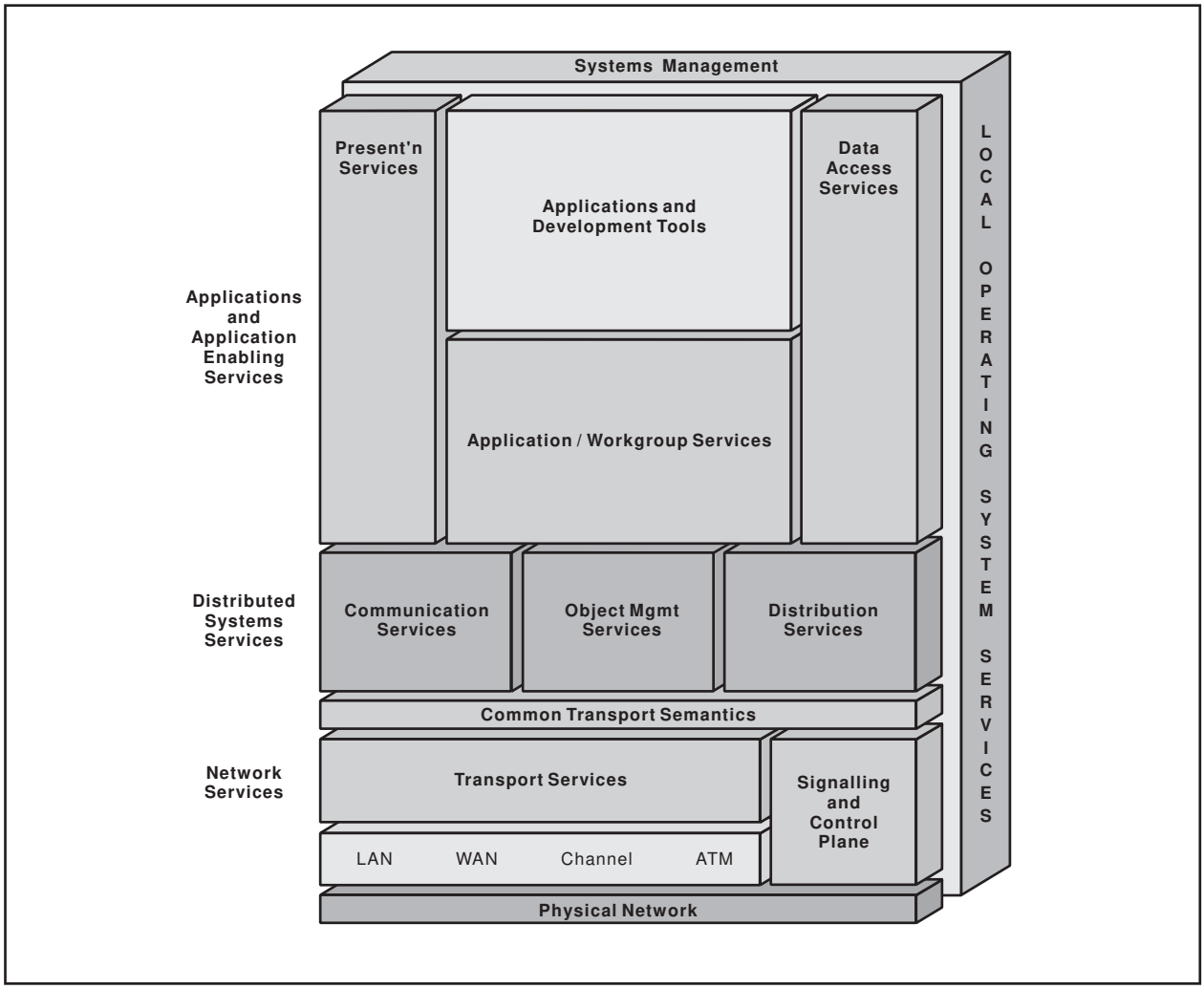
Open Blueprint

Web Browser Resource Manager



G325-6589-00

Open Blueprint

Web Browser Resource Manager

G325-6589-00

About This Paper

Open, distributed computing of all forms, including client/server and network computing, is the model that is driving the rapid evolution of information technology today. The Open Blueprint structure is IBM's industry-leading architectural framework for distributed computing in a multivendor, heterogeneous environment. This paper describes the Web Browser resource manager component of the Open Blueprint and its relationships with other Open Blueprint components.

The Open Blueprint structure continues to accommodate advances in technology and incorporate emerging standards and protocols as information technology needs and capabilities evolve. For example, the structure now incorporates digital library, object-oriented and mobile technologies, and support for internet-enabled applications. Thus, this document is a snapshot at a particular point in time. The Open Blueprint structure will continue to evolve as new technologies emerge.

This paper is one in a series of papers available in the *Open Blueprint Technical Reference Library* collection, SBOF-8702 (hardcopy) or SK2T-2478 (CD-ROM). The intent of this technical library is to provide detailed information about each Open Blueprint component. The authors of these papers are the developers and designers directly responsible for the components, so you might observe differences in style, scope, and format between this paper and others.

Readers who are less familiar with a particular component can refer to the referenced materials to gain basic background knowledge not included in the papers. For a general technical overview of the Open Blueprint, see the *Open Blueprint Technical Overview*, GC23-3808.

Who Should Read This Paper

This paper is intended for audiences requiring technical detail about the Web Browser Resource Manager in the Open Blueprint. These include:

- Customers who are planning technology or architecture investments
- Software vendors who are developing products to interoperate with other products that support the Open Blueprint
- Consultants and service providers who offer integration services to customers

© **Copyright International Business Machines Corporation 1996. 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

Web Browser Resource Manager 1
Web Browser Structure and Function 2
Protocols Supported 6
Web Browser Direction 6

Notices 7
Trademarks 7

Communicating Your Comments to IBM 9

© Copyright IBM Corp. 1996

Figures

1. A Web Browser Screen 1
2. The Web Browser Environment 2
3. A Simple Forms Interaction 4
4. The Use of Java Applets Within the Browser 5

© Copyright IBM Corp. 1996

Web Browser Resource Manager

Web browsers have become the client interface of choice for Internet computing. As the user's gateway into the World Wide Web (WWW), Web browsers provide a client based end user environment for information delivery and interaction with Web servers. The Web Browser resource manager provides the user with a point and click method for browsing and navigating Web content without overt knowledge of the underlying servers accessed. The Web Browser resource manager includes capabilities that provide an attractive electronic publishing medium and client interaction based on plug-in support and Java-enabled capabilities. For the convenience of the reader, this paper uses the term *Web browser* to refer to generic Web browser function, and *Browser* or *Web Browser resource manager* to refer to the Open Blueprint Web Browser resource manager.

From a user's perspective, the WWW is a collection of documents containing text, multimedia objects, or hyperlinks to other pages. Information providers set up hypermedia (Web) servers from which Web browsers can retrieve documents. Figure 1 below is an example of a Web browser screen.

Figure 1. A Web Browser Screen

Web browsers depend on hypertext and multimedia to provide an easy-to-use interface for exploring and retrieving information from the World Wide Web. Hypertext documents provide information about a

© Copyright IBM Corp. 1996

particular subject and also provide links to other relevant documents. Much of the current Web structure is page-oriented content and uses hyperlinks for navigation. Users click on the links in a document they are viewing to make the linked document available for further browsing (viewing). Users can also click on multimedia data icons for related functions, for example to listen to an audio stream.

Because the Web Browser resource manager can provide distributed presentation support for a wide variety of server based applications, it provides the capabilities of a *thin client*.

Web Browser Structure and Function

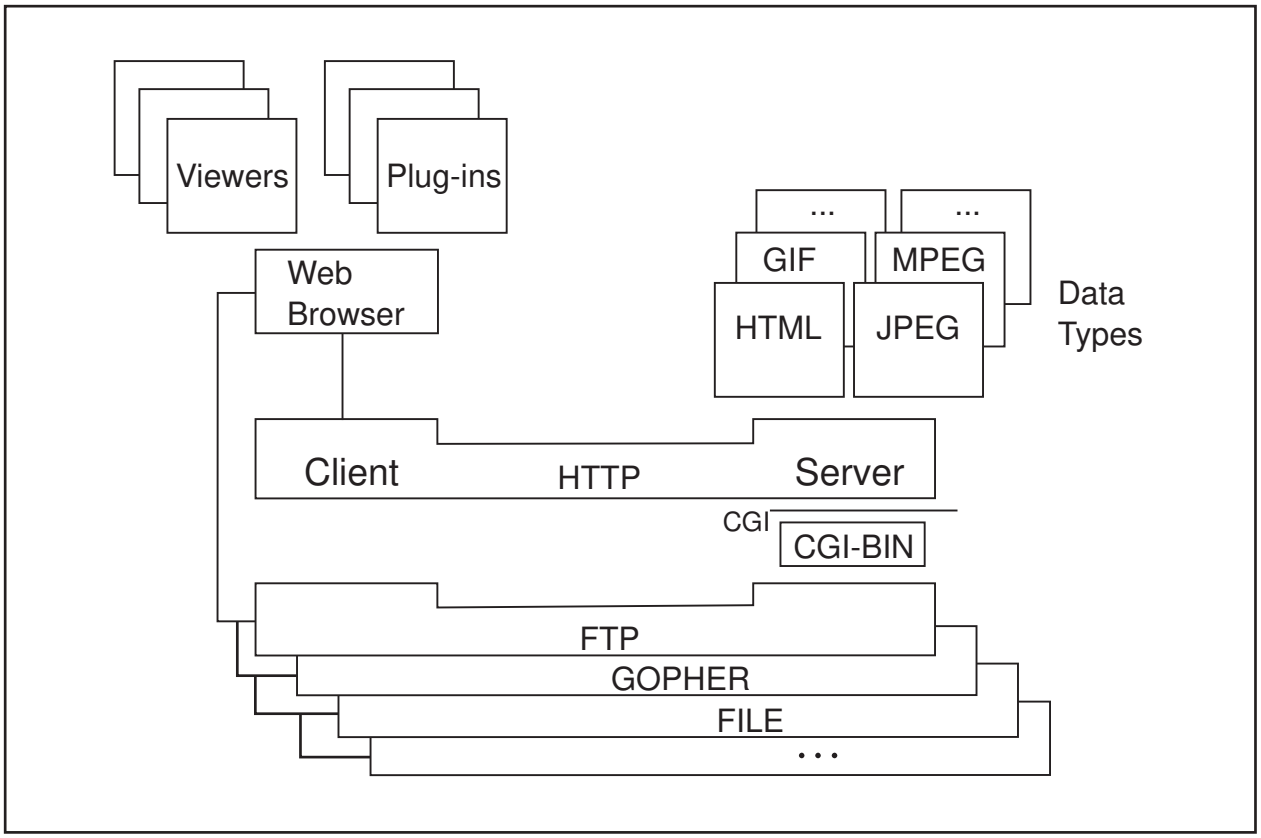


Figure 2. The Web Browser Environment

As shown in Figure 2 above, the Web Browser resource manager operates primarily through the HTTP resource manager to handle HyperText Markup Language (HTML) and other data types supported by the HyperText Transport Protocol (HTTP). HTTP, designated by `http://` in a Uniform Resource Locator (URL), is the most common protocol found on the Web. (See the Directory resource manager component description paper for details on the definition and format of URLs.) This protocol defines how content such as hypermedia files move from Web server systems to the Browser on the client system. However, the Browser can support other protocols such as FTP, Gopher, File and so on (see “Protocols Supported” on page 6).

The Browser primarily consists of:

- *A parsing engine* that requests, dissects, and interprets incoming data that is distributed through out the Internet
- A native platform *rendering engine* for viewing the information

The Browser uses:

- The HTTP resource manager and Open Blueprint Network Services to work with the multiple protocols the client supports (see the Network Services and HTTP resource manager component description papers)
- Security resource managers to ensure privacy of communications between the client and the server. (See the *Security in the Open Blueprint* and the *Open Blueprint HTTP Resource Manager* component description papers for additional information about Web security.)
- The Virtual Machine resource manager for execution of downloadable Java applications

The browser acts as an interpreter and viewer for specific data types (see “Data Type Support” on page 5). In addition, the browser includes plug-in and platform specific helper applications that enable viewing and interaction for data types not directly supported by the Browser. This extendability feature also allows enterprises to provide enterprise-specific helper applications that are configured to work with standard Web browsers. However, with current advances in Web browsers, more and more functions previously performed outside of the Web browsers are being performed within the actual Browser.

The Web Browser resource manager is enabled for international use, supporting the navigation and viewing of documents in various languages, including double-byte languages. The Browser client is, however, only capable of displaying content as it is provided by Web servers. To fully support the international market, Web content providers can establish multiple versions of their Web data, including multiple navigation paths based on the language preference of the end user.

Web Navigation

The Web Browser resource manager recognizes URLs contained within HTML documents (embedded URLs). When an end user clicks on an embedded URL, the browser accesses the designated server and returns the requested information. This process can be repeated such that chains of information can be accessed. The end user can always go back along the chain to a previously accessed information page.

Forms Processing

The browser provides support for forms processing. Forms processing, which is defined by the HTML 2.0 specification, enables applications to accept input from end users. An HTML form used in conjunction with a server CGI program (see the HTTP Resource Manager component description paper), can deliver dynamic content to the browser, based on user input from the client. A form can contain various input controls, including:

- Text fields
- Checkboxes
- Radio buttons
- Scrolling lists
- Popup menus
- Buttons

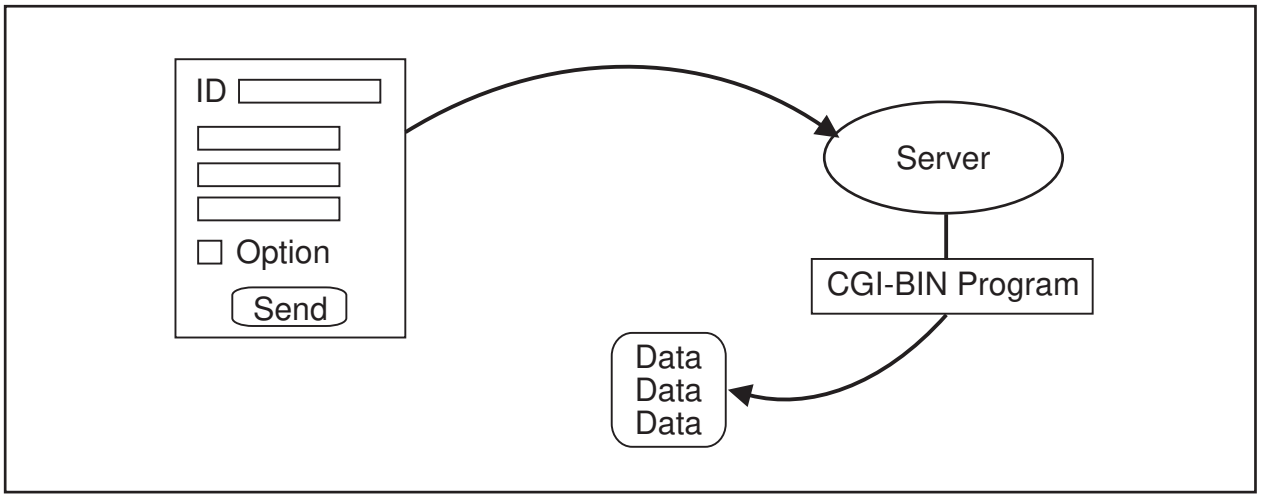


Figure 3. A Simple Forms Interaction

As shown in Figure 3, HTML forms sent to a client by a server are filled in by the end user and converted to HTML by the Browser. The Browser then uses the HTTP resource manager to flow the form to the server indicated in the original HTML form source. Typically, the form identifies a CGI-BIN program that will process the form.

Forms are potentially a greatly improved means of collecting user information, much more intuitive than for example, the alternative of a primitive mail interface. Forms within the browser provide a moderate level of customization and number of available controls. An interface requiring higher interactivity or additional control types would require alternative methods for interaction with the user.

Note: A draft proposal of HTML 3.2 is awaiting ratification by the W3C.[1] This proposal contains many enhancements to HTML that provide more control over Internet content formatting. With these improvements, the content developer will be able to more easily control appearance of data viewed on the Web, including sophisticated use of color, fonts, alignment, lists and stylesheets. The draft proposal can be viewed on the Web at URL: http://www.w3.org/pub/WWW/TR/WD-html32.html

Java Support

The Browser supports Java applets as defined in the HTML 3.2 Working Draft. This is a key capability of the Browser. Java provides a platform-neutral implementation for extending browser functions. This implementation frees the content developer from multiple platform development because Java bytecodes are sent to and interpreted by the Virtual Machine on the client system. The Virtual Machine resource manager supports the IIOP protocol, allowing object-oriented messaging between the Java environment and other object-oriented environments by using the Object Request Broker (ORB) resource manager. IIOP is the Internet Inter-ORB Protocol defined by Object Management Group (OMG) as part of the Common Object Request Broker Architecture (CORBA) specification. IIOP enables platform-independent object messaging for distributed object applications. This protocol enables enterprises to take advantage of existing CORBA 2.0 compliant enterprise applications.

For additional information about Java and ORB support, see the Virtual Machine resource manager and Object Request Broker resource manager component description papers.

Dynamic Content Support

An example of the use of Java applets is the support of dynamic content by the browser. Dynamic content is defined as information that is assembled at *run time* as opposed to predefined static content. With the use of Java applets within HTML, portions of the browser rendered view can be owned by an applet and updated by client control from the running applet. Java can potentially reduce the refresh of documents necessary to provide much greater use of dynamic content in the browser.

Note: Many server CGI-BIN programs also create content dynamically, typically by creating HTML files for the browser to interpret and render.

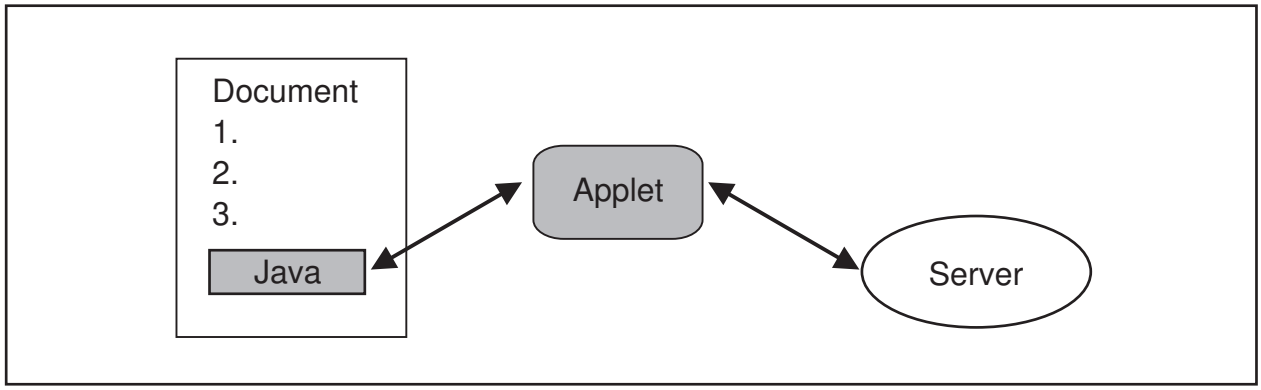


Figure 4. The Use of Java Applets Within the Browser

Figure 4 illustrates the movement (downloading) of a Java applet from a server to a client. The need for the applet is recognized by the browser based on an HTML tag within a document. As stated above, the browser and the server communicate using the HTTP resource manager.

Data Type Support

The World Wide Web contains many types of data. Increasingly, much of the data on the World Wide Web is multimedia content, which includes advanced graphics, audio, video, and animation formats. Data delivered from the server using the HTTP protocol is classified by Multipurpose Internet Mail Extension (MIME) type. The Web Browser resource manager either renders the data itself if it can do so appropriately, or it can invoke a helper or plug-in application based on the MIME type that is capable of handling the specific data type.

Examples of data types (MIME types) supported directly by the Web Browser resource manager are:

- GIF Image (image/gif)
- JPEG Image (image/jpeg)
- HTML Document (text/html)
- plain text (text/plain)

Examples of data types supported through the use of the View resource manager or specific format viewers are:

- Acrobat PDF document (application/pdf)
- PostScript formatted document (application/postscript)
- Rich-Text Format document (application/rtf)
- MO:DCA—P formatted document (application/afp)

Examples of data types supported by the Multimedia resource manager include:

- Microsoft WAV audio (audio/x-wav)

- RealAudio file (audio/ra)
- Apple QuickTime video (video/quicktime)
- MPEG video (video/mpeg)

Protocols Supported

In addition to HTTP, the Web Browser resource manager supports multiple protocols. The protocol is usually specified as part of the URL and designates the type of server that will handle a given request such as selecting a link. Other protocols the Browser supports are:

ftp The File Transfer Protocol (FTP) is used to transfer files from FTP servers using the FTP application. The Browser checks the file type to determine the appropriate actions such as unpacking compressed files and prompts the user to designate their choice for local storing.

file This protocol designation tells the Browser that the information desired resides on a local file system such as a hard drive, CD-ROM, or diskette. When the format of the URL is: `file:///<DriveName>:`], this indicates that the URL points to a file on the hard drive specified in the `<DriveName>:` parameter. Otherwise, it performs the same function as the FTP: protocol

gopher Gopher servers are precursors to today's web servers that enable Gopher clients to view remote file directories and files. Gopher client function is incorporated in the Browser. By employing the gopher protocol, the Browser can display a text view of the gopher server's file directories.

mailto The mailto protocol enables the Browser to invoke support for the Internet Simple Mail Transfer Protocol (SMTP), for example by invoking the Mail resource manager.

news The news protocol enables the Browser to access to the Usenet newsgroups. Newsgroups are topic discussion forums. The news protocol is supported through using Newsreader viewers.

telnet This protocol results in a telnet session. The Browser does not open telnet sessions; but it delegates launching to a separate telnet client application program.

wais Wide Area Information Server (WAIS) provides a technique for searching databases over the Internet. This protocol tells the Browser to invoke the WAIS application client, which allows the end user to enter search requests for execution by a WAIS application server. Results obtained by the WAIS application are returned to the end user through the Browser.

Web Browser Direction

The Web Browser resource manager provides the external appearance of the World Wide Web (and some local and distributed enterprise data). This style of end user support will be augmented in the future as the Human Computer Interface (HCI) supports the concept of customized places, which will supersede the current single desktop metaphor. Examples of places are (rooms within) the home and conference rooms. As this occurs, APIs will be able to invoke the Web Browser resource manager subcomponents, such that the Browser function will fit into the new customized places metaphor.

For more information on the places metaphor, see the Human Computer Interaction resource manager component description paper.

---

[1] The World Wide Web Consortium (W3C) was established to promote advances in the World Wide Web and encourage the adoption of a common set of World Wide Web protocols throughout the world.

Notices

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. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

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:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

IBM
IBMLink
MOD:CA
Open Blueprint

The following terms are trademarks or service marks of other companies:

Apple (name and logo)	Apple Computer, Incorporated
CORBA	Object Management Group, Incorporated
Gopher	University of Minnesota
Java	Sun Microsystems, Incorporated
Microsoft	Microsoft Corporation
PostScript	Adobe Systems, Incorporated
RealAudio	Progressive Networks
QuickTime	Apple Computer, Incorporated

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the U.S. and other countries.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

© Copyright IBM Corp. 1996

Communicating Your Comments to IBM

If you especially like or dislike anything about this paper, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply. Feel free to comment on specific error or omissions, accuracy, organization, subject matter, or completeness of this paper.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

- If you prefer to send comments by FAX, use this number:

  United States and Canada: 1-800-227-5088.

- If you prefer to send comments electronically, use one of these ID's:
  - Internet: **USIB2HPD@VNET.IBM.COM**
  - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
  - IBMLink: **CIBMORCF at RALVM13**

Make sure to include the following in your note:

- Title of this paper
- Page number or topic to which your comment applies

© Copyright IBM Corp. 1996

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

G325-6589-00