CTG) is a set of server-based software components that allow a Java program, to invoke programs or transactions running in a CICS region. The Java program can be an applet, a servlet or a custom Java application. We describe the architecture of using the CTG with applets and servlets, but not applications since they have no specific architecture.

The CICS Transaction Gateway is available for production use on OS/390, and on the following distributed platforms: AIX, Sun Solaris, and Windows NT. It is also available for development use on Windows 95 and Windows 98. When the CTG runs on a distributed platform it is supported for use with the following CICS servers:

- OS/390 CICS Transaction Server
- CICS/ESA V4.1
- VSE CICS Transaction Server
- CICS VSE V2.3
- OS/2 Transaction Server
- TXSeries CICS (AIX, HP/UX, Sun Solaris, Windows NT)

The CICS Transaction Gateway consists of the following components:

- Gateway process

This long running process is used to accept CTG request issued from remote Java applets.

- Basic Java methods

These are used to set up connectivity to a CTG gateway process or to invoke the underlying CICS Universal Client or OS/390 EXCI.

- ECI Java methods

These methods provide access to CICS COMMAREA-based programs in a similar fashion to the CICS Universal Client ECI or the OS/390 EXCI.

- JavaBeans

Code written to Sun's JavaBeans specification (beans) support development of applications from a number of visual development environments such as VisualAge for Java.

In addition, the following components are available on non-OS/390 versions of the CTG:

- EPI Java methods

These methods provide Java access to CICS 3270 based transactions in a similar fashion to the CICS Universal Client EPI interface.

- Terminal servlet

This supplied servlet dynamically converts 3270 output into HTML for display on a Web browser.

When the CICS Transaction Gateway for OS/390 is used, it is supported with any OS/390 CICS Transaction Server region. Note, however that only CTG V3.1 is supported with CICS Transaction Server V1.3, and if you wish to use CTG V3.1 with CICS Transaction Server V1.2 the fix for APAR PQ31270 must be applied to CICS Transaction Server. This does not apply to CICS Transaction Server V1.3.

The following sections will discuss the major components of the OS/390 CTG and how the architecture is different when using Java applets as opposed to Java servlets. For further details on using the CTG on distributed platforms refer to *Revealed! CICS Transaction Gateway with More CICS Clients Unmasked*, SG24-5277.

9.10.1 CICS Transaction Gateway applet architecture

Figure 93 on page 163 shows an implementation of the CICS Transaction Gateway applet architecture on OS/390. The request flow from a browser to the CICS server via the CTG would be as follows:

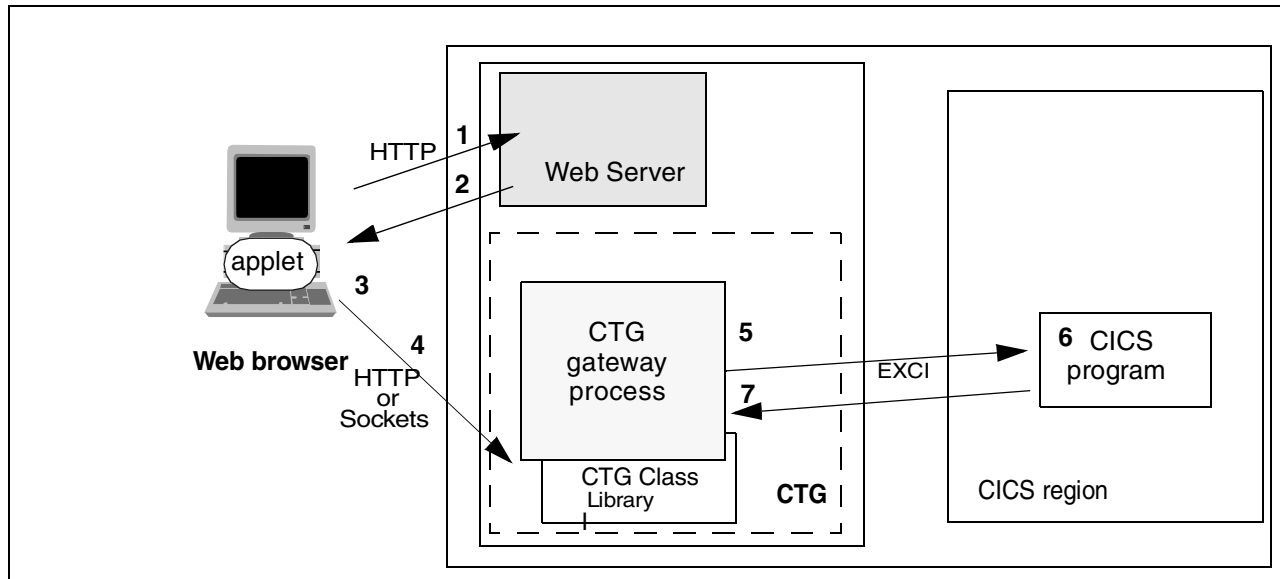


Figure 93. CICS Transaction Gateway applet architecture on OS/390

1. An HTTP request is sent from the Web browser to the URL of an HTML page containing an applet tag. The Web server sends this HTML page to the Web browser.
2. The Web browser finds the applet tag and requests a download of the applet and any associated class files.
3. The applet starts to run within the JVM of the Web browser. First of all the applet uses the `JavaGateway.open()` method to set up a network connection to the gateway process via one of the four supported network protocols (TCP/IP sockets, SSL sockets, HTTP or HTTPS).
4. Next the applet builds an ECI request using the `ECIRequest` class and then flows this to the CTG gateway process using the `JavaGateway.flow()` method.
5. The CTG gateway process receives the request, maps the ECI request to an EXCI request and flows this on to the CICS region.
6. The CICS program is invoked and retrieves the information from the COMMAREA, performs its business logic and then returns further information to the COMMAREA.
7. The response is now returned to the CTG. The CTG prepares the response from the CICS server passed on the ECI return call and then sends this back to the Java applet.

ASCII/EBCDIC data conversion of the CICS COMMAREA is handled by the CICS server data conversion templates (DFHCCNV). The HTTP headers require no data conversion since they are all in ASCII.

9.10.2 CICS Transaction Gateway servlet architecture

Figure 94 on page 164 illustrates the configuration for a Java-enabled Web browser communicating with an OS/390 with CICS Transaction Server region. The request flow from the browser to the CICS server via the CTG is as follows:

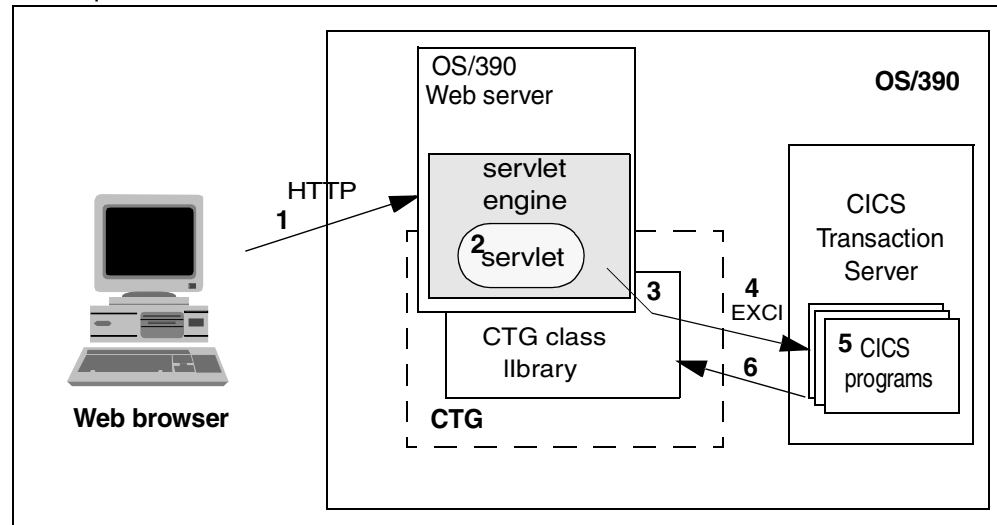


Figure 94. CICS Transaction Gateway servlet architecture on OS/390

1. An HTTP request is sent from the browser to the Web server.
2. The servlet is invoked by the HTTP request either via a URL, a form, or a server-side include. The servlet runs within the JVM of the WebSphere Application Server servlet engine and uses the `JavaGateway.open()` method to create a local connection to the underlying EXCI.
3. The servlet builds an ECI request using the `ECIRequest` class and then flows this to the CICS region using the `JavaGateway.flow()` method.
4. The CTG maps the ECI request to an EXCI request, and using the JNI interface calls the facilities of the CICS External Communications Interface (EXCI) to send the request to the OS/390 CICS Transaction Server using OS/390 XCF or MRO cross-memory or cross-Sysplex transport facilities.
5. The CICS program is invoked and retrieves the information from the COMMAREA, performs its business logic and then returns further information to the COMMAREA.
6. The response is now returned to the CTG via the EXCI. The CTG prepares the response from the CICS server passed on the EXCI return call, and sends this back to the browser.

9.11 Where to find more information

- *CICS Transaction Server for OS/390: Web Interface and 3270 Bridge*, SG24-5243
- *Revealed! Architecting Web Access to CICS* - SG24-5466
- *CICS Internet Guide* - SC34-5445
- *CICS Transaction Server for OS/390 Version 1 Release 3: Web Support and 3270 Bridge*, SG24-5480
- *Java Application Development for CICS*, SG24-5275

- *Revealed! CICS Transaction Gateway with More CICS Clients Unmasked, SG24-5277*

Chapter 10. Solutions for Accessing IMS from the Web

IMS continues to play a critical role in the information technology infrastructure of a number of enterprises around the world. It is evolving into the new network computing environment, allowing IMS customers to Web-enable their applications and make their data available to the world via the Internet.

In this chapter, we will take a look at some of the products and solutions developed to implement Web-to-IMS connectivity. An excellent reference for this material and more information on Web-to-IMS connectivity can be found in *IMS e-business Connect Using the IMS Connectors*, SG24-5427.

10.1 Solution Overview

Making a decision on what solution is best for your environment will depend on many things, including the network connectivity you have and the skill set available to you. You will also want to take into account your existing environment. For example, an enterprise with a Notes environment may want to use the Lotus Domino server and LotusScript programming. An enterprise with a variety of database source types may want to consider Host Publisher, Net.Data or Component Broker for access to multiple data types. Customers who have a need to access IMS data in their intranet may want to consider Host On-Demand.

- **Host On-Demand** provides terminal emulation in a Java environment. Access to IMS and other S/390, AS/400, or UNIX applications can be done using a Web browser as a terminal emulator. Host On-Demand is unique in that it provides a single Web interface to any 3270, 5250, or VT application. Setup is simple and requires no changes to the host applications. For accessing IMS, the appearance to the user is much like a traditional 3270 screen. It is the fastest transition path to make traditional 3270 applications available to Web users. Host On-Demand uses Java applets for host access and requires Web browsers to be Java-enabled with JVM 1.1 capability.
- **Screen Customizer** is a thin, intelligent Java applet that automatically converts host screens into graphical presentations that are easily customizable without any programming. It is based on screen recognition technology which requires no access to source code and generates no scripting or coding. It works as an add-on with IBM SecureWay Host On-Demand and IBM SecureWay Personal Communications.
- **Host Publisher** is very suitable for developing new Web applications that get data from existing 3270-based, 5250-based, VT-based, and JDBC-based applications, providing dynamic HTML pages for display. Access to host applications and databases is provided by Host Publisher applications consisting of Integration Objects. Host Publisher allows you to integrate data from several sources into the Web pages, giving more flexibility in building the data to be presented to the browser. Because the output of the applications is HTML there are no requirements for the client browsers.

Using Host Publisher to access IMS requires no changes to the IMS and the Host Publisher Studio makes application development easy. Using Host Publisher gives you the capability of extending Web access to other host applications.

Though based on the same technology for 3270 application access, Host On-Demand and Host Publisher differ in their capabilities and end user interaction. Host On-Demand provides a terminal front-end to host applications which can present either a traditional or GUI presentation of the host screens. Host Publisher provides a more sophisticated programmable interface, allowing logic, screen recognition, composite applications, and other application enhancements. The two products complement each other in providing a full range of host integration possibilities.

- **VisualAge Smalltalk Enterprise** offers a way to quickly develop Internet and intranet e-business applications. The VisualAge for Smalltalk IMS Connection extends the visual programming paradigm to include the wrapping of existing IMS transactions into business objects. VisualAge Smalltalk Web Connection is a tool for HTML page development and allows applications to quickly publish data to the Web.
- **IMS Web** provides an easy way to provide access to new and existing IMS transactions from the Web. The IMS Web Runtime component is installed as a CGI application on the Web server. Connectivity is simple, using TCP/IP and OTMA. IMS Web Studio provides development capabilities.
- **IMS Client for Java** provides templates for Java development. It uses the TCP/IP OTMA connectivity. Because it requires a Java-enabled browser and the Java applet may have Java-level dependencies, this solution is better suited for intranets/extranets.
- **IMS Object Connector** offers an object-oriented programming environment for accessing IMS. IMS message queues and DL/I segments are wrapped into object-oriented classes and connected to the IMS Transaction Manager and database.
- **Net.Data** provides dynamic Web page creation and interaction with relational and non-relational database management systems (DBMS), such as DB2, ODBC, and IMS databases, and applications written in programming languages such as Java, Perl, C++, and REXX. This is a good solution for enterprises that have multiple backend data sources.
- **Lotus Domino** environments have the ability to access IMS through MQSeries by using the MQLSX or MQEI LotusScript extensions.
- **Encina DE-Light** by Transarc provides DCE/RPC access to IMS through the AS/IMS interface. Encina DE-Light provides a Java applet to be downloaded to the Web browser that uses RPC calls to a gateway on the Web server to access IMS data on the host.
- **Component Broker** provides a development toolkit and run time environment for distributed three-tier applications, in which objects developed in different languages communicate with each other by way of the standard CORBA Object Request Broker and have a persistent image of traditional database and transaction environments through application adaptors. IBM Component Broker Connector (CBConnector) provides application adaptors to access CICS, IMS, and DB2. More information on Component Broker can be seen in "Component Broker" on page 124.

10.2 Connectivity

IMS has several options available for communication to the outside world. These are:

- TCP/IP sockets
- Advanced Program-to-Program Communication (APPC)
- MQSeries
- Distributed computing environment remote procedure calls (DCE/RPC)

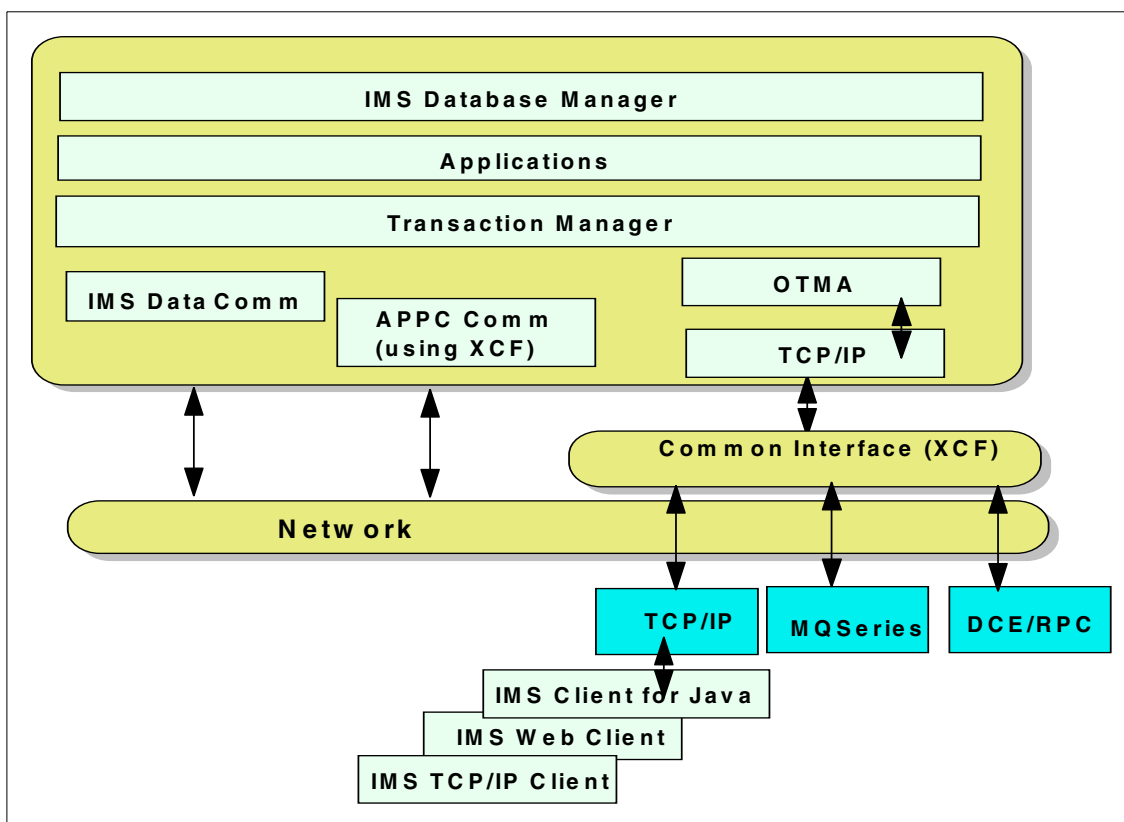


Figure 95. Web-to-IMS Access

10.2.1 Connecting to IMS with TCP/IP

We can divide IMS connectivity in a TCP/IP environment into four categories:

- Telnet 3270
- IMS TCP/IP Socket Listener
- IMS TCP/IP OTMA Connection and IMS Connect
- IMS TCP/IP OTMA Connector for Java and IMS Connector for Java

10.2.1.1 Telnet 3270 Support

The telnet protocol provides a standardized interface that allows terminal devices and terminal-oriented processes on hosts that support TCP/IP to communicate with each other. Using this protocol, IMS applications that currently use MFS, only require the use of telnet, which provides 3270 emulation services for TCP/IP-connected clients. Telnet is a part of the base TCP/IP for MVS product.

Telnet is usually used from a TCP/IP workstation that wants to access 3270 applications using a 3270 emulation window. It allows a user to log on to a remote or local TCP/IP host with a Telnet 3270 server and establish an SNA LU2 session with IMS. Security can be based on user ID and password.

Host On-Demand and the Host Publisher Host Access Integration Objects use this technique to establish a connection from a Web browser to a host.

10.2.1.2 IMS TCP/IP Feature

IMS TCP/IP, an optional feature of the TCP/IP for MVS product, provides an access path between TCP/IP-communicating devices and IMS Transaction Manager. It is the only solution for IMS versions earlier than IMS/ESA Version 5. Since IMS/ESA Version 5, the IMS TCP/IP OTMA Connection is recommended to provide communication between remote workstations and IMS Transaction Manager systems.

There are several restrictions when using the IMS TCP/IP feature:

- Transactions must be defined as MODE=SNGL in the IMS TRANSACT macro to ensure that the database buffers are emptied (flushed) to DASD when the second and subsequent GU calls are issued.
- Transactions must not reference other systems. Multiple systems coupling (MSC) is not supported.
- Transactions must not be conversational (that is, they must not use the IMS scratch pad area (SPA)).
- Security cannot be implemented through RACF (or any equivalent security product). The user ID available in the IMS transaction is the user ID of the IMS Listener. Thus security cannot be based on the requester's identity.
- To use the Assist module (implicit-mode support), the IMS application code has to be changed to issue xxxADLI calls instead of xxxTDLI calls.

Some of the above restrictions have been removed with the IMS TCP/IP OTMA Connection support.

10.2.1.3 Open Transaction Manager Access (OTMA)

Open Transaction Manager Access (OTMA) is a function of IMS that was introduced with IMS/ESA Version 5. OTMA is a transaction-based connectionless client/server protocol that provides an access path and an interface specification for sending and receiving transactions and data from IMS.

OTMA provides the facility for IMS to communicate very efficiently with MVS applications other than VTAM. These MVS applications are called OTMA clients. The MVS application can be user written, an independent software vendor's application or an IBM application. Examples of the OTMA client applications and the corresponding network client applications are shown in Figure 96.

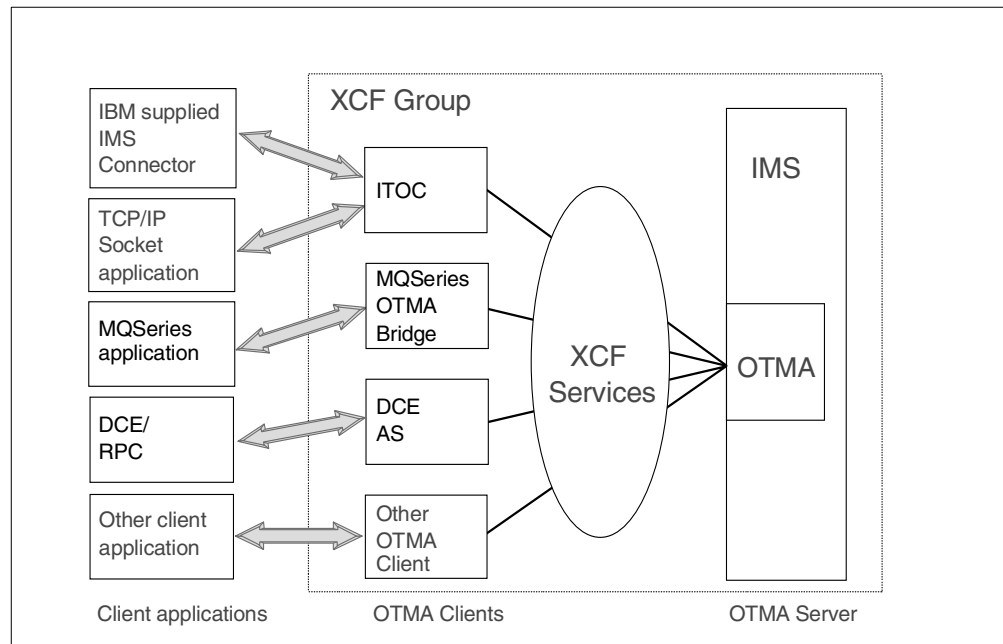


Figure 96. Examples of the different OTMA connections

One commonly used OTMA client is IMS TCP/IP Connection (IMS TOC). Most of the IBM supplied IMS e-business Connectors, such as IMS TOC Connector for Java, IMS Client for Java, and IMS Web, are based on the use of IMS TOC.

The OTMA client handles all device dependencies for a particular network protocol, which can be TCP/IP or SNA. IMS Transaction Manager can operate without needing any device-characteristic information. The majority of IMS message processing options such as nonresponse mode and response mode transactions, conversational processing, Fast Path transactions, Multiple Systems Coupling (MSC) processing, and IMS commands can be implemented with OTMA. OTMA does not support Message Format Services (MFS), although MODname can be provided within the messages.

IMS TCP/IP OTMA Connection (ITOC)

The IMS TCP/IP Open Transaction Manager Access Connection (IMS TOC or ITOC) provides communication links between TCP/IP clients and IMS Transaction Manager subsystems. IMS TOC supports multiple TCP/IP clients accessing multiple IMS resources. It runs on an MVS or OS/390 platform, using IMS/ESA Version 5 or Version 6.

IMS TOC consists of three core components:

- The TCP/IP Communication Component (CCC), which processes communications between TCP/IP clients and the ITOC
- The Datastore Communication Component (DCC), which handles communications between the ITOC and IMS
- The Command Component (CMD), which processes commands received from the MVS console operator

In addition to these core components, IMS TOC uses a communication driver facility that isolates the core components from the communication software. A

TCP/IP driver communicates with the IMS Web server and the client workstations. An IMS OTMA driver communicates with IMS.

IMS TOC is part of the IMS Web support, which is a set of functions on top of IMS/ESA Version 5 and 6 that can be downloaded from the IMS Internet home page. The ITOC architecture is designed to support IMS Web, as well as TCP/IP clients communicating with socket calls.

IMS TOC hardware and software requirements:

- Host processor capable of running IMS Version 5 or 6
- MVS Version 4.2 or later
- IMS/ESA Transaction Manager Version 5 or 6
- TCP/IP for MVS Version 3.2 or 3.4 and APAR PQ13154
- Resource Access Control Facility (RACF) Version 1.9.2 or an equivalent product

For more information, see:

- The IMS Internet home page at <http://www.software.ibm.com/data/ims>
- *IMS TCP/IP OTMA Connection Version 2.1 User's Guide*, found at <http://www.software.ibm.com/data/ims/about/imstoc/document/>

IMS Connect

IMS Connect is available on IMS Version 7 only. IMS Connect replaces IMS TOC and provides enhanced TCP/IP access to IMS. It also provides enhancements for SMP installability, usability, and performance. Future enhancements of IMS TCP/IP support will only be provided through the IMS V7 Connect feature.

IMS TCP/IP OTMA Connector for Java (JITOC)

The JITOC is a collection of Java beans, or objects, that enable a Java application to access IMS transactions using TCP/IP OTMA Connection. It is the Java communications side of the interface. JITOC is available for IMS Version 6 and earlier.

The JITOC provides a Common Connector Framework-compliant Java interface to IMS TOC. A Java application to access IMS is constructed using visual programming techniques to create an Enterprise Access Builder (EAB) command. Once the EAB command composite bean has been created and compiled into Java bytecode, it can be run from any platform that supports the Java Virtual Machine run time environment.

In conjunction with the VisualAge for Java development environment, IMS TOC Connector for Java (JITOC) enables you to rapidly develop Java applications that run your IMS transactions. With additional support from IBM WebSphere Studio and WebSphere Application Server, you can build and run Java servlets that access your IMS transactions from the Web.

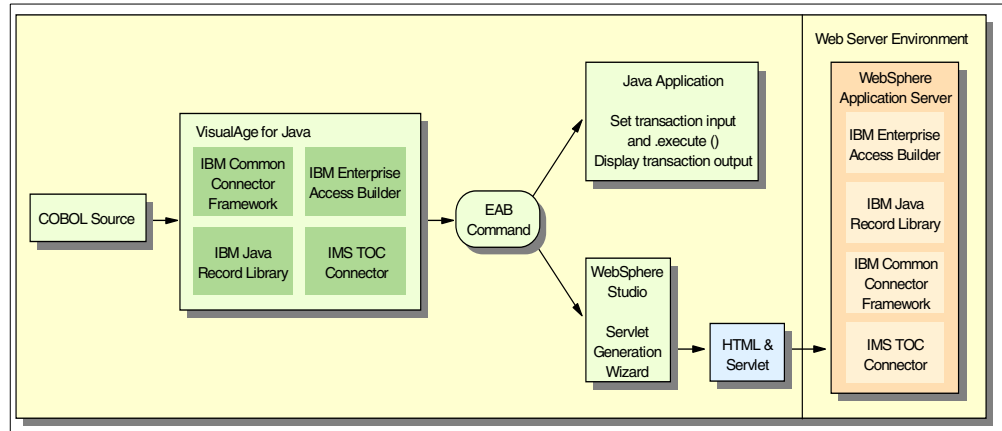


Figure 97. Developing Java applications and servlets using IMS TOC Connector

Requirements for the target IMS host machine are ITOC Version 2.1.3 or higher and IMS/ESA Version 5.1 or Version 6. Requirements for the Java development environment are the IBM enhanced JDK 1.1.6 and VisualAge for Java Enterprise Edition Version 2.1 or higher. The requirement for executing Java applications on your local workstation is a Java run time environment (JRE) 1.1.6B. The JRE is packaged with JDK 1.1.6. Requirements for creating and executing Java servlets are WebSphere Studio Version 1.0 and WebSphere Application Server Advanced Version 2.0 or higher.

JITOC is currently available from the VisualAge Developer Domain to registered owners of VisualAge for Java, Enterprise Edition, Version 2.0 for Windows NT.

More information can be found in the *IMS TOC Connector for Java* guide found at <http://www.software.ibm.com/data/ims/jitoc.html>

IMS Connector for Java

IMS Connector for Java provides similar functionality as the IMS TOC Connector for Java. IMS Connector for Java is only available on IMS Version 7, replacing IMS TOC Connector for Java (JITOC).

10.2.2 APPC Interface to IMS

Web Server programs can access IMS through APPC/IMS support. APPC is an implementation of the Systems Network Architecture (SNA).

10.2.3 MQSeries

IBM MQSeries provides communications between applications running across the Internet or a private network. Among the benefits that MQSeries brings are assured message delivery, time-independent (offline) communications, synchronization of messages into logical units of work, and connectivity between applications in many different system environments connected over IP or SNA networks.

All applications using the Message Queue Interface (MQI) can communicate with one another over the network to each other. If you implement a Web server program using MQI and implement the MQSeries connection in MVS as a client of OTMA for IMS, you can enable a Web browser or Java applet to access IMS data or transactions via MQSeries or MQSeries Client for Java.

10.3 Host Publisher

IBM SecureWay Host Publisher V2.1 provides access to 3270, 5250, VT, and JDBC backend data sources. Host Publisher applications run on the server, not as Java applets. The information is served to the clients as HTML. Because of this, there are no specific requirements placed on the client Web browsers, and therefore, it is appropriate for Internet use as well as intranet and extranet.

Host Publisher applications consist of Integration Objects that are actually Java beans. The Host Publisher studio is provided to make building the Integration Objects easy. Once the Integration Objects are built, Host Publisher Studio will guide you through building the application in the form of Java Server Pages (JSPs) that invoke the Integration Objects. Developing and deploying the applications is a simple and fast process.

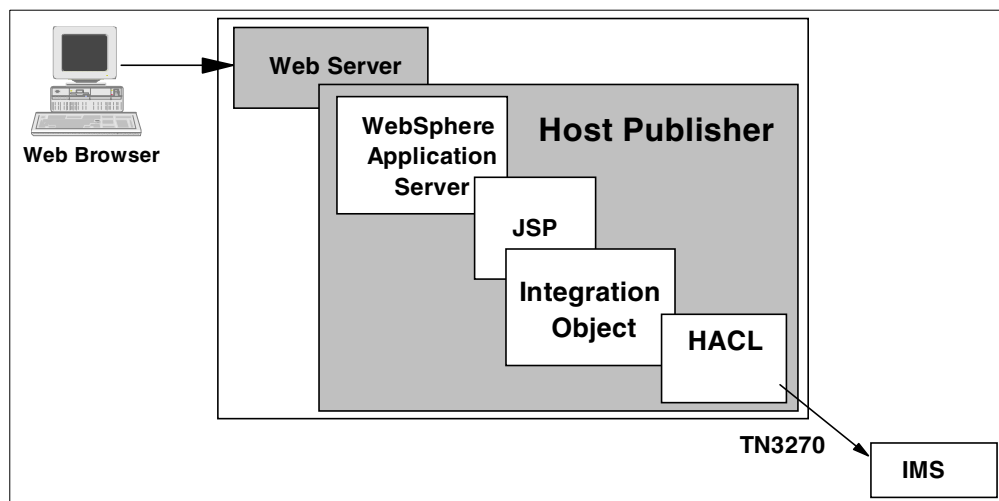


Figure 98. Using Host Publisher to access IMS

Simply put, Host Publisher provides you with host access capabilities without the need to build Java applications and applets yourself. If you want to take it one step further and enhance the applications built by Host Publisher, for instance by adding graphics to the pages, the application source is available for you to do so.

The Integration Objects can be used not only in the JSPs built by the Host Publisher Studio, but in any Java application, applet, or servlet. Integration objects built to access 3270, 5250, and VT data sources are called Host Access Integration Objects.

WebSphere Application Server Standard Edition is provided as an integral part of the Host Publisher server to support the Host Publisher applications. Host Publisher also ships with the IBM SecureWay Network Dispatcher for load balancing and failover capability among the Host Publisher servers.

More information about Host Publisher can be found in Chapter 3, "IBM SecureWay Host Publisher" on page 41. A brief overview of using Host Publisher to implement a CICS transaction can be seen in 3.6, "Creating host Integration Objects" on page 50. This same method would apply to accessing IMS.

10.4 Host On-Demand

Through the capabilities of a Host On-Demand (HOD) environment, Web users can now access IMS and other S/390, AS/400, and UNIX host functions using the emulation properties of the Host On-Demand clients to manipulate the application 3270, 5250, or VT data stream. In addition, Host On-Demand provides JDBC access to DB2 databases on AS/400 systems with Database On-Demand and access to CICS data with the CICS Gateway applet. We will quickly go over Host On-Demand as it relates to accessing IMS data. Host On-Demand is covered in more detail in Chapter 2, "IBM SecureWay Host On-Demand" on page 9.

For general use, Web access to IMS through HOD is recommended for intranet or extranet solutions for the following reasons:

1. Host On-Demand delivers to the browser a full function terminal emulator, which implies that any user accessing host functions through this method would need to understand the interface of the applications they are accessing. This is unlikely to be something that the general Internet user would be able to do. (The use of Screen Customizer can simplify this interface for novice users.)
2. It will require a Java enabled browser, and sometimes running specific levels of Java, to support the applets you are downloading from the Host On-Demand server. This is easy to control on the intranet browser and also for terminals of extranet users (such as business partners) but not so easy to control on the terminals of general Internet users.

This does not exclude access through the Internet if you are using the Internet as an external network connection environment for trained or trusted users who can use this path for logging on to your host systems. For example, you may have a need for occasional Web access to your IMS systems environment for IMS system programmers, database administrators or operations staff from home or while traveling.

10.4.1 Accessing IMS using HOD 3270 emulation

Host On-Demand terminal emulation functions are accomplished from a Java applet executing from a Java-enabled Web browser. There are three methods of getting the applet on the client machine:

- The applet can be downloaded each time the user accesses the HOD URL.
- It can be cached so it is only downloaded the first time the user accesses it and then only when updates have been made.
- It can be installed locally.

The last two options are ideal for situations where downloading the applet over the network is slow. HOD also provides an express client/server option and a Function On-Demand option to address slower networks.

Users connect to the OS/390 host using a Web browser as a 3270 emulator to access IMS. A TN3270 server is required for the connection to the host.

Host On-Demand server is available on:

- Windows NT 4.0 with SP3 or later
- AIX Version 4.2 or later
- OS/2 Warp Version 4 and Warp Server

- Novell NetWare Version 4.x
- Sun SolarisOS Release 2.5.1
- HP/UX 10.20
- RedHat Linux Version 5 Release 2
- OS/400 Version 4 Release 2 or later
- OS/390 Version 2 Release 1 or later

Figure 99 illustrates a simple configuration using HOD to access IMS on an OS/390 from a Web browser. This configuration can be varied by putting the telnet server and/or Web server/HOD server combination on other machines and other platforms.

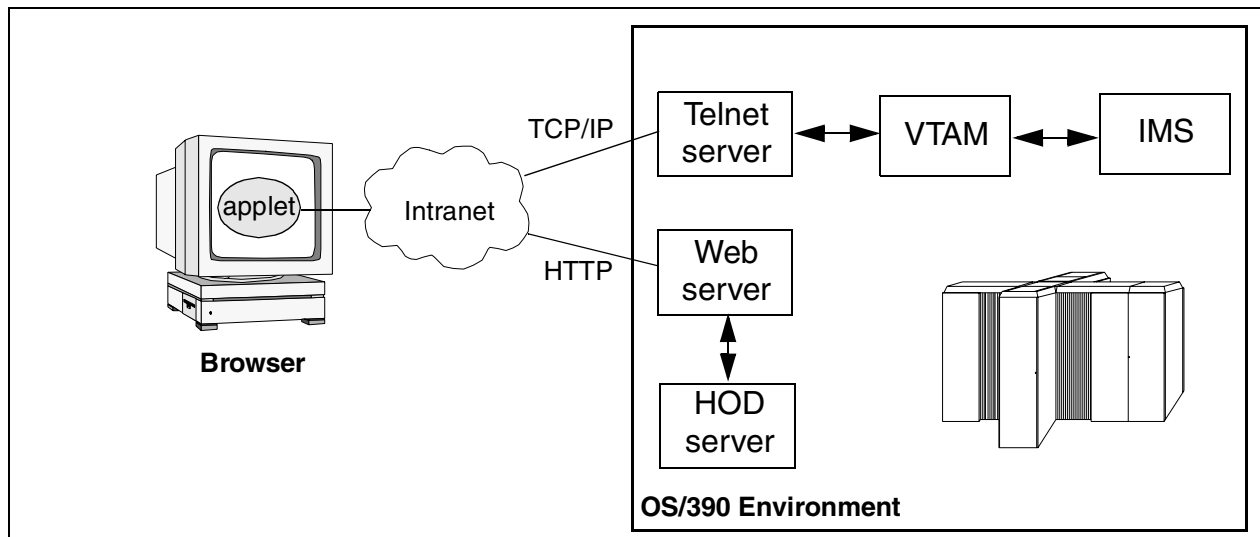


Figure 99. Host On-Demand - direct connection

In Figure 99, the Web browser enters the URL of the Host On-Demand server. Host On-Demand initiates the download of the Java applet if necessary. The Java applet, executing on the client machine, establishes a TN3270 session to the desired host by using a telnet server. The user now has the terminal emulation function on the Web browser and proceeds to log on to the host.

Screen Customizer/LE and the full Screen Customizer product can enhance the presentation by converting the “green screen” format into a graphical user interface (GUI). More information about Screen Customizer can be found in Chapter 4, “IBM SecureWay Screen Customizer” on page 69.

10.5 IMS Web

IMS Web provides an easy way to enable new and existing IMS transactions, both conversational and non-conversational, from your Internet and intranet sites while maintaining high performance and robust application development function, including a GUI code generator. IMS Web allows connections to IMS transactions from a variety of platforms, including Windows, OS/2, AIX, and Solaris.

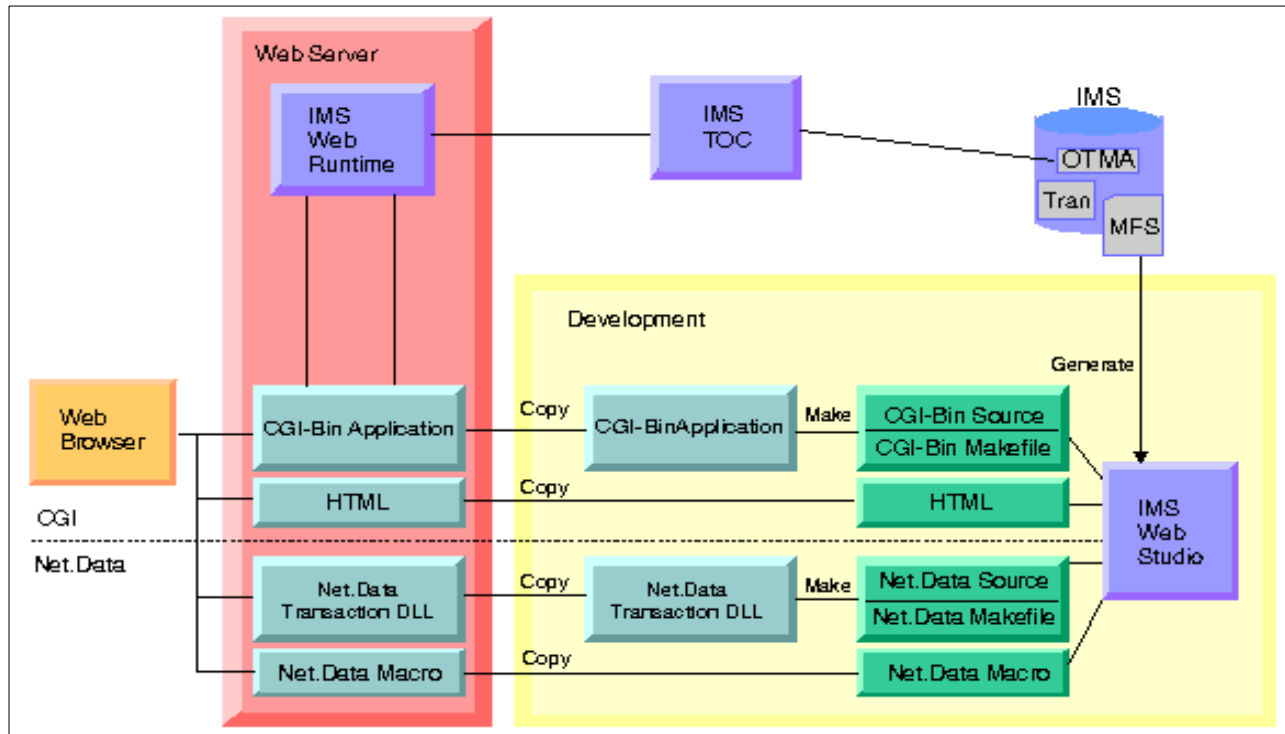


Figure 100. IMS Web

Features

IMS Web consists of the following components:

1. IMS Web Development

- **IMS Web Studio:** A GUI-based development tool (Java application that runs on JVM 1.1.4 or higher) which allows users to:
 - Select and download MFS (Message Format Service - IMS/DC Screen Mapper) source files for IMS transactions and parse them.
 - Generate source code and a platform-specific Makefile for a CGI-BIN executable program. The program uses the C++ classes to submit the transaction to IMS and dynamically generates HTML pages to display the output from the IMS transaction and to enter new data.
 - Generate source code for a Net.Data transaction DLL and macro for Net.Data-driven applications from the MFS source files. If Net.Data is used to submit IMS transactions, the IMS Web Runtime component is a prerequisite for Net.Data.

IMS Web Studio generates files in two directories, cgi and dtw, under the project directory. The cgi subdirectory contains the files needed for IMS Web-driven transactions. These files contain:

- C++ classes and source code for a CGI-BIN program that runs an IMS transaction from a workstation and dynamically creates an output HTML form to display the results of that transaction.
- Input HTML form for submitting transaction input data and for running the transaction CGI-BIN program.

The dtw subdirectory contains the files needed for the IMS Web backend of a Net.Data-driven transaction. These files contain:

- C++ classes and code for a transaction DLL that runs an IMS transaction from a workstation.
- Input and output HTML for the transaction packaged in a Net.Data macro.
- IMS Web Library: Contains C++ header and library files required to compile and link the generated C++ files for the IMS Web-driven CGI-BIN program or Net.Data-driven transaction DLL.

IMS Web Development supports the following platforms:

- Windows NT
- OS/2
- AIX
- OS/390 (IMS Web library only)
- Sun Solaris

2. IMS Web Runtime

The IMS Web Runtime component consists of libraries (which are installed on the Web server and are invoked by the generated CGI-BIN program or transaction DLL), conversion tables for the IMS Web supported languages, and an executable file, HWSCMD.EXE.

The conversion tables are used to convert text data in IMS messages so that it is displayed properly on all supported platforms, in all supported languages. HWSCMD.EXE is the command line executable file that is used to start and stop tracing of activity by the IMS Web Runtime component, as well as display and terminate IMS Web conversational transactions.

The IMS Web Runtime component supports the following platforms:

- Windows NT
- OS/2
- AIX
- OS/390
- Sun Solaris

The Runtime component libraries on the platforms include:

- DLLs: Windows NT, OS/2, OS/390
- Shared Libraries: AIX and Sun

IMS Web Runtime provides a set of functions that are called by the IMS Web Studio generated code.

The IMS Web Runtime component functions include:

- Communication to IMS TCP/IP OTMA Connection (IMS TOC) via TCP/IP.
- MFS-like message formatting and platform data conversion.
- TCP/IP listener program that accepts requests from servers and routes the messages to OTMA.

Requirements

- IMS Web Development component requires one of the following:
 - Windows NT Version 4.0 or later, running on an Intel processor
 - OS/2 Warp 4
 - AIX 4.1.4 or later
 - Sun Solaris 5.5.1 or later
 - OS/390 1.3 or later

- The workstation on which the IMS Web Development component is installed must have an appropriate C++ compiler:
 - For NT, Microsoft Visual C++ 4.2 or later
 - For AIX, IBM C Set++ for AIX
 - For OS/2, IBM VisualAge C++ for OS/2
 - For Sun, SPAR Compiler C++
 - For OS/390, IBM C/C++ Compiler for MVS/ESA
- IMS Web Studio requires:
 - The Java Development Kit (JDK) or Java Runtime Environment (JRE) 1.1.4 or later on any platform, except on Sun, which requires release 1.1.6 or later. To download the JDK or JRE, go to the following URL:
<http://www.javasoft.com/products>
- IMS Web Runtime component requires one of the following (must be installed on a Web server):
 - Windows NT Version 4.0 or later, running on an Intel processor
 - OS/2 Warp 4
 - AIX 4.1.4
 - Sun Solaris 5.5.1
 - OS/390 1.3
- Web server (IMS Web has been tested with the following):
 - IBM's Internet Connection Secure Server on Windows NT, OS/2, AIX, Sun Solaris, or OS/390
 - IBM HTTP Server (Domino Go servers) on Windows NT, AIX, OS/390, and Sun
 - Microsoft Internet Information Server on Windows NT
 - Netscape FastTrack Server on Windows NT, AIX, and Sun
- User created projects created by IMS Web Studio

Availability

IMS Web packages can be downloaded without charge from:

<http://www.software.ibm.com/data/ims/about/imsweb/download/>

Skills needed

- C/C++ programming
- Object-oriented design
- IMS conversational and non-conversational transactions
- Web programming and handling

Where to find more information:

- *Connecting the World Wide Web: A practical guide to IMS Connectivity*
SG24-2220
- <http://www.software.ibm.com/data/ims/about/imsweb>

10.6 IMS WWW Templates

The IMS WWW templates, also known as IMS Web Templates or (IWT), can be used to access IMS/ESA Version 4, Version 5, or Version 6 subsystems. They can be regarded as a gateway application that allows you to call IMS transactions from Web browsers through an Internet server.

Sample CGI and OS/390 Internet Connection API (ICAPI) programs are provided to communicate with IMS transactions. The requested IMS transaction information is provided, mapped to HTML, and returned to the Web browser for display or manipulation. The programs are written in C and call APPC routines to communicate with IMS, thus providing a flexible, interactive, portable interface into IMS.

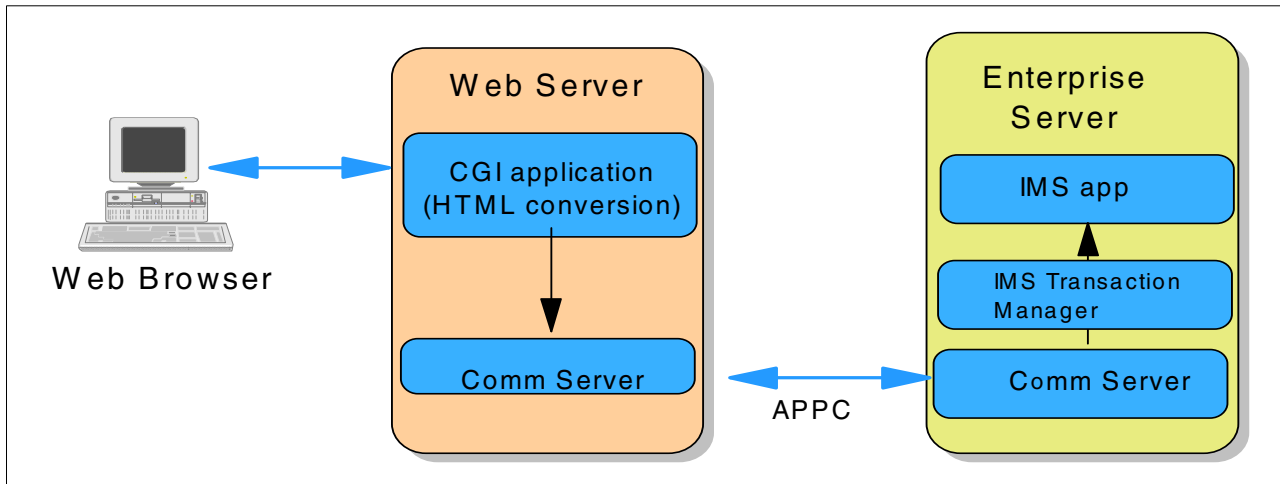


Figure 101. IMS WWW Templates

Features

- Communication with existing IMS transactions through the ICSS or IBM HTTP Server on OS/390.
- Request IMS transactions from an HTML page using a Web browser.
- The Web server can be directed to format the IMS requests and results in several ways:
 - By using libraries shipped with the product, to write custom formatting routines in C and have the OS/390 Web server call these routines for formatting the input (from browser to Web server to IMS), the output (opposite direction), or both.
 - By using an easy-to-understand macro language, the IWT Macro Language (IWTML) included in this package, to format the input, the output or both.
 - By combining the above techniques. For example, it's possible to format the input using a routine and format the output using a macro written in IWTML, or the other way around.
- Formatting can be done in CGI mode or in ICAPI/GWAPI mode.
- Sample programs are provided that can be used as canned solutions or as models to assist preparation of custom CGI or ICAPI programs for accessing IMS applications and data, mapping the results into an attractive page for users.
- Uses APPC communication to get to the IMS Transaction Manager (IMS TM).

Skill requirements

- Knowledge of OS/390 UNIX Systems Services is needed for installation of the IWT package.

- Knowledge of the IMS transactions is needed in order to know how to format the various fields sent to the transaction and how to retrieve data from the IMS output.
- Programming skills are not needed to use IWT. The IWT package includes an interpreted macro-language called IWTML that can be used for input/output formatting (IOF).
- HTML skills to produce HTML pages for IWTML templates. The IWTML statements are used to create, on the fly, the variable parts of the pages.

Availability

The IMS WWW Template package can be downloaded free at URL:

<http://www.s390.ibm.com/nc/sntc/IMS.html>

The tar files contain the source and HTML documentation of the IWT package, as well as the necessary Makefile and jobs for integrating into an MVS system. The package includes:

- Support for building the IMS input and parsing the IMS output in a macro-language called IWTML
- Significant macro-language extensions
- Support for conversational transactions
- APPC linkage to IMS

This package (and the code) supports both DLL/LoadModule connections between modules and the parsing routines that the customer can write. For simplicity (and limited test time), the Makefile only supports the DLL version in this package.

Requirements

The following software packages need to be installed and configured:

- OS/390 V1R3 UNIX System Services or later
- TCP/IP for S/390
- Web server (ICSS or IBM HTTP Server)
- IMS Version 4.1 or later (IWT needs an IMS version which supports APPC) with the sample PART transaction
- APPC communication
- Design IWT applications using libraries shipped with the product and/or built-in macro language

Where to find more information:

- *Connecting IMS to the World Wide Web: A Practical Guide to IMS Connectivity*, SG24-2220
- *How to Secure the Internet Connection Server for MVS/ESA*, SG24-4803
- IMS Connectors: <http://www.software.ibm.com/data/ims/about/imswwwc/>
- IMS Web Templates package: <http://www.s390.ibm.com/nc/sntc/IMS.htm>

For a comprehensive description of how IWT works, refer to:

<http://www.s390.ibm.com/nc/sntc/iwtdoc/html/index.html>

10.7 IMS Client for Java

IMS Client for Java is part of the IBM Application Framework for e-business, which provides a clear and comprehensive model for building and deploying e-business applications that link people, information, and business processes on the Web. The IMS Client for Java provides templates for preparing a Java program to access IMS applications and data running on S/390.

Features

- IMS Client for Java consists of the following:
 - Application and applet sample code to allow Java access from a workstation, S/390, or other JVM environment, to IMS applications and data. This code uses the IMS TOC to access IMS.
 - A user exit routine to translate messages into the format required by the IMS Open Transaction Manager Access (OTMA) interface.
- IMS Client for Java supports the following JVM environments:
 - Windows
 - OS/2
 - AIX
 - OS/400
 - OS/390
 - Sun Solaris
- The Java application accesses IMS applications and sends the information back to the Java machine. Additionally, a Java applet is provided for download from an OS/390 or workstation environment into a Web browser for direct access to IMS on OS/390. The Java applet can also be loaded locally.

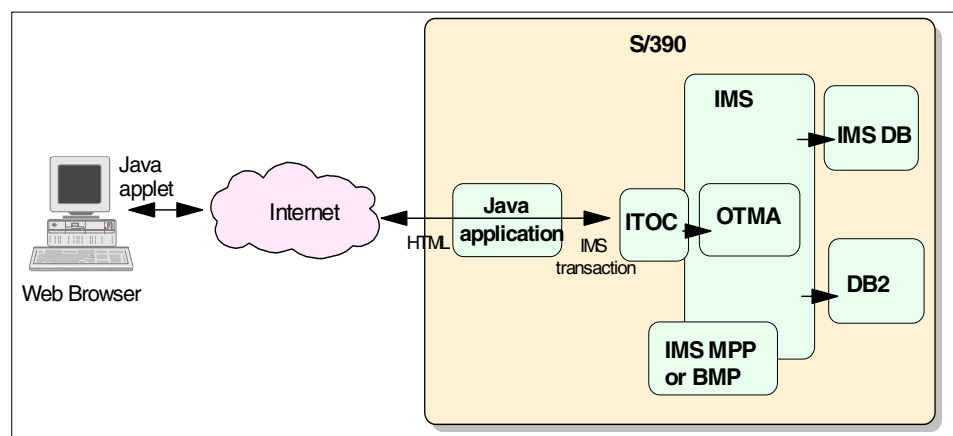


Figure 102. IMS Client for Java

Requirements

- OS/390 V1R3 UNIX System Services or later
- IMS TCP/IP OTMA Connection (IMS TOC)
- Web server
- IMS Version 5 or later
- APPC communication
- Web browser that supports Java V1.1

Availability:

IMS Client for Java is shipped at no charge with IMS TCP/IP Open Transaction Manager Access (OTMA) Connection 2.1.3. The whole package can be downloaded after registration from two sites:

1. <ftp://ps.software.ibm.com/ps/products/imsweb/r210/HWSMH210.exe>

The HWSMH210.exe is a self-extracting package, containing:

- HTML and .txt files, containing installation and configuration directions and the recent changes made to the product.
- Sequential .bin files to be downloaded into OS/390.
- Self-extracting file, JAVASAMPLE.exe.

2. <http://www.software.ibm.com/data/ims/about/imstoc/download>

Skills Needed:

- TCP/IP for OS/390 and IMS V5 knowledge for installation
- IMS application programming
- Java programming for building Java applets or applications

Where to find more information:

- *Connecting IMS to the World Wide Web: A practical Guide to IMS Connectivity*, SG24-2220
- IMS Connectors: <http://www.software.ibm.com/data/ims/imswwwc.html>
- IBM IMS TOC home page: <http://www.software.ibm.com/data/ims/imstoc.html>
- Online documentation shipped with downloadable self-extracting file:
<ftp://ps.software.ibm.com/ps/products/imsweb/r210/HWSMH210.exe>

10.8 IMS Object Connector

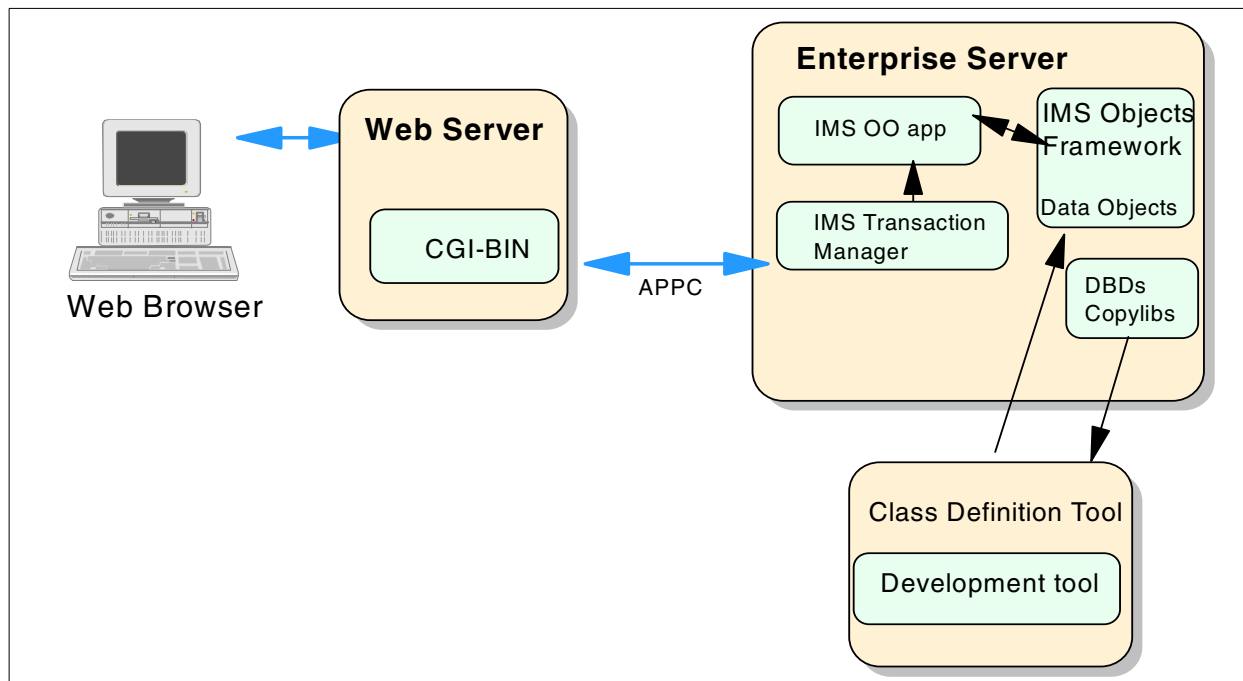


Figure 103. Overview of IMS Object Connector

IMS Object Connector provides a very convenient way to access IMS data and transactions by wrapping IMS I/O message queues, alternate message queues, and DL/I segments into C++ object-oriented classes (IMS Objects Framework) and connecting these objects to IMS TM and IMS DB.

IMS Object Connector components:

- Windows NT-based Class Definition Tool (CDT)
- IMS Objects Framework (OS/390 IMS OO run time environment)
- Internet Web server plug-in

Features

- Offers improved application programming productivity by supporting OO programming and object oriented (OO) programming tools.
- Eliminates DL/I coding.
- Windows NT-based Class Definition Tool (CDT) uses the database information from your existing IMS DBDs and COBOL copylib files to help you generate C++ data object classes in a project directory. You then upload the class files to your host OS/390 system for use with new IMS OO applications that you write using the IMS Object Connector programming model.
- The IMS Objects Framework provides run time support for IMS objects. Using these IMS objects, including CDT-generated data objects, the IMS OO application programs can access your IMS messages and IMS DB data with calls, such as transaction message calls (retrieve and write) and persistent database method calls (retrieve, update, insert, and delete). Application programs can also perform get and set method calls for each data object attribute.

Requirements

- Workstation:
 - Windows NT 4.0 for the CDT
 - IBM Database 2 Version 2 for Windows 95 and Windows NT or higher
 - Sun Microsystems JRE Version 1.1.4 (or higher) or JDK Version 1.1.4 (or higher)
- Host:
 - OS/390 Version 1 Release 2 or higher and IMS Version 5.1 or higher
 - OS/390 Version 1 Release 7 Language Environment or higher
 - C/C++ Compiler 3.2 or higher
- Internet support:
 - TCP/IP Version 3 Release 2 or higher
 - Web server program: IBM Internet Connection Secure Server (ICSS) Version 2.1 or higher with IBM HTTP Server
- User-defined IMS data object classes (using the CDT)
- User-written IMS OO applications

Availability

IMS Object Connector Version 1 is now available as a free download from the Internet, accessible through the IMS Object Connector home page at:

<http://www.software.ibm.com/data/ims/ioc.html>

Skills Needed

- Knowledge of the IMS databases
- Object-oriented programming

- C/C++ programming language

Where to find more information:

- *OS/390 C/C++ User's Guide*, SC09-2361
- *OS/390 C/C++ Programming Guide*, SC09-2362
- IMS Object Connector home page:
<http://www.software.ibm.com/data/ims/ioc.html>

10.9 Net.Data

Net.Data is an IBM software product that allows Web developers to build Web applications using Net.Data Web macros. Net.Data provides connectivity to a variety of data sources and applications including DB2, Oracle, Sybase and DRDA-enabled data sources and across a wide range of platforms, flat files, registries, system services, and IMS databases through IMS transactions.

How it works

The end user enters a URL into the Web browser and it is sent to the Web server. When the Web server receives a URL that refers to Net.Data, the Web server starts an instance of Net.Data and passes the information to it, including the name of the Web macro it needs. Net.Data reads and parses through the Web macro and interprets the statements. When a Net.Data %FUNCTION statement is encountered in the macro, it loads the requested language environment library and passes the information to the specified language environment for processing. The language environment returns the results to Net.Data. After these steps are complete, HTML text is what remains. The HTML text is sent to the Web server and Net.Data terminates. The HTML is then passed to the Web browser.

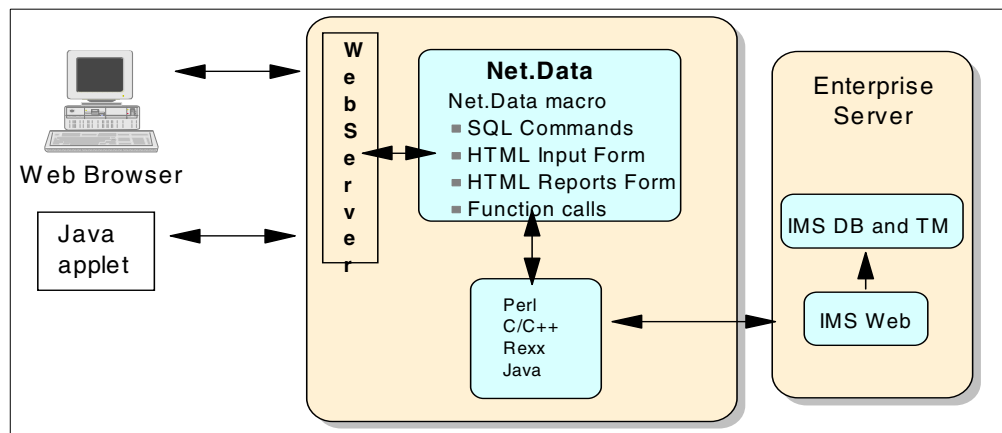


Figure 104. Net.Data and IMS

Net.Data supports HTTP server API interfaces for Netscape, Microsoft Internet Information Server, IBM HTTP Server, and IBM Internet Connection Server. Net.Data also supports CGI and FastCGI interfaces.

Features

- Access to IMS is through IMS Web.
- Supports client-side processing by Java applets and JavaScripts.
- Supports server-side processing by language environment definitions for languages such as SQL (for direct access to DB2), Java applications, Web Registry, REXX, Perl, Oracle and Sybase direct access languages and C++.

- Provides conditional logic and a rich macro language with the ability to run external programs.
- Provides a variety of servlets to help develop and manage macros.
- Supports ODBC and JDBC.
- Provides dynamic library loading.
- Provides pure HTML output.
- Live Connection feature to reduce overhead of re-establishing sessions.
- A Cache Manager can be set up to cache the most recently used pages in memory.
- A GUI panel-driven administration tool to allow users to automate the administration and configuration of Net.Data and Live Connection.
- Error message logging.
- UTF-8 and UTF-16 Unicoding support.
- FastCGI support.
- Supports the processing and display of multiple tables.
- Net.Data Task Guides provide a quick and easy way to create personalized Net.Data applications.

10.9.0.1 Net.Data IMS implementation

The interface to IMS from Net.Data is through IMS Web. In addition to providing the IMS Web language environment for Net.Data, the IMS Web Studio generates the Net.Data Web macro and the associated DLL or shared library used to process an IMS transaction through IMS Web.

The Net.Data Web macro contains a series of macro language, HTML, and language environment-specific statements. The macro includes both the input and output HTML for the IMS transaction, as well as a %FUNCTION statement to invoke the IMS Web language environment, which will, in turn, dynamically load the transaction DLL or shared library.

Net.Data can be used as a CGI application or as a Web server API application. When Net.Data is used as a CGI application, it is invoked as the executable DB2WWW. When Net.Data is used as a Web server API application, the server calls it as a DLL or shared library.

Language environments are Net.Data's interface to data and applications. Net.Data provides language environments to access DB2 and ODBC databases, REXX, Perl, C applications, file systems, Web registries, Java applets, and IMS transactions through IMS Web. Net.Data's IMS Web language environment is only supported when Net.Data runs as a CGI application. It is not supported with Net.Data as ICAP.

Requirements

- For hardware and software requirements, as well as for optional software for performances improvements, see updated information at <http://www.software.ibm.com/data/net.data/downloads.html>
- IMS Web
- The IMS Web Runtime component has to be installed on the Web server in order to use the IMS Web language environment.
- For most platforms, the IMS Web language environment is shipped with Net.Data; however, the LE for OS/390 is not. It is available from the Internet on the IMS Web home page at <http://www.software.ibm.com/data/ims/imsweb.html>

- The IMS Web Development component and/or the IMS Web Studio has to be installed to generate the Net.Data Web macro and transaction DLL to be used in conjunction with the IMS Web language environment.
- Net.Data is available for the following platforms:
 - Windows NT
 - OS/2
 - AIX
 - OS/400
 - OS/390
 - Sun Solaris
 - HP-UX
 - SCO UnixWare

Availability

Net.Data is currently shipped as a no-charge feature of:

- Universal Database UDB and DB2 Version 6.1 for OS/390
- IBM HTTP Server (formerly the Lotus Domino Go Webserver)
- Net.Data can also be downloaded free at:

<http://www.software.ibm.com/data/net.data/downloads.html>

Skills Needed

- Basic SQL, REXX, Perl, C++, Java programming language knowledge depending upon the implementation
- HTML knowledge

Where to find more information:

- <http://www.software.ibm.com/data/net.data/>

10.10 MQSeries Link LotusScript Extension (MQLSX)

The MQSeries Link LotusScript Extension is a set of LotusScript classes that incorporate the power of the MQSeries Message Queueing Interface (MQI), giving your LotusScript applications the ability to interact with any MQSeries capable application. Using these classes, Notes developers can open a session with an MQSeries queue manager, attach to a queue, and send or receive messages.

It is implemented as a LotusScript extension, allowing LotusScript to access the IBM MQSeries API so that MQSeries calls can be done directly from LotusScript code. The MQLSX code does not make any calls to Notes.

MQLSX supports the following platforms:

- Windows (3.x, 95, NT 3.51 and 4.0)
- OS/2 3.0 and 4.0
- AIX 4.1 and 4.2
- HP-UX V10
- OS/400
- MVS/ESA
- Sun Solaris 2.5

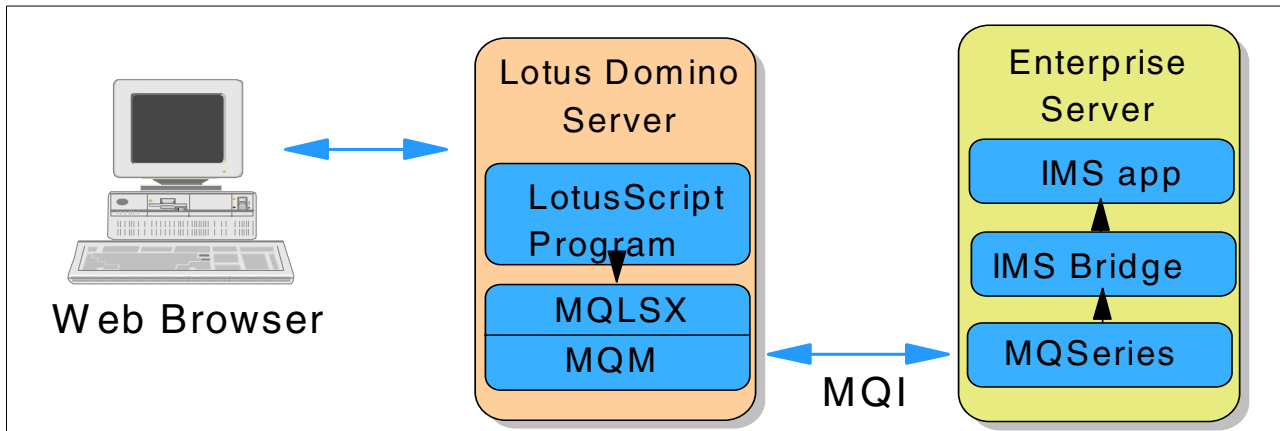


Figure 105. MQLSX to IMS

Features

- LotusScript Extension (LSX)
- Object wrapper for MQI
- Brings full programmability from LotusScript to MQSeries
- Better performance than MQEI because there is no database lookup at run time
- Allows MQSeries object model conformance

Requirements

- Lotus Notes Client 4.5.1 or higher or Lotus Domino Server 4.5.1 or higher
- Install appropriate level of MQSeries Server or MQSeries Client on the same machine as the LotusScript applications using the MQLSX.
 - For Intel or UNIX: V2 or V5
 - For AS/400: V4.2 (Domino Server implementation only)
 - For MVS/ESA: V1.2 (Domino Server implementation only)
- Installation of MQLSX:
 - If installed on an IBM HTTP Server (Domino Server), then Notes clients and Web browsers can use MQSeries via the server.
 - If installed on a Notes client, then the Notes client must have an MQSeries client installed.

Note: The MQLSX cannot be installed directly onto a Web browser.

- MQSeries IMS Bridge
- LotusScript programs

Availability

- Part of the Lotus Domino.Connect family
- Current release levels of MQLSX:
 - 1.3.1 for AS/400 and MVS/ESA
 - 1.3.2 for Windows 16-bit implementations
 - 1.3.3 for Windows, OS/2, and UNIX (32-bit implementations)
- IBM MQSeries SupportPacs for each platform:
 - <http://www.software.ibm.com/ts/mqseries/txppacs>
- 1.3.2 for Intel and UNIX is shipped as part of MQSeries & CICS Connections for Domino
- Shipped as part of MQSeries for Windows NT V5.1

Skills Needed

- LotusScript
- MQLSX API
- MQSeries knowledge
- IMS knowledge

Where to find more information:

- *Connecting the World Wide Web. A Practical Guide to IMS Connectivity*, SG24-2220

10.11 MQSeries Enterprise Integrator for Lotus Notes (MQEI)

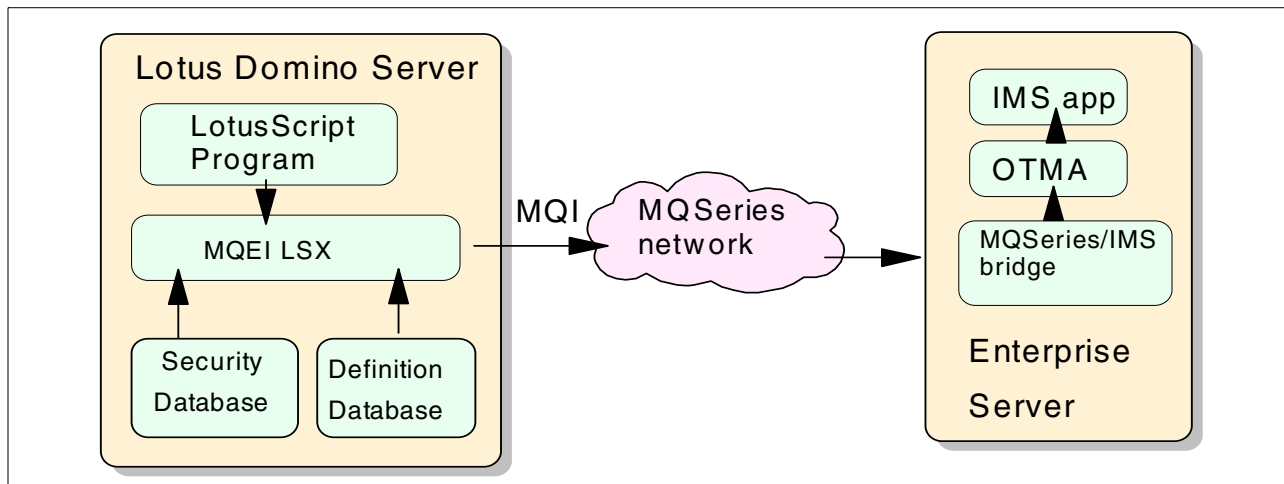


Figure 106. MQEI Access to IMS

MQSeries Enterprise Integrator is much the same idea as MQLSX, but the application programming interface is simplified and more intuitive. You will probably get better performance with MQLSX, but may find the features of MQEI make it worth staying with the simpler interface.

MQEI is a collection of Notes databases and a LotusScript extension (MQEI LSX) which provide transparent access to an organization's enterprise applications. It gives your Lotus Domino applications, written in LotusScript, the ability to communicate with applications running in non-Notes environments using MQSeries as the network protocol.

Features:

- LotusScript Extension (LSX).
- Provides a simplified, common API for enterprise access regardless of the enterprise system.
- The API has a common set of verbs independent from each enterprise system, eliminating the need for the programmer to learn more than one API.
- Uses Notes databases to store messages, enterprise application definitions, and security parameters.
- Integrated security features enabling seamless sign-on to enterprise systems.
- LotusScript independence from network configuration and message formats.

- Built-in capability to access unmodified enterprise applications running under IMS with object-oriented applications, as well as native MQSeries applications on other enterprise server platforms.
- Enhanced message building facilities. Notes programmers can build and interpret messages by reading and writing named fields within the message, without having to know precise details of the message format. The system will provide default values for fields that the programmer does not set explicitly, if this is appropriate. Messages for multiple destinations can be created. Common message structures can be re-used in several LotusScript applications.
- Integration of the Identification and Authentication services of Lotus Notes with the access controls provided by IMS. MQEI has its own security database under Notes, containing host user ID and authenticators.
- The programming model is simpler than the model provided by the standard MQLSX, as the Enterprise Integrator uses a Lotus Notes database to hold object-definition information, reducing the amount of in-line programming required.
- Allows access to IMS through the MQSeries IMS bridge, which provides OTMA access to IMS.
- Automatically builds MQSeries IMS bridge headers when sending a message.

Requirements

- Lotus Notes Client 4.5.1 or higher or Lotus Domino Server 4.5.1 or higher
- Whether running in the client or server environment, LotusScript applications require either V2 or V5 of the MQSeries client or an MQSeries server, installed on the same machine as MQEI.
- MQEI is available on the following platforms:
 - Windows 3.x, 95, and NT 3.51 and 4.0
 - OS/2 3.0 and 4.0
 - AIX 4.1, 4.2, and 4.3
 - HP-UX V10
 - Sun Solaris 2.5
- MQSeries on the target enterprise system
- MQSeries IMS Bridge, which is part of IMS V5
- User-written LotusScript applications

Availability

- MQSeries 1.1.4 and above include the IMS Bridge.
- MQEI is a part of the Lotus Domino.Connect family.
- MQEI is a part of MQSeries & CICS Connections for Domino 1.0a.
- Free download evaluation version for Windows NT, OS/2, and AIX.

Skills Needed

- LotusScript programming
- MQSeries implementation
- IMS platform and programming

References

- *Lotus Solution for the Enterprise, Volume 4 - Lotus Notes and the MQSeries Enterprise Integrator*, SG24-2217
- *Connecting the World Wide Web. A Practical Guide to IMS Connectivity* SG24-2220

10.12 AS/IMS, Java, and DCE/RPC

Host IMS applications can be accessed through the Internet through the Open Systems Foundation's (OSF's) Distributed Computing Environment (DCE) with a Java-enabled browser. Customers can write a simple Java applet that issues a Remote Procedure Call (RPC) to an IMS transaction through MVS OpenEdition DCE/RPC and AS/IMS.

Application Support/IMS (AS/IMS) transforms an incoming RPC into an IMS input message and uses the ISC link to place the message in the input message queue. The AS/IMS server gets the response from an output message and transforms it into RPC output parameters.

10.12.1 DCE/RPC

The Distributed Computing Environment (DCE) is a cross-platform set of services that supports the development, use, and maintenance of distributed computing applications. The services, DCE Secure Core and DCE Data Sharing Services, work together to act as a single application interface.

The DCE Secure Core services include multithreading application support, all RPCs for remote calls, directory services, security, clock synchronization, auditing, and gateway support.

The DCE Data Sharing Services include DFS, a high-performance distributed file system for sharing remote files, that appear as local file systems to the user.

The IBM DCE suite of products provides the following additional features:

- Incorporates public key infrastructure (PKI).
- Supports Lightweight Directory Access Protocol (LDAP).
- Simplifies administration with a Web browser-based interface.
- Enables enterprise growth for distributed processing, independent of hardware or software platforms.
- Provides the capability to integrate any DCE-compliant software or hardware platform into your enterprise.
- Year 2000 ready.

DCE RPC consists of a tool for the development of distributed applications and a run-time service that implements the network protocols.

IBM DCE runs on the following platforms:

- Windows NT
- OS/2
- AIX
- UNIX
- VM/ESA
- MVS/ESA
- OS/390
- Sun Solaris

For a detailed description of a DCE product from Transarc, see:

<http://www.transarc.com>

10.12.2 Web access to IMS through Application Support/IMS

Application Support/IMS (AS/IMS) uses an Intersystem Communication (ISC) parallel session to connect to IMS. AS/IMS can reside on a different OS/390 or MVS sysplex as IMS. ISC allows multiple connections, but only one at a time is dedicated to a specific user.

AS/IMS converts RPC calls into IMS messages and vice versa, giving a Java applet downloaded on a browser the ability to use RPC calls to an IMS application program.

The client does not need to know where the IMS programs are located; the underlying DCE directory service and DCE run-time service provide location information. The combination of AS/IMS and the OpenEdition DCE Base Services feature ensures that differences between the client and server systems are transparent to the application programmers.

From a security point of view, the combination of DCE and AS/IMS performs authentication, and RACF performs authorization. IMS V4 supports the use of the /SIGN command with ISC links. With IMS V4, AS/IMS can pass the RACF user ID associated with the DCE principal to IMS. IMS can in turn use this RACF user ID in combination with RACF to control access to IMS resources.

AS/IMS ensures that a particular DCE client is authorized for access to IMS resources purely on the basis of the existence of an identity mapping entry.

10.12.3 Encina DE-Light

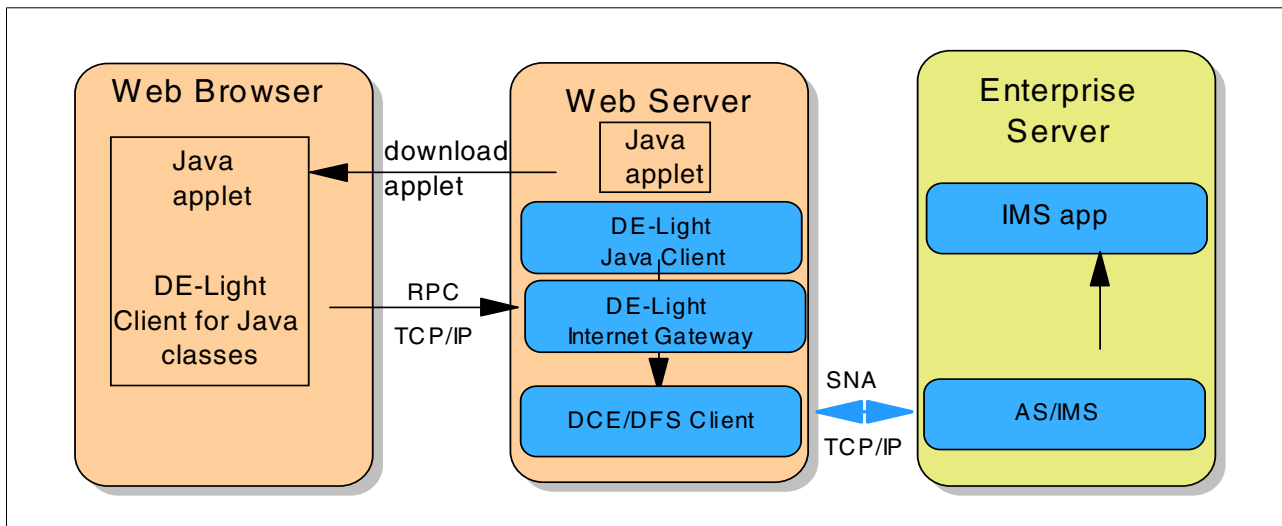


Figure 107. Encina DE-Light

An example of a Java implementation of IMS access via DCE is Encina DE-Light Web Client from Transarc. More detailed information can be found on Transarc's home page: <http://www.transarc.com>

DE-Light consists of two software components, a client and a gateway.

The DE-Light Web client:

- Consists of several small Java classes which use a unique API to issue DCE RPCs and Encina TRPCs and implement the DE-Light client run time.
- Submits the RPCs and TRPCs through the DE-Light gateway.

The DE-Light Gateway:

- Is a DCE client.
- Receives requests from DE-Light Web clients.
- Converts the requests into RPCs or TRPCs (acts as an RPC interpreter).

When a user visits a Web page that accesses DCE or Encina, the Java applet is downloaded to the client browser. The applet contains presentation logic, a link to a DE-Light Web client Java class, and a small amount of application logic. To execute an RPC, the client run time builds a string describing the RPC, then passes the string to the DE-Light gateway. A TCP/IP connection is used to exchange the string. The gateway builds a true DCE RPC or (Encina TRPC) and makes the appropriate call. Upon completion, the outbound arguments are sent by the gateway to the run time client. The results are stored in the data dictionary. In order to format and display the data, the presentation logic is invoked.

Features

The DE-Light Web client offers the following advantages over existing forms-based approaches for integrating with DCE and Encina applications:

- Ease of installation; only a Web browser has to be installed on the client side.
- Requires less than 100 KB of system memory.
- Run time written in Java allows for automatic download of the Web client when needed.
- Efficient communication protocol between the client and gateway, designed for users connecting on low bandwidth channels like dialup lines.
- Java-managed distribution of application logic that has been modified.
- Caching allows a large number of clients to use a single gateway.
- Application logic is downloaded from the Web server and executed on application servers. Presentation and initial data editing takes place on the client.
- Reliable operation of Web applications when implemented in a three-tiered architecture, which provides load-balancing, scalability, and server replication
- High performance from multi-threading allows for many concurrently executing calls.
- The Encina Transactional RPC (TRPC) API makes it easy to implement transactions with full integrity.
- Leverages existing programming staff skills since the DE-Light API is a simplified version of DCE and Encina API.
- All applications have access to DCE security services.

Requirements

- Gateway prerequisites:
 - DCE 1.1
 - Solaris 2.4, 2.5 or AIX 4.1.4 or HP-UX 10.20
 - 25 MB disk space
 - 15 MB swap space
 - Any secure Web server running on same host as gateway (the Internet DE-Light Client only)
- DE-Light client prerequisites:
 - MS-Windows 3.1 or higher

- MS-DOS 6.0 or higher
- 100 KB RAM
- 2 MB disk space
- Internet DE-Light client prerequisites:
 - Netscape 3.0 or Internet Explorer 3.0
 - Sun's JDK 1.0.2 or JDK 1.1
- Certified TCP stacks (DE-Light is designed to work with any Winsock-compliant TCP stack):
 - Windows for Workgroups
 - Novell LAN Workplace
 - FTP Software PC/TCP
 - Chameleon

Availability

The products must be ordered from Transarc Corp. <http://www.transarc.com>.
DE-Light and DCE patches can be downloaded from <http://www.transarc.com>.

Skills Needed

- AS/IMS
- DCE and Kerberos setup and administration
- RPC usage

References

- *MVS/ESA OpenEdition DCE: Application Support Servers CICS and IMS* - GG24-4482
- *Connecting the World Wide Web. A Practical Guide to IMS Connectivity*, SG24-2220
- <http://www.software.ibm.com/enetwork/dce/library/>
- <http://www.transarc.com/Product/Txseries/DELight/>
- <http://www.transarc.com/Product/Txseries/DELight/DELight1.0/docs/java/overview.html>
- <http://www.transarc.com/Product/Txseries/DELight/dltdwnld.html>

Chapter 11. Solutions for accessing DB2 from the Web

Among the highest priorities in DB2-based enterprises is the ability to access business information stored in DB2 from Internet, intranet, and extranet applications. Access to DB2 data from Web applications can be delivered through Host On-Demand, Host Publisher, Net.Data, Java Database Connectivity (JDBC), and SQLJ.

The DB2 family consists of:

- DB2 Universal Database (UDB) for mid-range and large systems on OS/400, VSE/VM, and OS/390.
- DB2 UDB for Intel and UNIX environments. The supported platforms include OS/2, Windows NT, Windows 98, Windows 95, AIX, HP-UX, Linux, SCO UnixWare, and Solaris.

Host Publisher

Host Publisher is very suitable for developing new Web applications that get data from existing 3270-based, 5250-based, VT-based, and JDBC-based applications, providing dynamic HTML pages for display. Access to host applications and databases is provided by Host Publisher applications consisting of Integration Objects invoked by Java Server Pages (JSPs).

Host Publisher delivers new applications without any need to change the existing applications. Host Publisher allows you to integrate data from several sources into the Web pages, giving more flexibility in building the data to be presented to the browser. Because the output of the applications is HTML there are no requirements for the client browsers.

Host Publisher provides several JDBC drivers, including DB2 UDB local and DB2 UDB remote.

For more details about Host Publisher, see Chapter 3, "IBM SecureWay Host Publisher" on page 41.

Host On-Demand

Host On-Demand is unique in that it provides a single Web interface to any 3270, 5250, or VT application, giving host emulation users easy Web access to DB2 data on the host. The Host On-Demand Java applet running on the Web browser connects to a TN3270 or TN5250 server, giving 3270/5250 access to the host. The appearance to the user is much like a traditional emulator screen. By adding Screen Customizer, the green screen presentation can be automatically changed to a GUI presentation.

In addition to the terminal emulation function, Host On-Demand also provides the Database On-Demand feature. This feature provides a JDBC driver for exchanging DB2 data with an AS/400 system.

Host On-Demand is a simple product to install and implement. It is the fastest transition path to make traditional terminal emulation applications available to Web users. However, there may be some restrictions on the browsers that can be used, making this more of an intranet or extranet solution.

Java

There are three alternatives for accessing DB2 data with a Java solution: Java applications, applets, and servlets.

Java applications are attractive to customers who would like applications to be portable among operating systems. A Java application does not need a browser to be executed. It can be executed in a client or server machine.

Java applets are Java programs imbedded in HTML for execution on the Web browser. This requires Java-enabled browsers and may have Java-related dependencies. It may be best used where the Web environment is controlled, such as an intranet or extranet.

Java servlets are pieces of Java code that run inside a Java-enabled Web server and extend the functions of the Web server. The Web server hands requests to the servlet, which replies to them. Servlets can be viewed as a good substitute for CGI applications.

DB2 Java support is provided through JDBC and SQLJ. Dynamic SQL for Java is provided by Java Database Connectivity (JDBC), which is a Java API for client access to any relational database. A JDBC driver is necessary to implement the JDBC method calls, converting them into calls that are understandable to the database engine. SQLJ provides support for embedded static SQL within Java.

Net.Data

Net.Data provides high performance Web applications with robust application development function. Net.Data provides connectivity to a variety of data sources and applications including DB2, Oracle, Sybase and DRDA enabled data sources, ODBC data sources, and across a wide range of platforms, flat files, registries, system services, and IMS databases through IMS transactions.

Net.Data exploits web server interfaces (APIs), providing higher performance than common gateway interface (CGI) applications. Net.Data supports client-side processing as well as server-side processing with languages such as Java, REXX, Perl and C++. Net.Data provides conditional logic and a rich macro language.

With Net.Data, you get full support for Java. You can use a Java applet to create a graphical chart, such as a pie chart, from the results of a Net.Data application. With Net.Data's support for JavaScripts, you can validate data entered at the client's Web browser.

Component Broker

Component Broker provides a development toolkit and run time environment for distributed three-tier applications, in which objects developed in different languages communicate with each other by way of the standard CORBA Object Request Broker and have a persistent image of traditional database and transaction environments through application adaptors. IBM Component Broker Connector (CBConnector) provides application adaptors to access CICS, IMS, Oracle, MQ, and DB2. The Relational Database Application Adaptor for DB2 provides support for embedded SQL on DB2. Support for DB2 CLI and ODBC is targeted for future releases. More information about Component Broker can be found in Chapter 7, "WebSphere" on page 111.

Lotus Domino environment

Enterprises that have a Lotus Domino environment have several choices when determining how to connect enterprise data to the Web. These choices primarily depend on the amount of programming they want to do to control the presentation of the data on the Web.

The Lotus Domino Server acts as both a Notes server and a Web server. It allows access from Web browsers to Notes applications by dynamically translating Notes applications into HTML. The following programming methods are available for integrating DB2 data into Notes applications:

- @DB Functions providing read access to ODBC databases
- Lotus Connector LotusScript Extensions (LC LSX)
- LotusScript Data Object (LS:DO) providing read/write access to ODBC databases
- DB2LSX provides native access to DB2 through the DB2 Call Level Interface (CLI)
- Lotus Connector API (LC API)
- JDBC Java classes allow a Notes agent to send SQL directly to a DB2 data source and retrieve results.
- The JDBC-ODBC bridge provided by Domino allows Java access to ODBC data sources. DB2 is both JDBC and ODBC compliant. If your installation already has an ODBC environment set up and wants to add JDBC, this is an easy option.

Lotus Enterprise Integrator (formerly called NotesPump) is used to exchange large volumes of data between Lotus Notes and relational database systems on an event-driven, scheduled, or ad-hoc basis. It can also be used for enterprise-wide connectivity. Lotus Enterprise Integrator (LEI) can access DB2 on a wide range of mainframe and client/server platforms. No programming effort is required to implement LEI. LEI offers three interfaces for Web access to DB2 data:

- LEI RealTime to access DB2 data to populate Notes forms
- Java classes for Java applet access to DB2 data
- Web server CGI for HTML access to DB2 data

Lotus Domino is covered more thoroughly in Chapter 6, "Lotus Domino" on page 95. An excellent reference is *Lotus Solutions for the Enterprise, Volume 2, Using DB2 in a Domino Environment*, SG24-4918.

11.1 Programming interfaces to DB2

Programming access to DB2 data is done primarily from one of three interfaces:

- ODBC
- JDBC
- SQLJ

We will take a quick look at each interface.

11.1.1 Java Database Connectivity (JDBC)

Java Database Connectivity (JDBC) is an object interface that allows Java applications, applets, servlets, and agents to retrieve and manipulate data in database management systems using SQL. The interface allows a single

application to connect to many different types of databases through a standard protocol. JDBC handles details such as connecting to a database, fetching query results, committing or rolling back transactions, and converting SQL types to and from Java program variables.

JDBC is implemented as a driver manager with multiple drivers. JDBC uses the driver manager to handle finding and connecting to a driver. Each driver links the application to a specific type of database. A JDBC data source consists of the data the user application wants to access and its associated parameters. Each JDBC driver processes JDBC method invocations, sends SQL statements to a specific data source, and returns the results to the application.

There are four types of JDBC drivers:

- **JDBC-ODBC bridge:** Provides JDBC access via ODBC drivers. NotesSQL (the Domino/Notes ODBC driver) may be used with this bridge.
- **JDBC Native driver:** Converts JDBC calls into calls on the client API for the DBMS in question. This type of driver requires some binary code to be loaded on the client machine.
- **JDBC Net driver:** Translates JDBC calls into a DBMS-independent net protocol which is then translated to a DBMS protocol by a server. This net server middleware is able to connect its all-Java clients to many different databases. This is the most flexible Java alternative.
- **JDBC thin driver:** Converts JDBC calls into the network protocol used by DBMSs directly. This allows a direct call from the client machine to the DBMS server and is a practical solution for Internet access.

11.1.1.1 DB2 JDBC support

DB2 is now distributed with JDBC and the DB2 JDBC interface to allow relational data access from Java code. It provides dynamic SQL for Java.

DB2 provides native support for Java at client workstations and DB2 servers.

Java is supported on the client workstations in two ways:

- A Java application uses the DB2 CAE, which must be installed on the client workstation, to communicate with the DB2 server. Customers with existing DB2 client/server configurations can now use Java as a database application development tool.
- Java applets allow applications to be developed that access DB2 servers without requiring DB2 CAE code to be installed on client workstations. Java applets can be automatically downloaded to the client workstation at invocation time. Java support in DB2 servers consists of the ability to create native Java-based, user-defined functions and stored procedures.

Using TCP/IP, the Java applet uses the JDBC applet driver to open a connection with the JDBC applet server. The JDBC applet server, using the DB2 CAE, then connects to the database and accesses the tables. It is also important to note that the JDBC applet server can be located on a machine other than the Web server if the applet is developed with JDK 1.1.

Java application support

A Java application can be executed on a client or server machine. The DB2 JDBC driver handles the JDBC API calls from your application and uses DB2 CAE to

communicate the requests to the server and receive the results. The same JDBC APIs used in an applet or a servlet can be used by the application to access DB2.

Java applet support

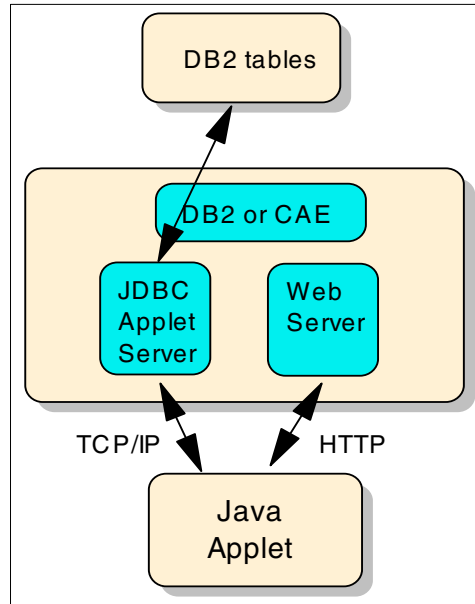


Figure 108. DB2 JDBC interface

To run an applet, you need a Java-enabled Web browser on the client machine. When the Web browser loads an HTML page with an applet tag, the Java applet is downloaded to the client machine, which then downloads the Java class files and the DB2 JDBC driver. When the applet calls the JDBC API to connect to DB2, the JDBC driver establishes separate communications with the DB2 database through the JDBC applet server residing on the DB2 server.

Java servlet support

A servlet can easily access DB2 via JDBC. It parses the HTTP requests coming from an HTML page or an applet in the client browser, calls DB2, and sends the results back to the client. However, this kind of application can be easily developed using Net.Data. You can use servlets if you do not want to write a macro language-based application or if you want to implement a more sophisticated program with complex logic accessing your DB2 data.

Two-tier versus a three-tier JDBC solution

In a two-tier JDBC solution, DB2 Connect, CAE, the JDBC driver and the Java application or applet run on the client. The DB2 database resides on the Web server. In a three-tier JDBC solution, CAE, JDBC driver, and the Java application or applet run on the client. DB2 Connect resides on the middle tier. The DB2 database resides on the Web server. For Java servlets in a two-tier JDBC environment, the servlet, JDBC driver, CLI and DB2 database reside on the Web server. The Web browser runs on the client.

In the three-tier environment, the Web browser resides on the client. The middle tier is the Web server, which is where the Java servlet, JDBC driver, CAE, and DB2 Connect reside. The DB2 database is on the host server.

Choosing between a two-tier and three-tier JDBC solution

Use two-tier solutions when you want to have:

- Only two platforms and a Web browser on the client. You must use signed applets and preinstall DB2 CAE and DB2 Connect Personal Edition on every client system as well as distribute the client certificates.
- A two-tier approach, but do not require a Web browser on the client. Use a preinstalled Java application on the client platform. You will need DB2 CAE and DB2 Connect Personal Edition on each client platform. You will not have to distribute client certificates.

Use three-tier solutions when you want:

- To avoid configuring or preinstalling software or certificates on the client system. Use CAE and DB2 Connect Enterprise Edition on the middle tier.
- To access multiple DB2 subsystems from a single applet, with a requirement that all updates form a single transaction. Use DB2 Connect in the middle tier.
- To avoid using TCP/IP on OS/390. Use DB2 Connect Enterprise Edition on the middle tier.

Requirements and skills needed

- JDBC server, DB2 Connect, and DB2 CAE
- Java-enabled browser
- Java programming

Where to find more information:

- *From Client/Server to Network Computing: A Migration to Java*, SG24-2247

11.1.2 SQLJ

DB2 SQLJ allows you to create, build, and run embedded static SQL for Java applications, applets, and stored procedures. SQLJ provides support for embedded SQL within Java applications and Java applets. SQLJ was developed initially by Oracle and now other DBMS vendors including IBM embrace SQLJ for embedding SQL in Java applications. The SQLJ specification has now been embraced by the SQLJ consortium and is currently under review by ANSI.

At this moment, IBM is the only vendor where embedded SQL in an SQLJ program is run as true static SQL in the database. Thus, DB2 applications using dynamic SQL continue to use JDBC while applications requiring the performance advantages of static SQL use SQLJ.

The SQLJ specification consists of three parts:

- Part0, which specifies the syntax of embedded SQLJ within Java
- Part1, which specifies the extensions for invoking static methods of Java classes as stored functions and stored procedures
- Part2, which specifies the extensions to support Java classes as SQL data types

The SQLJ specification consists of a set of programming extensions that define interactions between SQL and Java. It comprises a set of clauses that extend Java programs to include static SQL constructs. A utility, the SQLJ Translator, is needed to translate SQLJ into Java code that the Java compiler can understand.

The SQLJ Translator transforms SQLJ clauses into standard Java code that accesses the database through a call interface. One output of the SQLJ translator is a generated Java source program that can be compiled by any Java compiler. The other Translator output is one or more files that contain the SQL statements. These files are run through a DBMS-specific tool or binder, which modifies or extends the file. The extended file is then input to the bind process. In this way, the Java code can execute static SQL.

SQLJ supports only embedded SQL constructs. It relies on JDBC for support of dynamic SQL and does not attempt to replicate the features of JDBC. Rather, SQLJ contains mechanisms that enable a Java programmer to move easily between the two environments and share state information.

SQLJ allows a Java programmer to create, build, and run embedded SQL Java applications, applets, stored procedures, and user-defined functions (UDFs). DB2 for OS/390 will initially support Part0 of the ANSI SQLJ specification.

For non-OS/390 environments, DB2 supports SQLJ with the DB2 Software Developer's Kit (SDK) along with DB2 JDBC support provided by DB2 CAE. In OS/390 environments, DB2 Server for OS/390 Version 5 and DB2 UDB for OS/390 Version 6 support SQLJ.

For more information about SQLJ, see the SQLJ Webpage at:
<http://www.sqlj.org>.

11.1.3 Open Database Connectivity (ODBC)

Net.Data, Lotus Script Data Object (LS:DO), Lotus JDBC-ODBC bridge, Host Publisher JDBC-ODBC Bridge driver and others use the ODBC interface to access DB2 data.

The ODBC standard is a set of functions established by Microsoft to access Relational Database Management Systems (RDBMS), including DB2. ODBC implements and extends the standard callable SQL interface. The ODBC interface gives the user some flexibility. An application can access different database management systems by using different database drivers that link the application to the specific database management system at run time. ODBC has four components:

- **Application:** The application is responsible for making an ODBC function call to submit SQL requests and retrieve results. The application requests a database connection, sends SQL requests to the database, obtains the results, checks for errors in the data, reports back to the user, and terminates the database connection.
- **Driver Manager:** The driver manager provides an interface between the application and the ODBC drivers. It loads the necessary driver when an application requests to connect to a database.
- **Driver:** It implements the ODBC function set, interacting with the database and handling requests for the data source. RDBMS ODBC drivers are for specific applications like DB2, Oracle, Notes, and SQL. The DB2 ODBC driver uses DB2 Connect or DB2 CAE for connectivity to remote DB2 systems.
- **Data Source:** The data source is a particular database management system. It consists of the data needed to perform the SQL requests and returns the results back to the application.

11.1.4 Java Stored Procedures

Business logic can be implemented securely in the DB2 server with minimal network overhead using stored procedures which can now be written in Java in addition to more traditional programming languages.

Java stored procedures are supported in DB2 UDB Version 5 on workstation platforms. In the future, it will be possible to develop and run Java stored procedures for DB2 Version 5 on OS/390.

Java stored procedures can be invoked from DB2 SQLJ, JDBC, SQL, or ODBC requests. The invoking applications can be applets, servlets, stored procedures or regular applications running on any DB2 supported platform.

Java stored procedures are not required to detect the type of calling application or API used. You can read more about Java stored procedures in a workstation environment at URL:

<http://www.software.ibm.com/data/db2/java/v5/docs/>

DB2 for OS/390 Version 5, in conjunction with VisualAge for Java - High-Performance Compiler, supports Java stored procedures. The High-Performance Compiler produces a program object that is an executable. In this case, DB2 for OS/390 Version 5 is able to load such a load module and executes it. Parameters passing between DB2 and the high performance compiled Java stored procedure have been reviewed for this particular kind of programming.

11.2 DB2 connectivity

Network connectivity to DB2 is accomplished by using DB2 Connect. DB2 Connect allows clients to access data stored on any IBM DB2 UDB relational database and on database servers that implement the Distributed Relational Database Architecture (DRDA).

DRDA servers include:

- DB2 for MVS/ESA
- DB2 for AS/400
- DB2 for OS/390
- DB2 for VSE & VM systems

The target database server for a DB2 Connect installation is known as a DRDA application server. DB2 Connect supports network connectivity to DRDA application servers using APPC, TCP/IP, or AnyNet, depending on the platform. DB2 Connect acts as a gateway for DB2 clients. Currently, DB2 for OS/390 Version 5.1 (and higher) and DB2 for VM Version 6 are the only DRDA servers to support TCP/IP.

DB2 clients can use any of the following protocols supported on the platform to connect to the DB2 Connect gateway:

- NetBIOS (OS/2 and Windows environments)
- TCP/IP (all environments)
- IPX/SPX (Novell NetWare LANs)
- APPC (IBM operating environments)

- Named Pipes (Windows NT environments)

DB2 Connect Personal Edition

DB2 Connect Personal Edition provides single-user access to host databases from the Intel workstation where it is installed. It is available on the Intel platform for OS/2, Windows 98, Windows 95, Windows NT, and Windows 3.x. DB2 Connect Personal Edition includes Integrated SNA Support for use on Windows 3.x. Users of OS/2 and Windows NT also require IBM Communications Server. DB2 Connect Personal Edition was formerly known as DDCS Single-User.

DB2 Connect Enterprise Edition

DB2 Connect Enterprise Edition provides the ability for multiple clients to access host data by acting as a gateway. It supports the following Intel and UNIX workstations in a workgroup, department, or LAN setting: AIX, HP-UX, OS/2, SCO UnixWare, Solaris, and Windows NT. It routes each database request from the DB2 clients to the appropriate DRDA Application Server database. The client uses the Client Application Enabler (CAE). DB2 Connect Enterprise Edition was formerly known as DDCS Multi-User Gateway.

11.2.0.1 DB2 Client Application Enabler

All DB2 products have a common component called the DB2 Client Application Enabler (CAE). Once a DB2 application has been developed, the DB2 CAE component must be installed on each workstation executing the application. If the application and database are installed on the same workstation, the application is known as a local client. If the application is installed on a workstation other than the DB2 server the application is known as a remote client. There is a different CAE for each supported DB2 client operating system.

The CAE provides the ability to communicate with a DB2 UDB server or DB2 Connect gateway. From the CAE, you can also do the following:

- Issue an interactive SQL statement to access data on a remote UDB server.
- Graphically administer and monitor a UDB database server.
- Run applications that were developed to comply with the ODBC standard.
- Run Java applications that access and manipulate data in DB2 UDB databases using JDBC.

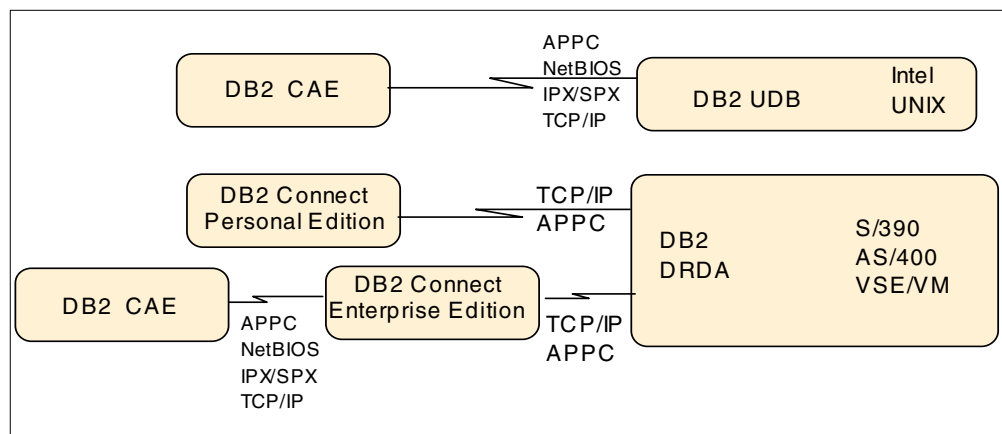


Figure 109. DB2 CAE and DB2 Connect

11.3 Host On-Demand

Through the capabilities of a Host On-Demand (HOD) environment, Web users can now access DB2 and other S/390, AS/400, and UNIX host functions using the emulation properties of the Host On-Demand clients to manipulate the application 3270, 5250, or VT data stream. In addition, Host On-Demand provides JDBC access to DB2 databases on AS/400 systems with Database On-Demand and access to CICS data with the CICS Gateway applet. We will quickly go over Host On-Demand as it relates to accessing DB2 data. Host On-Demand is covered in more detail in Chapter 2, “IBM SecureWay Host On-Demand” on page 9.

For general use, Web access to DB2 through HOD is recommended only for intranet or extranet solutions for the following reasons:

1. Host On-Demand delivers to the browser a full function terminal emulator, which implies that any user accessing host functions through this method would need to understand the interface of the applications they are accessing. This is unlikely to be something that the general Internet user would be able to do. (The use of Screen Customizer can simplify this interface for novice users.)
2. It will require a Java-enabled browser, and sometimes running specific levels of Java, to support the applets you are downloading from the Host On-Demand server. This is easy to control on the intranet browser and also for terminals of extranet users (such as business partners) but not so easy to control on the terminals of general Internet users.

This does not exclude access through the Internet if you are using the Internet as an external network connection environment for trained or trusted users who can use this path for logging on to your host systems. For example, you may have a need for occasional Web access to your DB2 systems environment for DB2 system programmers, database administrators or operations staff from home or while traveling.

Two different methods can be used from Host On-Demand to access DB2:

- HOD 3270 or 5250 terminal emulation
- Database On-Demand

11.3.1 Accessing DB2 using HOD 3270 or 5250 emulation

Host On-Demand terminal emulation functions are accomplished from a Java applet executing from a Java-enabled Web browser. There are three methods of getting the applet on the client machine:

- The applet can be downloaded each time the user accesses the HOD URL.
- It can be cached so it is only downloaded the first time the user accesses it and then only when updates have been made.
- It can be installed locally.

The last two options are ideal for situations where downloading the applet over the network is slow. HOD also provides an express client/server option and a Function On-Demand option to address slower networks.

Users connect to the OS/390 or AS/400 host using a Web browser as a 3270/5250 emulator to access DB2. This does not differ from any other scenario where Host On-Demand is used as a terminal emulator to access OS/390 or

AS/400 programs. A TN3270/TN5250 server is required for the connection to the host.

Figure 110 illustrates a simple configuration of using HOD to access DB2 on an OS/390 from a Web browser. This configuration can be varied by putting the telnet server and/or Web server/HOD server combination on other machines and other platforms.

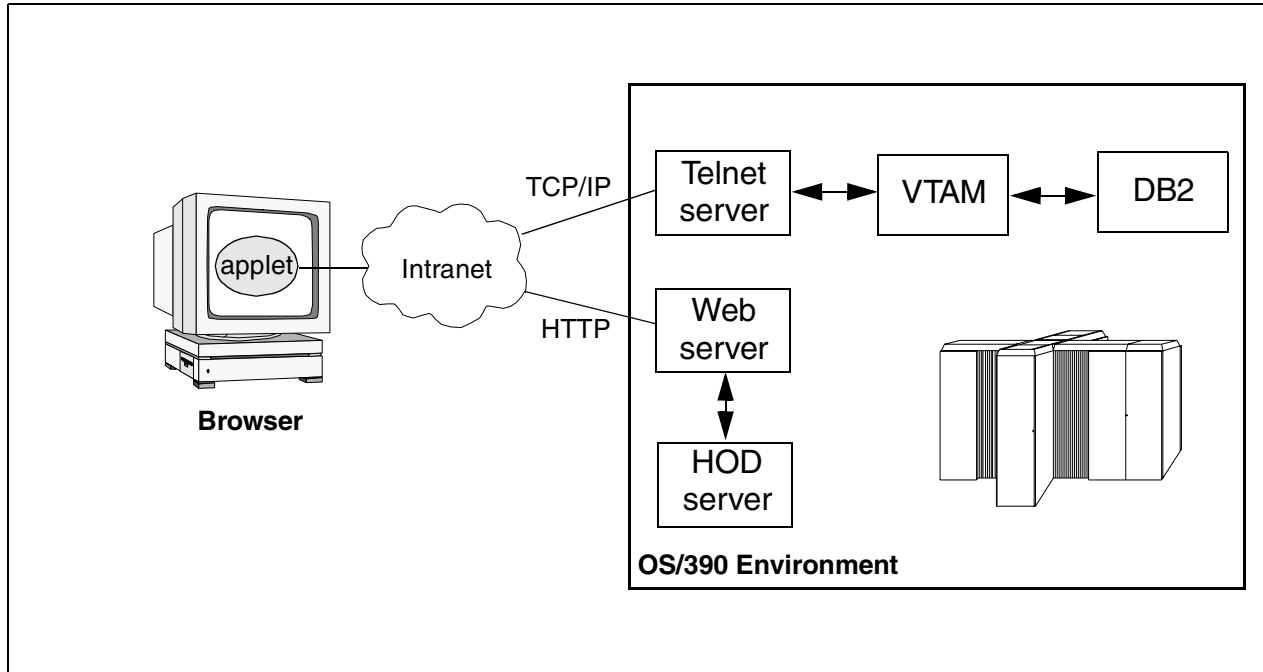


Figure 110. Host On-Demand - direct connection

In Figure 110, the Web browser enters the URL of the Host On-Demand server. Host On-Demand initiates the download of the Java applet if necessary. The Java applet, executing on the client machine, establishes a TN3270 or TN5250 session to the desired host by using a telnet server. The user now has the terminal emulation function on the Web browser and proceeds to log on to the host.

Screen Customizer/LE and the full Screen Customizer product can enhance the presentation by converting the “green screen” format into a graphical user interface (GUI). More information about Screen Customizer can be found in Chapter 4, “IBM SecureWay Screen Customizer” on page 69.

11.3.2 Database On-Demand

Host On-Demand includes the Database On-Demand feature. Like the terminal emulation functions, this feature is a Java applet downloaded or installed on the client machine. This feature allows users to access AS/400 by running SQL statements online using a JDBC driver. The AS/400 JDBC driver is shipped with Database On-Demand. You may use other drivers, such as the DB2 JDBC driver, but the following restrictions apply:

- JDBC drivers that access data locally will not work with Database On-Demand.

- Network JDBC drivers work with Database On-Demand only if the database resides on the same server as the Web server serving the Database On-Demand Web page and applet.

Database On-Demand provides a simple way for the user to extract DB2 data from the host. From the Web browser, the user can use simple GUI panels to define the target host, build SQL statements, and see the results. SQL statements can be saved for future use.

This feature is covered in more detail in Chapter 2, “IBM SecureWay Host On-Demand” on page 9.

11.4 Host Publisher

IBM SecureWay Host Publisher V2.1 provides access to 3270, 5250, VT, Java and JDBC backend data sources. Host Publisher provides several JDBC drivers to access DB2 data.

Host Publisher applications run on the server, not as Java applets. The information is served to the clients as HTML. Because of this, there are no specific requirements placed on the client Web browsers, and therefore, it is appropriate for Internet use as well as intranet and extranet.

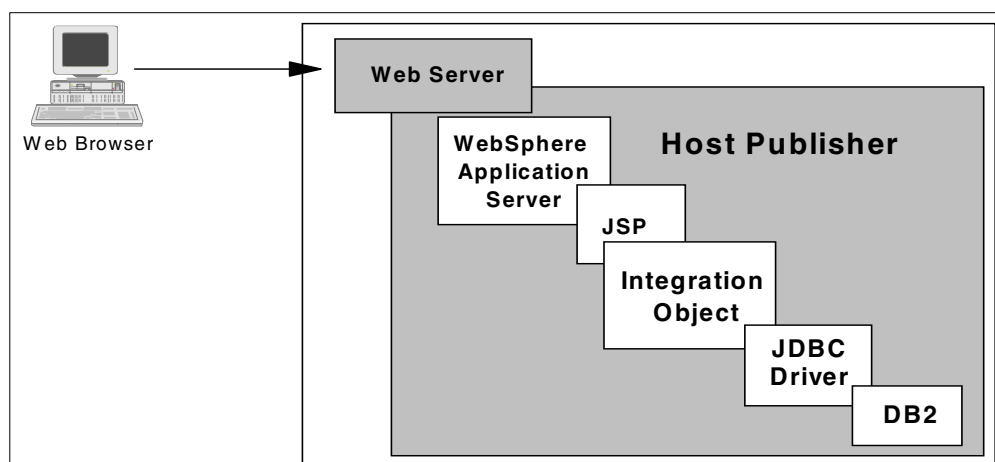


Figure 111. Using Host Publisher to access DB2

Host Publisher applications consist of Integration Objects that are actually Java beans. The Host Publisher Studio is provided to make building the Integration Objects easy. Once the Integration Objects are built, Host Publisher Studio will guide you through building the application in the form of Java Server Pages (JSPs) that invoke the Integration Objects. Developing and deploying the applications is a simple and fast process.

Simply put, Host Publisher provides you with JDBC access capabilities without the need to build Java applications and applets yourself. If you want to take it one step further and enhance the applications built by Host Publisher, for instance by adding graphics to the pages, the application source is available for you to do so. The following JDBC drivers come with Host Publisher:

- IBM DB2 UDB local

- IBM DB2 UDB remote
- AS/400 Toolbox for Java
- JDBC-ODBC bridge
- Oracle
- Sybase

The Integration Objects can be used not only in the JSPs built by the Host Publisher Studio, but in any Java application, applet, or servlet. Integration objects built to access JDBC data sources are called database Integration Objects.

WebSphere Application Server V2 Standard Edition is a requirement for Host Publisher and is provided as an integral part of the Host Publisher server to support the Host Publisher applications on the AIX, Windows NT, and Solaris platforms. For other platforms, the WebSphere Application Server for that operating system must be installed.

Host Publisher comes with the IBM SecureWay Network Dispatcher for load balancing and hot backup capability among the Host Publisher servers.

More information about Host Publisher can be found in Chapter 3, “IBM SecureWay Host Publisher” on page 41. A brief overview of using Host Publisher to create a Database Integration Object can be seen in 3.7, “Creating Database Access Integration Objects” on page 59.

11.5 Net.Data

Net.Data is an IBM software product that allows Web developers to build Web applications using Net.Data Web macros. Net.Data provides connectivity to a variety of data sources and applications including DB2, Oracle, Sybase and DRDA-enabled data sources, ODBC data sources, and across a wide range of platforms, flat files, registries, system services, and IMS databases through IMS transactions. Net.Data has an SQL interface for accessing DB2 through the CLI.

Net.Data can be used as a CGI application or as a Web server API application. When Net.Data is used as a CGI application, it is invoked as the executable DB2WWW. When Net.Data is used as a Web server API application, the server calls it as a DLL or shared library.

Language environments are Net.Data's interface to data and applications. Net.Data provides language environments to access DB2 and ODBC databases, REXX, Perl, C applications, file systems, Web registries, Java applets, and IMS transactions through IMS Web.

Net.Data supports HTTP server API interfaces for Netscape, Microsoft Internet Information Server, IBM HTTP Server, and IBM Internet Connection Server. Net.Data also supports CGI and FastCGI interfaces.

To give you an idea of how Net.Data works, consider the following case where Net.Data is used as a CGI application. When the end user enters a URL into the Web browser it is sent to the Web server. When the Web server receives a URL that refers to Net.Data, the Web server starts an instance of Net.Data and passes the information to it, including the name of the Web macro it needs. Net.Data reads and parses through the Web macro and interprets the statements. When a

Net.Data %FUNCTION statement is encountered in the macro, it loads the requested language environment library and passes the information to the specified language environment for processing. The language environment returns the results to Net.Data. After these steps are complete, HTML text is what remains. The HTML text is sent to the Web server and Net.Data terminates. The HTML is then passed to the Web browser.

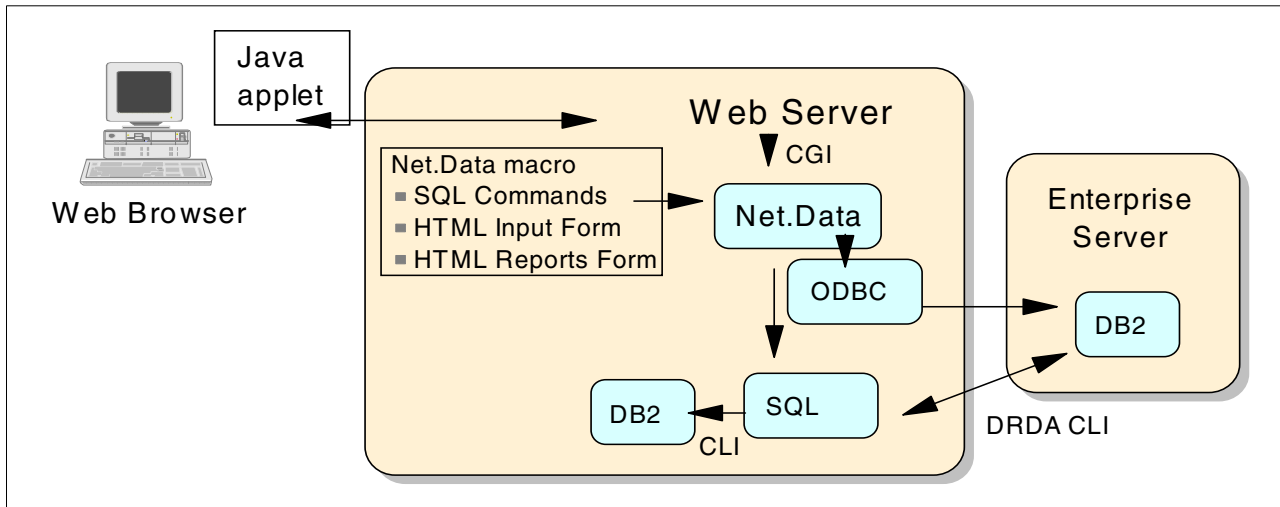


Figure 112. Net.Data

Features

- Net.Data helps you create dynamic Web pages with access to DB2 data whose content results are specified by input from the user, the current state of the database, or other factors that you design into your application.
- As a CGI, it processes input from HTML pages and sends SQL commands to a DB2 database.
- Supports client-side processing by Java applets and JavaScripts.
- Supports server-side processing by language environment definitions for languages such as SQL (for direct access to DB2), Java applications, Web Registry, REXX, Perl, Oracle and Sybase direct access languages and C++.
- Provides conditional logic and a rich macro language with the ability to run external programs. Key to application development using Net.Data is the notion of a macro file. Macro files are the mechanism for conveying information to Net.Data about gathering user input, queries to be executed, and the manner in which results are to be formatted.
- Provides a variety of servlets to help develop and manage macros.
- Supports ODBC and JDBC.
- Supports dynamic library loading.
- Provides pure HTML output.
- Live Connection feature to reduce overhead of re-establishing sessions.
- A GUI panel-driven administration tool to allow users to automate the administration and configuration of Net.Data and Live Connection.
- Error message logging.
- UTF-8 and UTF-16 Unicoding support.
- FastCGI support
- Supports the processing and display of multiple tables.

- Net.Data Task Guides provide a quick and easy way to create personalized Net.Data applications.
- A Cache Manager can be set up to the most recently used pages in memory.

Requirements

- For hardware and software requirements, as well as for optional software for performances improvements, see updated information at <http://www.software.ibm.com/data/net.data/downloads.html>
- Network connectivity to DB2
- Net.Data is available on the following platforms:
 - Windows NT
 - OS/2
 - AIX
 - OS/400
 - OS/390
 - Sun Solaris
 - HP-UX
 - UNIX

Availability

Net.Data is available with the DB2 product family:

- DB2 UDB for OS/390
- DB2 UDB for AS/400
- DB2 UDB

Where to find more information:

- <http://www.almaden.ibm.com/journal/sj/371/peng.html>
- <http://www.software.ibm.com/data/net.data>
- *Accessing DB2 for OS/390 Data from the World Wide Web*, SG24-5273

11.6 References and where to find more information

- <http://www.almaden.ibm.com/journal/sj/371/peng.html>
- <http://www.software.ibm.com/data/net.data>
- <http://www.sqlj.org>
- *Accessing DB2 for OS/390 Data from the World Wide Web*, SG24-5273
- *Lotus Solutions For the Enterprise, Volume 2, Using DB2 in a Domino Environment*, SG24-4918
- *From Client/Server to Network Computing: A Migration to Java*, SG24-2247
- *The Universal Connectivity Guide to DB2*, SG24-4894
- *DB2 Server for OS/390 Version 5 Recent Enhancements - Reference Guide*, SG24-5421

Appendix A. Special notices

This publication is intended to help enterprise developers who are looking for a way to integrate enterprise server data into the Web determine the options available. The information in this publication is not intended as the specification of any programming interfaces that are provided by any products mentioned. See the PUBLICATIONS section of the IBM Programming Announcement for each product 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 information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. 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.

This document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are

fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM ®	OpenEdition
OS/2	OS/390
OS/400	Proprinter
RACF	S/390
SecureWay	ServicPac
SP	SP1
SP2	SupportPac
System/390	TXSeries
VisualAge	VM/ESA
VTAM	WebSphere
400	

The following terms are trademarks of other companies:

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/or other countries licensed exclusively through X/Open Company Limited.

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 B. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

B.1 International Technical Support Organization publications

For information on ordering these ITSO publications see “How to get IBM Redbooks” on page 217.

- *IBM Component Broker Connector Overview*, SG24-2022
- *IBM SecureWay Host On-Demand: Enterprise Communications in the Era of Network Computing*, SG24-2149
- *Lotus Solutions for the Enterprise, Volume 4 Lotus Notes and the MQSeries Enterprise Integrator*, SG24-2217
- *Connecting IMS to the World Wide Web: A Practical Guide to IMS Connectivity*, SG24-2220
- *From Client/Server to Network Computing, A Migration to Java*, SG24-2247
- *MVS/ESA OpenEdition DCE: Application Support Servers CICS and IMS*, GG24-4482
- *Lotus Solutions for the Enterprise, Volume 3 Using the IBM CICS Gateway for Lotus Notes*, SG24-4512
- *How to Secure the Internet Connection Server for MVS/ESA*, SG24-4803
- *Lotus Solutions for The Enterprise, Volume 1 Lotus Notes: An Enterprise Application Platform*, SG24-4837
- *Lotus Solutions for the Enterprise, Volume 2: Using DB2 in a Domino Environment*, SG24-4918
- *The Universal Connectivity Guide to DB2*, SG24-4894
- *LDAP Implementation Cookbook*, SG24-5110
- *IBM Component Broker on System/390*, SG24-5127
- *IBM WebSphere Performance Pack Usage and Administration*, SG24-5233
- *CICS Transaction Server for OS/390: Web Interface and 3270 Bridge*, SG24-5243
- *Lotus Solutions for the Enterprise, Volume 5 NotesPump: The Enterprise Data Mover*, SG24-5255
- *Accessing DB2 for OS/390 Data from the World Wide Web*, SG24-5273
- *Java Application Development for CICS, Base Services and CORBA Client Support*, SG24-5275
- *Revealed! CICS Transaction Gateway with More CICS Clients Unmasked*, SG24-5277
- *Lotus Notes and Domino R5.0 Security Infrastructure Revealed*, SG24-5341
- *eBusiness Application Solutions on OS/390 Using Java, Volume 1*, SG24-5342
- *DB2 Server for OS/390 Version 5 Recent Enhancements - Reference Guide*, SG24-5421

- *Connecting Domino to the Enterprise Using Java*, SG24-5425
- *IMS e-business Connect Using the IMS Connectors*, SG24-5427
- *WebSphere Application Servers: Standard and Advanced Editions*, SG24-5460
- *Revealed! Architecting Web Access to CICS*, SG24-5466
- *CICS Transaction Server for OS/390 Version 1 Release3: Web Support and 3270 Bridge*, SG24-5480
- *Understanding IBM SecureWay FirstSecure*, SG24-5498
- *Lotus Domino Enterprise Integration: Architecture and Products* - SG24-5593
- *OS/390 e-business Infrastructure: IBM HTTP Server V5.1 for OS/390*, SG24-5603
- *OS/390 e-business Infrastructure: IBM WebSphere Application Server 1.1 for OS/390*, SG24-5604
- *Programming with the Host Access APIs*, SG24-5856

B.2 Redbooks on CD-ROMs

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 Format)	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

B.3 Other publications

These publications are also relevant as further information sources:

- *WebSphere Application Server R7 Getting Started for OS/390 V1.1*, GC34-4757
- *CICS TS for OS/390 V1R3 CICS Internet Guide*, SC34-5445
- *OS/390 V2R6.0 C/C++ User's Guide*, SC09-2361
- *OS/390 V2R8.0 C/C++ Programming Guide*, SC09-2362
- ITSO redpiece: *IBM SecureWay Host On-Demand 4.0: Enterprise Communications in the Era of Network Computing*, SG24-2149-01, available at <http://www.redbooks.ibm.com>

11.7 Referenced Web sites

- <http://www.ibm.com/software/network/hostondemand/>
- <http://www.ibm.com/software/network/hostintegration/>
- <http://java.sun.com/products/jsp/index.html>
- <http://www.ibm.com/software/network/hostpublisher/>
- <http://java.sun.com/100percent/>
- <http://www.ibm.com/software/network/technology/haci/samples.html>
- <http://www.javasoft.com/beans/docs/spec.html>
- <http://www.ibm.com/software/network/hostondemand/library/publications/beans.html>
- <http://www.edge.lotus.com>
- <http://www.lotus.com/enterpriseintegration>
- <http://www.lotus.com/home.nsf/welcome/ei>
- <http://www.s390.ibm.com/products/domino/>
- <http://www.software.ibm.com/ts/mqseries/>
- <http://www.ibm.com/software/ts/cics>
- <http://www.ibm.com/software/data>
- <http://www.ibm.com/s390>
- <http://www.ibm.com/software/websphere>
- <http://www.ibm.com/software/ebusiness>
- <http://java.sun.com/products>
- <http://www.ibm.com/software/developer/library>
- <http://www.transarc.com/Product/EFS/>
- <http://www.ibm.com/software/ts/cics/platforms/internet/cicsgw4j/announce/jgann201.html>
- <http://www.ibm.com/software/security/firstsecure>
- <http://www.software.ibm.com/data/ims/about/imsweb>
- <http://www.s390.ibm.com/nc/sntc/IMS.html>
- <http://www.software.ibm.com/data/ims/about/imswwwc/>
- <http://www.s390.ibm.com/nc/sntc/IMS.htm>
- <http://www.s390.ibm.com/nc/sntc/iwt doc/html/index.html>
- <http://www.software.ibm.com/data/ims/about/imstoc/download>
- <http://www.software.ibm.com/data/ims/imswwwc.html>
- <http://www.software.ibm.com/data/ims/imstoc.html>
- <ftp://ps.software.ibm.com/ps/products/imsweb/r210/HWSMH210.exe>
- <http://www.software.ibm.com/data/ims/ioc.html>
- <http://www.software.ibm.com/data/net.data/>
- <http://www.transarc.com>
- <http://www.software.ibm.com/data/ims/jitoc.html>
- <http://www.almaden.ibm.com/journal/sj/371/peng.html>
- <http://www.software.ibm.com/data/net.data>
- <http://www.sqlj.org>
- **White paper: IBM Component Broker Technical Overview**
(<http://www.software.ibm.com/ad/cb/litp.html>)

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about ITSO 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 redbooks fax order form to:

In United States	e-mail address usib6fpl@ibmmail.com
Outside North America	Contact information is in the "How to Order" section at this site: http://www.elink.ibm.link.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.ibm.link.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.ibm.link.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:

[illegible]

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.

Index

Symbols

@DB 97, 197

Numerics

3270 Bridge 138
3270 host print 19
3270 Web bridge 138, 143, 156, 157, 159, 160, 161
3287 9, 19
5250 file transfer 13
5250 host print 20
5250 printer 9, 20

A

ACCUM 160
ActiveX 84
AIX 43
Alias 136
Apache 43
applet 9, 12, 13, 14, 16, 24, 32, 34, 91, 92, 148, 163, 174, 182, 198, 199, 204, 208
AS/400 Toolbox for Java 36, 59, 207
AS/IMS 191
Automation Objects 87, 92, 93

B

Basic Mapping Support (BMS) 136
BDK 84
BMS 136
broadcast 17

C

C++ 87, 91, 92, 96, 168, 177, 178, 196
cached client 13
CAE 99, 198, 199, 200, 201, 203
CBConnector 138
CGI 177, 180, 196, 197, 208
CICS 7, 8, 13, 57, 58, 152, 175, 196
CICS Corba 138, 158
CICS External CICS Interface (EXCI) 135
CICS Gateway applet 150
CICS Gateway for Java 139
CICS Internet Gateway 139
CICS Java Gateway 9
CICS Native IIOP support
 CIOR transaction 159
 CSOL 159
 DFHIIOP 159
 DFHIIOPA 159
 DFJIIOP 159
 REQUESTMODEL 159
CICS provided ORB 159
CICS Sockets listener 156
CICS telnet daemon 147
CICS Transaction Gateway 161, 162, 163, 164

CICS Transaction Gateway (CTG) 139
CICS Universal Client 135, 147, 162
CICS Web Interface (CWI) 138
CICS Web support 138, 151, 154, 157, 159, 160, 161
cicsteld 147
CLI 98, 196, 197, 199, 207
client authentication 16
Color Remap bean 85
COM 93
COMMAREA 135, 136, 143, 154, 155, 156, 159, 161, 163
Communications Server 3, 15, 16, 17, 91, 146, 147
Communications Server for NT 27
Component Broker 4, 7, 137, 159, 168, 196
conditional statements 56
Connect Macro 52
connection pools 42
CORBA 7, 96, 116, 137, 138, 158, 159, 196
CSP 139
CTG 139, 143, 161, 162, 163, 164
CWI 138
CWS 138, 143, 151, 152, 153, 159
 CICS socket listener 153
 CSOL 153
 CWXN 153

D

Data compression 14
data looping 56
Data Macro 52
Database Access Object 49
Database On-Demand 7, 10, 13, 28, 34, 39, 40, 60, 175, 204, 205
DB2 7, 8, 43, 168, 175, 185, 195, 196, 197, 198, 199, 201, 204, 207
DB2 CAE 200
DB2 Client Application Enabler 203
DB2 Connect 200, 202
DB2 Connect Enterprise Edition 203
DB2 Connect Personal Edition 203
DB2 UDB local 59, 206
DB2 UDB remote 59, 207
DB2LSX 97, 98, 197
DCE 169, 191, 192
DECS 96
DES 42
DFHCCNV 153, 163
DFHMIRS 155
DFHWBAPI 155
DFHWWBLI 155, 161
DFHWWBTTA 161
Directory Agent 17
Directory assistance 110
directory catalog 109
directory information tree 23
Disconnect Macro 52
Dispatcher 46, 47

distinguished name 23
Distributed Program Link (DPL) 135
DIT 23
DNS 46, 47
Domino 43, 137
Domino Enterprise Connection Services 96
download client 13, 15
DPL 135, 159
DRDA 185, 196, 202, 207

E

EAB 172
ECI 135, 162, 163, 164
ECLApplets 91
EJB 4
Encina DE-Light 168, 192
Enterprise Access Builder 172
Enterprise beans 114
enterprise beans 112, 113
Enterprise JavaBeans 4, 112
EPI 135, 162
EXCI 135, 155, 162, 164
execution page (JSP) 60
Express 11, 24
Express Client 9, 14, 15, 16
Express Server 9, 10, 14, 16
Express Service 16
External Call Interface (ECI) 135
External Presentation Interface (EPI) 135

F

failover 17, 18, 42
FEPI 161
FileTransfer bean 85
firewall 3, 148
Front End Programming Interface 161
Function On-Demand 13, 14, 204

G

global screens 54
GWAPI 180

H

HACL 11, 20, 137
HACL Automation Objects 91
HOD 21, 136
 direct connection 145
 protocol isolation 146
HOD Service Manager 150
Host Access Beans 92
Host Access Beans for Java 11, 20, 91, 92
Host Access Class Libraries 20
Host Access Class Library 11, 146
Host Access Controls for ActiveX 91, 92
Host Access Integration Object 49
Host On-Demand 3, 5, 6, 9, 10, 11, 12, 13, 14, 16, 18, 21, 24, 25, 31, 91, 148, 149, 167, 175, 195, 204, 205
Host On-Demand Connector 83, 90, 121

Host On-Demand Express 14
Host Print Transform (HPT) 20
Host Publisher 3, 7, 42, 43, 45, 48, 52, 57, 136, 137, 143, 151, 167, 174, 195, 206
Host Publisher Server 41, 43, 66
Host Publisher Server Administration panel 66
Host Publisher Studio 41, 42, 43, 44, 45, 48, 50, 56, 65, 167, 174, 206
Host-On-Demand 136, 143
HPT 20
HTML 41, 42, 44, 64, 65, 174, 195, 196, 197, 199, 206, 208
HTML editor 57
HTTP 17
httpd.conf 155
HTTPS 17

I

IBM HTTP Server 42, 43, 95, 179, 207
IBM SecureWay Host Access Class Library 86
IBM SecureWay LDAP Server 21
IBM Visual Age for Java 92
IBM VisualAge C++ 92
ICAPI 180
IDL 158
IIOP 96, 138, 139, 158, 159
IMS 7, 8, 167, 169, 170, 172, 175, 185, 196, 207
IMS Client for Java 168, 182
IMS Connect 172
IMS Connector for Java 173
IMS Object Connector 168, 184
IMS TCP/IP 170
IMS TOC 171
IMS TOC Connector 171
IMS Web 168, 176
IMS Web Runtime 177, 178, 179
IMS Web Studio 177, 179
IMS Web Templates 179
IMS WWW templates 179
input page (JSP) 60
input variables 54, 62
Integration Object 44, 49, 50, 51, 52, 54, 59, 63, 64, 65
Interface Definition Language 158
intraNetWare for SAA 17
ISS 46, 47

J

Java 12, 19, 87, 101, 125, 167, 168, 182, 196
Java Development Kit 92
Java Server Pages 4, 7, 49, 195, 206
JavaBeans 44, 49, 83, 162
JavaBeans Development Kit 84
JavaGateway class 163
JavaScript 196
JCICS 138
JDBC 7, 9, 10, 28, 30, 36, 41, 42, 59, 60, 96, 101, 150, 195, 196, 197, 198, 199, 200, 203, 204, 205, 206, 208
JDBC Native driver 198
JDBC Net driver 198

JDBC thin driver 198
JDBC-ODBC bridge 8, 60, 197, 198, 201, 207
JITOC 172
JSP 4, 7, 49, 60, 63, 65, 112, 151, 174, 195, 206

K

KeyPad bean 85
KeyRemap bean 85

L

LaserPCL 19
LaserPPDS 19
LaunchPad 88
LC API 96, 97, 197
LC Java 96
LC LSX 96, 97, 197
LDAP 11, 24, 97
LDAP Server 12
LEI 96, 197
Lightweight Directory Access Protocol 110
LINK 136
Live Connection 208
load balancing 45, 151, 174
Locally installed client 13, 14
Lotus Bean Machine 84
Lotus Connector API 96, 97
Lotus Connector Java classes 96
Lotus Connector LotusScript Extension 96
Lotus Connector LotusScript Extensions 97
Lotus Domino 8, 109, 197
Lotus Domino Go 43
Lotus Enterprise Integrator 96
Lotus Notes 92
LotusScript 8, 87, 93
LotusScript Data Object 97
LotusScript Extension 87, 91, 92
LS
 DO 97, 98, 197, 201
LUM 24

M

Macro bean 85
MacroManager bean 85
Microsoft Front Page 44
Microsoft IIS 43
Microsoft Office 92
Microsoft Visual C++ 92
MQEI 8, 97, 137, 168, 189
MQLSX 8, 97, 137, 168, 187
MQSeries 4, 8, 42, 168, 169, 173, 188
MQSeries Client for Java 173
MQSeries Enterprise Integrator 97
MQSeries link LotusScript Extension 97
multicast 17

N

NDS 109
Net.Data 7, 168, 177, 178, 185, 195, 196, 199, 201,

207, 208
NetCICS 138, 160, 161
 NCSWCONV 161
NetObjects Fusion 44
Netscape Directory Server 21
Netscape Enterprise Server 43
Netscape FastTrack Server 43
Network Dispatcher 42, 43, 45, 151, 174, 207
NotesPump 8, 197
Novell Directory Services 97

O

ODBC 98, 168, 196, 197, 201, 203, 208
OHIO 11, 90
OIA 88
OO 90
Open Host Interface Objects 11, 83, 90
Open Transaction Manager Access (OTMA) 170
Operator Information Area 88
Oracle 43, 60, 185, 196, 201, 207, 208
ORB 158
OS/390 43
OS/400 16
OTMA 170, 171, 173, 178, 182, 190

P

Packaging 42
Perl 168, 196, 208
Personal Communications 3, 19, 20, 87, 91
PostScript 20
PowerBuilder 87
presentation space 88
Printer Definition Files (PDFs) 19
Printer Definition Tables (PDTs) 19
Proprinter 19
public key infrastructure 109

R

RACF 159, 172
RealTime 197
Redirector 10, 11, 12, 15, 16, 24
REXX 168, 196, 208
RFC 17

S

scope 17, 18
Screen Bean 85
Screen Customizer 10, 13, 16, 151, 167, 176
Screen Customizer/LE 6, 10, 13, 14, 149, 176, 205
screen recognition 53
Service Agent 17
Service Location Protocol 17
servlet 164, 196, 199
Session bean 85
SLP 11, 17, 18, 27
smartcard 17
SQL 10, 35, 39, 59, 61, 62, 97, 196, 197, 200, 201, 205, 206, 207, 208

SQL statement 60
SQLJ 197, 200, 201
SSL 10, 11, 14, 15, 16, 17, 42, 148, 163
SUN Solaris 43
Sybase 43, 60, 185, 196, 207, 208
Symantec Visual Cafe 92

T

Terminal bean 85
TN3270 9, 13, 17, 18, 27, 50, 176, 205
TN5250 9, 13, 17, 18, 50, 205
TXSeries 4, 161

U

URL 13, 57, 60
User Agent 17

V

Visual Basic 87
VisualAge for Java 95, 162, 172
VisualAge Generator 139
VisualAge Smalltalk Enterprise 168
VT 13, 50, 86, 150, 175, 195, 204, 206
VT100 9
VT220 9
VT52 9

W

Web bridge exit 157
WebSphere 4, 42, 43, 45, 95, 172
WebSphere Application Server 42, 43, 49, 125

X

X.500 109

IBM Redbooks evaluation

IBM Web-to-Host Integration Solutions
SG24-5237-01

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.ibm.com/>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?

☐ **Customer** ☐ **Business Partner** ☐ **Solution Developer** ☐ **IBM employee**
☐ **None of the above**

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

SG24-5237-01

Printed in the U.S.A.

